

Regression with R

Niels Richard Hansen

2026-08-25

Table of contents

Preface	6
1 Introduction	8
1.1 A brief history of regression	10
1.1.1 Linear regression	10
1.1.2 Beyond linear regression	12
1.1.3 Regression and machine learning . . .	15
1.1.4 Computation	16
1.1.5 The four components to regression modeling	17
1.2 What this book is—and what it is not	18
1.3 Organization	21
1.4 Using R	22
2 The linear model	23
2.1 Outline and prerequisites	23
2.2 Claim sizes	24
2.3 The linear model assumptions	27
2.4 Linear models of claim sizes	31
2.5 Estimation theory	36
2.5.1 Weighted linear least squares estimation	36
2.5.2 Algorithms	40
2.6 Sampling distributions	41
2.6.1 Moments	42
2.6.2 Tests and confidence intervals	43
2.7 Model assessment	46
2.7.1 Investigating the claim size models . .	48
Exercises	53
3 Birth weight—a case study	60
3.1 Outline and prerequisites	60
3.2 Exploratory data analysis	61
3.2.1 Descriptive summaries	62
3.2.2 Pairwise associations	66
3.2.3 Birth weight and gestational age . . .	71

3.3	Linear models of birth weight	72
3.3.1	Univariate regression models	72
3.3.2	A multivariate linear regression model	75
3.3.3	Including interactions	78
	Exercises	82
4	Exponential dispersion models	84
4.1	A motivating example	84
4.2	Exponential dispersion models	85
4.3	Deviance	96
4.3.1	Bregman divergences and quasi-deviances	98
	Exercises	101
5	Generalized linear models	105
5.1	Model assumptions	105
5.2	Estimation theory	108
5.2.1	Maximum likelihood estimation	108
5.2.2	Algorithms	112
5.3	Sampling distributions	115
5.3.1	An idealized least squares estimator	116
5.3.2	Tests and confidence intervals	118
5.4	Model assessment and diagnostic	121
	Exercises	123
6	Existence and uniqueness	129
	Exercises	129
7	Regression modeling	132
7.1	Outline and prerequisites	133
7.2	General regression models	133
7.2.1	Estimation methodologies	137
7.2.2	Regression modeling objectives	139
7.3	Nonlinear basis expansions	145
7.3.1	Nonlinear expansions for claims modeling	145
7.3.2	Nonlinear birth weight models	153
7.4	Practical data analysis	160
7.5	Causal interpretations	160
	Exercises	160
8	Model Assessment	163
8.1	Outline and prerequisites	163
8.2	Bike rentals	164

8.3	Quantifying model predictions	167
8.3.1	Loss and risk	168
8.3.2	Regression risk	175
8.4	Error estimates	176
8.4.1	Error estimation via validation data	177
8.4.2	Error estimation via data splitting	178
8.4.3	Analytic error estimates	184
8.5	Data splitting estimators compared	191
8.5.1	Implementations in R	191
8.5.2	Simulation study	192
8.6	Binary model assessment	197
8.6.1	Concordance parameters	197
	Exercises	199
9	Small for gestational age – a case study	203
9.1	Outline and prerequisites	203
9.2	Small for gestational age	204
9.3	Binary regression	207
9.3.1	Logistic regression	208
9.3.2	Other binary regression models	213
9.3.3	Calibration	217
9.3.4	Measuring and optimizing predictive performance	222
	Exercises	226
10	Advertising – a case study	227
10.1	Outline and prerequisites	227
10.2	Exploratory data analysis	228
10.2.1	Pairwise associations	232
10.3	A Poisson regression model	236
10.4	Improvements of the Poisson model	239
	Exercises	246
11	Survival analysis	250
11.1	Outline and prerequisites	250
11.2	Introduction to survival analysis	251
11.2.1	Survival distributions and censoring	254
11.2.2	Model assumptions	255
11.2.3	The Kaplan-Meier and Nelson-Aalen estimators	257
11.3	Parametric survival models	263
11.3.1	The likelihood	264
11.3.2	Regression models	267

11.4	Semiparametric survival models	277
11.4.1	Cox's partial likelihood	279
11.4.2	Poisson empirical likelihood	282
11.4.3	Ties	286
11.5	Survival residuals	287
11.5.1	Cox-Snell residuals	287
11.5.2	Martingale residuals	288
11.5.3	Deviance residuals	292
11.5.4	Prostate cancer continued	292
11.6	Discrete time survival models	297
11.6.1	Discrete hazards	297
11.6.2	Empirical likelihood estimators	299
11.6.3	The discrete proportional hazards model	300
	Exercises	301
12	Company default – a case study	308
12.1	Outline and prerequisites	308
12.2	Exploratory analysis	309
12.3	A Cox model	314
12.3.1	Model predictions	315
12.3.2	Model diagnostic	319
13	Interval estimation	322
13.1	Outline and prerequisites	322
13.2	Interval estimators	323
13.2.1	Standard confidence intervals	326
13.2.2	Likelihood intervals	327
13.2.3	Robust standard errors	334
13.3	Bootstrap	338
13.3.1	Bootstrap basics	338
13.3.2	Bootstrap algorithms	340
13.3.3	Bootstrap calibration	343
13.4	Advertising revisited	346
13.4.1	Model based intervals	347
13.4.2	Robust standard errors	349
13.4.3	Bootstrap intervals	351
	Exercises	356

Preface

Warning

The book is undergoing revision and some chapters are subject to change without notice.
If an individual chapter is still under revision this is marked at the top of the chapter.

This book was written as the textbook material for a graduate statistics course in regression analysis. The prerequisites include univariate linear regression, t -tests and constructions of standard confidence intervals, and knowledge of standard distributions such as the normal, the binomial and the Poisson distribution. The reader will also benefit from introductory statistics courses covering likelihood methods, one- and two-sided analysis of variance, and aspects of asymptotic theory. In addition, a solid knowledge of linear algebra is assumed.

The exposition is mathematical, but the emphasis is on data modeling as well as theory. Mathematics is used to make model descriptions and assumptions precise, and to give precise results relevant for the practical analysis of data and the computations required for carrying out data analysis.

The book attempts to be complete and thorough on the topics covered, yet to be practical and relevant for the applied statistician. The means for achieving the latter is by larger case studies using R. The R code included is complete and covers most aspects of the data analysis from reading data into R, cleaning and plotting data to data analysis, data modeling and model diagnostics.

The book source is available from the [RwR GitHub repository](#) and it is accompanied by the [R package RwR](#) containing datasets used in the book.

 Warning

The book is undergoing revision and some chapters are subject to change without notice.

If an individual chapter is still under revision this is marked at the top of the chapter.

1 Introduction

Adrien-Marie Legendre was a French mathematician, who in the early 19th century took a particular interest in predicting the trajectories of comets. The understanding of the movements of celestial objects represented one of the main applications of mathematics at that time. There were great scientific challenges in astronomy, such as the stability of the solar system, but there were also more earthbound applications, such as celestial navigation. Legendre wanted to describe the trajectory of a comet by a parabolic path, and he wanted to use observations of points from a comet's trajectory to infer the best path coefficients. When Legendre substituted the observed points into the path equation he was left with more linear equations in the unknown coefficients than the number of coefficients. There was no solution; the observations did not fall exactly on any parabolic path.

Legendre was not the first to deal with such a *path fitting* problem, and various techniques had been developed, none of which were completely satisfactory. One solution was to disregard some equations so that a unique solution could be found to the remaining. The resulting path would then match some observed points exactly. But which equations should be disregarded? And would the resulting path be a good approximation to the omitted observation points?

Legendre's parabolic paths could only produce an approximation of the observations and the future trajectory of the comet. It would not be an exact match. The deviations—or errors—from predicted trajectories of celestial objects were primarily due to unsystematic inaccuracies in the measurements. For Legendre and his contemporaries there was no clear solution as to how deviations from a path should be described and handled. The statistical theory was in its infancy, and the numerical computations required to obtain a descriptive statistical understanding of their nature was very labor intensive. Today it is well known that such deviations

can be described statistically, and the methods of path or curve fitting are beneficially studied using statistical theory.

Legendre found an elegant and applicable algebraic solution to his path fitting problem—as described in the section below—which was quickly adapted by astronomers and geodesists in France and Germany. He did, however, not realize the statistical nature of his method. This aspect was developed and expanded by others in the 19th and 20th century, and the theory evolved into the statistical subject we know today as *regression*.

Regression is a core pillar of statistics, and the statistical theory of regression is a theory about models and methods for relating observables, for example, the coordinates on the path of a comet. The general idea of relating observables through a combination of a systematic and an unsystematic component penetrates most areas of science, technology and business today. The observables are typically treated asymmetrically with one of them pointed out as the *outcome* variable and the others as *predictor variables*, and a regression model is a probability model of the outcome given the predictors. It is largely the same type of model that the much younger field of machine learning deals with under the term *supervised learning*.

In a regression model, the systematic component represents a predictable relation between the outcome and the predictors, while the unsystematic component represents an unpredictable residual variation. This unpredictable variation may be due to measurement errors, sampling inaccuracies, lack of information or other mechanisms, whose contribution cannot be captured by the systematic component.

In regression theory, the unsystematic component is treated statistically. That is, even if the residual variation of individual observations is unpredictable, it can be described and modeled in terms of probability distributions. From this perspective, the theory has to deliver regression models and methods for fitting such models to data that: i) adequately capture the systematic component; ii) adequately describe the unsystematic component; iii) adequately answer questions we might have about the relation between observables. Additionally, the theory investigates properties of such regression models and methods, e.g., how do we compare different

methodologies, and how do we choose among models and methods?

1.1 A brief history of regression

We give in this section a historical perspective on regression. Tracking the origin of regression methods and its terminology provides a historical context for the theory developed in this book. It is a complex story, and the introduction will provide a bird's-eye perspective on what this book is about.

1.1.1 Linear regression

Legendre's book (Legendre 1805) on the determination of the orbit of comets was published in 1805 containing the appendix *Sur la Méthode des moindres carrés*. It outlined a method for fitting a linear equation with unknown coefficients to observations. The English translation of the title of the appendix is *On the Method of least squares*, and it is the earliest known appearance of the method in print. Independently of Legendre, Carl F. Gauss (Gauss 1809) discovered the Gaussian distribution and used it to derive the method of least squares from Laplace's principle of inverse probability. Gauss was like Legendre motivated by problems in astronomy—specifically the description and prediction of the motions of planets. The invention of the method of least squares marks a turning point in the history of regression. Earlier work by Pierre-Simon Laplace, and related work by Roger J. Boscovich with applications to geodesy, using the method of least absolute deviation, had been less successful. The method of least absolute deviation leads to a nonlinear estimator that is hard to compute. By contrast, the least squares estimator is a linear estimator that solves the (linear) normal equation, and Gauss presented a useful method for its numerical computation. The generality of the linear model framework considered by Gauss is impressive and corresponds essentially to that treated in Chapter 2 in this book.

In the years after the invention of the method of least squares, Gauss and Laplace derived several theoretical results about the method including sampling properties of the least squares estimator and confidence intervals for the parameters. The

Legendre, Adrien Marie. 1805. *Nouvelles méthodes Pour La détermination Des Orbites Des Comètes*. Firmin Didot.

Gauss, Carl Friedrich. 1809. *Theoria Motus Corporum Coelestium in Sectionibus Conicis Solem Ambientium*. Perthes et Besser, Hamburg.

logical foundation of the method of least squares was, however, confusing at the time. Gauss's initial derivation was—as mentioned above—based on Laplace's principle of inverse probability. Later, Laplace as well as Gauss sought justification of the method in the sampling properties of the resulting estimator. Specifically by investigating if the estimator has a form of minimal estimation error. Gauss showed what is today known as the Gauss-Markov theorem: the least squares estimator has minimal variance among all linear unbiased estimators.

The models that Gauss fitted to data using his method of least squares were not known to him as regression models. That terminology originates from Sir Francis Galton (Galton 1886). In studies of hereditary traits Galton observed an apparent regression (in the sense of a movement) towards a mediocre state in the offspring compared to the parent(s). Originally he considered the size of plant seeds and later the height of human individuals. What Galton had discovered was the phenomenon known today as regression toward the mean. He explained the phenomenon statistically. The trait is partly inherited from the parent and partly a consequence of additional variation. This results in a bivariate distribution showing a co-relation of the trait in parent and offspring. Though Galton correctly described the phenomenon statistically, he wrongly explained the regression phenomenon as being caused by inheritance from the ancestry of the parent in addition to the parent themselves.

To understand the co-relation Galton observed in his hereditary studies, he contributed to the theory of the bivariate normal distribution and introduced the general notion of correlation. The technical details of what is known today as the *Pearson* correlation coefficient were worked out by Karl Pearson, though. Galton, nevertheless, realized the usefulness of regression and correlation in various application areas, and the regression terminology was adapted by others, notably Karl Pearson. Interestingly, the initial developments of regression analysis in British mathematical statistics were independent of the continental literature, and the regression models were, in particular, not framed in the linear model terminology of Gauss.

In the first half of the 20th century, Sir Ronald A. Fisher had an enormous impact on mathematical statistics. He

Galton, Francis. 1886. "Regression Towards Mediocrity in Hereditary Stature." *The Journal of the Anthropological Institute of Great Britain and Ireland* 15: 246–63.

developed Analysis of Variance (ANOVA), introduced the likelihood concept and advocated the general principle of maximum likelihood estimation among other things. The primary application of ANOVA was to designed experiments (Fisher 1935), and though far from Gauss's applications on the modeling of the motion of celestial objects, the ANOVA models fit into the framework of the linear model.

Fisher did not rely on or acknowledge the contributions of his predecessors to any great extent, and he rediscovered known results about the method of least squares. But he also contributed significantly to the theory, for instance by deriving sampling distributions of test statistics, notably the F -test statistic. His presentation and applications of ANOVA models have been highly influential on the current terminology, which is, for instance, reflected in the `anova()` function in R for testing linear hypotheses within the linear model. Subsequent developments with contributions from many people have turned the linear model into a unified and general statistical theory as outlined in one of the first unifying textbooks on the subject by Shayle R. Searle (Searle 1971).

The history of regression is intertwined with the general history of statistics, and for the history before 1935 the work by Anders Hald (Hald 2007) is a valuable source. For the more recent, and much richer, development of nonlinear regression we give a brief outline in the following sections that focus on a few selected original sources.

1.1.2 Beyond linear regression

Regression models, methods and applications proliferated in the second half of the 20th century. Computers played a pivotal role, first of all by easing the solution of nonlinear estimating equations by iterative methods, which paved the way for applications of nonlinear regression models and nonlinear estimation methods. The use of computer simulations and Markov Chain Monte Carlo (MCMC), in particular, also made Bayesian regression modeling a practical possibility. In this brief account of the history, it is impossible to do justice to all the different directions, and we will therefore narrow the focus to regression models treated in this book.

Fisher, R. A. 1935. *The Design of Experiments*. Oliver and Boyd.

Searle, S. R. 1971. *Linear Models*. John Wiley & Sons, Inc., New York-London-Sydney.

Hald, Anders. 2007. *A History of Parametric Statistical Inference from Bernoulli to Fisher, 1713–1935*. Sources and Studies in the History of Mathematics and Physical Sciences. Springer, New York.

Even if linear regression models are flexible, some regression problems do not fit into the linear modeling framework. One of the early applications of a nonlinear regression model was to the modeling of mortality of an organism as a function of the dosage of a toxic agent. Chester I. Bliss (Bliss 1935) proposed a transformation method where dosage is log-transformed and death frequency is transformed by the inverse distribution function for the normal distribution with mean 5 and unit variance. The resulting transformed death frequencies were called probits. The model posited that the transformed data should fall on a straight line, and Bliss suggested using the method of weighted least squares for fitting such a line.

Already one year earlier, Bliss (1934) presented a similar idea with the probits computed from an S-shaped curve inspired by, but not identical to, the distribution function for the normal distribution. His 1935-paper had, however, an important appendix written by Fisher, which showed how to appropriately deal with dosages for which none or all of the organisms survived. In this appendix, Fisher outlined how the method of maximum-likelihood could be used for estimation. In doing so he came up with a correction step in the weighted least squares fit, which is what we today know as the Fisher scoring algorithm.

The probit model introduced by Bliss is one among a number of nonlinear regression models—notably models of counts—that share a similar structure. These regression models were unified by John Nelder and Robert Wedderburn (Nelder and Wedderburn 1972) in the framework they called generalized linear models, which includes the linear model as a special case but extends it in two notable ways: by including a nonlinear *link function* between the predictors and the outcome, and by allowing for a more flexible description of the outcome distribution.

For generalized linear models, the method of least squares is replaced by maximum likelihood estimation. Nelder and Wedderburn showed that the maximum likelihood estimator can be computed iteratively by the Fisher scoring algorithm, where each step in this algorithm is, in fact, the solution of a weighted least squares problem, and they called it the Iterative Weighted Least Squares (IWLS) algorithm. They also paraphrased some terminology and methodology from

Bliss, C. I. 1935. “The Calculation of the Dosage-Mortality Curve.” *Annals of Applied Biology* 22 (1): 134–67.

Bliss, C. I. 1934. “The Method of Probits.” *Science* 79 (2037): 38–39.

Nelder, J. A., and R. W. M. Wedderburn. 1972. “Generalized Linear Models.” *J. R. Stat. Soc. (A)* 135 (3): pp. 370–384.

linear models, in particular the use of analysis of deviance as a generalization of ANOVA.

As a curious detail, Nelder and Wedderburn outlined in their original paper how generalized linear models could form a useful basis for courses in statistics, which would unify the treatment of a number of statistical models that had previously been treated separately. The theory of generalized linear models was further developed in the 70s and 80s, and an authoritative reference on this classical theory is the book by McCullagh and Nelder (1989). In addition to his theoretical contributions, Nelder initially chaired the GLIM Working Party, who developed the statistical software program GLIM (Generalized Linear Interactive Modeling). This program contributed to making generalized linear models applicable in practice, and it was a source of inspiration for the later implementations (Chambers and Hastie 1991) in S and R.

Another type of regression problem that does not fit directly into the linear modeling framework is the modeling of *survival times*. As for generalized linear models there is a question of modeling scale; at which scale is survival time most appropriately associated to other variables? It can be handled similarly as for generalized linear models via a link function, but there is an additional practical problem with observations being right censored.

Survival models and estimation of life tables have a long history, in particular in actuarial science and medical statistics, but a systematic approach to survival regression models was not made until the seminal contributions by Sir David R. Cox (Cox 1972). He introduced the semiparametric proportional hazards model, which has since become the most widely used survival regression model. Cox made two major contributions in his paper. He introduced the partial likelihood, which provided a means for deriving a sensible estimating equation, and he presented approximate sampling distributions of the resulting estimator.

The paper by Cox inspired a rapid development in the theory of survival regression models, which clarified some of his heuristic arguments and extended his model class considerably. Central to this development was the use of methods from the theory of stochastic processes, in particular continuous time martingale theory. Modern survival analysis theory is heavily

McCullagh, P., and J. A. Nelder.
1989. *Generalized Linear Models*.
Chapman & Hall, London.

Chambers, J. M., and T. J. Hastie,
eds. 1991. *Statistical Models in S*.
Chapman and Hall/CRC.

Cox, D. R. 1972. "Regression Models and Life-Tables." *Journal of the Royal Statistical Society. Series B (Methodological)* 34 (2): pp. 187–220.

based on the use of stochastic processes, and an authoritative account of this theory is the by now classical book *Statistical Models Based on Counting Processes* (Andersen et al. 1993).

On the practical side, the estimating equation derived from Cox’s partial likelihood in the semiparametric framework is nonlinear—and so are parametric likelihood based estimating equations. The estimator thus has to be computed by iterative methods. As a curious historical detail, practical computations could be carried out using the framework of generalized linear models, and the GLIM program could be used to fit the proportional hazards models in Cox’s semiparametric framework as well as for several parametric models. This made these models and methods applicable early on, but specialized software implementations were later developed, such as the survival package for Splus by Terry Therneau (Therneau and Grambsch 2000). The package was ported to R, and it is one of the classical core packages for survival analysis using R.

Andersen, Per Kragh, Ørnulf Borgan, Richard D. Gill, and Niels Keiding. 1993. *Statistical Models Based on Counting Processes*. Springer Series in Statistics. Springer-Verlag.

Therneau, Terry M., and Patricia M. Grambsch. 2000. *Modeling Survival Data: Extending the Cox Model*. Springer, New York.

1.1.3 Regression and machine learning

There were notable developments outside of mainstream statistics related to regression models, but little interaction between the different scientific communities. Frank Rosenblatt (Rosenblatt 1962) explored perceptrons and artificial neural networks for binary classification from multiple input variables. Support vector machines (Cortes and Vapnik 1995) were developed for similar purposes, with the conceptual framework established in 1965 by Vladimir Vapnik.

Rosenblatt, Frank. 1962. *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. Spartan Books.

Artificial neural networks and support vector machines are today core subjects in the field of machine learning. Both can be understood as regression models of a binary outcome variable, and both can be altered to model a continuous outcome variable as well. The models are typically presented as *classification* models rather than probabilistic models, and machine learning theory and practice focus on classification performance over interpretable relations between the predictors and the outcome. Statistics is historically connected to other scientific fields, where explanations, interpretations and answers to inferential questions are central. That is, questions about the nature and magnitudes of the relations between any specific predictor and the outcome. Inferential questions

Cortes, Corinna, and Vladimir Vapnik. 1995. “Support-Vector Networks.” *Machine Learning* 20 (3): 273–97.

have therefore driven the theoretical and methodological development in statistics, while classification performance has driven the development in machine learning.

In 2001 Leo Breiman (Breiman 2001) divided the data modeling communities into “two cultures”: those using stochastic (or probabilistic) data models and those using algorithmic models. Breiman ascribed the developments of algorithmic modeling largely to the machine learning community, and he strongly criticized the conventional statistical community for their (simplistic) probabilistic data modeling. Undoubtedly, contemporary regression modeling has benefited greatly from developments in machine learning, and the “two cultures” are linked much closer today than ever before.

The development of lasso (Tibshirani 1996) is an interesting early example. It was first developed for the linear regression model, but the lasso estimator is nonlinear and biased—as opposed to the least squares estimator. It required specialized algorithms for its computation, but it turned out to have certain computational and statistical benefits that made it possible to tackle large scale data analysis in new and useful ways.

Not long after the introduction of lasso, in 2001, the first edition of *The Elements of Statistical Learning* (Hastie et al. 2009) was published. It is hardly a coincidence that this was the same year as Breiman’s paper was published. The book had a notable impact on the statistics community by bridging the gap between the machine learning literature and the statistics literature. Along with lasso, the book introduced a range of nonlinear regression models and optimization methods to a broad statistical audience. This was a timely publication and in combination with new applications to large scale data modeling it spurred a substantial development of new regression modeling methods and exciting challenges.

1.1.4 Computation

In the long history of regression, one particularly influential development on how we work with regression models today is the growth of *computational resources*.

Breiman, Leo. 2001. “Statistical Modeling: The Two Cultures.” *Statistical Science* 16 (3): 199–231.

Tibshirani, Robert. 1996. “Regression Shrinkage and Selection via the Lasso.” *Journal of the Royal Statistical Society. Series B* 58 (1): 267–88.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. 2009. *The Elements of Statistical Learning*. 2nd ed. Springer Series in Statistics. Springer.

Data analysis with regression models has been completely transformed by the increase of computer power and the developments of statistical programming environments such as R. The computer serves all data analysts, and interactive and exploratory data analysis in the spirit of Tukey (1977) is now routine. Nonparametric and resampling based statistical methods, that require substantial computations, have also made data analysis possible in situations where we lack theoretical methods or where we want to impose minimal modeling assumptions.

Tukey, John. 1977. *Exploratory Data Analysis*. Pearson.

The computer has, in addition, become the lab bench of statistics, and it has brought an experimental component to statistics. Most methodologies and theoretical results are today scrutinized via simulation studies, which can inform us about the applicability and limitations of the methods and the theory.

When the history of regression is viewed from our vantage point, with the computational resources we have available today, some of the theory and methodologies may appear outdated and irrelevant. And some surely are! Nobody should use least squares methods because solving linear equation systems is possible by hand, and nobody should fit generalized linear models using the arcane syntax and model specification of GLIM. However, important residues of history are alive and central to regression today because they represent solid and well understood methodology, and these residues function as the foundation we build on.

1.1.5 The four components to regression modeling

The history of regression contains four recurrent components.

1. Models
2. Estimation methodology
3. Sampling properties
4. Model skepticism and critique

The first component—the model development—was, and is, driven by applications, and the second component was developed to compute estimates of model parameters from data, that is, to fit models to data. Over the course of history, a

supply of models, such as the linear and generalized linear models, have been developed together with methods or principles for parameter estimation, such as the methods of least squares or maximum likelihood.

The third component—the sampling properties—was developed to understand theoretical properties of the various estimation methodologies and to provide (frequentistic) quantification of uncertainty.

The developments of components one through three are all good and well, but from Gauss to Cox there has been grave concerns about whether the model assumptions that supported the theory could be justified for particular applications. Techniques of more or less formal character have been developed and used throughout history for validating if model assumptions are plausible. Though such techniques will never provide certainty, they serve as important sanity checks that keep models and methodology aligned with data.

The four components: models; estimation methodology; sampling properties; and model critique, are recurring components in this book as they are in history. The intention of this book is to teach the reader these four cornerstones of statistics in the framework of regression modeling.

1.2 What this book is—and what it is not

This is a textbook for a graduate-level course in regression modeling and regression theory. The purpose of the book is to make the reader develop their own independent understanding of what regression models are, what they are capable of, and how we can fit them to data and work with them to answer scientific questions. There exists today a very broad range of such models and related methodology, and this book had to be selective.

This book is therefore *not*

- a survey of regression models and methods
- a cookbook for regression modeling in practice
- an encyclopedia for answering all questions related to regression modeling
- a guide to state-of-the-art regression modeling

In fact, most models treated in this book are somewhat classical and not state of the art, though they are widely used in practice. It is a deliberate choice to use relatively simple and classical models as illustrative examples of general theory and methodology—even if new applications may need to look beyond this book for state-of-the-art solutions.

This book is not a *normative* prescription either of how a data analyst should analyze data using regression modeling techniques. The book presents a development of concrete model classes and practically usable methodologies, and a description of properties and justifications of various regression analysis techniques. The author is concerned with accurately developing the theory and methodologies in sufficient detail for the reader to completely absorb them. The author is certainly also concerned with developing good habits around practical data analysis but less convinced about having the definitive say on what the reader *should* do.

To be concrete, the book primarily covers the following three regression model classes:

- linear models
- generalized linear models
- parametric and proportional hazards regression models

These are classical subjects and an obligatory part of an education in statistics. At the same time they serve as good examples when developing general statistical principles and methodologies such as estimation methodology, likelihood based methods as well as methods for interval estimation and statistical tests.

The practical value of the three classical model classes is boosted substantially when combined with two techniques the book also covers:

- basis (or feature) expansions
- regularized estimation

The three primary questions we seek to answer with regression models are:

- quantifications of conditional associations
- tests of conditional independencies
- predictions of outcomes given the value of the predictors

The first question deals with quantification of the associational strength between the outcome and one predictor given the other predictors. The second question deals with formally testing the hypothesis that there is no association between the outcome and one predictor given the other predictors. These two questions are inferential and ask about specific properties of an unknown conditional distribution. The last question is directly asking for a (good) predictive model.

The book deals with the questions above exclusively in terms of *associational* regression modeling, where the regression model relates the distribution of the *outcome* variable to one or more *predictor variables* in terms of a full or partial specification of the conditional distribution of the outcome given the predictors. Importantly, associational regression models do *not* necessarily capture *causal* relations, that is, relations that persist under interventions on or manipulations of the observables.

Associational regression models do play a role when inferring causal relations, but we refer to the substantial literature on causal inference for a discussion of the *untestable structural assumptions* that we need to make to be able to draw causal inference. A thorough understanding of the subject matter field, or knowledge from other studies, might allow us to make such necessary structural assumptions, but they are typically untestable using only the dataset we have at hand for the regression analysis.

Associational models are, however, directly applicable for predicting outcomes given observations of the predictors, which is used for medical diagnosis and prognosis, in businesses to predict customer behavior, to predict risk in insurance companies, pension funds and banks, to make weather forecasts and in many other areas where it is of interest to know what we cannot (yet) observe.

It is fairly clear what makes a model good for prediction. Given a method for quantification of predictive accuracy, the best model is the most accurate model. The accuracy of a prediction is typically quantified by a *loss* function, which actually quantifies how *inaccurate* a prediction is. The smaller the loss is the more accurate is the prediction. The expected loss quantifies how accurate the predictive model is on average. Predictive performance is something we can measure, and the

book develops general methodology for quantification of a model's predictive strength.

It is slightly more tricky to answer the inferential questions, partly because methods for doing so generally rely on some level of correctness of the regression model, that is, does the model fit the data? The two final points, that we emphasize in this introduction as part of the book, are:

- model diagnostics
- model-robust inference

Diagnostics are post-model fitting procedures for discovering ways that data violate model assumptions. If the model does not fit we cannot trust inferential conclusions based on model assumptions. The two ways around such problems are: either alter the model so that it fits the data; or rely on inferential procedures that are robust to model misspecifications. Both perspectives will be developed in the book.

1.3 Organization

The book consists of a combination of theory and larger case studies. The theory chapters focus on presenting the mathematical framework for regression modeling together with the relevant mathematical theory such as derivations of likelihood functions, estimators, estimation algorithms and properties of estimators. Other chapters focus on more general methodological questions. This includes developing methods for dealing with the complex decision process that practical data analysis is.

While smaller data examples are presented in the theory chapters, the larger case studies are developed to better illustrate how theory supports applications. The hope is that this will ease the transition from theory to practice. The price to pay is that there are some distractions due to real world data challenges. Data rarely behaves well. There are missing values and outliers, the models do not fit the data perfectly, the data comes with a strange encoding of variables and many other issues. Issues that require decisions to be made and issues on which many textbooks on statistical theory are silent.

By working through the case studies in detail it is the hope that many relevant practical problems are illustrated and appropriate solutions are given in such a way that the reader is better prepared to turn the theory into applications on their own.

1.4 Using R

We use the programming language R¹ throughout to illustrate how good modeling strategies can be carried out in practice on real data, but the book will not provide an introduction to the R language. Consult the R manuals (R Core Team 2025+) or the many textbooks on R programming and data analysis (Wickham 2019; Wickham et al. 2023). The classical and influential book *Modern Applied Statistics with S* (Venables and Ripley 2002) and the corresponding MASS package (comes with the R distribution) should also be mentioned. Several classical statistical models and methods are supported by the MASS package and documented in that book.

The case studies in this book are complete with R code that covers all aspects of the analysis. They represent an integration of data analysis with documentation and reporting. This is an adaptation to data analysis of *literate programming* (Knuth 1984). The main idea is that the writing of a report that documents the data analysis and the actual data analysis are merged into one document. This supports the creation of *reproducible analysis*, which is a prerequisite for reproducible research. To achieve this integration the book was written using the Quarto² publishing system and the R package knitr³. Quarto and knitr allows you to integrate text with chunks of R code and produce an output document in various formats, e.g., a webpage or a pdf document, that include text, code and output from evaluating the code. The use of the RStudio⁴ or Positron⁵ integrated development environments (IDE) is also recommended. Both are developed by Posit PBC as open source projects and the desktop versions are free to use.

The book source is available from the [RwR GitHub repository](#), which also contains information on how to recreate the R environment used for writing the book and how to obtain the RwR package containing the datasets used.

¹ <https://www.r-project.org/>

R Core Team. 2025+. *R: A Language and Environment for Statistical Computing*. cran.r-project.org/manuals.html.

Wickham, Hadley. 2019. *Advanced R*. 2nd ed. The R Series. Chapman & Hall/CRC. adv-r.hadley.nz.

Wickham, Hadley, Mine Çetinkaya-Rundel, and Garrett Grolemund. 2023. 2nd ed. O'Reilly. r4ds.hadley.nz.

Venables, W. N., and B. D. Ripley. 2002. *Modern Applied Statistics with S*. 4th ed. Springer.

Knuth, D. E. 1984. "Literate Programming." *The Computer Journal* 27 (2): 97–111.

² <https://quarto.org/>

³ <https://yihui.name/knitr/>

⁴ <https://posit.co/download/rstudio-desktop/>

⁵ <https://positron.posit.co/>

2 The linear model

The linear model is the classical workhorse of regression modeling. It is easy to interpret and work with in practice and it has tractable theoretical properties. It may not be the optimal model choice in any given situation, but it is a good reference model whenever more sophisticated models are considered, and in combination with nonlinear variable transformations and basis expansions, as considered in Chapter 7, it can be surprisingly flexible.

2.1 Outline and prerequisites

This chapter introduces the linear model as well as aspects of the general regression modeling framework and terminology used throughout the book. An example on fire insurance claim size modeling is introduced in Section 2.2 and used throughout the chapter to illustrate the theory. Section 2.5 covers how the linear model is fitted to data, and Section 2.6 gives results on the sampling distributions of parameter estimators and test statistics that are used in relation to the linear model. Section 2.7 treats model assessment and comparison using the claim size models to illustrate the theory.

The chapter primarily relies on the following four R packages.

```
library(RwR)
library(ggplot2)
library(tibble)
library(broom)
library(splines)
```

A few functions from additional packages are used, but in those cases the package is made explicit via the `::` operator.

2.2 Claim sizes

In this section we take a first look at a dataset from a Danish insurance company with the purpose of studying the relation between the insurance sum and the claim size for commercial fire claims. The data is included as the `claims` dataset in the `RwR` package.

```
data(claims, package = "RwR")
```

We will use this dataset to build simple linear regression models that can predict potential claim sizes given knowledge of the insurance sums. More refined predictions as well as the possibility to differentiate policy prices can also be achieved by breaking down such a relation according to other variables.

```
tibble(claims) |> head()
```

```
# A tibble: 6 x 3
  claims      sum grp
  <dbl>    <dbl> <fct>
1 3853. 34570294. 1
2 1194.  7776469. 1
3 3917. 26343305. 1
4 1259.  5502915. 1
5 11594. 711453346. 1
6 33535. 716609368. 1
```

To inspect the data we can print out a small part of the dataset using the `head()` function. There is a similar `tail()` function that prints the last part of the dataset.

The first column printed shows the row names (in this case just the row numbers). The other columns show the values of the variables in the dataset. In this case the dataset contains the following three variables:

- `claims`: the claim size in Danish kroners (DKK)
- `sum`: the insurance sum in Danish kroners (DKK)
- `grp`: the trade group of the business

The `claims` and `sum` variables are numeric, while the `grp` variable is a categorical trade group. The trade group is represented by integers but encoded and stored as a factor. The trade group encoding is:

- 1 = residences
- 2 = ordinary industry
- 3 = hotel and restaurant
- 4 = production industry

The marginal distributions of claim size and insurance sum are both highly skewed and span several orders of magnitude—as is revealed by the histograms in Figure 2.1. For this reason we will log-transform both variables in the subsequent figures. In this section this is purely for the purpose of better visualization. In later sections we will consider regression models for the log-transformed variables as well. Skewness of the marginal distributions does not automatically imply that the variables should be log-transformed for the regression model, and we will return to this discussion later.

Throughout we will avoid constructing new variables in the dataset that represent transformations of other variables. We will instead use the functionalities in R for changing the scale in a plot or for transforming one or more variables directly when fitting a model. The scatter plot in Figure 2.2 of claim size against insurance sum on a log-log-scale was constructed using `ggplot()` from the R package `ggplot2`.

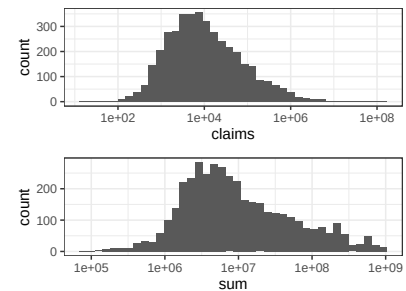


Figure 2.1: Histograms of claim size and insurance sum on log-scale.

```
p0 <- ggplot(
  data = claims,
  aes(sum, claims),
  alpha = I(0.2)
) +
  scale_x_log10(
    "Insurance sum (DKK)",
    breaks = 10^c(6, 7, 8),
    labels = c("1M", "10M", "100M")
  ) +
  scale_y_log10(
    "Claim size (DKK)",
    breaks = 10^c(2, 4, 6, 8),
    labels = c("100", "10K", "1M", "100M")
  ) +
  geom_point()
```

```
Warning in fortify(data, ...): Arguments in `...` must be used.
x Problematic argument:
* alpha = I(0.2)
i Did you misspell an argument name?
```

```
p <- p0 +
  geom_smooth(
    method = "lm",
    linewidth = 1,
    se = FALSE
  )
p
```

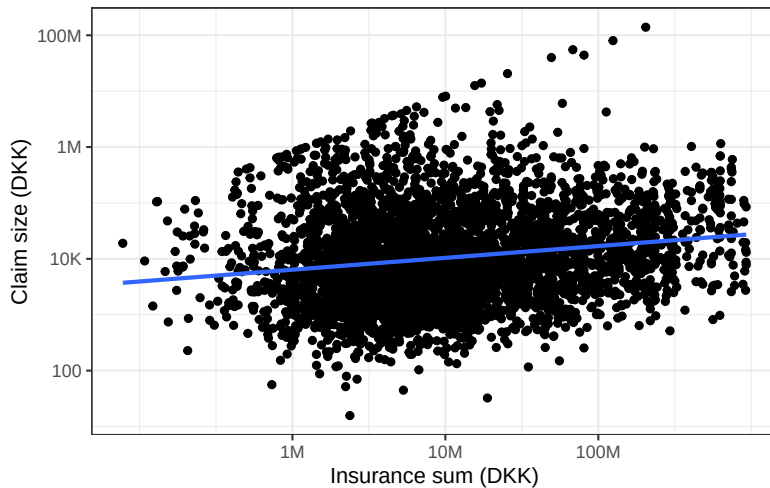


Figure 2.2: Scatter plot of claim size against insurance sum. Note the log-log axes.

The code used to generate the plot shows that the plot is first stored as an R object named `p` and then plotted by “printing” `p` (the line just containing `p`). This is useful for later reuse of the same plot with some modifications as will be illustrated below. Note that the regression line was added to the plot using the `geom_smooth()` function with `method = "lm"`.

From Figure 2.2 it appears that there is a positive correlation between the insurance sum and the claim size—the larger the insurance sum is, the larger is the claim size. Note the “line-up” of the extreme claim sizes, which (approximately) fall on the line with slope 1 and intercept 0. These claims are capped at the insurance sum, and we should pay some attention to the effect of these extremes in subsequent modeling steps.

The questions that we will focus on in this chapter are whether the trade group should be included into the model and whether a linear relation (on the log-log-scale) between the claim size and insurance sum is appropriate. As a first attempt to address the first question we stratify the scatter plot according to the trade group.

```
p + facet_wrap(~ grp, ncol = 4)
```

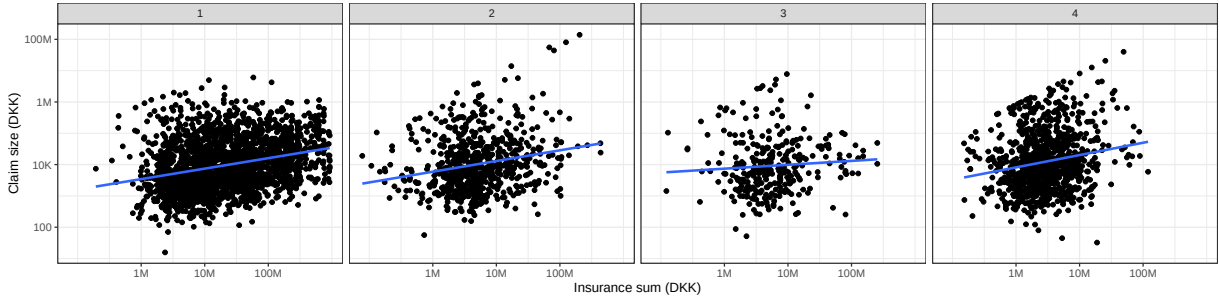


Figure 2.3: Scatter plots of claim size against insurance sum stratified according to trade group. A regression line is added to the plots for each group.

Figure 2.3 shows that there might be some differences, but it is difficult to tell from the figure alone if these differences are due to random variation or if they actually represent real differences between the groups.

The following sections will introduce the mathematical framework for fitting linear regression models and for investigating the models. Based on this theory we can tell if the model is an adequate model of the data and answer the questions put forward.

2.3 The linear model assumptions

This section briefly introduces the linear model and the typical assumptions made. This settles notation for the case study in the following section. In a first reading it can be read quickly and returned to later to better digest the model assumptions and their implications.

The linear model relates a numerical *outcome* variable Y to a p -dimensional vector X of *predictors* via the identity

$$\mathbf{E}(Y \mid X) = X^T \beta. \quad (2.1)$$

Here

$$X^T \beta = X_1 \beta_1 + \dots + X_p \beta_p$$

is a linear combination of the predictors weighted by the β -parameters and $\mathbf{E}(Y \mid X)$ denotes the conditional expectation

The entries of X go by many other names: explanatory variables, covariates, independent variables, regressors, inputs or features. Similarly, Y is also called the outcome or dependent variable.

of Y given X . Thus the linear⁶ model is a model of the conditional expectation of the outcome variable given the predictors.

An *intercept* parameter, β_0 , is often added,

$$\mathbf{E}(Y \mid X) = \beta_0 + X^T \beta.$$

It is notationally convenient to assume that the intercept parameter is included among the other parameters. This can be achieved by joining the predictor $X_0 = 1$ to X , thereby increasing the dimension to $p + 1$. In the general presentation we will not pay particular attention to the intercept. We will assume that if an intercept is needed, it is appropriately included among the other parameters, and we will index the predictors from 1 to p . Other choices of index set, e.g., from 0 to p , may be convenient in specific cases.

In addition to the fundamental linearity expressed by Equation 2.1 we will for various reasons need additional model assumptions. For later reference we collect three standard model assumptions here.

A1. The conditional expectation of Y given X is

$$\mathbf{E}(Y \mid X) = X^T \beta.$$

A2. The conditional variance of Y given X does not depend upon X ,

$$\mathbf{V}(Y \mid X) = \sigma^2.$$

A3. The conditional distribution of Y given X is a normal distribution,

$$Y \mid X \sim \mathcal{N}(X^T \beta, \sigma^2).$$

It is, of course, implicit in **A1** and **A2** that Y has finite expectation and variance, respectively. While **A1** explicitly specifies the modeling assumption about the conditional mean, the purpose of **A2** is less obvious. A good reason to want this assumption to hold, is to ensure that the predictors primarily affect the outcome distribution via a location shift, and that the spread of the distribution is unaffected by the predictors. This makes interpretations of the model easier. It also makes assessment of the accuracy of model predictions

⁶ The linearity that matters for estimation and statistical inference is the linearity in the unknown parameter vector β . It is not linearity in the predictor variables.

assumption **A2** is sometimes called *homoskedasticity*, which is derived from the Greek words “homo” (same) and “skedastios” (dispersion). A violation of **A2** is *heteroskedasticity*. Less pretentiously, we may call **A2** *variance homogeneity* and refer to violations of **A2** as *variance inhomogeneity*.

easier. Assumption **A2** has, moreover, technical consequences for the statistical analysis of data, which we return to in subsequent parts of the book.

Assumption **A3** is a strong distributional assumption, which entails **A1** and **A2** but is unnecessarily restrictive for many purposes. Often **A1** and **A2** are sufficient, but whether **A3** holds can be important when using the model to make probabilistic predictions, e.g., when computing prediction intervals. It is, in addition, only under assumption **A3** that an exact statistical theory can be developed for finite sample sizes. Some results used in practice are formally derived under this assumption, and they must be regarded as approximations when **A3** is violated.

There exists a bewildering amount of terminology related to the linear model in the literature. Notation and terminology has been developed differently for different submodels of the linear model. If the X -vector only represents continuous variables, the model is often referred to as the linear *regression* model. Since any categorical variable can be encoded via k binary dummy variables (also known as a one-hot encoding), the linear model includes all ANalysis Of VAriance (ANOVA) models. Combinations, known in parts of the literature as ANalysis of COVAriance (ANCOVA), are of course also possible.

Treating each special case separately and using special terminology is unnecessary and confusing—and most likely a consequence of historically different needs in different areas of applications. We give a unified treatment of the linear model. It is a fairly simple model with a rather complete theoretical basis. That said, many modeling questions still have to be settled for any specific real data analysis, which makes applications of even the simple linear model non-trivial.

For statistical applications we have a sample of n observations, and to develop the statistical theory we need additional distributional assumptions regarding their joint distribution. Letting $Y_1, \dots, Y_n$ denote the n observations of the outcome with corresponding predictors $X_1, \dots, X_n$, we collect the outcomes into a column vector $\mathbf{Y}$, and we collect the predictors into an $n \times p$ matrix $\mathbf{X}$ called the *model matrix*. The i -th row of $\mathbf{X}$ is X_i^T . The additional assumptions are:

Dummy variable encoding: A categorical variable on k levels is encoded as a k -dimensional binary vector, whose j -th dummy variable equals 1 if the value of the categorical variable equals the j -th level and 0 otherwise.

The outcome vector and model matrix:

$$\mathbf{Y} = \begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{pmatrix} \quad \mathbf{X} = \begin{pmatrix} X_1^T \\ X_2^T \\ \vdots \\ X_n^T \end{pmatrix}$$

A4. The conditional distribution of Y_i given $\mathbf{X}$ depends upon X_i only, and Y_i and Y_j are conditionally *uncorrelated* given $\mathbf{X}$,

$$\text{cov}(Y_i, Y_j \mid \mathbf{X}) = 0.$$

A5. The conditional distribution of Y_i given $\mathbf{X}$ depends upon X_i only, and $Y_1, \dots, Y_n$ are conditionally *independent* given $\mathbf{X}$.

A6. The pairs $(X_1, Y_1), \dots, (X_n, Y_n)$ are i.i.d.

There is a hierarchy among these three assumptions with **A6** implying **A5** and **A5** implying **A4**. Thus **A4** is the weakest distributional assumption about the joint distribution of the observations.

Assumption **A4** implies together with **A1** and **A2** that

$$\mathbf{E}(\mathbf{Y} \mid \mathbf{X}) = \mathbf{X}\beta, \quad (2.2)$$

and that

$$\mathbf{V}(\mathbf{Y} \mid \mathbf{X}) = \sigma^2 \mathbf{I} \quad (2.3)$$

where $\mathbf{I}$ denotes the $n \times n$ identity matrix.

Assumptions **A5** and **A3** imply that

$$\mathbf{Y} \mid \mathbf{X} \sim \mathcal{N}(\mathbf{X}\beta, \sigma^2 \mathbf{I}). \quad (2.4)$$

In summary, there are essentially two sets of distributional assumptions:

- the weak set **A1**, **A2** and **A4**, which imply the moment identities (2.2) and (2.3)
- the strong set **A3** and **A5**, which additionally imply the distributional identity (2.4).

Occasionally, **A6** combined with **A1** and **A2** is used to justify certain resampling based procedures, such as bootstrapping, or to give asymptotic justifications of inference methodologies.

The linear regression model and the different assumptions related to it can also be phrased in terms of an *additive error*, which is defined by

$$\varepsilon := Y - X^T \beta.$$

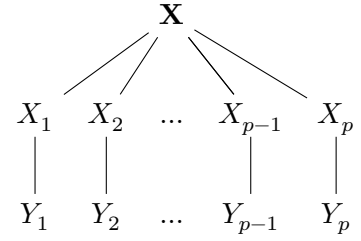


Figure 2.4: Graphical illustration of the assumptions on the joint distribution. The Y_i depends on $\mathbf{X}$ through X_i only, and the Y_i -s are independent given $\mathbf{X}$.

Definition of the *additive error*.

We can then write the outcome in terms of the predictors and the additive error as

$$Y = X^T\beta + \varepsilon.$$

The error term ε is sometimes also called the *noise* term. In terms of ε the model assumptions can be expressed as:

- **A1** is equivalent to $\mathbf{E}(\varepsilon \mid X) = 0$
- **A2** is equivalent to $\mathbf{V}(\varepsilon \mid X) = \sigma^2$
- **A3** is equivalent to $\varepsilon \mid X \sim \mathcal{N}(0, \sigma^2)$.

Note that under assumption **A3**, the conditional distribution of ε given X does not depend upon X , thus **A3** implies that ε and X are independent.

It is always a challenge in practice to justify model assumptions, and we need to accept that assumptions cannot generally be justified before the data analysis. There are no magic arguments or simple statistical summaries that imply that the assumptions are fulfilled. A histogram of the marginal distribution of the outcomes can, for instance, not be used as an argument for or against assumption **A3** on the conditional normal distribution of Y . Justifications and investigations of model assumptions are generally done *after* a model has been fitted to data, and this is called *model diagnostics*.

2.4 Linear models of claim sizes

Returning to the insurance claim sizes, a regression model of log claim size, $\log(Y_i)$, for insured i using log insurance sum, $\log(X_{i,\text{sum}})$, as predictor can be fitted to data using the R function `lm()`. The theory behind the computations is treated in the subsequent section.

```
claims_lm <- lm(
  log(claims) ~ log(sum),
  data = claims
)
coefficients(claims_lm)
```

```
(Intercept)    log(sum)
 5.8410060    0.2114886
```

This model is the linear model

$$\mathbf{E}(\log(Y_i) \mid X_{i,\text{sum}}) = \beta_0 + \beta_{\text{sum}} \log(X_{i,\text{sum}}).$$

In the R terminology the parameter β_0 is the `(Intercept)` coefficient and β_{sum} is the `log(sum)` coefficient. The model corresponds to the regression line shown in Figure 2.2.

Note that in concrete cases we will not write the conditioning variable explicitly as $\log(X_{i,\text{sum}})$. As long as the right hand side is a linear combination of transformations of the conditioning variables (and an intercept) this is a linear model. It would be bothersome to write out explicitly the vector of transformations on the left hand side. These are, however, computed explicitly by `lm` and stored in the model matrix $\mathbf{X}$.

```
model.matrix(claims_lm)[781:784, ]
```

	(Intercept)	log(sum)
781	1	19.42274
782	1	14.63276
783	1	14.90718
784	1	15.24711

In the general treatment of the linear model it is notationally beneficial to always condition explicitly on the model matrix or a row in the model matrix. Thus to have a smooth notation in the general theory as well as in the concrete examples we will have to live with this slight discrepancy in notation. This will usually not be a source of confusion once the nature of this discrepancy is understood clearly. The models treated below illustrate this point further.

We can condition on the trade group, grp_i , as well and thus consider the model

$$\begin{aligned} \mathbf{E}(\log(Y_i) \mid X_{i,\text{sum}}, \text{grp}_i) &= \beta_0 + \beta_{\text{sum}} \log(X_{i,\text{sum}}) \\ &+ \beta_{\text{grp2}} X_{i,\text{grp2}} + \beta_{\text{grp3}} X_{i,\text{grp3}} + \beta_{\text{grp4}} X_{i,\text{grp4}}. \end{aligned}$$

This model is known as the additive model. The dummy variable encoding of the categorical trade group is used, and since the model includes an intercept the dummy variable corresponding to the first group is removed. Otherwise the model would be overparametrized. The consequence is that

The rows in the model matrix shown were chosen to illustrate certain differences in subsequent examples.

The dummy variable encoding is

$$X_{i,\text{grpk}} = 1(\text{grp}_i = k).$$

the first group acts as a baseline, and that the parameters β_{grp2} , β_{grp3} and β_{grp4} are the differences from the baseline. These differences are also known as *contrasts*. We see, for example, that

$$\mathbf{E}(\log(Y_i) \mid X_{i,\text{sum}} = x, \text{grp}_i = 1) = \beta_0 + \beta_{\text{sum}} \log(x)$$

and

$$\mathbf{E}(\log(Y_i) \mid X_{i,\text{sum}} = x, \text{grp}_i = 2) = \beta_0 + \beta_{\text{sum}} \log(x) + \beta_{\text{grp2}},$$

thus the difference from the baseline trade group 1 to trade group 2 is β_{grp2} .

We can again fit this model using `lm()` in R, which automatically determines the model parametrization (using defaults) and computes the model matrix.

```
claims_lm_add <- lm(
  log(claims) ~ log(sum) + grp,
  data = claims
)
coefficients(claims_lm_add)
```

(Intercept)	log(sum)	grp2	grp3	grp4
3.5974043	0.3300459	0.5472730	0.4013465	0.9143058

```
model.matrix(claims_lm_add)[781:784, ]
```

	(Intercept)	log(sum)	grp2	grp3	grp4
781	1	19.42274	0	0	0
782	1	14.63276	1	0	0
783	1	14.90718	0	0	1
784	1	15.24711	1	0	0

Alternative parametrizations can be specified using the `contrasts` argument.

```
claims_lm_add <- lm(
  log(claims) ~ log(sum) + grp,
  data = claims,
  contrasts = list(grp = "contr.sum")
)
coefficients(claims_lm_add)
```

(Intercept)	log(sum)	grp1	grp2	grp3
4.06313567	0.33004591	-0.46573135	0.08154169	-0.06438483

```
model.matrix(claims_lm_add)[781:784, ]
```

	(Intercept)	log(sum)	grp1	grp2	grp3
781	1	19.42274	1	0	0
782	1	14.63276	0	1	0
783	1	14.90718	-1	-1	-1
784	1	15.24711	0	1	0

The `contr.sum` contrast gives the parametrization

$$\begin{aligned} \mathbf{E}(\log(Y_i) \mid X_{i,\text{sum}}, \mathbf{grp}_i) &= \beta_0 + \beta_{\text{sum}} \log(X_{i,\text{sum}}) \\ &\quad + \beta_{\text{grp1}} X_{i,\text{grp1}} + \beta_{\text{grp2}} X_{i,\text{grp2}} \\ &\quad + \beta_{\text{grp3}} X_{i,\text{grp3}} + \beta_{\text{grp4}} X_{i,\text{grp4}}. \end{aligned}$$

with the constraint $\beta_{\text{grp1}} + \beta_{\text{grp2}} + \beta_{\text{grp3}} + \beta_{\text{grp4}} = 0$. Only the parameters β_{grp1} , β_{grp2} and β_{grp3} are reported, and $\beta_{\text{grp4}} = -\beta_{\text{grp1}} - \beta_{\text{grp2}} - \beta_{\text{grp3}}$ as can also be seen from the model matrix. This latter parametrization can be more convenient if we want to compare trade groups to the overall intercept instead of to one specific baseline group.

The additive model gives a different intercept for each trade group but the slope is kept the same for all trade groups. Within the framework of linear models we can capture differences in slope as an *interaction* between the insurance sum and the trade group. Using the parametrization with trade group 1 as baseline this model is

$$\begin{aligned} \mathbf{E}(\log(Y_i) \mid X_{i,\text{sum}}, \mathbf{grp}_i) &= \beta_0 + \beta_{\text{sum}} \log(X_{i,\text{sum}}) \\ &\quad + (\beta_{\text{grp2}} + \beta_{\text{sum,grp2}} \log(X_{i,\text{sum}})) X_{i,\text{grp2}} \\ &\quad + (\beta_{\text{grp3}} + \beta_{\text{sum,grp3}} \log(X_{i,\text{sum}})) X_{i,\text{grp3}} \\ &\quad + (\beta_{\text{grp4}} + \beta_{\text{sum,grp4}} \log(X_{i,\text{sum}})) X_{i,\text{grp4}}. \end{aligned}$$

The parameter estimates and model matrix are:

This model corresponds to the regression lines shown in Figure 2.3. This example illustrates why we do not want to write out the eight-dimensional vector explicitly in the condition when in fact it is just a transformation of $X_{i,\text{sum}}$ and $\mathbf{grp}_i$. On the other hand, it would be equally bothersome to drag an abstract transformation around in the general notation.

Three linear models were considered in this section, and subsequent sections and chapters will treat different ways of

```
coefficients(claims_lm_int) |> print(width = 120)
```

```
(Intercept)      log(sum)      grp2      grp3      grp4 log(sum):grp2 log(sum):grp3 log(sum):grp4
3.525657166    0.334288387    0.434262343    3.666489986    0.040781595    0.007719206    -
0.209986484    0.059341891
```

```
model.matrix(claims_lm_int)[781:784, ] |> print(width = 120)
```

```
(Intercept) log(sum) grp2 grp3 grp4 log(sum):grp2 log(sum):grp3 log(sum):grp4
781          1 19.42274    0    0    0         0.00000          0         0.00000
782          1 14.63276    1    0    0         14.63276          0         0.00000
783          1 14.90718    0    0    1          0.00000          0        14.90718
784          1 15.24711    1    0    0         15.24711          0         0.00000
```

comparing models and for drawing inference about model parameters and hypotheses. It must be emphasized that the purpose of developing the theory and the corresponding statistical methods is **not** to be able to deduce from data what the “true model” is. There are several reasons for that. For once, all models should be considered as approximations only.

We can try to find an adequate model approximation that captures essential aspects of the data, and we may be able to justify that a model is adequate, but we will never be able to justify that a model is “true”.

All models are wrong but some are useful. George Box, 1978

Several models may also be valid and yet appear contradictory, and it may then depend on the purpose of the modeling which to use. It is, for instance, important to formulate precisely which predictors we condition on. Suppose that the conditional expectation $\mathbf{E}(\log(Y_i) \mid X_{i,\text{sum}}, \text{grp}_i)$ is given by the additive model, say, then it follows from the tower property of conditional expectations that

$$\begin{aligned} \mathbf{E}(\log(Y_i) \mid X_{i,\text{sum}}) &= \beta_0 + \beta_{\text{sum}} \log(X_{i,\text{sum}}) \\ &+ \beta_{\text{grp2}} \mathbf{P}(\text{grp}_i = 2 \mid X_{i,\text{sum}}) \\ &+ \beta_{\text{grp3}} \mathbf{P}(\text{grp}_i = 3 \mid X_{i,\text{sum}}) \\ &+ \beta_{\text{grp4}} \mathbf{P}(\text{grp}_i = 4 \mid X_{i,\text{sum}}). \end{aligned}$$

Now if

$$\mathbf{P}(\text{grp}_i = k \mid X_{i,\text{sum}}) \approx \delta_k + \gamma_k \log(X_{i,\text{sum}})$$

for $k = 2, 3, 4$ then the linear model

$$\mathbf{E}(\log(Y_i) \mid X_{i,\text{sum}}) = \beta'_0 + \beta'_{\text{sum}} \log(X_{i,\text{sum}})$$

The approximation can be justified locally using Taylor’s theorem if the probability depends smoothly on the predictor.

will be an approximately valid model as well with

$$\begin{aligned}\beta'_0 &= \beta_0 + \beta_{\text{grp2}}\delta_2 + \beta_{\text{grp3}}\delta_3 + \beta_{\text{grp4}}\delta_4 \\ \beta'_{\text{sum}} &= \beta_{\text{sum}} + \beta_{\text{grp2}}\gamma_2 + \beta_{\text{grp3}}\gamma_3 + \beta_{\text{grp4}}\gamma_4.\end{aligned}$$

That is, even if we do not condition on the group, the model is (approximately) valid, but the intercept and slope parameters for this model may differ from those in the model where we additionally condition on the group. This depends on whether the $\beta_{\text{grp}k}$, δ_k and γ_k parameters are 0. If the $\beta_{\text{grp}k}$ parameters were all 0 it would not matter if group is included in the conditioning. If group and insurance sum were independent then $\gamma_k = 0$, and both models would be equally valid models of the conditional expectation of the log claim size. The slope would then be the same in both models but the intercept would differ. If group and insurance sum were dependent then the slope would depend on whether we condition on insurance sum only or whether we condition on insurance sum as well as group.

The key take-home message is that a predictor variable may be excluded from the model even if it has a non-zero coefficient without necessarily invalidating the model. Another take-home message is that there is no such thing as an unqualified “effect” of a predictor variable. Effect sizes, as quantified by the β coefficients, are always depending upon exactly which variables we condition on, whether this is made explicit by the modeling or is implicit by the data sampling process.

2.5 Estimation theory

The theory that we will cover in this section is on the estimation of the unknown parameter β using least squares methods. We give theoretical results on the existence and uniqueness, and we provide characterizations of the least squares estimator. We also discuss how the estimator is computed in practice.

2.5.1 Weighted linear least squares estimation

We will consider the generalization of linear least squares that among other things allows for weights on the individual cases. Allowing for weights can be of interest in itself, but serves, in

particular, as a preparation for the methods we will consider in Chapter 5.

We introduce the *weighted squared error loss* as

$$\text{Loss}(\beta) = (\mathbf{Y} - \mathbf{X}\beta)^T \mathbf{W}(\mathbf{Y} - \mathbf{X}\beta) \quad (2.5)$$

where $\mathbf{W}$ is a positive definite matrix. An $n \times n$ matrix is positive definite if it is symmetric and

$$\mathbf{y}^T \mathbf{W} \mathbf{y} > 0$$

for all $\mathbf{y} \in \mathbb{R}^n$ with $\mathbf{y} \neq 0$. A special type of positive definite weight matrix is a diagonal matrix with positive entries in the diagonal. With

$$\mathbf{W} = \begin{pmatrix} w_1 & 0 & \dots & 0 \\ 0 & w_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & w_n \end{pmatrix}$$

we find that the weighted squared error loss becomes

$$\text{Loss}(\beta) = \sum_i w_i (Y_i - X_i^T \beta)^2.$$

That is, the i -th case receives the weight w_i .

We will estimate the parameter β via minimization of Loss—or via minimization of Loss plus a penalty term. With Ω a positive semidefinite $p \times p$ matrix we will seek to minimize the function

$$\text{Loss}_\Omega(\beta) := \text{Loss}(\beta) + \beta^T \Omega \beta. \quad (2.6)$$

The penalty term, $\beta^T \Omega \beta$, introduces a form of control on the minimization problem, and thus the estimator, by shrinking the minimizer toward 0. This can have beneficial numerical as well as statistical consequences. The purpose of introducing a penalty will be elaborated upon in subsequent chapters, where some concrete examples will be given.

With $\mathbf{W} = \mathbf{I}$ this loss is proportional to the negative log-likelihood loss under assumptions **A3** and **A5** as derived in Chapter 5.

Equation 2.7 is called the *normal equation*.

Theorem 2.1. *A solution to the equation*

$$(\mathbf{X}^T \mathbf{W} \mathbf{X} + \Omega) \beta = \mathbf{X}^T \mathbf{W} \mathbf{Y} \quad (2.7)$$

is a minimizer of the quadratic loss function (2.6). There is a unique minimizer if either $\mathbf{X}$ has full column rank p or if Ω is positive definite.

Proof. The derivative of Loss_Ω is

$$D_\beta \text{Loss}_\Omega(\beta) = -2(\mathbf{Y} - \mathbf{X}\beta)^T \mathbf{W} \mathbf{X} + 2\beta^T \Omega.$$

To compute this derivative, it may be useful to think of Loss as a composition,

$$\text{Loss}(\beta) = b(a(\beta)),$$

where $a(\beta) = (\mathbf{Y} - \mathbf{X}\beta)$ and $b(z) = z^T \mathbf{W} z$. Here a is a map $a : \mathbb{R}^p \rightarrow \mathbb{R}^n$ with derivative $D_\beta a(\beta) = -\mathbf{X}$. And b is a function $b : \mathbb{R}^n \rightarrow \mathbb{R}$ with derivative $D_z b(z) = 2z^T \mathbf{W}$. By the chain rule

$$D_\beta \text{Loss}(\beta) = D_z b(a(\beta)) D_\beta a(\beta) = -2(\mathbf{Y} - \mathbf{X}\beta)^T \mathbf{W} \mathbf{X}.$$

The derivative of $\beta \mapsto \beta^T \Omega \beta$ is found in the same way as the derivative of b . Note that the derivative is a *row vector*.⁷

We note that Equation 2.7 is equivalent to $D_\beta \text{Loss}_\Omega(\beta) = 0$, hence a solution of Equation 2.7 is a stationary point of Loss_Ω .

The second derivative of Loss_Ω is

$$D_\beta^2 \text{Loss}_\Omega(\beta) = 2(\mathbf{X}^T \mathbf{W} \mathbf{X} + \Omega).$$

This matrix is (globally) positive semidefinite, hence the function is convex and any stationary point is a global minimum.

If $\mathbf{X}$ has rank p or if Ω is positive definite, $D_\beta^2 \text{Loss}_\Omega(\beta)$ is positive definite and the minimizer of Loss_Ω is the unique solution of Equation 2.7. $\square$

When the solution to Equation 2.7 is unique it can, of course, be written as

$$\hat{\beta} = (\mathbf{X}^T \mathbf{W} \mathbf{X} + \Omega)^{-1} \mathbf{X}^T \mathbf{W} \mathbf{Y}.$$

This can be a theoretically convenient formula for the estimator, but as we discuss below, the practical computation of $\hat{\beta}$ does typically not rely on explicit matrix inversion.

A geometric interpretation of the weighted least squares estimator without penalization provides additional insights. The inner product induced by $\mathbf{W}$ on $\mathbb{R}^n$ is given by $\mathbf{y}^T \mathbf{W} \mathbf{x}$, and

⁷ The *gradient*,

$$\nabla_\beta \text{Loss}(\beta) = D_\beta \text{Loss}(\beta)^T,$$

is a column vector.

In the rest of this section we only consider the case with $\Omega = 0$.

the corresponding norm is denoted $\|\cdot\|_{\mathbf{W}}$. With this notation we see that

$$\text{Loss}(\beta) = \|\mathbf{Y} - \mathbf{X}\beta\|_{\mathbf{W}}^2.$$

If $L = \{\mathbf{X}\beta \mid \beta \in \mathbb{R}^p\}$ denotes the column space of $\mathbf{X}$, Loss is minimized whenever $\mathbf{X}\beta$ is the orthogonal projection of $\mathbf{Y}$ onto L in the inner product given by $\mathbf{W}$.

$\|\mathbf{y}\|_{\mathbf{W}}^2 = \mathbf{y}^T \mathbf{W} \mathbf{y}$ specifies a norm if and only if $\mathbf{W}$ is positive definite.

Lemma 2.1. *The orthogonal projection onto L is*

$$\Pi = \mathbf{X}(\mathbf{X}^T \mathbf{W} \mathbf{X})^{-1} \mathbf{X}^T \mathbf{W}$$

provided that $\mathbf{X}$ has full column rank p .

Proof. We verify that Π is the orthogonal projection onto L by verifying three characterizing properties:

$$\begin{aligned} \Pi \mathbf{X} \beta &= \mathbf{X} \beta \quad (\Pi \text{ is the identity on } L) \\ \Pi^2 &= \mathbf{X}(\mathbf{X}^T \mathbf{W} \mathbf{X})^{-1} \mathbf{X}^T \mathbf{W} \mathbf{X}(\mathbf{X}^T \mathbf{W} \mathbf{X})^{-1} \mathbf{X}^T \mathbf{W} \\ &= \mathbf{X}(\mathbf{X}^T \mathbf{W} \mathbf{X})^{-1} \mathbf{X}^T \mathbf{W} = \Pi \\ \Pi^T \mathbf{W} &= (\mathbf{X}(\mathbf{X}^T \mathbf{W} \mathbf{X})^{-1} \mathbf{X}^T \mathbf{W})^T \mathbf{W} \\ &= \mathbf{W} \mathbf{X}(\mathbf{X}^T \mathbf{W} \mathbf{X})^{-1} \mathbf{X}^T \mathbf{W} = \mathbf{W} \Pi. \end{aligned}$$

The last property is self-adjointness w.r.t. the inner product given by $\mathbf{W}$. $\square$

Note that for $\Omega = 0$, Theorem 2.1 is a consequence of Lemma 2.1. This follows from the identity $\Pi \mathbf{Y} = \mathbf{X} \hat{\beta}$, since the equation $\Pi \mathbf{Y} = \mathbf{X} \beta$ has a unique solution when the columns of $\mathbf{X}$ are linearly independent.

If $\mathbf{X}$ does not have rank p the projection is still well defined, and it can be written as

$$\Pi = \mathbf{X}(\mathbf{X}^T \mathbf{W} \mathbf{X})^- \mathbf{X}^T \mathbf{W}$$

where $(\mathbf{X}^T \mathbf{W} \mathbf{X})^-$ denotes a generalized inverse. This is seen by verifying the same three conditions as in the proof above. The solution to $\Pi \mathbf{Y} = \mathbf{X} \beta$ is, however, no longer unique, and the solution

$$\hat{\beta} = (\mathbf{X}^T \mathbf{W} \mathbf{X})^- \mathbf{X}^T \mathbf{W} \mathbf{Y}$$

is just one possible solution.

A generalized inverse of a matrix $\mathbf{A}$ is any matrix $\mathbf{A}^-$ with the property that

$$\mathbf{A} \mathbf{A}^- \mathbf{A} = \mathbf{A}.$$

2.5.2 Algorithms

The actual computation of the solution to the normal equation is typically based on a QR decomposition of $\mathbf{X}$ and not on a matrix inversion or a direct solution of the linear equation.

The R function `lm()`—or rather the underlying R functions `lm.fit()` and `lm.wfit()`—are based on the QR decomposition. We first treat the case $\mathbf{W} = \mathbf{I}$ and $\Omega = 0$ and then subsequently the more general cases.

The QR decomposition of $\mathbf{X}$ is the factorization

$$\mathbf{X} = \mathbf{Q}\mathbf{R},$$

where $\mathbf{Q}$ is an orthogonal matrix and $\mathbf{R}$ is an upper triangular matrix. Since

$$\mathbf{X}^T\mathbf{X} = \mathbf{R}^T \underbrace{\mathbf{Q}^T\mathbf{Q}}_{\mathbf{I}} \mathbf{R} = \mathbf{R}^T\mathbf{R}, \quad (2.8)$$

the normal equation becomes

$$\mathbf{R}^T\mathbf{R}\beta = \mathbf{X}^T\mathbf{Y}.$$

This equation can be solved efficiently and in a numerically stable way in a two-step pass by exploiting first that $\mathbf{R}^T$ is lower triangular and then that $\mathbf{R}$ is upper triangular. Moreover, the QR decomposition can be implemented in a rank revealing way. This is useful when $\mathbf{X}$ does not have rank p , in which case the QR decomposition effectively reparametrizes the model as if some columns were removed from $\mathbf{X}$. Such a version of the QR decomposition is used by `lm.fit()` and `lm.wfit()` in R, and one will obtain parameter estimates with the value NA when $\mathbf{X}$ has (numerical) rank strictly less than p .

For a general weight matrix we write $\mathbf{W} = \mathbf{L}\mathbf{L}^T$, in which case the normal equation can be rewritten as

$$(\mathbf{L}^T\mathbf{X})^T\mathbf{L}^T\mathbf{X}\beta = (\mathbf{L}^T\mathbf{X})^T\mathbf{L}^T\mathbf{Y}$$

Thus by replacing $\mathbf{X}$ with $\mathbf{L}^T\mathbf{X}$ and $\mathbf{Y}$ with $\mathbf{L}^T\mathbf{Y}$ we can proceed as above and solve the normal equation via a QR decomposition of $\mathbf{L}^T\mathbf{X}$.

The computations based on the QR decomposition do not involve the computation of the *cross product* $\mathbf{X}^T\mathbf{X}$ (or more

The normal equation can be solved by Gaussian elimination, which corresponds to an LU decomposition of

$$\mathbf{X}^T\mathbf{W}\mathbf{X} + \Omega.$$

The algorithm was named after Carl Friedrich Gauss, though he did not invent it. However, he promoted it for efficient computation of the solution of least squares problems.

This could be the Cholesky decomposition with $\mathbf{L}$ lower triangular, or $\mathbf{L}$ could be the symmetric square root of $\mathbf{W}$. For a diagonal $\mathbf{W}$, $\mathbf{L}$ is diagonal and trivial to compute by taking square roots. For unstructured $\mathbf{W}$ the computation of the Cholesky decomposition scales as n^3 .

generally, $\mathbf{X}^T \mathbf{W} \mathbf{X}$). The factorization (2.8) of the positive semidefinite matrix $\mathbf{X}^T \mathbf{X}$ as a lower and upper triangular matrix is called the Cholesky decomposition, and as outlined above it can be computed via the QR decomposition of $\mathbf{X}$. An alternative is to compute $\mathbf{X}^T \mathbf{X}$ directly and then compute its Cholesky decomposition. The QR decomposition is usually preferred for numerical stability. Computing $\mathbf{X}^T \mathbf{X}$ is essentially a squaring operation, and precision can be lost. However, if n is large compared to p it will be faster to compute the Cholesky decomposition from the cross product.

If $\Omega \neq 0$ the normal equation can be solved by computing the Cholesky decomposition

$$\mathbf{X}^T \mathbf{W} \mathbf{X} + \Omega = \mathbf{R}^T \mathbf{R}$$

directly and then proceed as above.

To use the QR decomposition we write $\Omega = \mathbf{D} \mathbf{D}^T$. Then the normal equation can be written as

$$\begin{pmatrix} \mathbf{X}^T \mathbf{L} & \mathbf{D} \end{pmatrix} \begin{pmatrix} \mathbf{L}^T \mathbf{X} \\ \mathbf{D}^T \end{pmatrix} \beta = \begin{pmatrix} \mathbf{X}^T \mathbf{L} & \mathbf{D} \end{pmatrix} \begin{pmatrix} \mathbf{L}^T \mathbf{Y} \\ 0 \end{pmatrix}.$$

The computations could again be done via the Cholesky decomposition of Ω , or $\mathbf{D}$ could be computed as the symmetric square root of Ω .

By replacing $\mathbf{X}$ with

$$\begin{pmatrix} \mathbf{L}^T \mathbf{X} \\ \mathbf{D}^T \end{pmatrix}$$

and $\mathbf{Y}$ with

$$\begin{pmatrix} \mathbf{L}^T \mathbf{Y} \\ 0 \end{pmatrix}$$

we can solve the normal equation via the QR decomposition as above.

2.6 Sampling distributions

In this section we give results on the distribution of the estimator, $\hat{\beta}$, and test statistics of linear hypotheses under the weak assumptions **A1**, **A2** and **A4** and under the strong assumptions **A3** and **A5**. Under the weak assumptions we can only obtain results on moments, while the strong distributional assumptions give exact sampling distributions. Throughout we restrict attention to the case where $\mathbf{W} = \mathbf{I}$ and $\Omega = 0$.

2.6.1 Moments

Some results involve the unknown variance parameter σ^2 (see assumption **A2**) and some involve a specific estimator $\hat{\sigma}^2$. This estimator of σ^2 is

$$\hat{\sigma}^2 = \frac{1}{n-p} \sum_{i=1}^n (Y_i - X_i^T \hat{\beta})^2 = \frac{1}{n-p} \|\mathbf{Y} - \mathbf{X}\hat{\beta}\|^2 \quad (2.9)$$

provided that $\mathbf{X}$ has full rank p . Defining the i -th *residual* as

$$\hat{\varepsilon}_i = Y_i - X_i^T \hat{\beta},$$

the variance estimator equals the empirical variance of the residuals—up to division by $n-p$ and not n . Since the residual is a natural estimator of the unobserved error ε_i , the variance estimator $\hat{\sigma}^2$ is a natural estimator of the error variance σ^2 . The explanation of the denominator $n-p$ is related to the fact that $\hat{\varepsilon}_i$ is an estimator of ε_i . A partial justification, as shown in the following theorem, is that division by $n-p$ makes $\hat{\sigma}^2$ unbiased.

Theorem 2.2. *Under the weak assumptions **A1**, **A2** and **A4**, and assuming that $\mathbf{X}$ has full rank p ,*

$$\begin{aligned} \mathbf{E}(\hat{\beta} \mid \mathbf{X}) &= \beta, \\ \mathbf{V}(\hat{\beta} \mid \mathbf{X}) &= \sigma^2 (\mathbf{X}^T \mathbf{X})^{-1}, \\ \mathbf{E}(\hat{\sigma}^2 \mid \mathbf{X}) &= \sigma^2. \end{aligned}$$

Proof. Using assumptions **A1** and **A4** we find that

$$\begin{aligned} \mathbf{E}(\hat{\beta} \mid \mathbf{X}) &= \mathbf{E}((\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y} \mid \mathbf{X}) \\ &= (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{E}(\mathbf{Y} \mid \mathbf{X}) \\ &= (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{X} \beta \\ &= \beta. \end{aligned}$$

Using assumptions **A2** and **A4** it follows that

$$\begin{aligned} \mathbf{V}(\hat{\beta} \mid \mathbf{X}) &= (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{V}(\mathbf{Y} \mid \mathbf{X}) \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1} \\ &= (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \sigma^2 \mathbf{I} \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1} \\ &= \sigma^2 (\mathbf{X}^T \mathbf{X})^{-1}. \end{aligned}$$

For the computation of the expectation of $\hat{\sigma}^2$, the geometric interpretation of $\hat{\beta}$ is useful. Since $\mathbf{X}\hat{\beta} = \Pi \mathbf{Y}$ with Π the

orthogonal projection onto the column space L of $\mathbf{X}$, we find that

$$\mathbf{Y} - \mathbf{X}\hat{\beta} = (\mathbf{I} - \Pi)\mathbf{Y}.$$

Because $\mathbf{E}(\mathbf{Y} - \mathbf{X}\hat{\beta} \mid \mathbf{X}) = 0$

$$\mathbf{E}(\|\mathbf{Y} - \mathbf{X}\hat{\beta}\|^2 \mid \mathbf{X}) = \sum_{i=1}^n \mathbf{V}(\mathbf{Y} - \mathbf{X}\hat{\beta} \mid \mathbf{X})_{ii}$$

and

$$\begin{aligned} \mathbf{V}(\mathbf{Y} - \mathbf{X}\hat{\beta} \mid \mathbf{X}) &= \mathbf{V}((\mathbf{I} - \Pi)\mathbf{Y} \mid \mathbf{X}) \\ &= (\mathbf{I} - \Pi)\mathbf{V}(\mathbf{Y} \mid \mathbf{X})(\mathbf{I} - \Pi)^T \\ &= (\mathbf{I} - \Pi)\sigma^2\mathbf{I}(\mathbf{I} - \Pi) \\ &= \sigma^2(\mathbf{I} - \Pi). \end{aligned}$$

The sum of the diagonal elements in $(\mathbf{I} - \Pi)$ is the trace of this orthogonal projection onto $L^\perp$ —the orthogonal complement of L —and is thus equal to the dimension of $L^\perp$, which is $n - p$. $\square$

2.6.2 Tests and confidence intervals

For the computation of exact distributional properties of test statistics and confidence intervals we need the strong distributional assumptions **A3** and **A5**.

Theorem 2.3. *Under the strong assumptions **A3** and **A5** it holds, conditionally on $\mathbf{X}$, that*

$$\hat{\beta} \sim \mathcal{N}(\beta, \sigma^2(\mathbf{X}^T\mathbf{X})^{-1})$$

and that

$$(n - p)\hat{\sigma}^2 \sim \sigma^2\chi_{n-p}^2.$$

Moreover, for the standardized Z -score

$$Z_j = \frac{\hat{\beta}_j - \beta_j}{\hat{\sigma}\sqrt{(\mathbf{X}^T\mathbf{X})_{jj}^{-1}}} \sim t_{n-p},$$

or more generally for any $a \in \mathbb{R}^p$

$$Z_a = \frac{a^T\hat{\beta} - a^T\beta}{\hat{\sigma}\sqrt{a^T(\mathbf{X}^T\mathbf{X})^{-1}a}} \sim t_{n-p}.$$

Proof. See Hansen (2012), Chapter 10. $\square$

The standardized Z -scores are used to test hypotheses about a single parameter or a single linear combination of the parameters. The Z -score is computed under the hypothesis (with the hypothesized value of β_j or $a^T\beta$ plugged in), and compared to the t_{n-p} distribution. The test is two-sided. The Z -scores are also used to construct confidence intervals for linear combinations of the parameters. A 95% confidence interval for $a^T\beta$ is computed as

$$a^T\hat{\beta} \pm z_{n-p} \cdot \hat{\sigma} \sqrt{a^T(\mathbf{X}^T\mathbf{X})^{-1}a} \quad (2.10)$$

where $\hat{\sigma} \sqrt{a^T(\mathbf{X}^T\mathbf{X})^{-1}a}$ is the estimated standard error of $a^T\hat{\beta}$ and z_{n-p} is the 97.5% quantile in the t_{n-p} -distribution.

For the computation of $a^T(\mathbf{X}^T\mathbf{X})^{-1}a$ we do not need to compute the $(\mathbf{X}^T\mathbf{X})^{-1}$ if we have already computed the QR-decomposition of $\mathbf{X}$ or the Cholesky decomposition of $\mathbf{X}^T\mathbf{X}$. With $\mathbf{X}^T\mathbf{X} = \mathbf{L}\mathbf{L}^T$ for a lower triangular⁸ $p \times p$ matrix $\mathbf{L}$ we find that

$$\begin{aligned} a^T(\mathbf{X}^T\mathbf{X})^{-1}a &= a^T(\mathbf{L}\mathbf{L}^T)^{-1}a \\ &= (\mathbf{L}^{-1}a)^T\mathbf{L}^{-1}a \\ &= b^Tb \end{aligned}$$

where b solves $\mathbf{L}b = a$. The solution of this lower triangular system of equations is *faster* to compute than the matrix-vector product $(\mathbf{X}^T\mathbf{X})^{-1}a$, even if the inverse matrix is already computed and stored. This implies that the computation of $(\mathbf{X}^T\mathbf{X})^{-1}$ is never computationally beneficial. Not even if we need to compute estimated standard errors for many different choices of a .

To test hypotheses involving more than a one-dimensional linear combination, we need the F -tests. Let $p_0 < p$ and assume that $\mathbf{X}'$ is an $n \times p_0$ -matrix whose p_0 columns span a p_0 -dimensional subspace of the column space of $\mathbf{X}$. With $\hat{\beta}'$ the least squares estimator corresponding to $\mathbf{X}'$ the F -test statistic is defined as

$$F = \frac{\|\mathbf{X}\hat{\beta} - \mathbf{X}'\hat{\beta}'\|^2/(p-p_0)}{\|\mathbf{Y} - \mathbf{X}\hat{\beta}\|^2/(n-p)}. \quad (2.11)$$

The Z -score statistics are in this setup also known as the t -test statistics. Under the hypothesis $H_0: \beta_j = 0$ we see that $Z_j = \hat{\beta}_j / \text{se}(\hat{\beta}_j)$ where $\text{se}(\hat{\beta}_j)$ is an estimate of the standard error, that is, the standard deviation of the estimator $\hat{\beta}_j$.

⁸ If we have computed the QR-decomposition of $\mathbf{X}$ we take $\mathbf{L} = \mathbf{R}^T$.

Note that the denominator is just $\hat{\sigma}^2$. The F -test statistic is one-sided with large values critical.

Theorem 2.4. *Under the hypothesis*

$$H_0 : \mathbf{Y} \mid \mathbf{X} \sim \mathcal{N}(\mathbf{X}'\beta_0, \sigma^2\mathbf{I})$$

the F -test statistic follows an F -distribution with $(p-p_0, n-p)$ degrees of freedom.

Proof. See Hansen (2012), Chapter 10. □

Hansen, Ernst. 2012. *Introduktion Til Matematisk Statistisk*. Institut for Matematiske Fag.

The terminology associated with the F -test and reported by the R function `anova()` is as follows (R abbreviations in parentheses). The squared norm $\|\mathbf{Y} - \mathbf{X}\hat{\beta}\|^2$ is called the residual sum of squares (**RSS**) under the model, and $n - p$ is the residual degrees of freedom (**Res. Df**). The squared norm $\|\mathbf{X}\hat{\beta} - \mathbf{X}'\hat{\beta}'\|^2$ is the sum of squares (**Sum of Sq**), and $p - p_0$ is the degrees of freedom (**Df**).

The squared norm $\|\mathbf{Y} - \mathbf{X}'\hat{\beta}'\|^2$ is the residual sum of squares under the hypothesis, and it follows from Pythagoras's theorem that

$$\|\mathbf{X}\hat{\beta} - \mathbf{X}'\hat{\beta}'\|^2 = \|\mathbf{Y} - \mathbf{X}'\hat{\beta}'\|^2 - \|\mathbf{Y} - \mathbf{X}\hat{\beta}\|^2.$$

Thus the sum of squares is the difference between the residual sum of squares under the hypothesis and under the model. The R function `anova()` computes and reports these numbers together with a p -value for the null hypothesis derived from the appropriate F -distribution.

It is important for the validity of the F -test that the column space, L' , of $\mathbf{X}'$ is a subspace of the column space, L , of $\mathbf{X}$. Otherwise the models are not nested and a formal hypothesis test is meaningless even if $p_0 < p$. It may not be obvious from the matrices if the models are nested. By definition, $L' \subseteq L$ if and only if

$$\mathbf{X}' = \mathbf{X}\mathbf{C} \tag{2.12}$$

for a $p \times p_0$ matrix $\mathbf{C}$ (of rank p_0), and we can verify that $L' \subseteq L$ by finding a $\mathbf{C}$ matrix such that (2.12) holds. The situation where the columns of $\mathbf{X}'$ is a subset of the columns of $\mathbf{X}$, which is a hypothesis on the complementary set of parameters being 0, corresponds to $\mathbf{C}$ having 0-s or 1-s appropriately placed

along the main diagonal and off-diagonal elements being 0.

In the ANOVA literature, where categorical predictors and their interactions are considered, a considerable amount of work has been invested on choosing the $\mathbf{X}$ matrix, and thus the parametrization, so that relevant scientific hypotheses can be formulated in terms of certain parameters being 0. Though this can ease communication in certain cases, it can also be quite confusing. Linear hypotheses considered within the linear model are not specific to a given choice of parametrization, and there is no requirement that a linear hypothesis is linked to certain parameters being 0. This will be illustrated in subsequent sections.

It is not necessary to compute the actual $\mathbf{C}$ matrix if theoretical arguments show that it exists.

2.7 Model assessment

This section introduces methods for investigating deviations from the model assumptions in the data and methods for assessing the descriptive or predictive strength of the model.

In the light of George Box's famous aphorism, *all models are wrong but some are useful*, it is appropriate to add a supplementary Box quote: *Since all models are wrong the scientist must be alert to what is importantly wrong* (Box 1976). This is done by investigating the assumptions **A1**, **A2** and **A3** after the model has been fitted to data and typically via the residuals

$$\hat{\varepsilon}_i = Y_i - X_i^T \hat{\beta}, \quad (2.13)$$

which are approximations of the errors ε_i .

We use the residuals to investigate the model assumptions mostly via graphical methods.

- Plots of $\hat{\varepsilon}_i$ against a predictor or against the *fitted values* $X_i^T \hat{\beta}$ show deviations from assumptions **A1** or **A2**
- A qq-plot of the residuals $\hat{\varepsilon}_i$ against the normal distribution shows deviations from assumption **A3**.

To measure the predictive strength of a linear model, we can compute the residual sum of squares

$$\text{RSS} = \|\mathbf{Y} - \mathbf{X}\hat{\beta}\|^2 = \sum_{i=1}^n (\hat{\varepsilon}_i)^2.$$

Box, George E. P. 1976. "Science and Statistics." *Journal of the American Statistical Association* 71 (356): 791–99.

It quantifies how well $X_i^T \hat{\beta}$ predicts Y_i for $i = 1, \dots, n$. The absolute value of RSS is difficult to interpret, and it also suffers from an optimism by being computed on the data used for estimating the parameters.

The (in)famous R^2 is a first attempt to introduce an interpretable index of predictive strength.

Definition 2.1. Let RSS_0 be the residual sum of squares for the intercept only model and RSS the residual sum of squares for the model of interest (of dimension p). The coefficient of determination is

$$R^2 = \frac{\text{RSS}_0 - \text{RSS}}{\text{RSS}_0} = 1 - \frac{\text{RSS}}{\text{RSS}_0}. \quad (2.14)$$

Observe that the F -test statistic for the intercept only model is

$$F = \frac{n-p}{p-1} \frac{R^2}{1-R^2}$$

Thus the coefficient of determination and the F -test are in one-to-one correspondence via a monotonely increasing function. The closer R^2 is to 1, the larger is the F -test. With a single numerical predictor, R^2 is the *square of the Pearson correlation* between the outcome and the predictor. For this reason, the square root of R^2 is called the *multiple correlation coefficient*.

For two non-nested models with the same complexity (same dimension p in this case), the one with the largest R^2 (smallest RSS) is the best predictor on the dataset. But R^2 is monotonely increasing in p for nested models, which implies that among nested models the more complex model will always predict better on the data. This does not reflect what will happen if we want to make predictions on an independent observation ($Y^{\text{new}}, X^{\text{new}}$). The average residual sum of squares, RSS/n , will generally underestimate the *prediction error*.

$$\mathbf{E} \left((Y^{\text{new}} - (X^{\text{new}})^T \hat{\beta})^2 \mid \hat{\beta} \right),$$

and the more complex the model is (the larger p is), the more serious is the underestimation. It is quite possible that $\text{RSS} \approx 0$ ($R^2 \approx 1$) even though the prediction error is still considerable.

The R^2 quantity has surely been widely overused and over interpreted, but it is a sensible quantity to consider, in particular for observational data. Note also the definition of the adjusted R^2 below.

To (partly) remedy the optimism of R^2 we introduce the *adjusted* R^2 . We can interpret $1 - R^2$ as a ratio of variance estimates,

$$1 - R^2 = \frac{\text{RSS}/n}{\text{RSS}_0/n} = \frac{\tilde{\sigma}^2}{\tilde{\sigma}_0^2}$$

using *biased* estimates of the variance. The adjusted R^2 is defined as

$$\overline{R}^2 = 1 - \frac{\text{RSS}/(n-p)}{\text{RSS}_0/(n-1)} = 1 - (1 - R^2) \frac{n-1}{n-p}, \quad (2.15)$$

where the variance ratio is based on *unbiased* estimates of the variance.

2.7.1 Investigating the claim size models

Figure 2.5 shows three typical diagnostic plots. The residuals plotted against the fitted values, $X_i^T \hat{\beta}$, can be used to detect if the mean value specification is adequate and if the variance is approximately constant as a function of the fitted values. The scatter plot smoother added to the residual plot bends up in the ends indicating that the mean is not spot on. There is no clear indication of variance inhomogeneity. The histogram of the residuals and the qq-plot of the standardized⁹ residuals can be used for comparing the residual distribution with the normal distribution. In this case the residuals have a slightly right skewed distribution when compared to the normal distribution. Of course we also note the pattern of the most extreme positive residuals that correspond to the capped claims. In this section we will not pursue the effect of those.

⁹ Standardized so as to make the residuals have unit variance under the model assumptions.

```
claims_diag <- augment(claims_lm) ## Residuals etc.
gridExtra::grid.arrange(
  ggplot(claims_diag, aes(.fitted, .resid), alpha = I(0.2)) + geom_point() + geom_smooth(),
  ggplot(claims_diag, aes(.resid)) + geom_histogram(bins = 40),
  ggplot(claims_diag, aes(sample = .std.resid)) + geom_qq() + geom_abline(),
  ncol = 3
)
```

```
Warning in fortify(data, ...): Arguments in `...` must be used.
x Problematic argument:
* alpha = I(0.2)
i Did you misspell an argument name?
```

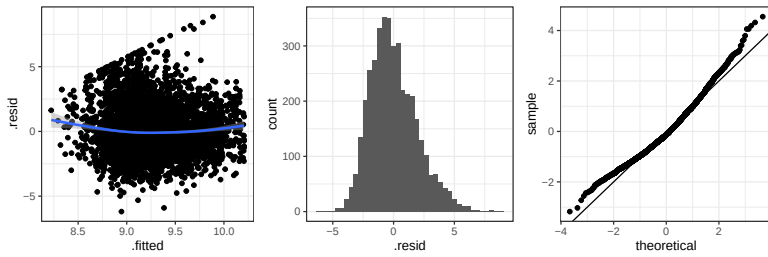


Figure 2.5: Diagnostic plots based on residuals for the regression model conditioning on insurance sum only. The residual plot (left) and the histogram (middle) use the raw residuals. The qq-plot (right) uses standardized residuals.

We proceed by looking at the model summary. The `lm` object `claims_lm` contains a substantial amount of information about the fitted linear model—in addition to the parameter estimates. The generic base R function `summary()` prints a summary of that information.

```
summary(claims_lm)
```

Call:

```
lm(formula = log(claims) ~ log(sum), data = claims)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-6.1922	-1.3836	-0.2523	1.2070	8.8644

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	5.84101	0.29487	19.81	<2e-16 ***
log(sum)	0.21149	0.01824	11.59	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.949 on 4034 degrees of freedom

Multiple R-squared: 0.03225, Adjusted R-squared: 0.03201

F-statistic: 134.4 on 1 and 4034 DF, p-value: < 2.2e-16

From the summary of the model, including only log insurance sum as a predictor, we find that the slope is different from 0 according to the t -test. However, R^2 (denoted **Multiple R-square** in the R output) is only around 3% and thus very small, which shows that the insurance sum is a quite weak predictor of the claim size. Note that for this regression model with only one numerical predictor, R^2 is the squared Pearson correlation between log claim size and log insurance sum, which we can verify directly.

```
cor(log(claims$claims), log(claims$sum))^2
```

```
[1] 0.03224911
```

Two alternatives to `summary()` are the functions `tidy()` and `glance()` from the broom package. They also extract model information but in a more specific and useful way. The `glance()` function gives a one-line summary of a range of model statistics, including R^2 , $\hat{\sigma}$ and degrees of freedom.

```
glance(claims_lm) |>
  knitr::kable()
```

r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC	deviance	coefficient
0.0322491	0.0320092	1.94927	134.4281	0	1	-8419.683	16845.37	16864.27	15327.8	0.03224911

Table 2.1: Summary statistics for the simple linear model of claim sizes

The `tidy()` function extracts parameter estimates, standard errors, test statistics and p -values, just as `summary()` does, but returns them as a tibble, which is useful for downstream computations. This feature is used below to generate a slightly nicer table via the `kable()` function from the knitr package.

```
tidy(claims_lm) |>
  knitr::kable()
```

term	estimate	std.error	statistic	p.value
(Intercept)	5.8410060	0.2948718	19.80863	0
log(sum)	0.2114886	0.0182407	11.59431	0

Table 2.2: Parameter estimates for the simple linear model of claim sizes

We then turn to the additive model that includes the trade group effect. We first consider the parameter estimates for the additive model. Recall that we used the sum contrast for the (final version of the) model.

```
tidy(claims_lm_add) |>
  knitr::kable()
```

term	estimate	std.error	statistic	p.value
(Intercept)	4.0631357	0.3351685	12.1226666	0.0000000
log(sum)	0.3300459	0.0212311	15.5454313	0.0000000
grp1	-0.4657314	0.0546124	-8.5279353	0.0000000
grp2	0.0815417	0.0648852	1.2567073	0.2089325
grp3	-0.0643848	0.0851485	-0.7561478	0.4496048

Table 2.3: Parameter estimates for the additive model of claim sizes

The table shows directly that group 1 has a lower intercept than the common intercept, while it can be inferred from the parametrization that group 4 has a higher intercept. The effect of including a trade group specific intercept is also an increased slope compared to the previous model.

```
glance(claims_lm_add) |>
  knitr::kable()
```

r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC	deviance
0.059088	0.0581544	1.922765	63.28537	0	4	-8362.926	16737.85	16775.67	14902.71

Table 2.4: Summary statistics for the additive model of claim sizes

The R^2 has increased to around 6% suggesting that the additive model is better for predictions than the model without trade group, though it is still far from a strong prediction model.

For the interaction model, see Table 2.5, the R^2 has increased a little compared to the additive model, but three additional parameters have also been added to the model. If we look at the adjusted R^2 the increase is miniscule.

```
glance(claims_lm_int) |>
  knitr::kable()
```

We can also compare the models by formally testing one model as a hypothesis within a larger model. This can be done in R with the `anova()` function, which uses the F -test statistic and the distributional results derived in the previous sections.

r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC	deviance
0.0606258	0.0589934	1.921908	37.13732	0	7	-8359.626	16737.25	16793.98	14878.35

Table 2.5: Summary statistics for the interaction model of claim sizes

To compare the three models we investigate the two hypotheses

$$H_0 : \beta_{\text{grp1}} = \beta_{\text{grp2}} = \beta_{\text{grp3}} = 0$$

in the additive model, and

$$H_1 : \beta_{\text{sum,grp2}} = \beta_{\text{sum,grp3}} = \beta_{\text{sum,grp4}} = 0$$

in the interaction model. These two hypotheses correspond to nested subspaces, and the F -tests can be computed by a single call of `anova()`.

```
anova(claims_lm, claims_lm_add, claims_lm_int) |>
  knitr::kable()
```

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
4034	15327.80	NA	NA	NA	NA
4031	14902.71	3	425.09024	38.361414	0.0000000
4028	14878.35	3	24.35674	2.198025	0.0861988

Table 2.6: Analysis of variance table comparing the three linear models of claim sizes.

The conclusion is that we cannot reject H_1 while H_0 can be clearly rejected. Thus using this approach, data does not provide evidence that the slope should be different in the four trade groups, but there is strong evidence that the intercept should. This is in concordance with the observations made above based on the estimated parameters and R^2 .

We conclude the section by revisiting the diagnostic plots, but this time for the interaction model, see Figure 2.6.

```
claims_diag_int <- augment(claims_lm_int) ## Residuals etc.
gridExtra::grid.arrange(
  ggplot(claims_diag_int, aes(.fitted, .resid), alpha = I(0.2)) + geom_point() + geom_smooth(),
  ggplot(claims_diag_int, aes(.resid)) + geom_histogram(bins = 40),
  ggplot(claims_diag_int, aes(sample = .std.resid)) + geom_qq() + geom_abline(),
```



```
ncol = 3
)
```

```
Warning in fortify(data, ...): Arguments in `...` must be used.
x Problematic argument:
* alpha = I(0.2)
i Did you misspell an argument name?
```

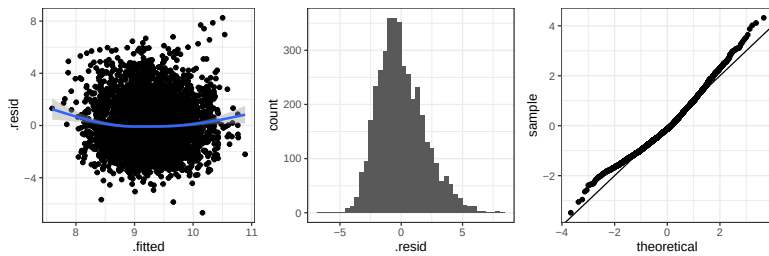


Figure 2.6: Diagnostic plots based on residuals for the regression model with an interaction of trade group and insurance sum. The residual plot (left) and the histogram (middle) use the raw residuals. The qq-plot (right) uses standardized residuals.

The conclusion is the same as for the simpler model that only includes insurance sum as predictor. The mean value model is not spot on and the residual distribution is slightly skewed. It is not a bad model fit, but it leaves room for improvements.

Exercises

Exercise 2.1. This exercise covers the relation between the formula interface, used to specify a linear model via the `lm()` function, and the model matrix $\mathbf{X}$ that a formula generates internally. The formula interface is convenient as it allows us to focus on variables instead of how they are encoded as columns in $\mathbf{X}$. It is, however, useful to know precisely how R interprets the formulas and translates them into matrices. The `lm()` function calls the function `model.matrix()` to create the model matrix from the formula and the data. We do not usually call this function ourselves, but understanding how it works gives some insight on how exactly formulas are parsed.

We simulate an example dataset to work with. Figure 2.7 shows the simulated data.

```
set.seed(04022016)
factor_labels <- rep(c("A", "B", "C"), each = 50)
formula_data <- tibble(
  x = rnorm(150, 0),
```

```

z = factor(sample(factor_labels)),
mu = (z == "A") * cos(x) + 0.5 * (z == "B") * sin(x),
y = rnorm(150, mu, 0.5)
)

```

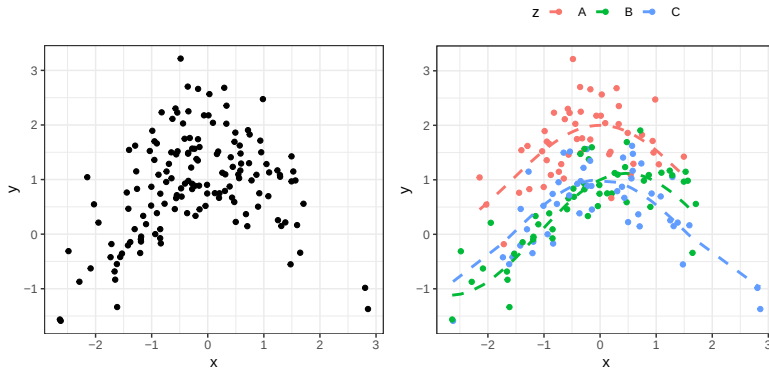


Figure 2.7: The left figure shows a scatter plot of the y variable against the x variable. The right figure adds color coding according to the group as encoded by the z variable. The dashed curves are the true means within the three groups.

1. Fit a linear regression model using `lm(y ~ x, formula_data)` where y is the outcome variable and x is the predictor variable. Investigate the residuals. Does the model fit the data?

The formula $y \sim x$ was used internally for the construction of a model matrix with two columns. A column of ones representing the intercept, and a column with the observed values of the predictor variable.

```

model.matrix(y ~ x, formula_data) |>
  head()

```

```

(Intercept)      x
1          1 -0.00976291
2          1 -1.64998340
3          1 -0.32014201
4          1 -1.42256848
5          1  1.24188519
6          1  1.08065770

```

2. How do you get rid of the intercept via the formula specification?

If we “add a factor” on the right hand side of the formula, e.g., $y \sim x + z$, the factor’s categorical levels will be represented via dummy variable encodings. If it has k levels the model matrix gets $k - 1$ additional columns.

```
model.matrix(y ~ x + z, formula_data) |>
  head()
```

	(Intercept)		x	zB	zC
1	1	-0.00976291	1	0	
2	1	-1.64998340	1	0	
3	1	-0.32014201	0	0	
4	1	-1.42256848	0	1	
5	1	1.24188519	1	0	
6	1	1.08065770	0	1	

The precise encoding is controlled by the `contrasts.arg` argument to `model.matrix()` (the `contrasts` argument to `lm()`) or by setting the contrasts associated with a factor. We can change the base level of `z` to be level 3 (value C), say.

```
contrasts(formula_data$z) <-
  contr.treatment(levels(formula_data$z), base = 3)
model.matrix(y ~ x + z, formula_data) |>
  head()
```

	(Intercept)		x	zA	zB
1	1	-0.00976291	0	1	
2	1	-1.64998340	0	1	
3	1	-0.32014201	1	0	
4	1	-1.42256848	0	0	
5	1	1.24188519	0	1	
6	1	1.08065770	0	0	

A change of the dummy variable encoding does not change the model (the column space) only the parametrization. This may occasionally be useful for interpretation or for testing certain hypotheses.

3. How can you check (numerically) that the two model matrices above span the same column space?

In formulas, the arithmetic operators `*` and `^` together with `:` have special meanings.

- Usage of `b * c` is equivalent to `b + c + b : c`.
- Usage of `b : c` formulates an interaction.
- Usage of `(a + b + c)^2` is equivalent to `(a + b + c) * (a + b + c)`, which is equivalent to `a + b + c + a : b + a : c + b : c`, etc.

The following two formulas generate two different model matrices.

```
model.matrix(y ~ z + x : z, formula_data) |>
  head()
```

	(Intercept)	zA	zB	zA:x	zB:x	zC:x
1	1	0	1	0.000000	-0.00976291	0.000000
2	1	0	1	0.000000	-1.64998340	0.000000
3	1	1	0	-0.320142	0.00000000	0.000000
4	1	0	0	0.000000	0.00000000	-1.422568
5	1	0	1	0.000000	1.24188519	0.000000
6	1	0	0	0.000000	0.00000000	1.080658

```
model.matrix(y ~ x * z, formula_data) |>
  head()
```

	(Intercept)	x	zA	zB	x:zA	x:zB
1	1	-0.00976291	0	1	0.000000	-0.00976291
2	1	-1.64998340	0	1	0.000000	-1.64998340
3	1	-0.32014201	1	0	-0.320142	0.00000000
4	1	-1.42256848	0	0	0.000000	0.00000000
5	1	1.24188519	0	1	0.000000	1.24188519
6	1	1.08065770	0	0	0.000000	0.00000000

- Are the two models above identical? That is, do the columns of the two model matrices span the same column space?

Transformations of predictor variables can be achieved by formally applying an R function to a variable in the formula specification. We happen to know that the data was generated with a mean value structure that is actually a linear model in $\sin(x)$ and $\cos(x)$.

- Run the following code and explain what it returns.

```
model.matrix(y ~ cos(x) + z * sin(x), formula_data) |>
  head()
```

- Fit the model specified by the formula $y \sim \cos(x) + z * \sin(x)$ to the simulated data, explain the parameter estimates you get and reproduce Figure 2.8.

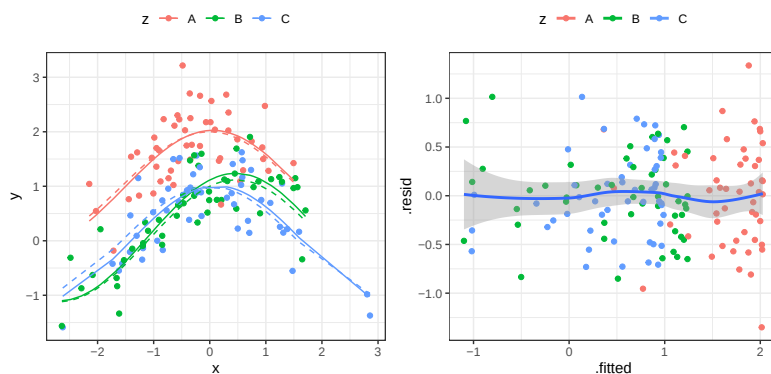


Figure 2.8: Left: The fitted values plotted as lines together with the data. The dashed lines are the true means. Right: a residual plot.

Exercise 2.2. This exercise is based on a dataset relating the mean verbal SAT scores for each of the 50 US states to the percentage of students taking the test. The data comes with the RWR package.

```
library(RWR)
data(sat, package = "RWR")
```

1. Fit a linear regression model (with an intercept) of the state SAT score as a function of percentage of students. Add the fitted line to the plot.
2. Fit a piecewise affine function with the “bend” at 50% of the students taking the test. How can you formally test the linearity assumption in Question 1 within this larger model?

Hint: you want to fit a linear model with a model matrix $\mathbf{X}$ with three columns representing an intercept and two piecewise linear functions. You can, for instance, generate those piecewise linear functions by an `ifelse()` construction in the formula. Try different choices of columns. Is there one choice particularly suitable for testing the linearity assumption?

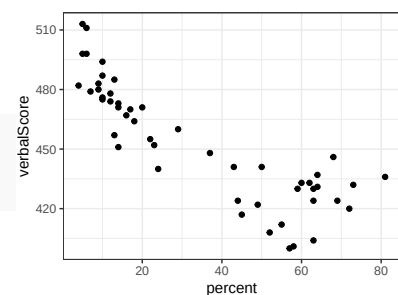


Figure 2.9: SAT scores for each US state against percentage of student taking the test.

Exercise 2.3. Show that if (2.12) holds then

$$\mathbf{C} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{X}'.$$

Explain how this can be used to check if the column space of an arbitrary $\mathbf{X}'$ is contained in the column space of $\mathbf{X}$.

Exercise 2.4. Let $\mathbf{C}$ be any $n \times p$ matrix and consider the linear estimator

$$\tilde{\beta} = \mathbf{C}^T \mathbf{Y}.$$

1. Show that $\tilde{\beta}$ is unbiased as an estimator of β if **A1** and **A4** hold and if

$$\beta = \mathbf{C}^T \mathbf{X} \beta$$

for all β .

2. Show that if the estimator is unbiased and **A1**, **A2** and **A4** hold then

$$\mathbf{V}(\tilde{\beta} \mid \mathbf{X}) = \sigma^2 \mathbf{C}^T \mathbf{C}$$

and that

$$\mathbf{V}(\hat{\beta} - \tilde{\beta} \mid \mathbf{X}) = \sigma^2 (\mathbf{C}^T \mathbf{C} - (\mathbf{X}^T \mathbf{X})^{-1})$$

where $\hat{\beta}$ is the least squares estimator.

3. Explain that this shows the Gauss-Markov theorem: Under the assumptions **A1** + **A2** + **A4** the least squares estimator of β has minimal variance among all linear, unbiased estimators of β .

Exercise 2.5. This exercise relates the empirical correlations between the predictor variables to the variance of the least squares estimator. The setup is as follows. The matrix $\mathbf{X}$ denotes an $n \times p$ matrix of predictor variables, $\mathbf{1}$ denotes an n dimensional column vector of ones and $(\mathbf{1} \mid \mathbf{X})$ denotes the $n \times (p + 1)$ matrix with the first column being $\mathbf{1}$ and the remaining $n \times p$ block being $\mathbf{X}$. We let

$$\bar{X} = \frac{1}{n} \mathbf{X}^T \mathbf{1}$$

denote the p dimensional vector of mean values for the predictor, and we let

$$\hat{\Sigma} = \frac{1}{n} (\mathbf{X} - \mathbf{1} \bar{X}^T)^T (\mathbf{X} - \mathbf{1} \bar{X}^T)$$

denote the empirical covariance matrix. It is assumed that $\hat{\Sigma}$ is invertible.

1. Show that

$$n \hat{\Sigma} = \mathbf{X}^T \mathbf{X} - n \bar{X} \bar{X}^T.$$

2. Use block matrix inversion to show that

$$\left((\mathbf{1} \ \mathbf{X})^T (\mathbf{1} \ \mathbf{X}) \right)^{-1} = \begin{pmatrix} * & * \\ * & \frac{1}{n} \hat{\Sigma}^{-1} \end{pmatrix}.$$

Here the *-s mean some values that we are not interested in computing in this exercise.

3. Show that if we fit a linear model with an intercept (which is denoted β_0) using the least squares estimator, then the variance of $\hat{\beta}_i$, conditionally on $\mathbf{X}$, for $i = 1, \dots, p$ is

$$\frac{\sigma^2}{n} (\hat{\Sigma}^{-1})_{ii}.$$

With $\hat{\Sigma}_{-1}$ denoting the empirical covariance matrix with the first row and column removed we can write $\hat{\Sigma}$ in block form as

$$\hat{\Sigma} = \begin{pmatrix} \hat{\sigma}_1^2 & \hat{\gamma}^T \\ \hat{\gamma} & \hat{\Sigma}_{-1} \end{pmatrix}.$$

4. Use block inversion (again) to show that

$$(\hat{\Sigma}^{-1})_{11} = \frac{1}{\hat{\sigma}_1^2 - \hat{\gamma}^T (\hat{\Sigma}_{-1})^{-1} \hat{\gamma}}.$$

5. Show that in the case $p = 2$, the formula above reduces to

$$(\hat{\Sigma}^{-1})_{11} = \frac{1}{\hat{\sigma}_1^2 (1 - \widehat{\text{corr}}(X_1, X_2)^2)}.$$

6. Show that if $\mathbf{X}_{\cdot 1} - \mathbf{1} \bar{X}_1$ is perpendicular to the remaining columns in $\mathbf{X} - \mathbf{1} \bar{X}^T$, then

$$(\hat{\Sigma}^{-1})_{11} = \frac{1}{\hat{\sigma}_1^2}.$$

7. Interpret the results obtained in this exercise in terms of how the empirical variance of and empirical correlation between predictor variables affect the variances of the estimators $\hat{\beta}_i$ for $i = 1, \dots, p$.

3 Birth weight—a case study

The question that we address in this case study is how birth weight of children is associated with a number of other observable variables. The dataset comes from a sub-study of The Danish National Birth Cohort Study. The Danish National Birth Cohort Study was a nationwide study of pregnant women and their offspring (Olsen et al. 2001). The included pregnant women completed a computer assisted telephone interview, scheduled to take place in pregnancy weeks 12–16 (for some, the interview took place later). We consider data on women interviewed before pregnancy week 17, who were still pregnant in week 17. One of the original purposes of the sub-study was to investigate if fever episodes during pregnancy were associated with fetal death.

Olsen, Jørn, Mads Melbye, Sjurður F. Olsen, et al. 2001. “The Danish National Birth Cohort - Its Background, Structure and Aim.” *Scandinavian Journal of Public Health* 29 (4): 300–307.

We focus on birth weight as the outcome variable of interest. If Y denotes the birth weight of a child, the objective is to find a good model of Y given a relevant set of predictor variables X . What we believe to be relevant can depend upon many things, for instance, that the variables used as predictors should be observable when we want to make a prediction. If we want to answer causal questions, e.g., does coffee drinking during pregnancy affect the birth weight of the child, we need to pay attention to assumptions required to answer such a question. Importantly, in a regression modeling context we should then also condition on confounders (often called adjusting in the literature).

In this chapter, we focus particularly on the relation between birth weight and the gestational age (age of fetus) at birth, and, given gestational age, how much the additional predictors add to the model.

3.1 Outline and prerequisites

The chapter primarily relies on the following R packages.


```
library(ggplot2)
library(tibble)
library(broom)
library(splines)
library(readr)
library(skimr)
library(dplyr)
library(RwR)
```

The chapter covers an exploratory analysis of the birth weight data included in the `RwR` package and then a number of univariate and multivariate linear regression models.

3.2 Exploratory data analysis

First, we obtain the dataset by reading it from the csv file included in the `RwR` package (Requires `RwR` version $\geq 0.3.1$). The data is also available in the package as a dataset, but it is instructive to see how it is imported from a file as well.

```
birth_weight_file <- system.file("extdata/birth_weight.txt", package = "RwR")
birth_weight <- read_table(
  birth_weight_file,
  col_types = cols(
    interviewWeek = col_character(),
    fetalDeath = col_factor(levels = c("0", "1")),
    abortions = col_factor(levels = c("0", "1", "2", "3")),
    children = col_factor(levels = c("0", "1")),
    smoking = col_factor(levels = c("1", "2", "3")),
    coffee = col_factor(levels = c("1", "2", "3")),
    feverEpisodes = col_integer()
  )
)
```

The `read_table()` function infers the column types automatically if they are not specified. For this dataset, several of the categorical variables are encoded using integers, and the column classes therefore require explicit specification above.

Mistakes are easily made if the classes of the columns in the data frame are not appropriate.

It is always a good idea to check that the data was read correctly, that the columns of the resulting data frame have the correct names and are of the correct classes, and to check the dimensions of the resulting data frame using `dim()`, `nrow()` or `ncol()`. This dataset has 12 variables and 11817 cases (rows).

interviewWeek	fetalDeath	age	abortions	children	gestationalAge	smoking	alcohol	coffee
14	0	36.73374	0	1	40	1	0.0	1
12	0	34.98700	0	1	41	3	2.0	2
13	1	33.70021	0	0	35	1	0.0	1
16	0	33.06229	0	1	38	1	4.0	2
15	0	31.83299	0	0	40	1	1.0	1
13	0	33.02943	0	1	42	1	0.5	1

Table 3.1: First six rows of the birth weight data. Some variables have missing values represented as **NA**, partly due to fetal death.

The 12 variables and their encoding in the birth weight dataset:

- **interviewWeek**: Pregnancy week at interview.
- **fetalDeath**: Indicator of fetal death (1 = death).
- **age**: Mother's age at conception in years.
- **abortions**: Number of previous spontaneous abortions (0, 1, 2, 3+).
- **children**: Indicator of previous children (1 = previous children).
- **gestationalAge**: Gestational age in weeks at end of pregnancy.
- **smoking**: Smoking status; 0, 1–10 or 11+ cigs/day encoded as 1, 2 or 3.
- **alcohol**: Number of weekly drinks during pregnancy.
- **coffee**: Coffee consumption; 0, 1–7 or 8+ cups/day encoded as 1, 2 or 3.
- **length**: Birth length in cm.
- **weight**: Birth weight in gram.
- **feverEpisodes**: Number of mother's fever episodes before interview.

3.2.1 Descriptive summaries

The first step is to summarize the variables in the dataset using simple descriptive statistics. This is to get an idea about the data and the variable ranges, but also to discover potential issues that we need to take into account in the further analysis. The list of issues we should be aware of includes, but is not limited to,

- extreme observations and potential outliers
- missing values
- skewness or asymmetry of marginal distributions

Anything worth noticing should be noticed. It should not necessarily be written down in a final report, but figures and tables should be prepared to reveal and not conceal.

A quick summary of the variables in a data frame can be obtained with the `summary()` function. It prints quantile information for numeric variables and frequencies for factor variables. This is the first example where the class of the columns matter for the result that R produces. Information on the number of missing observations for each variable is also given. The `skim()` function from the `skimr` package provides a more detailed view on the data with a better formatting, and it clearly breaks down the summary according to the variable type.

```
skim(birth_weight)
```

Variable type: character									
skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace		
interviewWeek	0	1	1	2	0	10	0		
Variable type: factor									
skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts				
fetalDeath	39	1	FALSE	2	0: 11659, 1: 119				
abortions	0	1	FALSE	4	0: 9598, 1: 1709, 2: 395, 3: 115				
children	0	1	FALSE	2	1: 6513, 0: 5304				
smoking	0	1	FALSE	3	1: 8673, 2: 1767, 3: 1377				
coffee	4	1	FALSE	3	1: 7821, 2: 3624, 3: 368				
Variable type: numeric									
skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
age	0	1.00	29.63	4.20	16.29	26.64	29.48	32.48	44.00
gestationalAge	0	1.00	39.44	2.82	17.00	39.00	40.00	41.00	47.00
alcohol	1	1.00	0.51	0.92	0.00	0.00	0.00	1.00	15.00
length	538	0.95	51.78	5.89	0.00	51.00	52.00	54.00	99.00
weight	538	0.95	3572.45	593.45	0.00	3250.00	3600.00	3950.00	6140.00
feverEpisodes	0	1.00	0.20	0.48	0.00	0.00	0.00	0.00	10.00

Table 3.2: Quick data summary using the `skim()` function from the `skimr` package.

Further investigations of the marginal distributions of the variables in the dataset can be obtained by using histograms, density estimates, tabulations and barplots. Barplots are preferable over histograms for numerical variables that take only a small number of different values, e.g., counts. This is the case for the `feverEpisodes` variable. Before such figures and tables are produced—or perhaps after they have been produced once, but before they enter a final report—we may prefer to clean the data a little.

We can observe from the data summary in Table 3.1 that for some cases weight or length is missing or registered as 0—and in some other cases weight or length is unrealistically small. These are most likely registration mistakes. Likewise, some lengths are registered as 99, and further scrutiny, e.g., by a scatter plot, reveals an observation with `weight` 3550 gram but `gestationalAge` registered as 18. We clean the data by excluding the cases with unrealistic values for some variables. This will leave four cases with missing values of coffee and one with a missing value of alcohol. We exclude those five cases as well, and we also remove the columns `interviewWeek` and `fetalDeath` as they will not be used subsequently.

```
birth_weight <- filter(  
  birth_weight,  
  weight > 32,  
  length > 10 & length < 99,  
  gestationalAge > 18  
) |>  
  na.omit() |>  
  select(- c(interviewWeek, fetalDeath))
```

We present density or histogram estimates of five numerical variables, see Figure 3.1. The density estimates, as the majority of the figures presented in this book, were produced using the `ggplot2` package. Readers familiar with ordinary R graphics can easily produce histograms with the `hist()` function or density estimates with the `density()` function. For this simple task `ggplot()` function does not offer much of an advantage—besides the fact that figures have the same style as other `ggplot` figures. However, the well-thought-out design and entire functionality of `ggplot` has resulted in plotting methods that are powerful and expressive. The benefit is that with `ggplot` it is possible to produce quite complicated

figures with clear and logical R expressions—and without the need to mess with low-level technical plotting details.

What is most noteworthy in Figure 3.1 is that the distribution of `alcohol` is extremely skewed, with more than half of the cases not drinking alcohol at all. This is noteworthy since little variation in a predictor makes it more difficult to detect whether it is associated with the outcome.

```
tmp <- lapply(
  names(birth_weight),
  \(x) ggplot(data = birth_weight[, x, drop = FALSE]) +
    aes_string(x) + xlab(x) + ylab("")
)
gd <- geom_density(adjust = 2, fill = gray(0.7))
gb <- geom_bar(fill = gray(0.7))

gridExtra::grid.arrange(
  tmp[[1]] + gd,
  tmp[[4]] + gb,
  tmp[[6]] + gb,
  tmp[[8]] + gd,
  tmp[[9]] + gd,
  ncol = 5
)
```

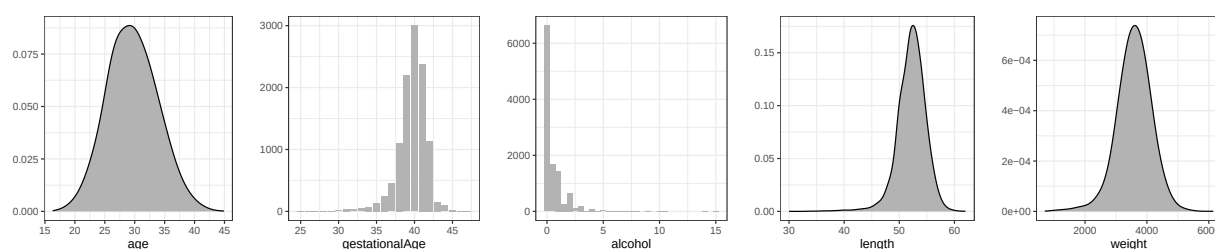


Figure 3.1: Density estimates or bar plots of continuous variables or variables taking many values.

For the categorical or discrete count variables we produce barplots instead of density estimates. Figure 3.2 shows that all these variables except `children` have quite skewed distributions.

In summary, the typical pregnant woman does not smoke or drink alcohol or coffee, nor has she had any previous spontaneous abortions or any fever episodes. About one-third drinks coffee or alcohol or smokes. These observations may not be surprising—they reflect what is to be expected for a random sample of cases. Little variation of a predictor

can, however, make estimation and detection of associations between the outcome and the predictors more difficult. In this case the dataset is quite large, and cases with the least frequently occurring predictor values are present in reasonable numbers.

```
gridExtra::grid.arrange(
  tmp[[2]] + gb,
  tmp[[3]] + gb,
  tmp[[5]] + gb,
  tmp[[7]] + gb,
  tmp[[10]] + aes(x = factor(feverEpisodes)) + gb + xlab("feverEpisodes"),
  ncol = 5
)
# The coercion of 'feverEpisodes' to a factor fixes the x-axes.
```

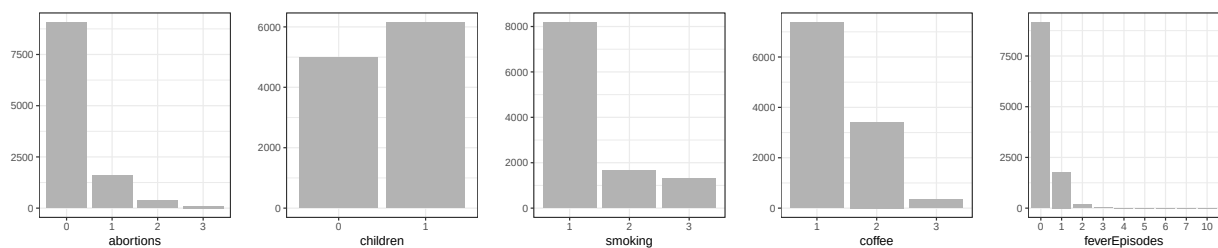


Figure 3.2: Barplots of variables taking few values.

3.2.2 Pairwise associations

The next step is to investigate associations of the variables. We are still not attempting to build a predictive model and the outcome does not yet play a special role. One purpose is again to get better acquainted with the data—this time by focusing on covariation—but there is also one particular issue that we should be aware of.

- Near collinearity of predictors.

Add this bullet point to the previous list of issues. If two or more predictors in the dataset are strongly correlated, they contain, from a predictive point of view, more or less the same information, but perhaps encoded in slightly different ways. Strongly correlated predictors result in the same problem as predictors with little variation. It can become difficult to estimate and detect joint association of the predictors with the outcome.

A technical consequence of near collinearity is that statistical tests of whether one of the predictors could be excluded become non-significant if the other is included, whereas a test of joint exclusion of the predictors can be highly significant. Thus it will become difficult to determine, on statistical grounds, if one predictor should be included over the other. It is best to know about such potential issues upfront. Perhaps it is, by subject matter arguments, possible to choose one of the predictors over the other as the most relevant to include in the model.

A scatter plot matrix is a useful graphical summary of the pairwise association of numerical variables. It can be supplemented with computations of Pearson correlations.

```
ggally_hexbin <- function(data, mapping, ...)  
  ggplot(data, mapping) + stat_binhex(...)  
  
GGally::ggpairs(  
  birth_weight,  
  columns = c("age", "alcohol", "gestationalAge", "length", "weight"),  
  lower = list(continuous = GGally::wrap("hexbin", bins = 20)),  
  diag = list(continuous = "blankDiag"),  
)
```

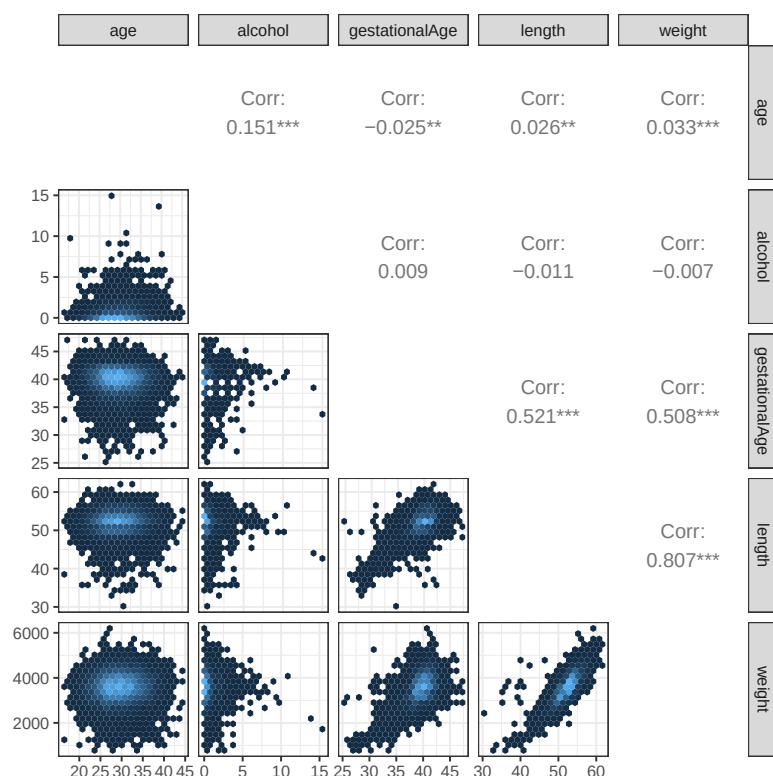


Figure 3.3: Scatter plot matrix of the numerical variables and corresponding Pearson correlations.

The scatter plots, Figure 12.1, show that `length` and `weight` are (not surprisingly) very correlated, and that both of these variables are also highly correlated with `gestationalAge`. The `alcohol` and `age` variables are mildly correlated, but they are virtually uncorrelated with the other three variables.

The scatter plot matrix was produced using the `ggpairs()` function from the `GGally` package. The dataset is quite large and a naive plot of a scatter plot matrix results in a lot of overplotting and large pdf graphics files. Figures can be saved as high-resolution png files instead of pdf files to remedy problems with file size. The actual plotting may, however, still be slow, and the information content in the plot may be limited due to the overplotting. A good way to deal with overplotting is to use hexagonal binning of data points. This was done using the custom `ggally_hexbin()` function based on the `stat_hexbin()`.

If two categorical variables are strongly dependent the corresponding vectors of dummy variable encoding of the categorical levels will be collinear. In extreme cases where only certain pairwise combinations of the categorical variables are

observed, the resulting dummy variable vectors will be perfectly collinear. Dependence between categorical variables may be investigated by cross-tabulation and formal tests of independence can, for instance, be computed. However, neither test statistics of independence nor corresponding p -values are measures of the degree of dependence—they scale with the size of the dataset and become more and more extreme for larger datasets. There is no single suitable substitute for the Pearson correlation that applies to categorical variables in general. In this particular example all the categorical variables are, in fact, ordinal. In this case we can use the Spearman correlation. The Spearman correlation is simply the Pearson correlation between the ranks of the observations. Since we only need to be able to sort observations to compute ranks, the Spearman correlation is well defined for ordinal as well as numerical variables.

```
cp <- cor(
  data.matrix(birth_weight),
  use = "complete.obs",
  method = "spearman"
)

corrplot::corrplot(
  cp,
  diag = FALSE,
  order = "hclust",
  addrect = 4,
  tl.srt = 45,
  tl.col = "black",
  tl.cex = 0.8
)
```

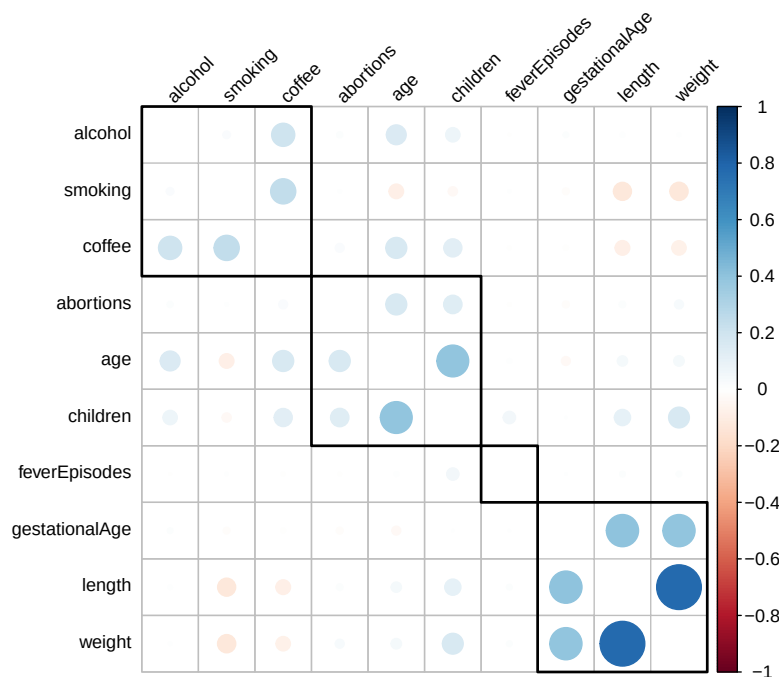


Figure 3.4: Spearman correlation matrix. Variables are ordered according to a hierarchical clustering. A rectangular grouping of variables into four clusters is shown.

Figure 3.4 shows Spearman correlations of all variables—categorical as well as numerical. For continuous variables the Spearman correlation is invariant to monotone transformations and thus less sensitive to outliers than the Pearson correlation. This invariance and its broader applicability make the Spearman correlation attractive as a means for exploratory investigations of pairwise association.

For the production of the plot of the correlation matrix, Figure 3.4, we used a hierarchical clustering of the variables. The purpose was to sort the variables so that the large correlations are concentrated around the diagonal. Since there is no natural order of the variables, the correlation matrix could be plotted using any order. We want to choose an order that brings highly correlated variables close together to make the figure easier to read. Hierarchical clustering can be useful for this purpose.

What we see most clearly from Figure 3.4 are three groupings of positively correlated variables. The **weight**, **length** and **gestationalAge** group, a group consisting of **age**, **children** and **abortions** (not surprising), and a grouping of **alcohol**, **smoking** and **coffee** with mainly coffee being correlated with the two others.

An alternative way to study the relation between a continuous and a categorical variable is to look at the distribution of the continuous variable stratified according to the values of the categorical variable. This can be done using boxplots.

Figure 3.5 shows boxplots for the continuous variable `weight`. From this figure we see that `weight` seems to be larger if the mother has had children before and to be negatively related to coffee drinking and smoking.

```
birth_weight_long <- tidyr::pivot_longer(
  birth_weight[, c("weight", "abortions", "children", "smoking", "coffee", "feverEpisodes")],
  cols = !weight,
  values_transform = as.factor
)
ggplot(birth_weight_long, aes(x = value, y = weight)) +
  geom_boxplot(fill = gray(0.8), outlier.alpha = 0.1) + xlab("") +
  facet_wrap(~ name, scale = "free_x", ncol = 5)
```

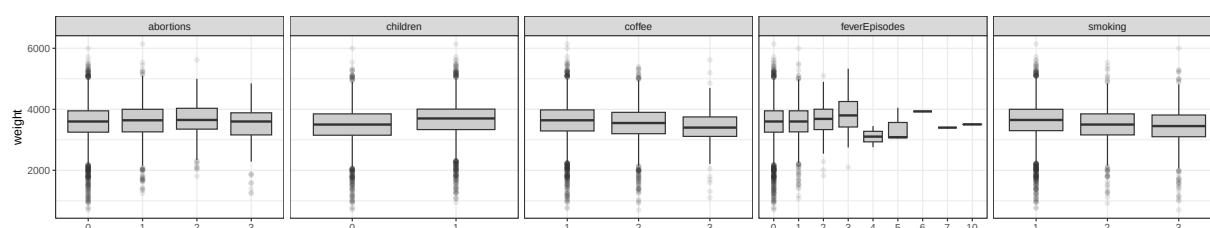


Figure 3.5: Boxplots of weight stratified according to the values of five of the other variables.

3.2.3 Birth weight and gestational age

As mentioned in the introduction, our primary interest is to build a model of the relation between gestational age at birth and the birth weight. The scatter plot of weight against gestational age, which is part of the scatter plot matrix in Figure 3.7, shows the strong marginal association. Figure 3.6 gives an alternative graphical illustration of this association.

```
p_bar <- ggplot(birth_weight, aes(as.factor(gestationalAge))) +
  geom_bar(fill = gray(0.7)) + theme(
    axis.title.x = element_blank(),
    axis.text.x = element_blank(),
  )
p_box <- ggplot(birth_weight, aes(x = as.factor(gestationalAge), y = weight)) +
  geom_boxplot(fill = gray(0.8), outlier.alpha = 0.1) + xlab("Gestational age")
gridExtra::grid.arrange(p_bar, p_box, heights = c(1, 4))
```

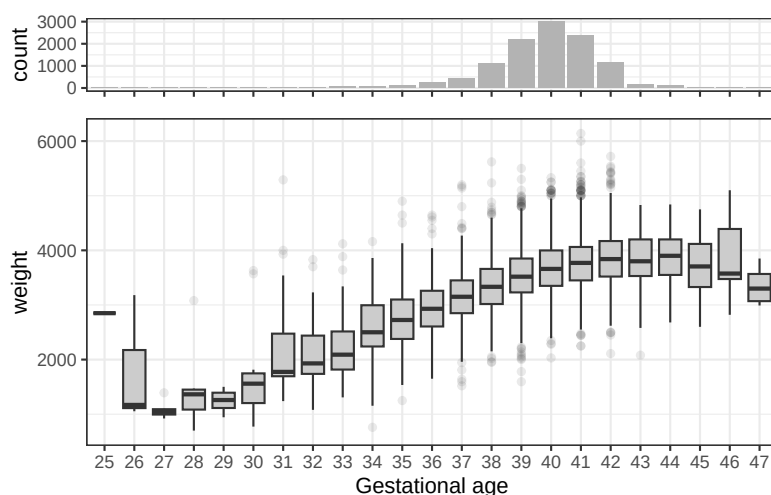


Figure 3.6: An alternative to a scatter plot, where the weight distribution is shown as a boxplot for each integer value of gestational age. The barplot above shows the marginal distribution of gestational age.

Figure 3.6 corroborates what we have already observed. There is a strong relation between birth weight and gestational age. The overall trend is also clear: birth weight increases with fetus age. However, the figure also indicates that this relation is not linear in the gestational age variable, and it even suggests that past week 44, the weight might decrease.

3.3 Linear models of birth weight

This section will build a multivariate linear regression model with **weight** as the outcome variable. Before building a joint regression model we will explore the marginal associations between the other variables and **weight** in greater details via univariate regression models with one model for each predictor.

3.3.1 Univariate regression models

We analyze the data here via a number of separate univariate linear regression models. The purpose of this analysis is not to do variable selection or draw any strong conclusions—it serves instead as a supplement to the exploratory analysis in the previous sections. In contrast to correlation considerations, the univariate regression models allow us to study single predictor associations with the outcome for categorical as well as numerical predictors, and the analysis can be generalized to models where the outcome is categorical.

The analysis uses the `add1()` function to add single predictors from a template formula (`form`) to avoid tedious repetitions of fitting the models using `lm()`.

```
form <- weight ~ gestationalAge +
  length +
  age +
  children +
  coffee +
  alcohol +
  smoking +
  abortions +
  feverEpisodes

null_model <- lm(weight ~ 1, data = birth_weight)

one_term_models <- add1(null_model, form, test = "F")
```

Table 3.3 shows the result of testing if inclusion of each of the predictors by themselves is significant. That is, we test the model with only an intercept against the alternative where a single predictor is included. The test used is the F -test—see Section 2.6.2 for details on the theory. For each of the categorical predictor variables the encoding requires `df` (degrees of freedom) dummy variables in addition to the intercept to encode the inclusion of a variable with `df + 1` levels.

```
one_term_models |>
  tidy() |>
  arrange(p.value, desc(Statistic)) |>
  filter(term != "<none>") |>
  select(-AIC) |>
  knitr::kable()
```

term	df	sumsq	rss	statistic	p.value
length	1	2351722932.3	1262184175	2.077487e+04	0.0000000
gestationalAge	1	932314502.1	2681592605	3.876542e+03	0.0000000
children	1	97619557.6	3516287550	3.095475e+02	0.0000000
smoking	2	53321111.0	3560585996	8.348023e+01	0.0000000
coffee	2	21990295.9	3591916811	3.412799e+01	0.0000000
abortions	3	6272999.9	3607634107	6.461428e+00	0.0002294
age	1	3954121.7	3609952986	1.221303e+01	0.0004764
feverEpisodes	1	1085988.3	3612821119	3.351611e+00	0.0671660
alcohol	1	169962.2	3613737145	5.244098e-01	0.4689818

term	df	sumsq	rss	statistic	p.value

Table 3.3: Marginal association tests sorted according to the p -value.

Figure 3.7 shows the scatter plots of `weight` against the 4 continuous predictors. This is actually just the last row in the scatter plot matrix in Figure 12.1, but this time we have added the regression line. For the continuous variables the tests reported in Table 3.3 are tests of whether the regression line has slope 0.

```
birth_weight_long <- tidyr::pivot_longer(
  select(birth_weight, c(age, alcohol, gestationalAge, length, weight)),
  cols = !weight,
  names_to = "variable"
)

binScale <- scale_fill_continuous(
  breaks = c(1, 10, 100, 1000),
  low = "gray80",
  high = "black",
  trans = "log",
  guide = "none"
)

ggplot(birth_weight_long, aes(value, weight)) +
  stat_binhex(bins = 20) + binScale + xlab("") +
  facet_wrap(~ variable, scales = "free_x", ncol = 4) +
  geom_smooth(linewidth = 1, method = "lm")
```

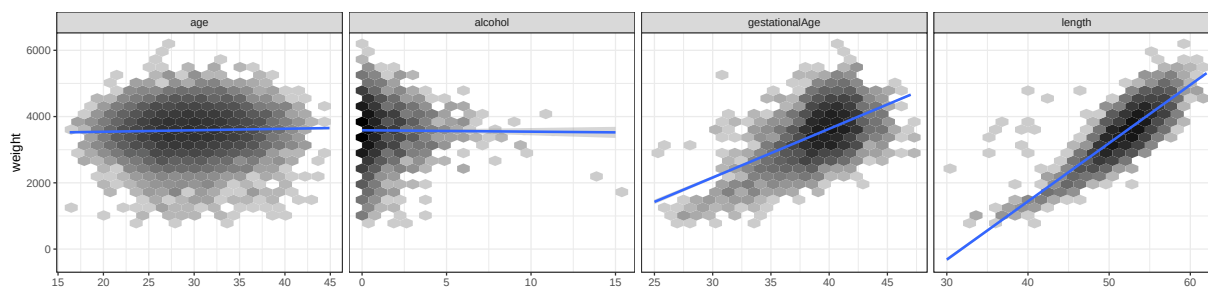


Figure 3.7: Scatter plots including linear regression lines.

Figure 3.7 illustrates the univariate linear regression models of `weight` for each of the four continuous predictor variables together with the scatter plot. It is clear from these plots that

birth weight is only weakly, if at all, marginally associated with age of the mother or alcohol consumption, while birth weight is strongly associated with gestational age and birth length.

3.3.2 A multivariate linear regression model

To build a multivariate regression model of the outcome variable `weight`, we need to decide which predictors we want to include. Our initial analyses showed that `length` is (unsurprisingly) a very good predictor of `weight`, but it is also close to being an equivalent “body size” measurement, and it will be affected in similar ways as `weight` by variables that affect fetus growth. From a predictive modeling point of view it is in most cases useless, as it will not be observable unless `weight` is also observable.

We are particularly interested in the association of `weight` with `gestationalAge`, but one might argue that `gestationalAge` is likewise of little interest if we want to predict `weight` early in pregnancy. We are, however, interested in what the other predictors add on top of `gestationalAge`. The exploratory analysis showed that `gestationalAge` is virtually unrelated to the other predictors, and the remaining predictors are not strongly correlated, and we have not found reasons to exclude any of them. We will thus fit a main effects linear model with eight predictors. We include all the predictors as they are, that is, we do not transform any of them.

The main effects model.

```
form <- update(form, . ~ . - length)
birth_weight_lm <- lm(form, data = birth_weight)
tidy(birth_weight_lm) |>
  knitr::kable()
```

term	estimate	std.error	statistic	p.value
(Intercept)	-2169.444992	98.598974	-22.0027137	0.0000000
gestationalAge	145.160059	2.303741	63.0105736	0.0000000
age	-2.002013	1.204674	-1.6618710	0.0965668
children1	185.954556	9.896906	18.7891615	0.0000000
coffee2	-65.539288	10.390137	-6.3078367	0.0000000
coffee3	-141.784319	27.241898	-5.2046418	0.0000002

term	estimate	std.error	statistic	p.value
alcohol	-2.749242	5.088446	-0.5402910	0.5890072
smoking2	-101.952291	13.049484	-7.8127448	0.0000000
smoking3	-131.188927	14.910537	-8.7984041	0.0000000
abortions1	27.842979	13.088798	2.1272373	0.0334223
abortions2	48.763060	25.445584	1.9163663	0.0553440
abortions3	-50.029260	45.803151	-1.0922668	0.2747395
feverEpisodes	6.363578	9.393500	0.6774448	0.4981378

Table 3.4: Summary table of parameter estimates, standard errors and t -tests for the linear model of weight fitted with 8 predictors.

Table 3.4 shows the estimated β -parameters among other things. Note that all categorical variables (specifically, those that are encoded as factors in the data frame) are included via a dummy variable representation. The precise encoding is determined by a linear constraint, known as a *contrast*. By default, the first factor level is constrained to have parameter 0, in which case the remaining parameters represent differences to this base level. In this case it is only occasionally of interest to look at the t -tests for testing if a single parameter is 0.

Table 3.5 shows instead F -tests of excluding any one of the predictors. It shows that the predictors basically fall into two groups; the strong predictors `gestationalAge`, `children`, `smoking` and `coffee`, and the weak predictors `abortions`, `age`, `feverEpisodes` and `alcohol`. The table was obtained using the `drop1()` function. We resist the temptation to use the tests for model reduction or model selection, as this will generally invalidate subsequent statistical analyses unless properly accounted for.

```
drop1(birth_weight_lm, test = "F") |>
  tidy() |>
  arrange(p.value, desc(statistic)) |>
  filter(term != "<none>") |>
  select(-AIC) |>
  knitr::kable()
```

term	df	sumsq	rss	statistic	p.value
gestationalAge	1	904499510.74	3442125857	3970.3323795	0.0000000

term	df	sumsq	rss	statistic	p.value
children	1	80425963.18	2618052310	353.0325909	0.0000000
smoking	2	26628256.86	2564254603	58.4428345	0.0000000
coffee	2	12916211.03	2550542557	28.3480811	0.0000000
abortions	3	2072100.55	2539698447	3.0318527	0.0280891
age	1	629181.69	2538255528	2.7618151	0.0965668
feverEpisodes	1	104551.28	2537730898	0.4589315	0.4981378
alcohol	1	66502.33	2537692849	0.2919143	0.5890072

Table 3.5: Tests of models excluding each term from the full model one at a time.

To investigate the model assumptions, we turn to model diagnostics. Several aspects of the statistical analysis presented so far rely on these assumptions, though the theory is postponed to the subsequent sections. Most notably, the distribution of the test statistics, and thus the p -values, depend on the strong set of assumptions, **A3** + **A5**. We cannot hope to prove that the assumptions are fulfilled, but we can check—mostly using graphical methods—that they are either not obviously wrong, and if they appear to be wrong, the methods should reveal how we can improve the model.

Model diagnostics for the linear model are mostly based on the residuals, which are estimates of the unobserved errors ε_i , or the *standardized* residuals, which are estimates of ε_i/σ . Plots of the standardized residuals against the fitted values, or against any one of the predictors, are useful to detect deviations from **A1** or **A2**. For **A3** we consider a qq-plot against the standard normal distribution. The assumptions **A4** or **A5** are more difficult to investigate. If we do not have a specific idea about how the errors, and thus the observations, might be correlated, it is very difficult to do anything.

There exists a plot method for `lm` objects that can produce a range of standard diagnostic plots. The method is useful for quick interactive usage, but the resulting plots are difficult to customize for publication quality. We develop alternatives based on `ggplot2`.

```
birth_weight_diag <- augment(birth_weight_lm)
p1 <- ggplot(birth_weight_diag, aes(.fitted, .std.resid)) +
  stat_binhex(bins = 20) + binScale + geom_smooth(linewidth = 1, fill = "blue") +
  xlab("fitted values") + ylab("standardized residuals")
p2 <- ggplot(birth_weight_diag, aes(gestationalAge, .std.resid)) +
  stat_binhex(bins = 20) + binScale + geom_smooth(linewidth = 1, fill = "blue") +
  xlab("gestationalAge") + ylab("")
p3 <- ggplot(birth_weight_diag, aes(sample = .std.resid)) +
  geom_abline(intercept = 0, slope = 1, color = "blue", linewidth = 1) +
  geom_qq() +
```

```
xlab("theoretical quantiles") + ylab("")
gridExtra::grid.arrange(p1, p2, p3, ncol = 3)
```

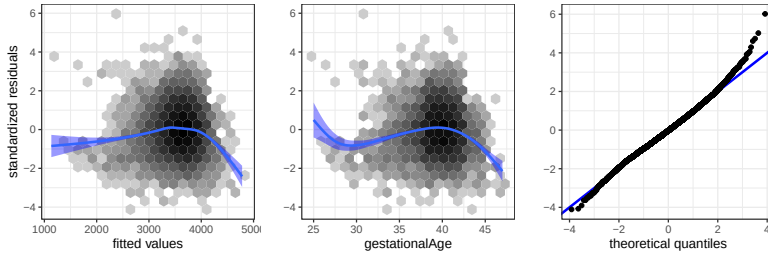


Figure 3.8: Diagnostic plots. Standardized residuals plotted against fitted values, the predictor `gestationalAge`, and a qq-plot against the normal distribution.

The residual plot in Figure 3.8 shows that the model is not spot on. The plot of the residuals against `gestationalAge` shows that there is a nonlinear effect that the linear model does not catch. Thus **A1** is not fulfilled. We address this specific issue in Section 7.3.2, where we solve the problem using splines. The qq-plot shows that the tails of the residuals are heavier than the normal distribution and right skewed. However, given the problems with **A1**, this issue is of secondary interest.

The diagnostics considered above address if the data set as a whole does not comply to the model assumptions. Single observations can also be extreme and, for instance, have a large influence on how the model is fitted. For this reason we should also be aware of single extreme observations in the residual plots and the qq-plot.

3.3.3 Including interactions

One standard way to try to improve on the multivariate linear regression model from the previous section is to include *interactions* between the predictors. The inclusion of interactions results in a substantial increase in the complexity of the models, even if we have only a few predictors. Moreover, it becomes possible to construct an overwhelming number of comparisons of models. Searching haphazardly through thousands of models with various combinations of interactions is not recommended. It will result in spurious discoveries that will be difficult to reproduce in other studies. The strategy we follow here is to include interactions of the strongest predictors from the main effects model and investigate if such interactions add any value to the model.

To comprehend an interaction model it is helpful to visualize the model to the extent it is possible. This is a point where the ggplot2 package is really strong. It supports a number of ways to stratify a plot according to different variables.

```
ggplot(birth_weight, aes(gestationalAge, weight)) +
  facet_grid(coffee ~ children + smoking, label = label_both) +
  binScale + stat_binhex(bins = 20) +
  geom_smooth(method = "lm", linewidth = 1, se = FALSE, fullrange = TRUE)
```

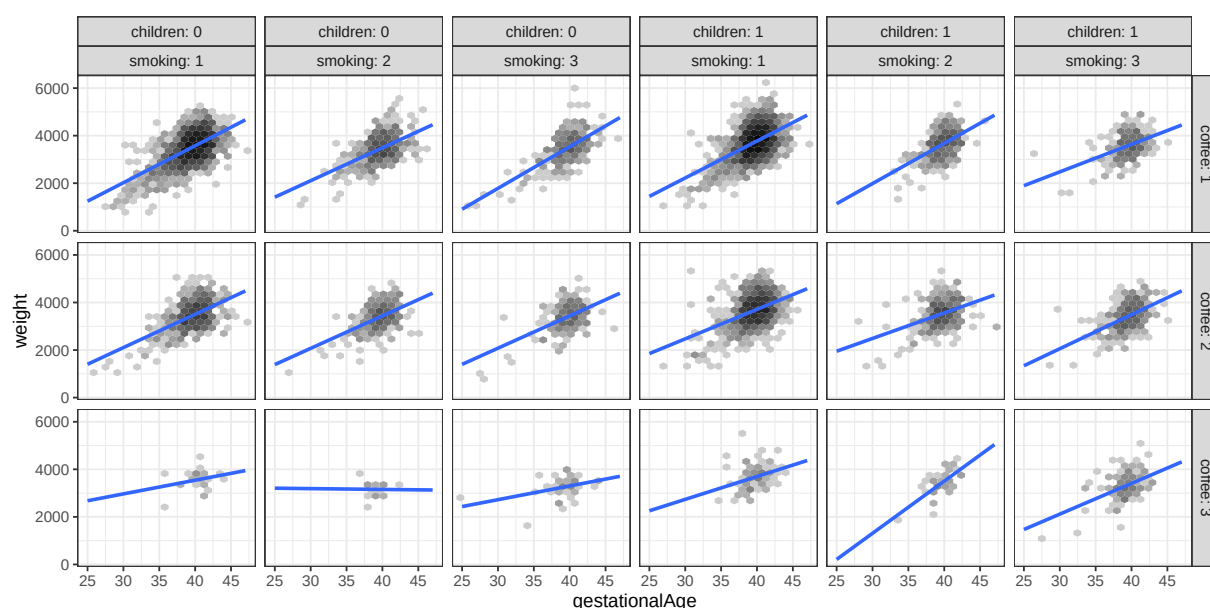


Figure 3.9: Scatter plots of **weight** against **gestationalAge** stratified according to the values of **smoking**, **children** and **coffee**

Figure 3.9 shows a total of 18 scatter plots where the stratification is according to **children**, **smoking** and **coffee**. A regression line was fitted separately for each plot. This corresponds to a model with a third order interaction between the four strong predictors (and with the weak predictors left out). Variations between the regression lines are seen across the different plots, which is an indication of interaction effects. For better comparison of the regression lines it can be beneficial to plot them differently. Figure 3.10 shows an example where the stratification according to **coffee** is visualized by color coding the levels of **coffee**.

```
ggplot(birth_weight, aes(gestationalAge, weight, color = coffee)) +
  facet_grid(. ~ children + smoking, label = label_both) +
  geom_smooth(method = "lm", linewidth = 1, se = FALSE, fullrange = TRUE)
```

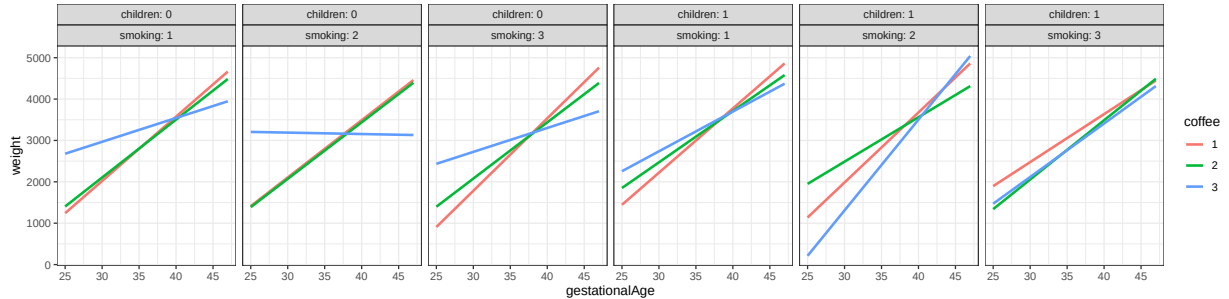


Figure 3.10: Comparison of estimated regression lines for `gestationalAge` stratified according to the values of `smoking`, `coffee` and `children`

Below we specify a linear regression model via the formula interface that includes a third order interaction among the strong predictors. The parameters are estimated via `lm()`, but we will not print out all of the 42 parameters. Instead we compare the model to the main effects model with no interactions via an *F*-test, which is done with the `anova()` function.

```
form <- weight ~ smoking * coffee * children * gestationalAge +
  age +
  alcohol +
  abortions +
  feverEpisodes

birth_weight_lm_int <- lm(form, data = birth_weight)

anova(birth_weight_lm, birth_weight_lm_int) |>
  knitr::kable()
```

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
11139	2537626346	NA	NA	NA	NA
11110	2514285831	29	23340516	3.55641	0

Table 3.6: Test of the model including a third order interaction against the main effects model.

Table 3.6 shows that the *F*-test of the third order interaction

model against the main effects model is clearly significant, that is, we can reject the null hypothesis of no interactions against the interaction model.

We reconsider model diagnostics for the extended model, where we have included the interactions. Figure 3.11 shows the residual plot. The inclusion of the interactions did not solve the earlier observed problems with the model fit. This is hardly surprising as the problem with the model appears to be related to a nonlinear relation between **weight** and **gestationalAge**. An apparent nonlinearity could be explained by interaction effects, but this would require a strong correlation between some predictors, e.g., that heavy coffee drinkers (**coffee** = 3) have large values of **gestationalAge**. We already established that this was not the case.

Since there is some lack of model fit, we should be skeptical about the conclusions from formal hypothesis tests. However, deviations from **A1** result in an increased residual variance, which will generally result in more conservative tests. That is, it will become harder to reject a null hypothesis, and thus, in this case, conclude that inclusion of the interactions is significant.

Before we conclude the analysis, we test the hypothesis where we exclude the four weak predictors together. Table 3.7 shows that the test results in a borderline *p*-value of around 5%. On the basis of this we might exclude the four weak predictors even though Table 3.5 suggested that the number of abortions is related to **weight**. The highly skewed distribution of **abortions** resulted in large standard errors and low power despite the size of the data set. In combination with the different signs on the estimated parameters in Table 3.4, depending upon whether the woman had had 1, 2 or 3+ spontaneous abortions, the study is inconclusive on how **abortions** is related to **weight**.

```
form <- weight ~ smoking * coffee * children * gestationalAge
birth_weight_lm_int_sub <- lm(form, data = birth_weight)
anova(birth_weight_lm_int_sub, birth_weight_lm_int) |>
  knitr::kable()
```

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
11116	2517158539	NA	NA	NA	NA

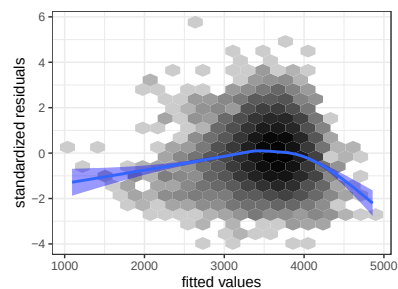


Figure 3.11: Residual plot for the third order interaction model.

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
11110	2514285831	6	2872708	2.11563	0.0482525

Table 3.7: Test of the full third order interaction model against the model excluding the 4 weak predictors.

In conclusion, we have developed a predictive model of `weight` given in terms of a third order interaction of the four predictors `gestationalAge`, `smoking`, `coffee` and `children` and additive linear effects of the four weak predictors `age`, `alcohol`, `abortions` and `feverEpisodes`. The model is not a perfect fit, as it does not catch a nonlinear relation between `weight` and `gestationalAge`. The fitted model can be visualized as in the Figure 3.9 or Figure 3.10.

We note that the formal F -test of the interaction model against the main effects model justifies the need for the increased model complexity. It is, however, clear from the figures that the actual differences in slopes are small, and the significance of the test reflects that we have a large data set, which makes us able to detect even small differences.

There is no clear-cut interpretation of the interactions either. The regression lines in the figures should, preferably, be equipped with confidence bands. This can be achieved by removing the `se = FALSE` argument to the `geom_smooth()` function, but this will result in a separate variance estimate for each combination of `smoking`, `coffee` and `children`. If we want to use the pooled variance estimate obtained by our model, we have to do something else. How this is achieved is shown in Section 7.3.2, where we also consider how to deal with the nonlinearity using spline basis expansions.

Exercises

Exercise 3.1. This exercise is on the modeling of annual wage. The data set `wage` from the R package `RwR` is from a US survey with 1125 respondents. It contains five variables. The outcome variable is `wage`, which is the respondent's annual wage in USD. In addition to `wage` the dataset includes the variables: `hours` (annual hours worked), `educ` (education, 1 = no high school diploma, 2 = high school graduate, 3

= college graduate), **age** (age of respondent in years) and **children** (number of children living with the respondent). The respondents with **hours** and **wage** equal to 0 are unemployed.

The objective of this exercise is to fit a predictive regression model of wage using some or all of the other variables in the dataset.

1. Fit and report a linear, additive model of the mean wage given some of the other variables using data for the employed respondents only.
2. Investigate how well the linear, additive model fits the data.
3. Investigate the effect of nonlinear variable transformations and inclusion of interaction effects. Choose a model to report.
4. Use the model chosen to predict the wage distribution for the unemployed respondents should they get a job. Compare with the wage distribution for the employed respondents. Discuss reasons that would invalidate such wage predictions for the unemployed.

There are few respondents with many children, and it might therefore be a good idea to collapse respondents with three or more children, say, into one group.

4 Exponential dispersion models

This chapter introduces a class of response distributions more general than the normal distribution, which will form the basis of an extension of the linear model in Chapter 5. Among the possible alternatives to the normal distribution are discrete distributions like the binomial and the Poisson distribution and continuous distributions like the Gamma distribution.

4.1 A motivating example

To motivate the introduction of the exponential dispersion models, we consider an example.

Example 4.1 (Bernoulli model). For a binary response variable $Y \in \{0, 1\}$ with $p = \mathbf{P}(Y = 1)$ the log-odds parameter is

$$\theta := \log \frac{\mathbf{P}(Y = 1)}{\mathbf{P}(Y = 0)} = \log \frac{p}{1 - p}.$$

Rearranging this equation gives

$$p = \frac{e^\theta}{1 + e^\theta} \quad \text{and} \quad 1 - p = \frac{1}{1 + e^\theta}.$$

From this,

$$\mathbf{P}(Y = y) = p^y(1 - p)^{(1-y)} = \frac{e^{y\theta}}{1 + e^\theta}$$

for $y \in \{0, 1\}$. The probability mass function thus takes the form

$$\frac{1}{c(\theta)} e^{y\theta}$$

with $c(\theta) = 1 + e^\theta$. By direct computations we may observe that

$$\mathbf{E}(Y) = p = \frac{c'(\theta)}{c(\theta)} = (\log c(\theta))'$$

and that

$$\mathbf{V}(Y) = p(1 - p) = \left(\frac{c'(\theta)}{c(\theta)} \right)'.$$

That is, the mean can be found by differentiation of $\log c$, and the variance is the derivative of the mean as a function of θ .

The identities relating the mean, variance and log-odds parameter θ for the Bernoulli model above may seem coincidental for this particular model. In fact, they are examples from the general theory of exponential families. The Bernoulli model can be parametrized by the mean value parameter (the probability p) or the log-odds parameter θ . The function

$$\theta \mapsto \frac{e^\theta}{1 + e^\theta} = p$$

is a one-to-one correspondence mapping $\mathbb{R}$ onto $(0, 1)$. The inverse of this function is the *logit function*

$$\text{logit}(p) = \log \frac{p}{1 - p}.$$

Moreover, the variance can be expressed in terms of the mean as $p(1 - p)$ or in terms of the log-odds parameter. The latter is obtainable from the function c via differentiation.

As mentioned, the Bernoulli model is an example of a univariate exponential family. The normal distribution with fixed variance is another example. This chapter investigates the class of exponential families on the real line with the extension that a dispersion parameter is also allowed. By introducing a dispersion parameter the resulting class of exponential dispersion models includes a number of two-parameter families of distributions on the real line such as the normal distribution and the Gamma distribution.

4.2 Exponential dispersion models

An exponential dispersion model is a two-parameter family of so-called exponential dispersion distributions determined by an exponential family and a dispersion parameter. The theory of exponential dispersion models provides a framework for a number of widely applied univariate distributions. The theory is interesting in itself, but it is treated here first of all to provide

the theoretical foundation for generalized linear regression models as treated in Chapter 5. In this section the models and their associated terminology are introduced, and the general relation between mean and variance is established. In subsequent chapters the statistical theory for generalized linear models is developed based on the framework of exponential dispersion models.

Every model is determined by a single σ -finite measure on the real line. With ν a σ -finite measure on $\mathbb{R}$ we define $c : \mathbb{R} \rightarrow [0, \infty]$ by

$$c(\theta) = \int e^{\theta y} \nu(dy), \quad (4.1)$$

and we let

$$I = \{\theta \in \mathbb{R} \mid c(\theta) < \infty\}^\circ$$

denote the interior of the set of θ -s for which $c(\theta) < \infty$.

Note that it is possible that $I = \emptyset$ —take, for instance, ν to be Lebesgue measure. If ν is a finite measure then $c(0) < \infty$, but it is still possible that $c(\theta) = \infty$ for all $\theta \neq 0$, which results in $I = \emptyset$. The case where I is empty is not of any relevance. There are two other special situations that are not relevant either. If ν is the zero measure, $c(\theta) = 0$, and if ν is a one-point measure, that is, $\nu = a\delta_y$ for $a \in (0, \infty)$ and δ_y the Dirac measure in y , then $c(\theta) = ae^{\theta y}$. Neither of these two cases will be of any interest, and they result in pathological problems that we want to avoid. We will therefore make the following regularity assumptions about ν throughout.

- The measure ν is not the zero measure, nor is it a one-point measure.
- The open set I is non-empty.

By the assumption that ν is not the zero measure, it follows that $c(\theta) > 0$ for $\theta \in I$. This makes the following definition possible.

Definition 4.1. The exponential family with structure measure ν is the one-parameter family of probability measures, P_θ for $\theta \in I$, defined by

$$\frac{dP_\theta}{d\nu}(y) = \frac{1}{c(\theta)} e^{\theta y}.$$

The parameter θ is called the canonical parameter.

Note that the exponential family is determined completely by the choice of the structure measure ν . Introducing $\kappa(\theta) = \log c(\theta)$ we have that

$$\frac{dP_\theta}{d\nu}(y) = e^{\theta y - \kappa(\theta)}$$

for $\theta \in I$. The function κ is called the *unit cumulant function* for the exponential family. It is closely related to the cumulant generating functions for the probability measures in the exponential family, see Exercise 4.4.

Under the regularity assumptions we have made on ν we can obtain a couple of very useful results about the exponential family.

Lemma 4.1. *The set I is an open interval, the parametrization $\theta \mapsto P_\theta$ is one-to-one, and the function $\kappa : I \mapsto \mathbb{R}$ is strictly convex.*

Proof. We first prove that the parametrization is one-to-one. If $P_{\theta_1} = P_{\theta_2}$ their densities w.r.t. ν must agree ν -almost everywhere, which implies that

$$(\theta_1 - \theta_2)y = \kappa(\theta_1) - \kappa(\theta_2)$$

for ν -almost all y . Since ν is assumed not to be the zero measure or a one-point measure, this can only hold if $\theta_1 = \theta_2$.

To prove that I is an interval let $\theta_1, \theta_2 \in I$, and let $\alpha \in (0, 1)$. Then by Hölder's inequality

$$\begin{aligned} c(\alpha\theta_1 + (1-\alpha)\theta_2) &= \int e^{\alpha\theta_1 y} e^{(1-\alpha)\theta_2 y} \nu(dy) \\ &\leq \left(\int e^{\theta_1 y} \nu(dy) \right)^\alpha \left(\int e^{\theta_2 y} \nu(dy) \right)^{1-\alpha} \\ &= c(\theta_1)^\alpha c(\theta_2)^{1-\alpha} < \infty. \end{aligned}$$

This proves that I is an interval. It is by definition open.

Finally, it follows directly from the inequality above that

$$\kappa(\alpha\theta_1 + (1-\alpha)\theta_2) \leq \alpha\kappa(\theta_1) + (1-\alpha)\kappa(\theta_2),$$

which shows that κ is convex. If we have equality in this inequality, we have equality in Hölder's inequality. This happens only if $e^{\theta_1 y}/c(\theta_1) = e^{\theta_2 y}/c(\theta_2)$ for ν -almost all y , and just as above we conclude that this implies $\theta_1 = \theta_2$. The conclusion is that κ is strictly convex. $\square$

Hölder's inequality says that

$$\int |fg| \leq \left(\int |f|^p \right)^{\frac{1}{p}} \left(\int |g|^q \right)^{\frac{1}{q}}$$

for $p, q \geq 1$ and $\frac{1}{p} + \frac{1}{q} = 1$. It is used with

$$\begin{aligned} f(y) &= e^{\alpha\theta_1 y}, \\ g(y) &= e^{(1-\alpha)\theta_2 y}, \end{aligned}$$

$$p = \frac{1}{\alpha} \text{ and } q = \frac{1}{1-\alpha}.$$

The structure measure ν determines the unit cumulant function by the formula

$$\kappa(\theta) = \log \int e^{\theta y} \nu(dy).$$

The fact that I is open implies that κ determines ν uniquely, see Exercise 4.5.

It turns out that a number of univariate exponential families can be embedded in a two-parameter family of probability measures with a common structure. This embedding exists when the structure measure belongs to a family of σ -finite measures ν_φ parametrized by $\varphi > 0$, whose cumulant functions are $\theta \mapsto \kappa(\varphi\theta)/\varphi$.

It is not at all obvious whether there exists a σ -finite measure ν_φ with cumulant function $\kappa(\varphi\theta)/\varphi$ for a given unit cumulant function κ . We will find this to be the case for all $\varphi > 0$ for several concrete examples, but there are examples, such as the Bernoulli model, where the extension is only possible for $\varphi^{-1} \in \mathbb{N}$. We will not pursue a systematic study, but see (Jørgensen 1997, chap. 4).

When the extension is possible, we have $\nu_1 = \nu$ and

$$\frac{\kappa(\varphi\theta)}{\varphi} = \log \int e^{\theta y} \nu_\varphi(dy).$$

Changing parameter from $\varphi\theta$ to θ , we can rewrite the above identity as

$$\frac{\kappa(\theta)}{\varphi} = \log \int e^{\frac{\theta y}{\varphi}} \nu_\varphi(dy) = \log \int e^{\theta y} \bar{\nu}_\varphi(dy),$$

where $\bar{\nu}_\varphi$ is a scale transformation of ν_φ . Since these cumulant functions are defined on the same open interval I as κ , they uniquely determine $\bar{\nu}_\varphi$, and thus ν_φ , for all $\varphi > 0$ by Exercise 4.5. The extension of the structure measure ν to the family ν_φ for $\varphi > 0$ is therefore unique—if there exists such an extension—and all the measures ν_φ and $\bar{\nu}_\varphi$ satisfy the same regularity conditions we have required of ν .

Whenever ν is embedded in such a family $(\nu_\varphi)_{\varphi>0}$ of structure measures we introduce the *exponential dispersion model* determined by ν —a two-parameter family of probability measures—by

$$\frac{dP_{\theta,\varphi}}{d\nu_\varphi}(y) = e^{\frac{\theta y - \kappa(\theta)}{\varphi}}. \quad (4.2)$$

Jørgensen, B. 1997. *The Theory of Dispersion Models*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Taylor & Francis.

The parameter φ is called the dispersion parameter, and we call $\nu = \nu_1$ the unit structure measure for the exponential dispersion model.

For fixed φ the exponential dispersion model is an exponential family with structure measure ν_φ and canonical parameter θ/φ . We abuse the terminology slightly and call θ the canonical parameter for the exponential dispersion model. Thus, whether θ/φ or θ is canonical depends upon whether we regard the model as an exponential family with structure measure ν_φ or as an exponential dispersion model with unit structure measure ν . Whenever we consider an exponential dispersion model, the measure ν will always denote the unit structure measure ν_1 .

Definition 4.2. The probability distribution $P_{\theta,\varphi}$ is called the (θ, ν_φ) -exponential dispersion distribution and is denoted $\mathcal{E}(\theta, \nu_\varphi)$.

In practice we check that a given parametrized family of distributions is, in fact, an exponential dispersion model by checking that its density can be brought on the form (4.2).

Example 4.2. The normal distribution $\mathcal{N}(\mu, 1)$ has density

$$\frac{1}{\sqrt{2\pi}} e^{-\frac{(y-\mu)^2}{2}} = e^{y\mu - \frac{\mu^2}{2}} \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}}$$

w.r.t. Lebesgue measure m . We identify this as an exponential family with¹⁰

$$\theta = \mu, \quad \kappa(\theta) = \frac{\theta^2}{2} \quad \text{and} \quad \frac{d\nu}{dm}(y) = \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}}.$$

Thus the structure measure ν is the standard normal distribution $\mathcal{N}(0, 1)$.

Introducing a dispersion parameter $\varphi > 0$ we observe that

$$\frac{\kappa(\theta)}{\varphi} = \frac{1}{\varphi} \frac{\theta^2}{2} = \frac{(\theta/\sqrt{\varphi})^2}{2} = \log \int e^{\frac{\theta y}{\sqrt{\varphi}}} \nu(dy) = \log \int e^{\frac{\theta y}{\varphi}} \nu_\varphi(dy)$$

where ν_φ is a scale transformation¹¹ of ν . That is, ν_φ is the $\mathcal{N}(0, \varphi)$ distribution with

$$\frac{d\nu_\varphi}{dm}(y) = \frac{1}{\sqrt{2\pi\varphi}} e^{-\frac{y^2}{2\varphi}}.$$

The parameters φ and θ are only uniquely determined up to a scale transformation, which corresponds to specifying which of the structure measures ν_φ is the unit structure measure.

¹⁰ The κ and ν are not unique. The constant

$$\frac{1}{\sqrt{2\pi}}$$

in ν can be moved to a $-\log \sqrt{2\pi}$ term in κ .

¹¹ With $S_\sigma(x) = \sigma x$ denoting a scale transformation, $\nu_\varphi = S_{\sqrt{\varphi}}(\nu)$.

The corresponding exponential dispersion distribution is $\mathcal{N}(\mu, \varphi)$. With the notation introduced in Definition 4.2,

$$\mathcal{E}(\mu, \nu_{\sigma^2}) = \mathcal{N}(\mu, \sigma^2),$$

where we, as is usual for the normal distribution, denote the dispersion parameter by σ^2 .

Example 4.3. The Poisson distribution with mean $\mu > 0$ has density (probability mass function)

$$e^{-\mu} \frac{\mu^y}{y!}$$

w.r.t. the counting measure τ for $y \in \mathbb{N}_0$. With $\theta = \log \mu$ we can rewrite this density as

$$e^{y \log \mu - \mu} \frac{1}{y!} = e^{y\theta - e^\theta} \frac{1}{y!},$$

and we identify the distribution as an exponential family with

$$\theta = \log \mu, \quad \kappa(\theta) = e^\theta \quad \text{and} \quad \frac{d\nu}{d\tau}(y) = \frac{1}{y!}.$$

We conclude that

$$\mathcal{E}\left(\log \mu, \frac{1}{y!} \cdot \tau\right) = \text{Pois}(\mu).$$

The next example is the Γ -family of distributions, whose representation as an exponential dispersion model is less obvious. These distributions can be parametrized by a shape and a scale parameter, and we have to rewrite their density to identify the parametrization in terms of the canonical parameter and the dispersion parameter. For this family of distributions there is yet another useful parametrization in terms of mean and variance.

Example 4.4. The Γ -distribution with shape parameter $\lambda > 0$ and scale parameter $\alpha > 0$ has density

$$\frac{1}{\alpha^\lambda \Gamma(\lambda)} y^{\lambda-1} e^{-y/\alpha} = e^{\frac{-y/(\lambda\alpha) - \log(\lambda\alpha)}{1/\lambda}} \frac{\lambda^\lambda}{\Gamma(\lambda)} y^{\lambda-1}$$

It is possible to introduce a dispersion parameter via a scaled Poisson distribution, see Exercise 4.8, but the resulting distribution will only have integer support if $\varphi \in \mathbb{N}$. It is therefore not useful in general for modeling counts.

w.r.t. Lebesgue measure m on $(0, \infty)$. We identify this family of distributions as the exponential dispersion model with dispersion parameter $\varphi = 1/\lambda$, canonical parameter

$$\theta = -1/(\lambda\alpha) < 0,$$

$$\kappa(\theta) = -\log(-\theta),$$

and structure measure given by

$$\frac{d\nu_\varphi}{dm} = \frac{1}{\varphi^{1/\varphi}\Gamma(1/\varphi)} y^{\frac{1}{\varphi}-1}$$

on $(0, \infty)$. We have $I = (-\infty, 0)$. The mean value of the Γ -distribution is $\mu := \alpha\lambda = -\frac{1}{\theta}$, and the variance is $\alpha^2\lambda = \varphi\mu^2 = \varphi/\theta^2$. Observe that the unit structure measure, ν_1 , is Lebesgue measure on $(0, \infty)$, and the corresponding exponential family is the exponential distribution.

The next theorem shows that the mean and variance for exponential dispersion models are always computable via differentiation of the cumulant function.

For the Γ -distribution:

Theorem 4.1. *The function κ is infinitely often differentiable on I . If $Y \sim \mathcal{E}(\theta, \nu_\varphi)$ for $\theta \in I$ then*

$$\kappa'(\theta) = -\frac{1}{\theta}$$

and

$$\kappa''(\theta) = \frac{1}{\theta^2}.$$

$$\mathbf{E}(Y) = \kappa'(\theta) \tag{4.3}$$

and

$$\mathbf{V}(Y) = \varphi\kappa''(\theta). \tag{4.4}$$

Proof. It follows by a suitable domination argument that, for $\theta \in I$,

$$\frac{d^n}{d\theta^n} c(\theta) = \int y^n e^{\theta y} \nu(dy) = c(\theta) \mathbf{E}(Y^n).$$

Since $\kappa(\theta) = \log c(\theta)$, it follows that also κ is infinitely often differentiable. In particular, if $Y \sim \mathcal{E}(\theta, \nu_1)$,

$$\mathbf{E}(Y) = \frac{c'(\theta)}{c(\theta)} = (\log c)'(\theta) = \kappa'(\theta),$$

and

$$\mathbf{E}(Y^2) = \frac{c''(\theta)}{c(\theta)},$$

and we find that

$$\mathbf{V}(Y) = \mathbf{E}(Y^2) - (\mathbf{E}(Y))^2 = \frac{c''(\theta)c(\theta) - c'(\theta)^2}{c(\theta)^2} = \kappa''(\theta).$$

The general case for $\varphi \neq 1$ can be proved by similar arguments. Alternatively, observe that for an arbitrary dispersion parameter $\varphi > 0$, the distribution of Y is an exponential family with canonical parameter $\theta_0 = \theta/\varphi$ and with corresponding cumulant function

$$\kappa_0(\theta_0) = \frac{\kappa(\varphi\theta_0)}{\varphi}.$$

The conclusion follows by differentiating κ_0 twice w.r.t. θ_0 . $\square$

We know from Lemma 4.1 that κ is strictly convex. Since κ was found to be differentiable on I , this implies that κ' is a strictly increasing function. This can also be seen directly from the previous theorem. Indeed, by the regularity assumptions on ν_φ , $P_{\theta,\varphi}$ is not a Dirac measure, which implies that $\mathbf{V}(Y) > 0$ in (4.4), see Exercise 4.3. Hence κ'' is strictly positive, and it follows that κ' is strictly increasing, and that κ is strictly convex.

That κ' is continuous and strictly increasing imply that the range of κ' ,

$$J := \kappa'(I),$$

is an open interval. This range is the range of possible mean values for the exponential dispersion model. Since κ' bijectively maps I onto J we can always express the variance in terms of the mean.

Definition 4.3. The variance function $\mathcal{V} : J \rightarrow (0, \infty)$ is defined as

$$\mathcal{V}(\mu) = \kappa''((\kappa')^{-1}(\mu)).$$

From the considerations above, the map

$$(\theta, \varphi) \mapsto (\kappa'(\theta), \varphi\kappa''(\theta)) = (\mu, \varphi\mathcal{V}(\mu))$$

is a bijection from $I \times (0, \infty)$ onto $J \times (0, \infty)$, and this shows that the exponential dispersion model can be parametrized

Theorem 4.1 only implies that κ is (strictly) convex because I is already known to be an interval. The argument based on Hölder's inequality is necessary to show this.

either in terms of the canonical and dispersion parameters or in terms of the mean and variance.

Occasionally, a third parametrization exists of an exponential dispersion model that may be useful or even natural, depending on the model.

With $H \subseteq \mathbb{R}$ and $\theta : H \rightarrow I$, the canonical parameter, $\theta(\eta)$, is in such cases expressed in terms of an *arbitrary* parameter $\eta \in H$. We call η arbitrary because we do not make any assumptions about what this parameter is¹², besides that it parametrizes the exponential dispersion model. The density for the exponential dispersion model w.r.t. ν_φ in the arbitrary parametrization is

$$e^{\frac{\theta(\eta)y - \zeta(\eta)}{\varphi}}$$

where $\zeta(\eta) = \kappa(\theta(\eta))$ and $\eta \in H$. The parametrization, $\eta \mapsto \theta(\eta)$, is allowed to depend upon additional (nuisance) parameters. We suppress any such dependences in the abstract notation. It is, however, not allowed to depend upon the dispersion parameter¹³, because then the dispersion parameter cannot be eliminated from subsequent estimation equations.

The *mean value function*, in the arbitrary parametrization, is given as

$$\mu(\eta) = \kappa'(\theta(\eta)).$$

We can then express the mean and the variance functions in terms of the function $\eta \mapsto \zeta(\eta)$.

Corollary 4.1. *If θ is twice differentiable as a function of η with $\theta'(\eta) \neq 0$, then*

$$\mu(\eta) = \frac{\zeta'(\eta)}{\theta'(\eta)}$$

and

$$\mathcal{V}(\mu(\eta)) = \frac{\zeta''(\eta)\theta'(\eta) - \zeta'(\eta)\theta''(\eta)}{\theta'(\eta)^3} = \frac{\mu'(\eta)}{\theta'(\eta)}.$$

Proof. From the definition $\zeta(\eta) = \kappa(\theta(\eta))$ we get by differentiation and (4.3) that

$$\zeta'(\eta) = \kappa'(\theta(\eta))\theta'(\eta) = \mu(\eta)\theta'(\eta),$$

¹² In regression applications, η will be $X^T\beta$, known as the linear predictor, for a vector of predictors X .

¹³ In technical terms, the dispersion parameter must be *variation independent* of nuisance parameters entering $\eta \mapsto \theta(\eta)$.

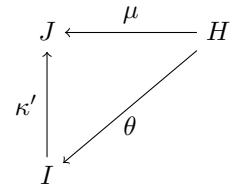


Figure 4.1: Relations between the three parametrizations.

and the first identity follows. An additional differentiation yields

$$\begin{aligned}\zeta''(\eta) &= \kappa''(\theta(\eta))\theta'(\eta)^2 + \kappa'(\theta(\eta))\theta''(\eta) \\ &= \mathcal{V}(\mu(\eta))\theta'(\eta)^2 + \frac{\zeta'(\eta)\theta''(\eta)}{\theta'(\eta)},\end{aligned}$$

and the second identity follows by isolating $\mathcal{V}(\mu(\eta))$. $\square$

Example 4.5. The canonical parameter in the Γ -model can be parametrized by $H = \mathbb{R}$ via the parametrization

$$\theta(\eta) = -e^{-\eta}.$$

Then $\zeta(\eta) = \eta$ and from Corollary 4.1 it follows that

$$\mu(\eta) = e^\eta.$$

When the mean value function, defined above as a function of the arbitrary parameter, is bijective, its inverse is called *the link function* and denoted g . It maps the mean value μ to the arbitrary parameter η , that is,

$$\eta = g(\mu).$$

The choice of the link function (equivalently, the mean value function) completely determines the θ -map and vice versa. The choice for which $\theta = \eta$ plays a particularly central role. If $\theta = \eta$ then $\mu = \kappa'$, and the corresponding link function is $g = (\kappa')^{-1}$.

Definition 4.4. The canonical link function is the link function

$$g = (\kappa')^{-1}.$$

The variance function is

$$\mathcal{V}(\mu) = \kappa''(g(\mu))$$

in terms of the canonical link g .

Example 4.6. For the Poisson distribution we have $\kappa(\theta) = e^\theta$, see Example 4.3. This implies that $\kappa'(\theta) = e^\theta$ and the canonical link function is, for the Poisson distribution, $g(\mu) = \log \mu$.

A regression model with $\eta = X^T\beta$ and a log-link function is often called a log-linear model. Thus the canonical link function for the Poisson distribution gives a log-linear regression model. Example 4.5 gives a log-linear regression model with a Γ -distributed response, but the log-link is not the canonical link for the Γ -distribution as the following example will reveal.

Example 4.7. For the Γ -model we have from Example 4.4 that $\kappa(\theta) = -\log(-\theta)$ and

$$\kappa'(\theta) = -\frac{1}{\theta}.$$

This gives that the canonical link function for the Γ -model is

$$g(\mu) = -\frac{1}{\mu}.$$

The canonical link for the Γ -model is not that attractive. Its range is $I = (-\infty, 0)$, and not the entire real line, which can be a nuisance. In practice, the Γ -model is often used with the log-link instead.

In generalized linear regression models, the response distribution will be specified in terms of the linear predictor $\eta = X^T\beta$, that determines the mean, and an exponential dispersion model, that determines the remaining aspects of the distribution. The specification of the mean is given in terms of the mean value function, or equivalently in terms of the link function. That is, with link function g ,

$$g(\mathbf{E}(Y \mid X)) = \eta = X^T\beta$$

or

$$\mathbf{E}(Y \mid X) = \mu(\eta).$$

This specifies implicitly the canonical parameter in the exponential dispersion model, and results in a model with

$$\mathbf{V}(Y \mid X) = \varphi\mathcal{V}(\mu(\eta))$$

where $\mathcal{V}$ is the variance function.

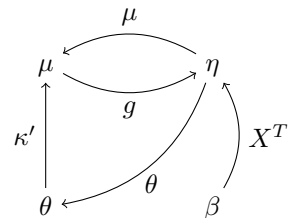


Figure 4.2: The linear predictor enters the exponential dispersion model through the mean value function.

4.3 Deviance

The (unit) *deviance* for an exponential dispersion model generalizes the squared error

$$(y - \mu)^2$$

for the $\mathcal{N}(\mu, 1)$ distribution. For a unit cumulant function $\kappa : I \rightarrow \mathbb{R}$ we let

$$\ell_y(\theta) = \theta y - \kappa(\theta)$$

for $y \in \bar{J}$ and $\theta \in I$ denote the log-likelihood of the corresponding exponential family, see also Lemma 5.1 in Chapter 5. The deviance is defined as twice the negative log-likelihood up to an additive constant. In terms of the canonical parameter the *deviance* is defined as

$$2 \left(\sup_{\theta_0 \in I} \ell_y(\theta_0) - \ell_y(\theta) \right)$$

for $y \in \bar{J}$ and $\theta \in I$. The supremum in the definition is attained in $g(y)$ for $y \in J$ where $g = (\kappa')^{-1}$ is the canonical link. For practical computations it is more convenient to express the deviance in terms of the mean value parameter than the canonical parameter.

Definition 4.5. With g denoting the canonical link function, the unit¹⁴ deviance is defined as

$$d(y, \mu) = 2 \left(\sup_{\mu_0 \in J} \left\{ g(\mu_0)y - \kappa(g(\mu_0)) \right\} - g(\mu)y + \kappa(g(\mu)) \right)$$

for $y \in \bar{J}$ and $\mu \in J$.

For $y \in J$, the supremum is attained in $\mu_0 = y$ and the unit deviance can be expressed as

$$d(y, \mu) = 2 (y(g(y) - g(\mu)) - \kappa(g(y)) + \kappa(g(\mu))). \quad (4.5)$$

Note that by the definition of the unit deviance we have that $d(y, y) = 0$ and

$$d(y, \mu) > 0$$

for $\mu \neq y$.

The deviance has simple analytic expressions for many concrete examples, which are interpretable as measures of how the observation y deviates from the expectation μ .

¹⁴ Here *unit* refers to the dispersion parameter being 1.

Example 4.8. For the normal distribution the canonical link is the identity and $\kappa(\theta) = \theta^2/2$, hence

$$d(y, \mu) = 2(y(y - \mu) - y^2/2 + \mu^2/2) = (y - \mu)^2.$$

Example 4.9. For the Poisson distribution the canonical link is the logarithm, and $\kappa(\theta) = e^\theta$, hence

$$d(y, \mu) = 2(y(\log y - \log \mu) - y + \mu) = 2(y \log(y/\mu) - (y - \mu))$$

for $y, \mu > 0$. It is clear from the definition that $d(0, \mu) = 2\mu$, and the identity above can be maintained even for $y = 0$ by the convention $0 \log 0 = 0$.

Example 4.10. For the binomial distribution it follows directly from the definition that for $y = 1, \dots, m-1$ and $\mu \in (0, m)$

$$\begin{aligned} d(y, \mu) &= 2 \left(y \log y/m + (m-y) \log(1-y/m) \right. \\ &\quad \left. - y \log \mu/m - (m-y) \log(1-\mu/m) \right) \\ &= 2 \left(y \log(y/\mu) + (m-y) \log((m-y)/(m-\mu)) \right). \end{aligned}$$

Again, by the convention $0 \log 0 = 0$ the identity is seen to extend also to the extreme cases $y = 0$ or $y = m$.

The unit deviance is approximately a quadratic form for $y \approx \mu$.

Theorem 4.2. *For the unit deviance it holds that*

$$d(y, \mu) = \frac{(y - \mu)^2}{\mathcal{V}(\mu)} + o((y - \mu)^2)$$

for $y, \mu \in J$.

Proof. We consider the function

$$y \mapsto d(y, \mu)$$

around μ . Since we know that $d(\mu, \mu) = 0$, that $y = \mu$ is a local minimum, and that d is twice continuously differentiable in $J \times J$, we get by Taylor's formula that

$$d(y, \mu) = \frac{1}{2} \partial_y^2 d(\mu, \mu) (y - \mu)^2 + o((y - \mu)^2).$$

Using that $\partial_y \kappa(g(y)) = yg'(y)$ we find that

$$\begin{aligned} \frac{1}{2} \partial_y^2 d(y, \mu) &= \partial_y^2 \left\{ yg(y) - \kappa(g(y)) - yg(\mu) + \kappa(g(\mu)) \right\} \\ &= \partial_y \left\{ g(y) + yg'(y) - yg'(y) - g(\mu) \right\} \\ &= g'(y) = \frac{1}{\kappa''(g(y))} = \frac{1}{\mathcal{V}(y)}. \end{aligned}$$

Plugging in $y = \mu$ completes the proof. $\square$

4.3.1 Bregman divergences and quasi-deviances

There is a close connection between the unit deviances introduced above and the concept of Bregman divergences, which are general distance measures determined by convex functions. Indeed, the unit deviance is effectively twice the Bregman divergence determined by κ .

Definition 4.6. Let $I \subseteq \mathbb{R}$ be an open interval and let $K : I \rightarrow \mathbb{R}$ be a continuously differentiable and strictly convex function. Then its associated Bregman divergence is defined by

$$D_K(\theta, \theta_0) = K'(\theta_0)(\theta_0 - \theta) - K(\theta_0) + K(\theta)$$

for $\theta, \theta_0 \in I$.

Strict convexity of K implies that $D_K(\theta, \theta_0) \geq 0$ and $D_K(\theta, \theta_0) = 0$ if and only if $\theta = \theta_0$. This justifies interpreting D_K as a distance measure, though it is neither symmetric nor does it fulfill the triangle inequality, so D_K is not a metric.

With $\theta = g(\mu)$ and $\theta_y = g(y)$ we see that $\kappa'(\theta_y) = g^{-1}(g(y)) = y$, and it then follows directly from Equation 4.5 that

$$d(y, \mu) = 2D_\kappa(\theta, \theta_y)$$

in terms of the canonical parameters θ and θ_y .

For any twice continuously differentiable and strictly convex K , we can define the quasi-link and quasi-variance functions by

$$g_K(\mu) := (K')^{-1}(\mu) \quad \text{and} \quad \mathcal{V}_K(\mu) := K''(g_K(\mu))$$

for $\mu \in J := K'(I)$. Here we have used that strict convexity of K implies that K' is strictly increasing, whence has an inverse on J . Note that J is an open interval.

We can express the Bregman divergence given by K in the mean value parametrization as

$$d_K(y, \mu) := 2D_K(g_K(\mu), g_K(y))$$

for $y, \mu \in J$. The following theorem gives a representation of any Bregman divergence in terms of its corresponding quasi-variance function.

Theorem 4.3. *With $\mathcal{V}_K$ a strictly positive quasi-variance function for a twice continuously differentiable and strictly convex K it holds that*

$$d_K(y, \mu) = 2 \int_{\mu}^y \frac{y-t}{\mathcal{V}_K(t)} dt$$

for $y, \mu \in J$.

Proof. Using that $K'(g_K(y)) = y$, the definition of d_K gives that

$$d_K(y, \mu) = y(g_K(y) - g_K(\mu)) - K(g_K(y)) + K(g_K(\mu)).$$

Differentiation twice gives, exactly as in the proof of Theorem 4.2, that

$$\frac{1}{2} \partial_y^2 d_K(y, \mu) = \frac{1}{\mathcal{V}_K(y)}.$$

It then follows by integrating twice, and changing the order of integration, that

$$d_K(y, \mu) = 2 \int_{\mu}^y \int_{\mu}^s \frac{1}{\mathcal{V}_K(t)} dt ds + ay + b = 2 \int_{\mu}^y \frac{y-t}{\mathcal{V}_K(t)} dt + ay + b.$$

Since $\partial_y d_K(\mu, \mu) = d_K(\mu, \mu) = 0$ we conclude that $a = b = 0$. $\square$

Exercise 4.10 shows that *any* continuous and strictly positive $\mathcal{V} : J \rightarrow \mathbb{R}$, defined on an open interval J , is a quasi-variance function corresponding to Bregman divergence. The quasi-deviance, defined by

$$d(y, \mu) = 2 \int_{\mu}^y \frac{y-t}{\mathcal{V}(t)} dt, \quad (4.6)$$

equals twice the Bregman divergence by Theorem 4.3 and is identical to the deviance given by Equation 4.5 if $\mathcal{V}$ is the variance function for an exponential dispersion model.

The upshot of this chapter is that for a (quasi-)variance function $\mathcal{V} : J \rightarrow \mathbb{R}$, specifying the relation between the mean and the variance of the outcome, there is a corresponding (quasi-)deviance, expressed by Equation 4.6, that can be used to quantify the deviation of the observation y from the mean μ . If the variance function happens to be the variance function of an exponential dispersion model, the deviance equals twice the negative log-likelihood up to an additive constant. And through the variance function a full probabilistic model of the outcome is specified in terms of the mean up to the dispersion parameter. Depending on what we want to achieve in practice, it may or may not be important to have a full probabilistic model, and Equation 4.6 shows that only the mean-variance relation is actually needed to define the deviance.

Exercises

Exercise 4.1. The probability mass function for any probability measure on $\{0, 1\}$ can be written as

$$p^y(1-p)^{1-y}$$

for $y \in \{0, 1\}$. Show that this family of probability measures for $p \in (0, 1)$ form an exponential family. Identify I and how the canonical parameter depends on p .

Exercise 4.2. Show that the binomial distributions on $\{0, 1, \dots, n\}$ with success probabilities $p \in (0, 1)$ form an exponential family.

Exercise 4.3. Show that if the structure measure $\nu = \delta_y$ is the Dirac-measure in y then for the corresponding exponential family we have $P_\theta = \delta_y$ for all $\theta \in \mathbb{R}$. Show next, that if ν is not a one-point measure, then P_θ is not a Dirac-measure for $\theta \in I$ and conclude that its variance is strictly positive.

Exercise 4.4. If P is a probability measure on $\mathbb{R}$, its cumulant generating function is defined as the function

$$K(s) = \log \int e^{sy} P(dy)$$

for s such that the integral is finite. Let $P_{\theta, \varphi}$ be an exponential dispersion distribution with unit cumulant function κ . Show that

$$K_{\theta, \varphi}(s) = \frac{\kappa(\theta + \varphi s) - \kappa(\theta)}{\varphi}$$

is the cumulant generating function for $P_{\theta, \varphi}$.

Exercise 4.5. Let P_θ be the exponential family given by the structure measure ν . Show that

$$\frac{d\nu}{dP_{\theta_0}} = e^{\kappa(\theta_0) - \theta y}$$

for any $\theta_0 \in I$. Then show that ν is uniquely specified by the unit cumulant function κ .

Hint: Use that the cumulant generating function for P_{θ_0} is determined by κ , and that a probability measure is uniquely determined by its cumulant generating function if it is defined on an open interval around 0.

Exercise 4.6. This exercise is about the *inverse Gaussian distribution*, which has density

$$f(y) = \sqrt{\frac{\lambda}{2\pi y^3}} e^{-\frac{\lambda(y-\mu)^2}{2\mu^2 y}} \quad (4.7)$$

w.r.t. Lebesgue measure on $(0, \infty)$. In this parametrization, $\lambda, \mu > 0$ are positive parameters.

1. Show that the family of inverse Gaussian distributions parametrized by $\lambda, \mu > 0$ is an exponential dispersion model. Identify how the canonical parameter and the dispersion parameter are given in terms of λ and μ —with the lack of uniqueness resolved by letting the unit structure measure correspond to $\lambda = 1$.
2. Show that the mean is μ and show that the variance function is

$$\mathcal{V}(\mu) = \mu^3.$$

3. Find the canonical link function and the unit deviance.

The use of μ in the “Wikipedia” parametrization in Equation 4.7 is sensible!

Exercise 4.7. This exercise is about the exponential family with structure measure $\nu = m_{[-1,1]}$ being Lebesgue measure restricted to the interval $[-1, 1]$.

1. Compute the unit cumulant function κ and show that $I = \mathbb{R}$. Plot the density for the corresponding exponential family w.r.t. Lebesgue measure for $\theta = -1$ and $\theta = 1$.

Assume that $Y \sim P_\theta$ where

$$\frac{dP_\theta}{d\nu} = e^{\theta y - \kappa(\theta)}.$$

2. Compute the mean value function $\mu(\theta)$ and show that

$$\mathbf{V}(Y) = \frac{1}{\theta^2} - \frac{1}{\sinh(\theta)^2}$$

for $\theta \neq 0$. Plot the variance function $\mathcal{V}(\mu)$ as a function of μ .

Hint: For the plot, it is not necessary to compute an analytic expression for the inverse of the mean value map.

Exercise 4.8. The unit cumulant function for the Poisson model is $\kappa(\theta) = e^\theta$, see Example 4.3. Let $S_\varphi(y) = \varphi y$ denote the scale transformation determined by $\varphi > 0$.

1. Show that

$$\frac{\kappa(\theta)}{\varphi} = \log \int e^{\theta y} \bar{\nu}_\varphi(dy) = \log \int e^{\frac{\theta y}{\varphi}} \nu_\varphi(dy)$$

with

$$\frac{d\bar{\nu}_\varphi}{d\tau}(y) = \frac{1}{\varphi^y y!}.$$

and $\nu_\varphi = S_\varphi(\bar{\nu}_\varphi)$.

2. Argue that the Poisson model in Example 4.3 can be extended to an exponential dispersion model where the support of the probability distribution $P_{\theta, \varphi}$ is $\varphi \mathbb{N}_0$.
3. Show that $S_{\varphi^{-1}}(P_{\theta, \varphi}) = \text{Pois}(e^\theta/\varphi)$. That is, $Y \sim \text{Pois}(\mu/\varphi)$ if and only if $\varphi Y \sim \mathcal{E}(\log \mu, \nu_\varphi)$, which shows that the members of the exponential dispersion model are scaled Poisson distributions.

Hint: You can compute the probability mass function explicitly for the distribution $S_{\varphi^{-1}}(P_{\theta, \varphi})$ and identify it as the probability mass function for a Poisson distribution. Alternative, you can compute the cumulant generating function of $P_{\theta, \varphi}$ using Exercise 4.4 and identify it as the cumulant generating function for a scaled Poisson distribution.

4. Compute $\mathbf{E}(\varphi Y)$ and $\mathbf{V}(\varphi Y)$ for $\varphi Y \sim \mathcal{E}(\log \mu, \nu_\varphi)$ directly, using that $Y \sim \text{Pois}(\mu/\varphi)$, and demonstrate that the results agree with Theorem 4.1.

Exercise 4.9. Let $N \sim \text{Pois}(\mu_0)$ and let $Y_1, Y_2, \dots \in \mathbb{N}_0$ be i.i.d. variables independent of N and with finite second moment. Define

$$Y = \sum_{n=0}^N Y_n,$$

whose distribution is a *compound Poisson distribution*.

1. Show that

$$\mu := \mathbf{E}(Y) = \mathbf{E}(Y_1)\mu_0.$$

2. Show that

$$\mathbf{V}(Y) = \frac{\mathbf{E}(Y_1^2)}{\mathbf{E}(Y_1)}\mu.$$

Exercise 4.10. Let $J \subseteq \mathbb{R}$ be an open interval and let $\mathcal{V} : J \rightarrow \mathbb{R}$ be continuous and strictly positive. Fix $\mu_0 \in J$ and define

$$g(\mu) = \int_{\mu_0}^{\mu} \frac{1}{\mathcal{V}(t)} dt.$$

1. Show that g is a strictly increasing and continuously differentiable function.
2. Argue that $I := g(J)$ is open and use the inverse function theorem to conclude that g^{-1} is differentiable with

$$(g^{-1})'(\theta) = \mathcal{V}(g^{-1}(\theta)).$$

3. Let $\theta_0 = g(\mu_0)$ and show that

$$K(\theta) := \int_{\theta_0}^{\theta} g^{-1}(s) ds$$

for $\theta \in I$ is twice continuously differentiable with

$$K''(\theta) = \mathcal{V}(g^{-1}(\theta)).$$

4. Argue that K is strictly convex, and that its corresponding quasi-link and quasi-variance functions are g and $\mathcal{V}$, respectively. That is, $g_K = g$ and $\mathcal{V}_K = \mathcal{V}$.

This particular example is a distribution on the non-negative integers with the ratio

$$\varphi = \frac{\mathbf{E}(Y_1^2)}{\mathbf{E}(Y_1)}$$

playing the role as a dispersion parameter.

5 Generalized linear models

Warning

This chapter is still a draft and minor changes are made without notice.

This chapter treats a generalization of the linear model that preserves much of its tractability, but can be adapted to cases where the linear model is inappropriate. This generalization is tightly linked to the exponential dispersion distributions treated in Chapter 4.

Section 5.1 introduces the fundamental model assumptions as generalizations of the model assumptions for the linear model. Section 5.2 covers how generalized linear models can be fitted to data via maximum-likelihood estimation. Section 5.3 covers the standard constructions of confidence intervals and test statistics and their distributional approximations that are used for statistical inference. The content of these sections is an extension of similar methods and results for the linear model as treated in Chapter 2.

5.1 Model assumptions

To motivate the general model assumptions we will consider three examples. In all three examples X denote a vector of predictor variables, and we introduce a model of the response given the predictors via a linear combination $X^T\beta$ of the predictors for some parameter vector β .

Example 5.1. For a binary response $Y \in \{0, 1\}$ we know that

$$\mathbf{E}(Y \mid X) = \mathbf{P}(Y = 1 \mid X) \in [0, 1]$$

and

$$\mathbf{V}(Y \mid X) = \mathbf{P}(Y = 1 \mid X)(1 - \mathbf{P}(Y = 1 \mid X)) = \mathcal{V}(\mathbf{E}(Y \mid X))$$

for the variance function $\mathcal{V}(\mu) = \mu(1 - \mu)$.

A linear model, $\mathbf{P}(Y = 1 \mid X) = X^T\beta$, of the conditional expectation is problematic. The constraint that a probability must be in $[0, 1]$ can only be enforced by restricting the parameter space of β . The restriction will depend on the values of the observed predictors, and it is generally impossible to ensure that all future predictions will be in $[0, 1]$. Moreover, the variance is not constant but depends upon the expectation via the variance function.

One solution is to consider a model of the form

$$\mathbf{P}(Y = 1 \mid X) = \mu(X^T\beta)$$

for $\mu : \mathbb{R} \rightarrow [0, 1]$ a given mean value function¹⁵. One possible choice is the logistic function, which corresponds to the canonical link for the Bernoulli distribution, and which results in the *logistic regression model*. Thus the model of the expectation is given in terms of the linear combination $X^T\beta$, but it is (nonlinearly) transformed using μ . By assumption, μ takes values in $[0, 1]$. The variance is then $\mathcal{V}(\mu(X^T\beta))$.

¹⁵ Typically a continuous and monotone function – a distribution function for instance.

Example 5.2. If Y is Poisson distributed, and thus takes values in $\mathbb{N}_0$, we know that

$$\mathbf{E}(Y \mid X) \geq 0,$$

and

$$\mathbf{V}(Y \mid X) = \mathbf{E}(Y \mid X) = \mathcal{V}(\mathbf{E}(Y \mid X))$$

for the variance function $\mathcal{V}(\mu) = \mu$. As for the Bernoulli model a linear model of the mean can produce values outside the distributions range of possible mean values. One solution is the *log-linear* model

$$\mathbf{E}(Y \mid X) = e^{X^T\beta}.$$

That is, the mean value function is the exponential function.

Example 5.3. If Y is a positive random variable given by

$$\log(Y) = X^T\beta + \varepsilon$$

for $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ then

$$\mathbf{E}(Y \mid X) = e^{X^T\beta} \mathbf{E}(e^\varepsilon).$$

Example 5.3 is an example of a model satisfying the two assumptions **GA1** and **GA2** below, but with a response distribution that is *not* an exponential dispersion distribution.

The distribution of e^ε is a log-normal distribution with mean $e^{\sigma^2/2}$, hence

$$\mathbf{E}(Y | X) = e^{X^T\beta + \sigma^2/2},$$

which shows that the model is a log-linear model. The variance is

$$\begin{aligned} \mathbf{V}(Y | X) &= e^{2X^T\beta} \mathbf{V}(e^\varepsilon) = e^{2X^T\beta} (e^{\sigma^2} - 1) e^{\sigma^2} \\ &= (e^{\sigma^2} - 1) (\mathbf{E}(Y | X))^2. \end{aligned}$$

Thus by introducing the dispersion parameter $\varphi = e^{\sigma^2} - 1$ and the variance function $\mathcal{V}(\mu) = \mu^2$ we see that

$$\mathbf{V}(Y | X) = \varphi \mathcal{V}(\mathbf{E}(Y | X)).$$

In conclusion, the log-normal model is a log-linear model with a quadratic variance function. It follows from Example 4.4 that the exponential dispersion model with quadratic variance function is the Gamma model. Thus an alternative to the log-normal model with the same mean and variance structure is a Gamma model with the log-link.

The linear model assumptions **A1–A3** can be generalized by allowing for nonlinear relations between a linear combination of the predictors and the mean and variance. This also allows for a non-constant variance as a function of the mean. For exponential dispersion distributions such a relation is a mathematical necessity as two of the examples above show.

The assumptions **GA1–GA3** below, that replace A1–A3 for generalized linear models, follow as abstractions of the structure found for exponential dispersion models. To ease notation we introduce

$$\eta = X^T\beta$$

to denote the *linear predictor* for a vector of predictors X and a vector of parameters β .

GA1. The conditional expectation of Y given X is

$$\mathbf{E}(Y | X) = \mu(\eta),$$

with $\mu : \mathbb{R} \rightarrow J$ the *mean value function*. The set $J \subseteq \mathbb{R}$ denotes the range of the mean value function.

GA2. The conditional variance of Y given X is

$$\mathbf{V}(Y | X) = \varphi \mathcal{V}(\mu(\eta)),$$

with $\mathcal{V} : J \rightarrow (0, \infty)$ the *variance function* and $\varphi > 0$ the *dispersion parameter*.

GA3. The conditional distribution of Y given X is the $(\theta(\eta), \nu_\varphi)$ -exponential dispersion distribution,

$$Y \mid X \sim \mathcal{E}(\theta(\eta), \nu_\varphi).$$

The exponential dispersion distributions referred to in **GA3** are introduced in Chapter 4. The normal distribution, as in Assumption A3, is a special case. As for the linear model, the assumption **GA3** is a strong distributional assumption, which implies **GA1** and **GA2** for *specific* choices of μ and $\mathcal{V}$ that depend on the map $\eta \mapsto \theta(\eta)$ and the chosen exponential dispersion model.

The situation is, however, a little more complicated than for the linear model. With arbitrary choices of μ , $\mathcal{V}$ and φ we cannot always find a corresponding exponential dispersion model. Moreover, Example 5.3 shows that there exist models fulfilling **GA1** and **GA2**, which are not exponential dispersion models. For the log-normal model there is an exponential dispersion model with an equivalent mean and variance structure, but the response distribution is different. In this light it is noteworthy that the methodology developed does not hinge on **GA3** to any great extent. Generalized linear models is largely a semiparametric modeling framework of mean and variance.

5.2 Estimation theory

In this section we cover the theory behind estimation in generalized linear models. We introduce maximum likelihood estimation for generalized linear models based on the exponential dispersion distributions. This includes the derivation of the nonlinear estimating equation, known as the score equation, and the iterative weighted least squares (IWLS) algorithm that is used in practice to fit the models to data.

5.2.1 Maximum likelihood estimation

We first consider the simple case where $Y \sim \mathcal{E}(\theta(\eta), \nu_\varphi)$, that is, the distribution of Y is given by the exponential dispersion

model with canonical parameter $\theta(\eta)$ and structure measure ν_φ . Derivations of the score equation and Fisher information in this case can then be used to derive the score equation and Fisher information in the general case when we have observations $Y_1, \dots, Y_n$ that are conditionally independent given the predictors with $Y_i \mid \mathbf{X} \sim \mathcal{E}(\theta(\eta_i), \nu_\varphi)$ and $\eta_i = X_i^T \beta$.

Definition 5.1. With ℓ denoting a generic C^2 log-likelihood function the score statistic is

$$U(\eta) := \nabla_\eta \ell(\eta).$$

The Fisher information,

$$\mathcal{J}(\eta) = -\mathbf{E}(D_\eta U(\eta)),$$

is minus the expectation of the Jacobian of the score statistic, or, equivalently, the expectation of the Hessian of the negative log-likelihood.

The score equation is obtained by equating the score statistic to 0. In all that follows, the dispersion parameter φ is regarded as fixed.

Lemma 5.1. *If $Y \sim \mathcal{E}(\theta(\eta), \nu_\varphi)$ then the log-likelihood function is*

$$\ell_Y(\eta) = \frac{\theta(\eta)Y - \zeta(\eta)}{\varphi},$$

the score statistic is $U(\eta) = \theta'(\eta)(Y - \mu(\eta))/\varphi$, and the Fisher information is

$$\mathcal{J}(\eta) = \frac{\theta'(\eta)\mu'(\eta)}{\varphi}.$$

Proof. The density for the distribution of Y w.r.t. ν_φ is by definition

$$e^{\frac{\theta(\eta)y - \zeta(\eta)}{\varphi}},$$

and it follows that $\ell_Y(\eta)$ has the stated form. Differentiation of $\varphi \ell_Y(\eta)$ yields

$$\varphi U(\eta) = \varphi \ell'(\eta) = \theta'(\eta)Y - \zeta'(\eta) = \theta'(\eta) \left(Y - \underbrace{\frac{\zeta'(\eta)}{\theta'(\eta)}}_{\mu(\eta)} \right),$$

where we have used Corollary 4.1. Furthermore, we find that

$$\varphi U'(\eta) = \theta''(\eta)(Y - \mu(\eta)) - \theta'(\eta)\mu'(\eta),$$

and since $E_\eta(Y) = \mu(\eta)$ it follows that

$$\mathcal{J}(\eta) = -E_\eta(U'(\eta)) = \frac{\theta'(\eta)\mu'(\eta)}{\varphi}.$$

□

With only a single observation, the score equation is equivalent to $\mu(\eta) = Y$, and it follows that there is a solution to the score equation if $Y \in J = \mu(I)$. However, the situation with a single observation is not relevant for practical purposes. The result is only given as an intermediate step towards the next result.

The score statistic U above is a function of the univariate parameter η . We adapt in the following the convention that for a vector $\eta = (\eta_1, \dots, \eta_n)^T$

$$U(\eta) = (U(\eta_1), \dots, U(\eta_n))^T.$$

That is, as a map the score statistic is applied coordinatewisely to the vector η . Note that the derivative (the Jacobian) of $\eta \mapsto U(\eta)$ is an $n \times n$ diagonal matrix.

Theorem 5.1. *Assume that $Y_1, \dots, Y_n$ are conditionally independent given $\mathbf{X}$ and that $Y_i \mid \mathbf{X} \sim \mathcal{E}(\theta(\eta_i), \nu_\varphi)$ where $\eta_i = X_i^T \beta$. Then with $\eta = \mathbf{X}\beta$ the score statistic expressed in the β -parameter is*

$$\mathcal{U}(\beta) = \mathbf{X}^T U(\eta).$$

The Fisher information is

$$\mathcal{J}(\beta) = \mathbf{X}^T \mathbf{W} \mathbf{X},$$

with the entries in the diagonal weight matrix $\mathbf{W}$ being

$$w_{ii} = \frac{(\mu'_i)^2}{\varphi \mathcal{V}(\mu_i)} = \frac{\theta'_i \mu'_i}{\varphi}$$

where $\mu_i = \mu(\eta_i)$, $\mu'_i = \mu'(\eta_i)$ and $\theta'_i = \theta'(\eta_i)$.

$$\partial_{\eta_i} U(\eta)_j = \begin{cases} U'(\eta_j) & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

The diagonal weight matrix $\mathbf{W}$ is

$$\frac{1}{\varphi} \begin{pmatrix} \frac{(\mu'_1)^2}{\mathcal{V}(\mu_1)} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \frac{(\mu'_n)^2}{\mathcal{V}(\mu_n)} \end{pmatrix}$$

Proof. By the independence assumption the log-likelihood is

$$\ell_{\mathbf{Y}}(\beta) = \sum_{i=1}^n \ell_{Y_i}(\eta_i)$$

where $\eta_i = X_i^T \beta$. By the chain rule,

$$\mathcal{U}(\beta) = \nabla_{\beta} \ell_{\mathbf{Y}}(\beta) = \sum_{i=1}^n X_i U(\eta_i) = \mathbf{X}^T U(\eta).$$

As argued above, $-D_{\eta} U(\eta)$ is diagonal, and the expectation of the diagonal entries are according to Lemma 5.1

$$w_{ii} = \frac{\theta'(\eta_i) \mu'(\eta_i)}{\varphi}.$$

The alternative formula for the weights follows from Corollary 4.1. Thus

$$\mathcal{J}(\beta) = -\mathbf{E}(D_{\beta} \mathcal{U}(\beta)) = -\mathbf{E}(\mathbf{X}^T D_{\eta} U(\eta) \mathbf{X}) = \mathbf{X}^T \mathbf{W} \mathbf{X}.$$

□

By observing that

$$U(\eta)_i = \frac{\theta'(\eta_i)(Y_i - \mu_i)}{\varphi},$$

it follows that the score equation $\mathcal{U}(\beta) = 0$ is equivalent to the system of equations

$$\sum_{i=1}^n \theta'_i(Y_i - \mu_i) X_{ij} = 0$$

for $j = 1, \dots, p$. Note that the score equation and thus its solution does not depend upon the dispersion parameter. Note also, that for the canonical link function we have $\theta_i = \eta_i$, hence $\theta'_i = 1$, and the weights simplify to

$$w_{ii} = \frac{\mu'_i}{\varphi} = \frac{\mathcal{V}(\mu_i)}{\varphi}.$$

We write $\theta'_i = \theta'(\eta_i)$ and use similar shorthand notation for parameter functions throughout.

Chapter 6 explores existence and uniqueness of the solution to the score equation. For an arbitrary link function it can be difficult to determine if there is a solution and whether it is unique, but for certain choices the negative log-likelihood becomes convex and some guarantees can be given. The canonical link function is one such choice, and Chapter 6 completely characterizes existence and uniqueness for the canonical link function.

Example 5.4. For the normal distribution $\mathcal{N}(\mu, \sigma^2)$ and with the canonical link function the log-likelihood function becomes

$$\begin{aligned}\ell(\beta) &= \frac{1}{\varphi} \sum_{i=1}^n Y_i X_i^T \beta - \frac{(X_i^T \beta)^2}{2} \\ &= \frac{1}{2\varphi} (2\mathbf{Y}^T \mathbf{X} \beta - \beta^T \mathbf{X}^T \mathbf{X} \beta) \\ &= \frac{1}{2\varphi} (\|\mathbf{Y}\|^2 - \|\mathbf{Y} - \mathbf{X} \beta\|^2).\end{aligned}$$

Up to the term $\|\mathbf{Y}\|^2$ – that doesn't depend upon the unknown β -vector – the log-likelihood function is proportional to the squared error loss with proportionality constant $-1/(2\varphi)$. The maximum likelihood estimator is thus equal to the least squares estimator.

5.2.2 Algorithms

The score equation is nonlinear, and it does not in general have a closed form solution. Solutions must thus be found by iterative methods. Newton's algorithm is based on a first order Taylor approximation of the score statistic. The resulting approximation of the score equation is a linear equation. Newton's algorithm consists of iteratively computing the first order Taylor approximation and solving the resulting linear equation. A slight modification of Newton's algorithm is often used, where the derivative of the score is replaced by its expectation, that is, by the Fisher information. To present the idea we consider a simple example of estimation in the exponential distribution with i.i.d. observations.

Example 5.5. Consider the parametrization $\theta(\eta) = -\eta^{-k}$ for $\eta > 0$ (and a fixed $k > 0$) of the canonical parameter in the exponential distribution. That is, the density is

$$e^{\theta(\eta)y - k \log \eta}$$

w.r.t. Lebesgue measure on $(0, \infty)$. The mean value function is

$$\mu(\eta) = -\frac{1}{\theta(\eta)} = \eta^k.$$

If Z is Weibull distributed with shape parameter k and scale parameter η then $Y = Z^k$ is exponentially distributed with scale parameter η^k . This explains the interest in this particular parametrization as it allows us to fit models with Weibull distributed responses.

With $Y_1, \dots, Y_n$ i.i.d. observations from this distribution and with

$$\bar{Y} = \frac{1}{n} \sum_{i=1}^n Y_i$$

the score statistic amounts to

$$U(\eta) = \sum_{i=1}^n \theta'(\eta)(Y_i - \mu(\eta)) = nk\eta^{-1}(\eta^{-k}\bar{Y} - 1)$$

where we have used that $\theta'(\eta) = k\eta^{-k-1}$. The score equation is thus equivalent to

$$\eta^{-k}\bar{Y} - 1 = 0.$$

It is straightforward to solve this equation analytically¹⁶, and the solution is

$$\eta = \bar{Y}^{1/k}.$$

However, to illustrate the general techniques we Taylor expand the left hand side of the equation around η_m to first order and obtain the linear equation

$$\eta_m^{-k}\bar{Y} - 1 - k\eta_m^{-k-1}\bar{Y}(\eta - \eta_m) = 0.$$

The solution of this linear equation is

$$\eta_{m+1} = \eta_m - \frac{\eta_m^{-k}\bar{Y} - 1}{-k\eta_m^{-k-1}\bar{Y}} = \eta_m - \frac{U(\eta_m)}{U'(\eta_m)}$$

provided that $U'(\eta_m) \neq 0$. This is Newton's algorithm. With a suitable choice of starting value η_1 we iteratively compute η_m until convergence.

If we replace the derivative of the score statistic in the approximating linear equation by its expectation we arrive at the linear equation

$$\eta_m^{-k}\bar{Y} - 1 - k\eta_m^{-1}(\eta - \eta_m) = 0,$$

whose solution is

$$\eta_{m+1} = \eta_m - \frac{\eta_m^{-k}\bar{Y} - 1}{-k\eta_m^{-1}} = \eta_m + \frac{U(\eta_m)}{\mathcal{J}(\eta_m)}.$$

The general technique of replacing the derivative of the score statistic with its expectation in Newton's algorithm is known as Fisher scoring.

¹⁶ Which shows that if $Z_1, \dots, Z_n$ are i.i.d. Weibull distributed with known shape parameter k and scale parameter η the MLE of η is

$$\hat{\eta} = \left(\frac{1}{n} \sum_{i=1}^n Z_i^k \right)^{1/k}.$$

The iterative weighted least squares algorithm in the general case follows the one-dimensional example above closely. First note that the dispersion parameter enters as a multiplicative constant in the log-likelihood, and its value does not affect the maximum likelihood estimate of β . We take it to be equal to 1 for the subsequent computations. The derivative of minus the score statistic¹⁷ is found as in the proof of Theorem 5.1 to be

$$-D_{\beta}\mathcal{U}(\beta) = \mathbf{X}^T \mathbf{W}^{\text{obs}} \mathbf{X}$$

where

$$\mathbf{W}^{\text{obs}} = - \begin{pmatrix} U'_1(\eta_1) & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & U'_n(\eta_n) \end{pmatrix}. \quad (5.1)$$

A first order Taylor expansion of the score statistic around β_m results in the following linearization of the score equation

$$\mathbf{X}^T U(\eta_m) - \mathbf{X}^T \mathbf{W}_m^{\text{obs}} \mathbf{X} (\beta - \beta_m) = 0.$$

Its solution is

$$\beta_{m+1} = \beta_m + (\mathbf{X}^T \mathbf{W}_m^{\text{obs}} \mathbf{X})^{-1} \mathbf{X}^T U(\eta_m),$$

provided that $\mathbf{X}^T \mathbf{W}_m^{\text{obs}} \mathbf{X}$ has full rank p . Replacing $\mathbf{W}_m^{\text{obs}}$ by $\mathbf{W}_m$ from Theorem 5.1 we get the Fisher scoring algorithm. We may note that the diagonal entries in $\mathbf{W}_m$ are always strictly positive if the mean value function is strictly monotone, which implies that $\mathbf{X}^T \mathbf{W}_m \mathbf{X}$ is positive definite and has rank p if and only if $\mathbf{X}$ has rank p . By contrast, the diagonal weights in $\mathbf{W}_m^{\text{obs}}$ may be negative.

We can rewrite the update formula for the Fisher scoring algorithm as follows

$$\begin{aligned} \beta_{m+1} &= \beta_m + (\mathbf{X}^T \mathbf{W}_m \mathbf{X})^{-1} \mathbf{X}^T U(\eta_m) \\ &= (\mathbf{X}^T \mathbf{W}_m \mathbf{X})^{-1} \mathbf{X}^T \mathbf{W}_m \underbrace{(\mathbf{X} \beta_m + \mathbf{W}_m^{-1} U(\eta_m))}_{\mathbf{Z}_m}. \end{aligned}$$

The vector $\mathbf{Z}_m$ is known as the *working response*, and its coordinates can be written out as

$$Z_{m,i} = X_i^T \beta_m + \frac{Y_i - \mu_{m,i}}{\mu'_{m,i}}. \quad (5.2)$$

¹⁷ Often called the observed Fisher information.

Note that $\mathbf{W}_m^{\text{obs}}$ as well as $\mathbf{W}_m$ depend upon the current β_m through $\eta_m = \mathbf{X} \beta_m$, hence the subscript m .

In terms of the working response, the vector β_{m+1} is the minimizer of the weighted squared error loss

$$(\mathbf{Z}_m - \mathbf{X}\beta)^T \mathbf{W}_m (\mathbf{Z}_m - \mathbf{X}\beta), \quad (5.3)$$

see Theorem 2.1. The Fisher scoring algorithm for generalized linear models is known as iterative weighted least squares (IWLS), since it can be understood as iteratively solving a weighted least squares problem. To implement the algorithm we can also rely on general solvers of weighted least squares problems. This results in the following version of IWLS. Given β_1 we iterate over the steps 1–3 until convergence:

1. Compute the working response vector $\mathbf{Z}_m$ based on β_m using (5.2).
2. Compute the weights

$$w_{m,ii} = \frac{(\mu'_{m,i})^2}{\mathcal{V}(\mu_{m,i})}.$$

3. Compute β_{m+1} by minimizing the weighted sum of squares (5.3).

It is noteworthy that the computations only rely on the mean value function μ , its derivative μ' and the variance function $\mathcal{V}$. Thus the IWLS algorithm depends on the mean and variance structure, as specified in the assumptions **GA1** and **GA2**, and not on any other aspects of the response distribution.

Exercise 5.5 explores how the IWLS algorithm can be derived for general response distributions. From a general viewpoint of maximum-likelihood estimation in a regression model there is nothing special about the exponential dispersion distributions. However, for the exponential dispersion distributions the algorithm becomes – as demonstrated above – an algorithm that only depends on the mean and variance structure of the model.

The dispersion parameter is eliminated (by taking $\varphi = 1$) in the IWLS algorithm. It doesn't mean that the dispersion parameter is irrelevant. It matters for the subsequent statistical analysis, but not for the estimation.

5.3 Sampling distributions

It is not possible to derive exact distributional results about the estimators or the various test statistics that are used for generalized linear models – except for the special case of the linear model with normal errors. We will need to

rely on approximations. By introducing an idealized least squares estimator and reinterpreting the IWLS algorithm in this context, it is possible to introduce sensible *approximate* moment results under the weak **GA1** and **GA2** assumptions. Approximate confidence intervals are subsequently introduced based on Z -scores, and tests of linear hypotheses on β under the **GA3** assumption are discussed using likelihood ratio (deviance) tests. The operational procedures for using the tests are the same as for the linear model – the only difference being that the distributions used to compute p -values are approximations. The formal asymptotic justifications will only be treated briefly.

5.3.1 An idealized least squares estimator

We introduce a weighted least squares estimator with weights and responses depending on the unknown parameters for which we can derive exact moment results. We can then interpret the IWLS algorithm as computing an approximate solution to this idealized least squares problem. The derivation emphasizes the fact that the algorithm only depends on the assumptions **GA1** and **GA2**.

Supposing that we know the true parameter, β , we can form the weighted quadratic loss

$$\tilde{\ell}(\tilde{\beta}) = \sum_{i=1}^n \mathcal{V}(\mu_i)^{-1} (Y_i - \mu(X_i^T \tilde{\beta}))^2$$

where $\mu_i = \mu(X_i^T \beta)$. Besides the fact that the weights depend upon the unknown true parameter, the $\tilde{\beta}$ enters nonlinearly through $\mu(X_i^T \tilde{\beta})$. We can Taylor expand the mean once¹⁸ around β to obtain

$$\mu(X_i \tilde{\beta}) \approx \mu_i + \mu'_i X_i^T (\tilde{\beta} - \beta)$$

and plug this approximation back into the loss. This yields the approximating quadratic loss

$$\begin{aligned} \tilde{\ell}(\tilde{\beta}) &= \sum_{i=1}^n \mathcal{V}(\mu_i)^{-1} \left(Y_i - \mu_i - \mu'_i X_i^T (\tilde{\beta} - \beta) \right)^2 \\ &= \sum_{i=1}^n \frac{(\mu'_i)^2}{\mathcal{V}(\mu_i)} \left(X_i^T \beta + \frac{Y_i - \mu_i}{\mu'_i} - X_i^T \tilde{\beta} \right)^2. \end{aligned}$$

¹⁸ This is the key step in the Gauss-Newton algorithm for nonlinear least squares estimation.

By introducing the *idealized* responses

$$Z_i = X_i^T \beta + \frac{Y_i - \mu_i}{\mu'_i} \quad (5.4)$$

and the idealized weights

$$w_{ii} = \frac{(\mu'_i)^2}{\mathcal{V}(\mu_i)},$$

the approximating quadratic loss can be written as

$$\tilde{\ell}(\tilde{\beta}) = (\mathbf{Z} - \mathbf{X}\tilde{\beta})^T \mathbf{W} (\mathbf{Z} - \mathbf{X}\tilde{\beta}) = \|\mathbf{Z} - \mathbf{X}\tilde{\beta}\|_{\mathbf{W}}^2,$$

with $\mathbf{W}$ the diagonal weight matrix with the w_{ii} s in the diagonal. Under the assumptions **GA1** and **GA2** we can observe that

$$\mathbf{E}(Z_i \mid X_i) = X_i^T \beta$$

and

$$\mathbf{V}(Z_i \mid X_i) = \varphi w_{ii}^{-1},$$

but since the weights as well as $\mathbf{Z}$ depend upon the unknown β , we cannot compute $\tilde{\ell}(\tilde{\beta})$. The IWLS algorithm can be understood as iteratively approximating the idealized response by the working response – by plugging in the current estimate of β . In this way we get approximations of the idealized loss, and the idealized weighted least squares estimator

$$\hat{\beta}^{\text{ideal}} = (\mathbf{X}^T \mathbf{W} \mathbf{X})^{-1} \mathbf{X}^T \mathbf{W} \mathbf{Z}.$$

It is worth observing that under **GA1**, **GA2** and A4 we have the following exact distributional results on $\hat{\beta}^{\text{ideal}}$

$$\begin{aligned} \mathbf{E}(\hat{\beta}^{\text{ideal}} \mid \mathbf{X}) &= \beta, \\ \mathbf{V}(\hat{\beta}^{\text{ideal}} \mid \mathbf{X}) &= \varphi (\mathbf{X}^T \mathbf{W} \mathbf{X})^{-1}, \\ \mathbf{E}(\|\mathbf{Z} - \mathbf{X}\hat{\beta}^{\text{ideal}}\|_{\mathbf{W}}^2 \mid \mathbf{X}) &= (n - p)\varphi. \end{aligned}$$

If we plug the maximum likelihood estimator, $\hat{\beta}$, in for β in the definition of the idealized quadratic loss, and thus replacing the idealized responses by

$$\hat{Z}_i = X_i^T \hat{\beta} + \frac{Y_i - \hat{\mu}_i}{\hat{\mu}'_i}$$

we obtain a computable loss function

$$\beta \mapsto \|\hat{\mathbf{Z}} - \mathbf{X}\beta\|_{\mathbf{W}}^2. \quad (5.5)$$

The fact that $\hat{\beta}$ is a fixed point for the IWLS algorithm implies that $\hat{\beta}$ is also the minimizer of (5.5). Provided that (5.5) is a good approximation of the idealized quadratic loss, we can expect that the distributional results on the idealized estimator will be good approximations for $\hat{\beta}$ as well. Observe that with

$$\mathcal{X}^2 := \|\hat{\mathbf{Z}} - \mathbf{X}\hat{\beta}\|_{\mathbf{W}}^2 = \sum_{i=1}^n \frac{(Y_i - \hat{\mu}_i)^2}{\mathcal{V}(\hat{\mu}_i)},$$

where $\mathcal{X}^2$ is known as the Pearson χ^2 -statistic, the results above suggest the estimator

$$\hat{\varphi} = \frac{1}{n-p} \mathcal{X}^2 \quad (5.6)$$

of the dispersion parameter.

5.3.2 Tests and confidence intervals

The distributional results for the idealized weighted least squares estimator suggest the approximation

$$\sqrt{\varphi(\mathbf{X}^T \mathbf{W} \mathbf{X})_{jj}^{-1}}$$

of the standard error of $\hat{\beta}_j$. By plugging in the estimate of the dispersion parameter and the estimate of the weights we define the j -th Z -score as

$$Z_j := \frac{\hat{\beta}_j - \beta_j}{\sqrt{\hat{\varphi}(\mathbf{X}^T \hat{\mathbf{W}} \mathbf{X})_{jj}^{-1}}}.$$

More generally, we can introduce

$$Z_a := \frac{a^T \hat{\beta} - a^T \beta}{\sqrt{\hat{\varphi} a^T (\mathbf{X}^T \hat{\mathbf{W}} \mathbf{X})^{-1} a}}$$

for $a \in \mathbb{R}^p$.

The Z -score is used exactly as it is used for linear models. First, it is used to test a one-dimensional restriction on the parameter vector – typically a hypothesis of the form $H_0 : \beta_j = 0$ for some index j . The square of a Z -statistic, Z_a^2 , is known as a univariate Wald statistic. It is possible to

In reality, $\hat{\beta}$ is β_m , the value of β in the m -th iteration of IWLS, when the algorithm is judged to have converged.

construct multivariate Wald statistics to test hypotheses about multivariate parameter constraints, but we will not pursue the construction here. Second, the Z -score is used to compute confidence intervals for $a^T\beta$ of the form

$$a^T\hat{\beta} \pm z \cdot \sqrt{\hat{\varphi}a^T(\mathbf{X}^T\mathbf{W}\mathbf{X})^{-1}a} \quad (5.7)$$

where z is chosen suitably.

A distributional approximation of Z_a is needed for computing either a p -value associated with a hypothesis test or for choosing the quantile z in the construction of the confidence interval. Asymptotic theory supports using the $\mathcal{N}(0, 1)$ -approximation, so if a confidence interval with 95% nominal coverage is sought, and $z = 1.96$ is chosen as the 97.5% quantile for the standard normal distribution. The actual coverage depends upon how well the distribution of Z_a is approximated by $\mathcal{N}(0, 1)$.

An alternative to using theoretical approximations is to use bootstrapping to compute approximating quantiles, see Section 13.3. Occasionally, z is chosen as the 97.5% quantile in the t_{n-p} -distribution instead – still aiming for a 95% coverage. The theoretical support for this practice is weak, but the practical consequence is clear. Since the tail quantiles for the t -distribution are larger than the tail quantiles for the normal distribution, the use of the t -distribution results in wider confidence intervals and more conservative conclusions. The difference is, however, negligible unless $n - p$ is small – less than 20, say.

The likelihood ratio tests are alternatives to z -tests and Wald tests. Their derivation is based on the stronger distributional assumption **GA3**. That is, the responses follow an exponential dispersion distribution conditionally on the predictors. In the framework of generalized linear models the likelihood ratio tests are usually formulated in terms of *deviances* as introduced in Section 4.3.

For a generalized linear model we have response observations $Y_1, \dots, Y_n$, and we let $\hat{\mu}_i$ denote the maximum likelihood estimate of the mean of Y_i . That is,

$$\hat{\mu}_i = \mu(X_i^T\hat{\beta})$$

with $\hat{\beta}$ the maximum likelihood estimate of $\beta \in \mathbb{R}^p$. Likewise, if $\hat{\beta}^0$ denotes the maximum likelihood estimate of β under the

null hypothesis

$$H_0 : \beta \in L$$

where $L \subseteq \mathbb{R}^p$ is a p_0 dimensional subspace of $\mathbb{R}^p$, we let $\hat{\mu}_i^0$ denote the corresponding maximum likelihood estimate of the mean for the i -th observation under the hypothesis. Recall that all such hypotheses can be rephrased in terms of a $p \times p_0$ matrix $\mathbf{C}$ of rank p_0 such that

$$H_0 : \mathbf{E}(Y_i | X_i) = \mu(X_i^T \mathbf{C} \beta^0).$$

Definition 5.2. The total deviances for the model and the null hypothesis are

$$D = \sum_{i=1}^n d(Y_i, \hat{\mu}_i) \quad \text{and} \quad D_0 = \sum_{i=1}^n d(Y_i, \hat{\mu}_i^0),$$

respectively. The deviance test statistic is

$$D_0 - D$$

and the F -test statistic is

$$F = \frac{(D_0 - D)/(p - p_0)}{D/(n - p)}.$$

The deviance test statistic is simply the log-likelihood ratio test statistic for the null hypothesis. The F -test statistic is inspired by the F -test for linear models. In both cases, large values of the test statistic are critical, and thus evidence against the null hypothesis. In contrast to the linear model, it is not possible to derive the exact distributions of the deviance test statistic or the F -test statistic under the null hypothesis. To compute p -values we need to rely on approximations or simulations. The general theoretical approximation of the deviance test statistic is

$$D_0 - D \stackrel{\text{approx}}{\sim} \varphi \chi_{p-p_0}^2, \quad (5.8)$$

which is exact for the linear model with normal errors (under the assumptions A3 + A5). A purpose of the F -test is, as for linear models, to (approximately) remove the dependence upon the unknown dispersion parameter. The approximation of the F -test statistic is

$$F \stackrel{\text{approx}}{\sim} F_{p-p_0, n-p},$$

which is exact for the linear model with normal errors. The approximation is generally good when φ is small.

A formal justification of the approximations is quite involved, and the typical strategy is to consider asymptotic scenarios with $n \rightarrow \infty$ combined with suitable conditions on the predictors X_i . We will explain the basis for these approximations later, but we will not consider formal asymptotic arguments.

5.4 Model assessment and diagnostic

The definition of the adjusted R^2 can easily be extended to generalized linear models by replacing $\text{RSS}/(n-p)$ with $\mathcal{X}^2/(n-p)$, where

$$\mathcal{X}^2 = \sum_{i=1}^n \frac{(Y_i - \hat{\mu}_i)^2}{\mathcal{V}(\hat{\mu}_i)}$$

is the Pearson χ^2 -statistic, or by $D/(n-p)$, where

$$D = \sum_{i=1}^n d(Y_i, \hat{\mu}_i)$$

is the total deviance.

As for the linear model, model diagnostics can be based on residuals, but for generalized linear models there are several possible choices. With observations $Y_1, \dots, Y_n$ and fitted mean values $\hat{\mu}_1, \dots, \hat{\mu}_n$ the i -th *raw residual* is

$$Y_i - \hat{\mu}_i.$$

In the R terminology the raw residual is called the *response residual*.

Whenever the variance function is not constant, the raw residual is not particularly useful. To take a non-constant variance function into account, a natural choice of residual is the *Pearson residual* defined as

$$\frac{Y_i - \hat{\mu}_i}{\sqrt{\mathcal{V}(\hat{\mu}_i)}}.$$

Pearson residuals are used in the same way as the raw or standardized residuals are used for the linear model. We plot the Pearson residuals against the fitted values or a predictor

The sum of the squared Pearson residuals is the Pearson χ^2 -statistic.

variable to visually check the model assumptions – the residuals should show no distributional dependence upon what we plot it against. Systematic trends show that **GA1** is not fulfilled, and variance inhomogeneity shows that the variance function in **GA2** is not appropriate.

The Pearson residual is based on the mean and variance assumptions **GA1** and **GA2** only. Based on the distributional assumption **GA3** we can also introduce the *deviance residual* for the i -th observation as

$$\text{sign}(Y_i - \hat{\mu}_i) \sqrt{d(Y_i, \hat{\mu}_i)}.$$

The deviance residuals can be used like the Pearson residuals, and Theorem 4.2 gives an approximate relation between them.

The total deviance is the sum of the squared deviance residuals.

Finally, the *working residual* should be mentioned. It is defined for the i -th observation as

$$\frac{Y_i - \hat{\mu}_i}{\mu'(\hat{\eta}_i)},$$

and is the residual from the weighted least squares problem (5.5).

Example 5.6. For the binomial case we find that the raw residual is

$$Y_i - m_i \hat{p}_i$$

where $\hat{p}_i$ is the estimate of the success probability for the i -th observation.

The Pearson residual is

$$\frac{Y_i - m_i \hat{p}_i}{\sqrt{m_i \hat{p}_i (1 - \hat{p}_i)}},$$

and the deviance residual is

$$\text{sign}(Y_i - m_i \hat{p}_i) \sqrt{2Y_i \log \frac{Y_i}{m_i \hat{p}_i} + 2(m_i - Y_i) \log \frac{m_i - Y_i}{m_i (1 - \hat{p}_i)}}.$$

The distribution of the residual is difficult to characterize in general. We cannot expect that the Pearson residuals follow a normal distribution, say. For generalized linear models there is thus no simple way to check if the strong distributional

assumptions – that the responses follow an exponential dispersion distribution given the predictors – are fulfilled, and the residuals are primarily used to investigate the mean-variance structure.

If the responses have a continuous distribution with distribution function $F_{\mu_i, \varphi}$ we can, however, construct a pp-plot by comparing

$$F_{\hat{\mu}_i, \hat{\varphi}}(Y_i)$$

to the quantiles of the uniform distribution. This can be useful for gamma distributed responses but not for Poisson distributed responses, say, unless the counts are quite large.

Exercises

Exercise 5.1. Define the probability distribution on $\{0, 1, \dots, m\}$ by the probability mass function

$$p_\eta(k) = \frac{1}{c(\eta)} \eta^k$$

with $c(\eta) = \sum_{k=0}^m \eta^k$ for $\eta > 0$.

1. Show that the family of probability distributions given by p_η for $\eta > 0$ is an exponential family. Find the structure measure and the unit cumulant function.
2. Compute the mean value function $\mu(\eta)$ and show that

$$\mathcal{V}(\mu(\eta)) = \frac{\eta}{(1-\eta)^2} - \frac{(m+1)^2 \eta^{m+1}}{(1-\eta^{m+1})^2}$$

for $\eta \neq 1$. Plot the link function for $m = 5, 10, 20$.

Hint: *You can plot the link function without a closed form expression.*

Consider the following data set with one explanatory variable and $n = 10$ observations.

i	1	2	3	4	5	6	7	8	9	10
X_i	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2
Y_i	1	5	2	8	7	5	9	7	10	10

Consider the generalized linear model with response distribution as given above, with the linear predictor

$$\eta_i = \beta_0 + \beta_1 x_i.$$

3. Implement the Fisher scoring algorithm to estimate β_0 and β_1 . Use as initial values $\beta_0 = 0$ and $\beta_1 = 1$ and report the estimated parameters for each iteration of the algorithm with a total of 10 iterations. Compute also an estimate of the standard error of the MLE $\hat{\beta}_1$ (assuming that the algorithm has converged to the MLE after 10 iterations).
4. Compute and plot the Pearson residuals against the fitted values.
5. Introduce a dispersion parameter $\varphi > 0$ and estimate it using the Pearson residuals. Recompute an estimate of the standard error of $\hat{\beta}_1$ taking the estimated dispersion parameter into account. Comment on the result.

Exercise 5.2. Define a probability distribution on the non-negative integers, $0, 1, 2, \dots$, as having probability mass function

$$p_\eta(k) = G(k, \rho) \left(\frac{1}{\rho + \eta} \right)^\rho \left(\frac{\eta}{\rho + \eta} \right)^k \quad (5.9)$$

for $\eta, \rho > 0$. The explicit expression for $G(k, \rho)$ is not needed – only that $\sum_{k=0}^{\infty} p_\eta(k) = 1$. In the following, ρ is considered a fixed nuisance parameter.

For the curious, the distribution is the negative binomial distribution.

1. Show that the family of probability distributions given by p_η for $\eta > 0$ is an exponential family for any fixed $\rho > 0$. Find the structure measure and the unit cumulant function.
2. Show that the mean value function is $\mu(\eta) = \eta$ and that

$$\mathcal{V}(\mu(\eta)) = \eta + \eta^2/\rho.$$

Consider the following data set with one explanatory variable and $n = 10$ observations.

i	1	2	3	4	5	6	7	8	9	10
X_i	5	10	15	20	25	30	35	40	45	50

i	1	2	3	4	5	6	7	8	9	10
Y_i	6	11	16	38	82	22	108	61	55	64

Consider the generalized linear model with response distribution as given above and with the linear predictor

$$\eta_i = \beta_0 + \beta_1 X_i.$$

3. Take $\rho = 1$ and compute the MLE of β_0 and β_1 . You can use the Fisher-scoring algorithm. How will it affect the result if ρ is changed?
4. Still taking $\rho = 1$ compute and plot the Pearson residuals against the fitted values and compute the χ^2 -statistic.
5. Write an R function that computes the χ^2 -statistic as a function of ρ using the general variance formula. Plot this χ^2 -statistic as a function of ρ and use it to argue that the data suggests a value of $\rho = 4$.
6. Compute a nominal 95% confidence interval of β_1 based on the MLE when assuming $\rho = 1$. Then compute a nominal 95% confidence interval assuming that $\rho = 4$ is the true value of ρ while the MLE is still based on assuming $\rho = 1$. Comment on the result.

Exercise 5.3. Recall from Example 5.3 that for the log-normal random variable, Y , whose distribution is given as

$$\log(Y) = \eta + \varepsilon \quad (5.10)$$

for $\eta \in \mathbb{R}$ the linear predictor and $\varepsilon \sim \mathcal{N}(0, \sigma^2)$, **GA1** holds with mean value function

$$\mu(\eta) = e^{\eta + \frac{\sigma^2}{2}}$$

and **GA2** holds with variance function $\mathcal{V}(\mu) = \mu^2$. The dispersion parameter is given as

$$\varphi = e^{\sigma^2} - 1$$

in terms of σ^2 .

Introduce the parameter $\tilde{\eta} = \eta + \frac{\sigma^2}{2}$. Whenever the linear predictor includes an intercept, the term $\frac{\sigma^2}{2}$ is simply absorbed into the intercept parameter, and we can regard $\tilde{\eta}$ as the linear predictor with this reparametrization of the intercept.

1. Identify the exponential dispersion model with mean value function $\mu(\tilde{\eta}) = e^{\tilde{\eta}}$ and variance function $\mathcal{V}(\mu) = \mu^2$. Show that the unit deviance for this model is

$$d(y, \mu) = 2 \left(\log \left(\frac{\mu}{y} \right) + \frac{y}{\mu} - 1 \right).$$

In general, the *coefficient of variation* of Y is defined as $\sqrt{\mathbf{V}(Y)}/\mu$. For the log-normal model as well as the exponential dispersion model above, this coefficient is constantly equal to $\sqrt{\varphi}$ and independent of μ . Both models are thus models using a log-link function and with a constant coefficient of variation.

Deviations in the model given by (5.10) are naturally measured in terms of the squared error $(\log(y) - \eta)^2$, which is the unit deviance for the normal distribution, and the model is typically fitted on this log-scale. The exponential dispersion model is typically fitted using the IWLS algorithm.

2. First fix $y = 1$ and plot $\tilde{\eta} \mapsto d(1, e^{\tilde{\eta}})$ for $\tilde{\eta} \in (-2, 2)$. Compare with $\tilde{\eta} \mapsto \tilde{\eta}^2$. Then show that in general

$$d(y, e^{\tilde{\eta}}) = (\log(Y) - \tilde{\eta})^2 + o((\log(y) - \tilde{\eta})^2).$$

What does this result imply about the relation between the least squares fit based on $\log(y)$ and the IWLS fit based on y ?

Exercise 5.4. This exercise is on the Danish fire insurance data introduced in Chapter 2. You should remove claims larger than 10 millions to avoid numerical problems. Recall that the response variable Y is the claim size and there are two potential predictors in the data set.

1. Refit the linear, additive model

$$\mathbf{E}(\log(Y_i)) = \beta_0 + \beta_{X_{i,\text{grp}}} + \beta_{\text{sum}} \log X_{i,\text{sum}} \quad (5.11)$$

as considered in Chapter 2, but using the reduced dataset. Construct model diagnostic plots for this model.

As shown in Example 5.3, and further elaborated on in Exercise 5.3, the log-normal regression model is also a generalized linear model with a log-link and a quadratic variance function. That is,

$$\log \mathbf{E}(Y_i) = \tilde{\beta}_0 + \beta_{X_{i,\text{grp}}} + \beta_{\text{sum}} \log X_{i,\text{sum}}. \quad (5.12)$$

2. Show that with a log-link and a quadratic variance function, $\mathcal{V}(\mu) = \mu^2$, the weights in the IWLS algorithm are always 1.
3. Fit the model given by (5.12). Investigate if the model fits the data. Compare the fitted values with those from the log-normal model.

An alternative to the log-normal model is the log-gamma model, where (5.11) still specifies the mean value, but where the log-response is assumed to have a gamma distribution. Such a model can be fitted as a generalized linear model with a gamma response, but with the identity link.

4. Fit the log-gamma model and discuss the model fit.
5. Construct pp-plots for both the log-normal and log-gamma model and discuss which response distribution appears most appropriate.

Exercise 5.5. Assume in this exercise that $Y_i \mid X_i$ has distribution with density $f_i(y_i, \eta_i)$ for $i = 1, \dots, n$, where $\eta_i = X_i^T \beta$ is the linear predictor. The generalized linear model with an exponential dispersion distribution is a special case with

$$\log f_i(y_i, \eta_i) = \frac{\theta(\eta_i)y_i - \zeta(\eta_i)}{\varphi}.$$

The purpose of the exercise is to investigate the computations of the maximum-likelihood estimator in this more general setup, and to understand which results are specific to exponential dispersion models. It is assumed that $f_i(y_i, \eta_i) > 0$, and we define

$$U_i(\eta_i) = (\log f_i(Y_i, \eta_i))'$$

and

$$w_i = -\mathbf{E}((\log f_i(Y_i, \eta_i))'')$$

where the differentiation is w.r.t. η_i . Introduce also the score vector $U(\eta) = (U_1(\eta_1), \dots, U_n(\eta_n))^T$ and the diagonal matrix $\mathbf{W}$ with $\mathbf{W}_{ii} = w_i$.

1. Prove that the score statistic equals $\mathbf{X}^T U(\eta)$ and that the Fisher information equals $\mathbf{X}^T \mathbf{W} \mathbf{X}$.

2. Prove that the m -th Fisher scoring step for solving the score equation amounts to solving the linear equation

$$\mathbf{X}U(\eta_m) - (\mathbf{X}^T\mathbf{W}_m\mathbf{X})(\beta - \beta_m) = 0.$$

Here $\mathbf{W}_m$ is defined in terms of $\eta_m = \mathbf{X}\beta_m$.

3. Show that if $w_{m,i} \geq 0$ then β_{m+1} is the solution of a weighted least squares problem with diagonal weight matrix $\mathbf{W}_m$ and working responses $X_i^T\beta_m + U_i(\eta_m)/w_{m,i}$.

Since w_i is the variance of the score statistic (and thus non-negative) under some regularity conditions, estimation can be carried out by iteratively weighted least squares also in the more general case considered in this exercise. However, this requires formulas for the score statistic and the weights, which cannot generally be obtained from the first two moments.

6 Existence and uniqueness

! Important

This chapter is currently undergoing conversion and substantial changes will be made without notice.

Exercises

Exercise 6.1. This exercise is about an additive genetic effects model. The response Y is an indicator of a phenotype, e.g., whether an individual has brown eyes or whether the individual has a particular disease. Some phenotypes are determined completely by well known genotypes, while others have a less well understood genetic component.

A genotype is an unordered pair of possible alleles (at a given genomic location) and can usually take three different values – here denoted bb , Bb and BB . The brown eye color is known to be dominant over the blue, so if B denotes the brown eye allele, then BB as well as Bb will result in brown eyes whereas only bb will result in blue eyes¹⁹.

For the purpose of this exercise you don't need to understand the genetic details, but genetics provides a context where the model considered is relevant. We recode the genotype as a numerical variable X as follows:

genotype	bb	Bb	BB
X	0	1	2

Thus X denotes the number of B alleles, and we will use it to construct a probabilistic model of the phenotype response Y

¹⁹ In reality, the genetics of eye color is somewhat more complicated, but this simple version of the story illustrates the principle.

given the genotype predictor X . The additive genetic effects model is given by

$$\text{logit}(\mathbf{P}(Y = 1 \mid X = x)) = \beta_0 + \beta_1 x. \quad (6.1)$$

Note that this additive model will not be a good model of eye color, but it can be a useful model of the genetic component of a disease.

Having observed n individuals we can summarize the data into a table of the following form:

X	0	1	2	Total
number of 1-s	m_0	m_1	m_2	m
number of 0-s	$n_0 - m_0$	$n_1 - m_1$	$n_2 - m_2$	$n - m$
Total	n_0	n_1	n_2	n

It is assumed that at least two of n_0, n_1 and n_2 are nonzero.

Introduce also the vectors

$$\mathbf{a} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad \text{and} \quad \mathbf{c} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}.$$

The first two questions deal with existence and uniqueness of the MLE in this model. Define

$$C = \{\mu_0 \mathbf{a} + \mu_1 \mathbf{b} + \mu_2 \mathbf{c} \mid \mu_0 \in (0, n_0), \mu_1 \in (0, n_1), \mu_2 \in (0, n_2)\}$$

1. Show that the MLE in the model given by Equation 6.1 exists and is unique if and only if

$$m_0 \mathbf{a} + m_1 \mathbf{b} + m_2 \mathbf{c} \in C.$$

Consider the following concrete data set:

X	0	1	2	Total
number of 1-s	0	1	5	6
number of 0-s	100	99	95	294
Total	100	100	100	300

In this data set there are $n = 300$ individuals and $m = 6$ of them had the disease. Of these, 5 had the BB allele, 1 had the Bb allele and 0 had the bb allele.

2. Make a figure illustrating the set C for the concrete data set in the table above. Show that in this case the MLE exists and is unique.
3. Find a general criterion in terms of n_0, n_1, n_2 and m_0, m_1, m_2 for the existence of the MLE.

Hint: You can simplify the problem to the case $n_0 = n_1 = n_2 = 100$.

4. Fit the model given by Equation 6.1 to the concrete data set in the table above.

Hint: You can fit the model using `glm()` based on the summary data in the table. This requires a specification of a two-column response in the formula. See also the remark about fitting binomial and quasibinomial models in the Details section in the help page for `glm()`.

7 Regression modeling

! Important

This chapter is currently undergoing conversion and substantial changes will be made without notice. For the sections on practical data analysis we refer, for the time being, to the 2020-edition.

This chapter deals with a range of general problems and questions related to regression modeling and the practical process of data analysis. Some of the general considerations apply to statistics and data analysis in a broad sense, but to keep focus, data analysis is treated in the framework of regression models only.

In the first part of the chapter we position the linear model from Chapter 2 within a more general framework. This makes it possible to formulate some of the central objectives in regression analysis without relying on specific model assumptions.

In the subsequent parts we turn to the process of practical data analysis, which involves a lot of decision making. For an inexperienced data analyst the decision making can be a daunting task, first of all because there is never a set of clearly correct decisions. Instead it appears as if many decisions have to be made on the basis of little knowledge, and whenever a decision is made, it may be open to criticism. It is therefore tempting to hide the decision process, but this can be detrimental to the scientific process.

We will develop methods for understanding and controlling the decision making process involved in practical data analysis. This ranges from grand decisions, like what the overall purpose of the analysis is, to technical decisions, such as how to handle missing values. Central to the decision process is a solid understanding of how we can interpret an associational regression model and what it can ultimately be used for.

For this reason we will also briefly treat some central *causal* questions and how and when we can use regression models to answer them.

The chapter concludes in Section 7.3.1 by introducing basis expansion techniques to capture nonlinear relations within the framework of the linear model. The techniques are illustrated by a spline basis expansion of the relation between claim size and insurance sum in the fire insurance example.

7.1 Outline and prerequisites

The chapter primarily relies on the following four R packages.

```
library(RwR)
library(ggplot2)
library(tibble)
library(broom)
library(splines)
library(dplyr)
```

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

`filter`, `lag`

The following objects are masked from 'package:base':

`intersect`, `setdiff`, `setequal`, `union`

7.2 General regression models

There exists a range of generalizations of the linear regression model. It would be misleading to lump them all together under the umbrella term *nonlinear* regression. Some generalizations focus specifically on nonlinear relations between the response and the predictors, while other generalizations focus on other targets than the conditional expectation, e.g., the median or certain quantiles.

We will consider various generalizations in this book. In Chapter 5 the target is still the conditional expectation – with a particular form of nonlinearity. But just as importantly, the chapter generalizes both the estimation methodology and the statistical theory to account for variance heterogeneity.

In Chapter 11 the target is the conditional hazard function of a survival distribution. One particular class of such survival regression models are models of the conditional expectation of the *log-survival time*, but most models are not directly targeting a conditional expectation.

In this chapter we treat some general aspects of regression modeling without making any of the linear modeling assumptions A1, A2 or A3, but for clarity, the modeling target will remain the conditional expectation throughout.

Definition 7.1. Let $X \in \mathbb{R}^p$ and $Y \in \mathbb{R}$ be random variables and suppose that $\mathbf{E}(|Y|) < \infty$. The regression function $m : \mathbb{R}^p \rightarrow \mathbb{R}$ is defined by

$$m(x) = \mathbf{E}(Y \mid X = x) \quad (7.1)$$

The regression function m is in general only uniquely defined almost surely w.r.t. to the distribution of X . Though if it has a continuous version, that version is unique on the support of X . In most applications it is reasonable to assume that m is continuous.

A general regression model is a set $\mathcal{M}$ of regression functions. That is,

$$\mathcal{M} \subseteq \{m \mid m : \mathbb{R}^p \rightarrow \mathbb{R}\}.$$

We see that the linearity assumption A1 can be expressed in terms of the regression function as

$$m(x) = x^T \beta,$$

which means that the corresponding linear regression model can be expressed as

$$\mathcal{M}_{\text{linear}} = \{x \mapsto x^T \beta \mid \beta \in \mathbb{R}^p\}.$$

The linear regression model is parametrized by the parameter β belonging to the parameter set $\mathbb{R}^p$.

Most regression models are parametrized by a parameter, abstractly denoted $\theta \in \Theta$, and we then write the model as

$$\mathcal{M} = \{m_\theta \mid \theta \in \Theta\}.$$

Example 7.1 (Dose-response curves). A simple classical example is the *logistic function* for $p = 1$:

$$m_{(a,b,c)}(x) = \frac{\beta}{1 + e^{-(bx+a)}} + \alpha \quad (7.2)$$

The parameter $\theta = (a, b, \alpha, \beta)$ is 4-dimensional and the parameter set is $\Theta = \mathbb{R}^4$. The S-shaped graph of a logistic function can, for instance, model the relation between a dosage (x) and a continuous response (y).

Example 7.2 (Neural networks). Let

$$\mu, h : \mathbb{R} \rightarrow \mathbb{R}$$

be two fixed (nonlinear) functions. With $\mathbf{B}$ a real $p \times q$ matrix, $\beta \in \mathbb{R}^q$ and $\alpha \in \mathbb{R}$ we define the regression function

$$m_{\mathbf{B},\alpha,\beta}(x) = \mu(\alpha + h(x^T \mathbf{B})\beta)$$

parametrized by $(\mathbf{B}, \alpha, \beta)$. The convention above is that h is applied coordinatewisely to the q -dimensional (row)-vector $x^T \mathbf{B}$.

This is an example of a neural network with a single hidden layer. The matrix $\mathbf{B}$ parametrizes the *input layer* mapping the predictor vector $x \in \mathbb{R}^p$ to $h(x^T \mathbf{B}) \in \mathbb{R}^q$. It is a composition of the parametrized linear map $x \mapsto x^T \mathbf{B}$ and a fixed coordinatewise nonlinear function h . The number α and the vector β parametrize the *output layer*, which is likewise a composition of a linear map and the nonlinear function μ .

We introduce the notation $L_{\alpha,\mathbf{B}}^h(x) = h(\alpha + x^T \mathbf{B})$ for an activation function h , a $p \times q$ matrix $\mathbf{B}$ and a vector $\alpha \in \mathbb{R}^q$. Then $L_{\alpha,\mathbf{B}}^h : \mathbb{R}^p \rightarrow \mathbb{R}^q$ and it represents a single layer in a neural network. Using this notation we can write the neural network regression function above as a composition of two such maps:

$$m_{\mathbf{B},\alpha,\beta} = L_{\alpha,\beta}^\mu \circ L_{0,\mathbf{B}}^h$$

The functions h and μ are known as *activation functions* in the literature on neural networks.

A general neural network with k hidden layers can be defined as

$$m_{\alpha_1, \mathbf{B}_1, \dots, \alpha_k, \mathbf{B}_k, \alpha, \beta} = L_{\alpha, \beta}^{\mu} \circ L_{\alpha_k, \mathbf{B}_k}^{h_k} \circ \dots \circ L_{\alpha_1, \mathbf{B}_1}^{h_1}$$

The model is parametrized by $(\alpha_1, \mathbf{B}_1, \dots, \alpha_k, \mathbf{B}_k, \alpha, \beta)$ where $\mathbf{B}_j$ is a $p_{j-1} \times p_j$ real matrix ($p_0 = p$), $\alpha_j \in \mathbb{R}^{p_j}$, $\beta \in \mathbb{R}^{p_k}$ and $\alpha \in \mathbb{R}$.

It is quite common to use the same activation function for all layers, that is, $h_1 = h_2 = \dots = h_k = \mu$. A couple of standard choices are:

$$\begin{aligned} \mu(\eta) &= (1 + e^{-\eta})^{-1} && \text{(logistic)} \\ \mu(\eta) &= \log(1 + e^{\eta}) && \text{(softplus)} \\ \mu(\eta) &= \max(0, \eta) && \text{(ReLU)} \end{aligned}$$

We will in Chapter 5 treat *generalized linear models*, which can be regarded as neural networks with no hidden layers. That is, the regression function is of the form

$$m_{\beta}(x) = \mu(x^T \beta)$$

We will in that context see canonical choices of the activation function μ dictated by the distribution of the response variable.

Even though the parameter space of a neural network can be very large – in particular for deep neural networks with many hidden layers – the parameter space is formally a finite dimensional space. We might want to choose a model $\mathcal{M}$ that is even larger, e.g., the set of all continuous functions, the set of all Lipschitz functions, or the set of all smooth functions. These sets cannot be parametrized in a nice way by a finite dimensional parameter²⁰, and we call the corresponding regression models *nonparametric*.

There are substantial theoretical differences between trying to estimate a regression function in a parametric model or in a nonparametric model. From a practical perspective, though, the transition from small parametric models to nonparametric models via large parametric models is much more gradual. We will not deal with estimation methods or theory that are

The 4-parameter dose-response model above is an example of a neural network with $p = q = 1$ and activation function being the logistic function.

²⁰ More precisely, they are not finite dimensional differentiable manifolds.

explicitly nonparametric, but it can be conceptually advantageous to define and discuss the purposes and objectives of a regression analysis in a nonparametric way. Doing so, we will be able to define our aims clearly without reference to a particular parametric model – even though we eventually, in this book, develop solutions in terms of specific parametric models.

7.2.1 Estimation methodologies

The most widely used estimator of the regression function is the least squares estimator minimizing the squared error loss. With a model $\mathcal{M}$ and observations $(X_1, Y_1), \dots, (X_n, Y_n)$ it is formally defined as

$$\hat{m} = \arg \min_{m \in \mathcal{M}} \sum_{i=1}^n (Y_i - m(X_i))^2. \quad (7.3)$$

If the model is parametrized by $\theta \in \Theta$, we can formulate the estimator in terms of estimating θ

$$\hat{\theta} = \arg \min_{\theta \in \Theta} \sum_{i=1}^n (Y_i - m_{\theta}(X_i))^2. \quad (7.4)$$

Example 7.3 (Birth weight).

There are several practical problems with these formal definitions, though. First, there may be no global minimizer of the sum of squares or there may be many. Second, it may be practically impossible to compute any such global minimizer unless $\mathcal{M}$ is particularly nice, e.g., $\mathcal{M} = \mathcal{M}_{\text{linear}}$ as in Theorem 2.1. Third, even if we could compute $\hat{m}$ it might not be a good estimator if $\mathcal{M}$ is very large relative to the sample size n .

Many practical estimation algorithms do not attempt to literally compute $\hat{m}$ or $\hat{\theta}$ as defined by (7.3) and (7.4), respectively. Typically, they iteratively find and improve candidate regression functions in the model – using the squared error loss as guide rather than trying to drive it to its minimal value. This can be done in many ways, e.g., by explicitly restricting the model space to an n -dependent subset $\mathcal{M}_n \subseteq \mathcal{M}$, or by regularizing the estimator via a penalty function $\text{pen} : \mathcal{M} \rightarrow \mathbb{R}$. In the latter case, the

For sufficiently flexible models $\mathcal{M}$, e.g., models containing all degree n polynomials, there is a subset $\mathcal{M}_0 \subseteq \mathcal{M}$ such that for all $m \in \mathcal{M}_0$, $m(X_i) = Y_i$ for all $i = 1, \dots, n$. All regression functions in $\mathcal{M}_0$ are said to *interpolate* the data perfectly, and they all achieve the globally minimal squared error loss of 0.

algorithm would then (try to) compute

$$\hat{m}^{\lambda_n} = \arg \min_{m \in \mathcal{M}} \sum_{i=1}^n (Y_i - m(X_i))^2 + \lambda_n \text{pen}(m) \quad (7.5)$$

for an n -dependent parameter $\lambda_n > 0$ controlling the tradeoff between the squared error loss and the penalty term. Typically, $\lambda_n \rightarrow 0$ for $n \rightarrow \infty$ so that the regularization has less and less impact on the estimator as the sample size increases. Section 7.3.1 demonstrated by example the use of penalization within the framework of a linear model, where the penalization was used to control the fluctuations of nonlinear interaction terms.

We can interpret the term $\lambda_n \text{pen}(m)$ as an *inductive bias*, where we bias the estimator toward small values of $\text{pen}(m)$. Another way to introduce an inductive bias is by *early stopping*. If we start the optimization algorithm in a particular subset $\mathcal{M}_0$ and then stop it before convergence, the result is implicitly biased toward models in $\mathcal{M}_0$. An algorithm could implement early stopping by monitoring the squared error loss on an *independent dataset* and stop when that error does not improve anymore. Similarly, the parameter λ_n could be selected by minimizing the squared error loss on an independent dataset over a range of possible values of λ_n .

For a notable part of this book we consider parametrized models and loss functions that are nice enough for the estimators we consider to actually be global minimizers, such as $\hat{\theta}$ in (7.4). Moving beyond those models, regression function estimators used in practice are mostly the result of running a complicated algorithm on data, which does not have a simple analytic representation as, e.g., a minimizer or a solution to an equation. This does not make such estimators bad, but it does make it somewhat more complicated to understand their theoretical properties. The least squares estimator in $\mathcal{M}_{\text{linear}}$, as treated in detail in Chapter 2, stands out as having a particularly well understood theory, which we can leverage for practical statistical data analysis. We cannot obtain a similarly simple theory for general regression function estimators, but we formulate our modeling objectives in terms of general regression functions in clear and simple way that does not hinge on simplistic model assumptions.

We use *inductive bias* to mean a model assumption or algorithmic constraint that steers the estimator in a particular direction. The concept is philosophically close to a Bayesian prior, and Bayesian statistics offers a principled way to transform prior beliefs to inductive biases. The folklore interpretation of the *No Free Lunch theorem* is that we cannot estimate m to generalize beyond data without some inductive bias.

7.2.2 Regression modeling objectives

As discussed at some length in the introduction, Chapter 1, one purpose of estimating the regression function m is to use $\hat{m}$ to make predictions about Y from observation of X . This is the primary focus in most of the machine learning literature. Though we will cover this aspect in later chapters – where we discuss quantification and optimization of predictive strength of a model – we also deal at length with other objectives.

One main question we consider is:

Does m depend on x_i ?

To formulate this precisely, let $x_{-j} = (x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_p)^T$ denote the $(p-1)$ -dimensional vector with the j -th coordinate removed from x , and let

$$m_{-j}(x_{-j}) = \mathbf{E}(Y \mid X_{-j} = x_{-j})$$

denote the regression function based on the X_{-j} -predictors only. We can then formulate the hypothesis

$$H_0 : \mathbf{E}(Y \mid X) = m_{-j}(X_{-j}). \quad (7.6)$$

This hypothesis means that m does not depend on the i -th coordinate, and it is an example of a conditional (mean) independence hypothesis. We will develop statistical theory to test H_0 based on various regression modeling frameworks.

Another main objective we consider is to:

Quantify (and estimate) how much m depends on a particular coordinate x_i .

Obviously, this objective is closely related to the hypothesis above – any sensible quantification should show that m does not depend on x_i if H_0 is true. Additionally, such a quantification is naturally tied together with the development of a test statistic of H_0 . Testing H_0 is, however, a simpler problem, and the challenge is to develop easily interpretable quantifications that are simultaneously easy to estimate. The regression modeling frameworks we consider will have some natural ways of quantifying such conditional associations.

In this section we focus for clarity on how m depends on a single coordinate. Similar questions can, of course, be formulated for any group of coordinates.

To aid the general discussion of the two objectives above, we introduce partial dependence functions and partial errors below. To ease notation we will use the convention that

$$m(x_j, x_{-j}) = m((x_1, \dots, x_{j-1}, x_j, x_{j+1}, \dots, x_p)^T),$$

where $m(x_j, x_{-j})$ is simply a shorthand notation for the right hand side above. Recall that $m(x) = \mathbf{E}(Y \mid X = x)$ denotes the joint regression function of Y on X .

Definition 7.2. The partial dependence function $m_j : \mathbb{R} \rightarrow \mathbb{R}$ for the j -th coordinate is

$$m_j(x_j) = \mathbf{E}(m(x_j, X_{-j})) = \mathbf{E}(\mathbf{E}(Y \mid X_j = x_j, X_{-j})). \quad (7.7)$$

The partial dependence function is also known as the adjusted regression function. It is essential for its definition that we integrate out over the *marginal* distribution of X_{-j} and not the conditional distribution of X_{-j} given $X_j = x_j$.

Recall that $m_{-j}(x_{-j}) = \mathbf{E}(Y \mid X_{-j} = x_{-j})$ is the regression function of Y on X_{-j} excluding the j -th coordinate of X .

Definition 7.3. The j -th partial error is defined as

$$\varepsilon_{-j} = Y - m_{-j}(X_{-j}). \quad (7.8)$$

Note that the tower property of conditional expectations imply the identity

$$m_{-j}(x_{-j}) = \mathbf{E}(m(X_j, x_{-j}) \mid X_{-j} = x_{-j}),$$

which relate m_{-j} to m . We can now show the following result on implications of the hypothesis H_0 for the partial dependence function and the partial error.

Proposition 7.1. *If the regression function $m(x) = \mathbf{E}(Y \mid X = x)$ does not depend on x_j then*

1. *The partial dependence function m_j is constant.*
2. *$m_{-j}(x_{-j}) = m(x_j, x_{-j})$ for any x_j .*
3. *$\text{cov}(\varepsilon_{-j}, X_j) = 0$.*

Proof. It follows directly from (7.7) that if m does not depend on x_j then $m_j(x_j) = \mathbf{E}(m(X_{-j}))$ is constant.

To prove the second claim we note that if m does not depend on the j -th coordinate, $m(X_j, x_{-j}) = m(x_j, x_{-j})$ for any fixed x_j , whence

$$m_{-j}(x_{-j}) = \mathbf{E}(m(X_j, x_{-j}) \mid X_{-j} = x_{-j}) = m(x_j, x_{-j}).$$

Finally, if m does not depend on x_j , $\varepsilon_{-j} = \varepsilon = Y - \mathbf{E}(Y \mid X)$. Whence by the tower property of conditional expectations,

$$\begin{aligned} \text{cov}(\varepsilon_{-j}, X_j) &= \text{cov}(\varepsilon, X_j) \\ &= \mathbf{E}(\varepsilon X_j) \\ &= \mathbf{E}(\mathbf{E}(\varepsilon X_j \mid X)) \\ &= \mathbf{E}(\mathbf{E}(\varepsilon \mid X) X_j) = 0, \end{aligned}$$

where we have used that $\mathbf{E}(\varepsilon \mid X) = 0$. $\square$

There are various procedures, motivated by Proposition 7.1, that are used in practice to better understand how m depends on a particular predictor. To describe them, let $\hat{m}$ and $\hat{m}_{-j}$ denote estimates of m and m_{-j} , respectively.

1. The *partial dependence plot* is simply a plot of $\hat{m}_j(X_{i,j})$ against $X_{i,j}$, where

$$\hat{m}_j(X_{i,j}) = \frac{1}{n} \sum_{i'=1}^n \hat{m}(X_{i,j}, X_{i',-j}).$$

It is a direct graphical check of whether m_j is constant.

2. For a q -dimensional model $\mathcal{M}$ of m and a q_0 dimensional model $\mathcal{M}_{-j}$ of m_{-j} , the F -test statistic is

$$F = \frac{\sum_{i=1}^n (\hat{m}_{-j}(X_{i,-j}) - \hat{m}(X_i))^2 (q - q_0)}{\sum_{i=1}^n (Y_i - \hat{m}(X_i))^2 / (n - q)}.$$

It quantifies directly any differences between m_{-j} and m .

3. Defining the j -th *partial residuals* as

$$\hat{\varepsilon}_{i,-j} = Y_i - \hat{m}_{-j}(X_{i,-j}),$$

we can plot $\hat{\varepsilon}_{i,-j}$ against $X_{i,j}$.

4. Letting $\hat{r}_j$ denote an estimate of the regression function

$$r_j(x_{-j}) = \mathbf{E}(X_j \mid X_{-j} = x_{-j}),$$

we can also plot the partial residuals $\hat{\varepsilon}_{i,-j}$ against $X_{i,j} - \hat{r}_j(X_{i,-j})$. This plot is known as an *added variable plot*.

The suggested plots of the partial residuals against either the j -th predictor or its residual are both attempts to visualize dependencies between ε_{-j} and X_j . The plots can be used to graphically detect if $\text{cov}(\varepsilon_{-j}, X_j) = 0$ or not. We can also use the partial residuals to directly estimate the covariance, either as

$$\text{cov}\tilde{v}_j = \frac{1}{n} \sum_{i=1}^n \hat{\varepsilon}_{i,-j} X_{i,j} \quad (7.9)$$

or as

$$\text{cov}\hat{v}_j = \frac{1}{n} \sum_{i=1}^n \hat{\varepsilon}_{i,-j} (X_{i,j} - \hat{r}_j(X_{i,-j})). \quad (7.10)$$

Both of these covariance estimates quantify deviations from H_0 , and in this way indirectly also how much m depends on the j -th coordinate. The estimator (7.9) has the advantage that we do not need to estimate the regression function r_j , but it can be severely biased and $\text{cov}\hat{v}_j$ is preferred in practice.

To better appreciate the general definitions and procedures outlined above, we go through four different examples in increasing complexity, where most of the definitions and quantities above have some explicit expressions in terms of model parameters or components.

Example 7.4. If $m(x) = x^T \beta$ is linear we find that the partial dependence function is

$$m_j(x_j) = \beta_j x_j + \mathbf{E}(X_{-j})^T \beta_{-j}.$$

We see that the partial dependence function is affine with slope β_j . It is constant if and only if $\beta_j = 0$, which is the case if and only if m does not depend on x_j .

The F -test statistic defined above coincides for this model with the F -test statistic defined by Equation 2.11, and it can be used to formally test H_0 .

We also find that

$$m_{-j}(x_{-j}) = x_{-j}^T \beta_{-j} + \beta_j \mathbf{E}(X_j \mid X_{-j} = x_{-j}) = x_{-j}^T \beta_{-j} + \beta_j r_j(x_{-j}).$$

Therefore

$$m(x) - m_{-j}(x_{-j}) = \beta_j(x_j - r_j(x_{-j}))$$

This difference is non-zero only if $\beta_j \neq 0$. Moreover,

$$\varepsilon_{-j} = \varepsilon + \beta_j(X_j - r_j(X_{-j}))$$

and we find that

$$\text{cov}(\varepsilon_{-j}, X_j) = \beta_j \mathbf{V}(X_j - r_j(X_{-j})).$$

This covariance is thus proportional to β_j , with a proportionality constant $\mathbf{V}(X_j - r_j(X_{-j}))$ depending on the distribution of X only.

Example 7.5. The regression function is called *partially linear* in the j -th variable if it is of the form

$$m(x) = \beta_j x_j + h_{-j}(x_{-j}).$$

In this case, similarly as above, the partial dependence function is an affine function

$$m_j(x_j) = \beta_j x_j + \mathbf{E}(h_{-j}(X_{-j}))$$

with slope β_j . Moreover,

$$m_{-j}(x_{-j}) = h_{-j}(x_{-j}) + \beta_j \mathbf{E}(X_j \mid X_{-j} = x_{-j}) = h_{-j}(x_{-j}) + \beta_j r_j(x_{-j}),$$

and thus

$$m(x) - m_{-j}(x_{-j}) = \beta_j(x_j - r_j(x_{-j}))$$

exactly as if m were linear. By the same argument as above, we get that

$$\text{cov}(\varepsilon_{-j}, X_j) = \beta_j \mathbf{V}(X_j - r_j(X_{-j})).$$

The computations above for a partially linear regression function reveal that the partial dependence plot as well as procedures based on the partial residuals should reveal the same affine dependence of m on x_j whether m is linear or only partially linear. The subtle, but important, practical difference is that this is only true if the estimators used for estimating the partial dependence function or for computing the partial residuals are flexible enough to correctly model the potentially nonlinear functions h_{-j} and m_{-j} .

Example 7.6. If the regression function has the form

$$m(x) = \beta_j(x_{-j})x_j + h_{-j}(x_{-j}),$$

it is affine in x_j for any fixed x_{-j} . It is partially linear in the j -th variable but with a coefficient that depends on x_{-j} . The partial dependence function is the affine function

$$m_j(x_j) = \mathbf{E}(\beta_j(X_{-j}))x_j + \mathbf{E}(h_{-j}(X_{-j}))$$

with slope $\mathbf{E}(\beta_j(X_{-j}))$ being the mean slope of x_j in m .

$$m_{-j}(x_{-j}) = h_{-j}(x_{-j}) + \beta_j(x_{-j})\mathbf{E}(X_j | X_{-j} = x_{-j}) = h_{-j}(x_{-j}) + \beta_j(x_{-j})r_j(x_{-j}),$$

and thus

$$m(x) - m_{-j}(x_{-j}) = \beta_j(x_{-j})(x_j - r_j(x_{-j})).$$

The covariance becomes

$$\begin{aligned} \text{cov}(\varepsilon_{-j}, X_j) &= \mathbf{E}(\beta_j(X_{-j})(X_j - r_j(X_{-j}))^2) \\ &= \mathbf{E}(\beta_j(X_{-j})\mathbf{V}(X_j - r_j(X_{-j}) | X_{-j})) \\ &= \mathbf{E}(\beta_j(X_{-j})v_j(X_{-j})), \end{aligned}$$

which we can interpret as a weighted mean of the individual slopes with weights $v_j(X_{-j}) = \mathbf{V}(X_j - r_j(X_{-j}) | X_{-j})$ being conditional variances.

Example 7.7. The regression function is called *partially additive* in the j -th variable if it is of the form

$$m(x) = m_j^0(x_j) + h_{-j}(x_{-j}).$$

The partial dependence function is then

$$m_j(x_j) = m_j^0(x_j) + \mathbf{E}(h_{-j}(X_{-j})),$$

which up to an additive constant coincides with m_j^0 . The regression function m_{-j} becomes

$$m_{-j}(x_{-j}) = h_{-j}(x_{-j}) + \mathbf{E}(m_j^0(X_j) | X_{-j} = x_{-j}) = h_{-j}(x_{-j}) + w_{-j}(x_{-j})$$

where $w_{-j}(x_{-j}) = \mathbf{E}(m_j^0(X_j) | X_{-j} = x_{-j})$. Therefore

$$m(x) - m_{-j}(x_{-j}) = m_j^0(x_j) - w_{-j}(x_{-j}).$$

Finally

$$\begin{aligned} \text{cov}(\varepsilon_{-j}, X_j) &= \text{cov}(m_j^0(X_j) - w_{-j}(X_{-j}), X_j) \\ &= \mathbf{E}(\text{cov}(m_j^0(X_j), X_j | X_{-j})). \end{aligned}$$

7.3 Nonlinear basis expansions

7.3.1 Nonlinear expansions for claims modeling

We found in the earlier analysis of the insurance claims data a lack of model fit. This could be due to a nonlinear relation between $\log(\text{claims})$ and $\log(\text{sum})$. To handle nonlinear relations between the response and one or more predictors, the predictors (as well as the response) can be nonlinearly transformed before they enter into the linear model. Such pre-modeling transformations extend the scope of the linear model considerably. For the insurance data the original variables were already log-transformed, and it is not obvious which other transformation to use.

```
x <- seq(11, 21, 0.05)
pol <- as_tibble(cbind(x, poly(x, degree = 5)))
long_pol <- tidyr::pivot_longer(pol, cols = !x, names_to = "basis_fct")
p <- ggplot(NULL, aes(x, value, colour = basis_fct)) +
  scale_color_discrete(guide = "none") + xlab("") + ylab("")
p + geom_line(data = long_pol, linewidth = 1)
```

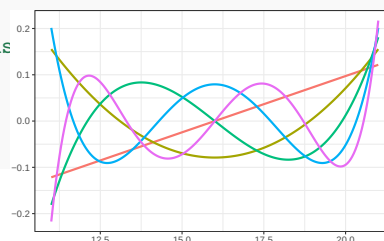


Figure 7.1: A basis of orthogonal polynomials.

An alternative to data transformations is to use a small but flexible class of *basis functions*, which can capture the non-linearity. One possibility is to use low degree polynomials. This can be done by simply including powers of a predictor as additional predictors. [Due to the parsing of formulas in R, powers have to be wrapped into an `I()` }-call, e.g., $y \sim x + I(x^2)$.] Depending on the scale and range of the predictor, this may work just fine. However, raw powers can result in numerical difficulties. An alternative is to use orthogonal polynomials, which are numerically more well behaved.

Figure 7.1 shows an example of an orthogonal basis of degree 5 polynomials on the range 11–21. This corresponds approximately to the range of $\log(\text{sum})$, which is a target for basis expansion in the example. What should be noted in Figure 7.1 is that the behavior near the boundary is quite erratic. This is characteristic for polynomial bases. To achieve flexibility in the central part of the range, the polynomials become erratic close to the boundaries. Extrapolation beyond the boundaries cannot be trusted.

An alternative to polynomials is splines. A spline²¹ is piecewisely a polynomial, and the pieces are joined together in a

²¹ A spline is also a thin and flexible wood or metal strip used for smooth curve drawing.

sufficiently smooth way. The points where the polynomials are joined together are called *knots*. The flexibility of a spline is determined by the number and placement of the knots and the degree of the polynomials. A degree k spline is required to be $k - 1$ times continuously differentiable. A degree 3 spline, also known as a *cubic spline*, is a popular choice, which thus has a continuous (piecewise linear) second derivative.

```
b_spline <- as_tibble(cbind(x, splines::bs(x, knots = c(14, 18))
long_b_spline <- tidyr::pivot_longer(b_spline, cols = !x, names
p + geom_line(data = long_b_spline, linewidth = 1) +
  geom_rug(aes(x = c(11, 14, 18, 21), y = NULL, color = NULL))
```

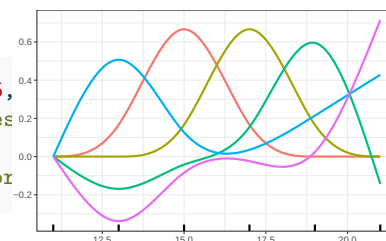


Figure 7.2 shows a basis of 5 cubic spline functions. They are so-called *B-splines* (basis splines). Note that it is impossible to visually detect the knots where the second derivative is non-differentiable. The degree k *B-spline* basis with r internal knots has $k + r$ basis functions. The constant function is not included then, and this is precisely what we want when the basis expansion is used in a regression model including an intercept. As seen in Figure 7.2, the *B-spline* basis is also somewhat erratic close to the boundary. For a cubic spline, the behavior close to the boundary can be controlled by requiring that the second and third derivatives are 0 at the boundary knots. The result is known as a *natural cubic spline*. The extrapolation (as a spline) of a natural cubic spline beyond the boundary knots is linear.

Figure 7.2: A basis of cubic *B-splines* computed using the `bs()` function with two internal knots at 14 and 18 in addition to two boundary knots at 11 and 21.

Due to the restriction on the derivatives of a natural cubic spline, the basis with r internal knots has $r + 1$ basis functions. Thus the basis for the natural cubic splines with $r + 2$ internal knots has the same number of basis functions as the raw cubic *B-spline* basis with r internal knots. This means in practice that, compared to using raw *B-splines* with r internal knots, we can add two internal knots, and thus increase the central flexibility of the model, while retaining its complexity in terms of $r + 3$ parameters.

```
n_spline <- as_tibble(cbind(x, splines::ns(x, knots = c(13, 15,
long_n_spline <- tidyr::pivot_longer(n_spline, cols = !x, names
p + geom_line(data = long_n_spline, linewidth = 1) +
  geom_rug(aes(x = c(11, 13, 15, 17, 19, 21), y = NULL, color = NULL))
```



It is possible to use orthogonal polynomial expansions as well as spline expansions in R together with the `lm()` function via

Figure 7.3: A *B-spline* basis for natural cubic splines computed using the `ns()` function with internal knots at 13, 15, 17 and 19 in addition to the two boundary knots at 11 and 21.

the functions `poly()`, `bs()` and `ns()`. The latter two are in the splines package, which thus has to be loaded.

We will illustrate the usage of basis expansions for claim size modeling using natural cubic splines.

```
claims_lm_spline_add <- lm(
  log(claims) ~ ns(log(sum), knots = c(13, 15, 17, 19)) + grp,
  data = claims
)
claims_lm_spline_int <- lm(
  log(claims) ~ ns(log(sum), knots = c(13, 15, 17, 19)) * grp,
  data = claims
)
```

When we fit a model using basis expansions the coefficients of individual basis functions are rarely interpretable or of interest. Thus reporting the summary including all the estimated coefficients is not particularly informative. Below we only extract the last lines from the summary to report the residual variance and the R^2 quantities.

```
glance(claims_lm_spline_add) |>
  knitr::kable()
```

r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC	deviance	df.residual
0.06416	0.0623	1.9185	34.509	0	8	-8352	16724	16787	14822	4027

Table 7.1: Summary statistics for the additive model based on a spline basis expansion.

```
glance(claims_lm_spline_int) |>
  knitr::kable()
```

r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC	deviance	df.residual
0.07294	0.06762	1.9131	13.724	0	23	-8333	16716	16874	14683	4012

Table 7.2: Summary statistics for the interaction model based on a spline basis expansion.

Compared to the additive model without the basis expansion, R^2 (also the adjusted one) is increased a little by the basis expansion. A further increase is observed for the model that includes interactions as well as a basis expansion, and thus gives a unique nonlinear relation between insurance sum and

claim size for each trade group. Figure 7.4 shows some diagnostic plots for the additive model, which show that the mean value model is now pretty good, while the residuals are still slightly right skewed. The lack of normality is not a problem for using the sampling distributions to draw inference. However, if we compute prediction intervals, say, then we must account for the skewness in the residual distribution. This can be done by using quantiles for the empirical distribution of the residuals instead of quantiles for the normal distribution.

```
claims_diag_spline <- augment(claims_lm_spline_add) ## Residuals etc.
gridExtra::grid.arrange(
  ggplot(claims_diag_spline, aes(.fitted, .resid), alpha = 0.2) + geom_point() + geom_smooth(),
  ggplot(claims_diag_spline, aes(.resid)) + geom_histogram(bins = 40),
  ggplot(claims_diag_spline, aes(sample = .std.resid)) + geom_qq() + geom_abline(),
  ncol = 3
)
```

```
Warning in fortify(data, ...): Arguments in `...` must be used.
x Problematic argument:
* alpha = 0.2
i Did you misspell an argument name?
```

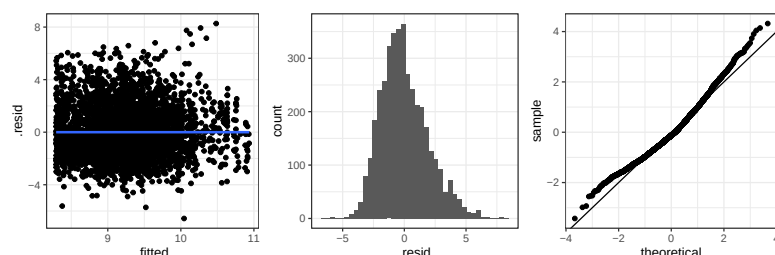


Figure 7.4: Residual plot (left), histogram (middle) of the raw residuals and qq-plot (right) of the standardized residuals for the additive model with a basis expansion of log insurance sum.

Graphical comparisons are typically preferable over parameter comparisons when models involving nonlinear effects and basis expansions are compared.

```
p0 <- ggplot(
  data = claims,
  aes(sum, claims),
  alpha = I(0.2)
) +
  scale_x_log10(
    "Insurance sum (DKK)",
    breaks = 10^c(6, 7, 8),
    labels = c("1M", "10M", "100M")
  ) +
  scale_y_log10(
```



```

    "Claim size (DKK)",
    breaks = 10~c(2, 4, 6, 8),
    labels = c("100", "10K", "1M", "100M")
  ) +
  geom_point()

```

Warning in fortify(data, ...): Arguments in `...` must be used.
 x Problematic argument:
 * alpha = I(0.2)
 i Did you misspell an argument name?

```

pred_add <- cbind(
  claims,
  exp(predict(claims_lm_spline_add, interval = "confidence"))
)
pred_int <- cbind(
  claims,
  exp(predict(claims_lm_spline_int, interval = "confidence"))
)
p0 <- p0 + facet_wrap(~ grp, ncol = 4) + coord_cartesian(ylim = c(100, 1e6))
p0 + geom_line(aes(y = fit), pred_add, color = "blue", linewidth = 2) +
  geom_ribbon(aes(ymin = lwr, ymax = upr), pred_add, fill = "blue", alpha = 0.2) +
  geom_line(aes(y = fit), pred_int, color = "red", linewidth = 2) +
  geom_ribbon(aes(ymin = lwr, ymax = upr), pred_int, fill = "red", alpha = 0.2)

```

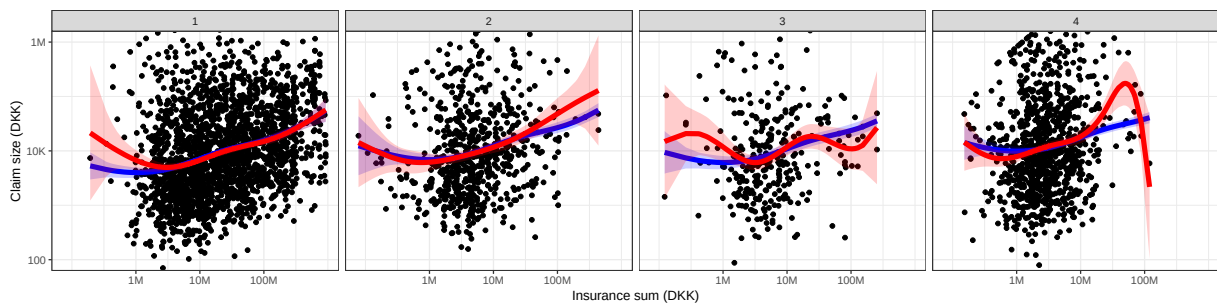


Figure 7.5: Scatter plots and fitted values for the claim size models using natural cubic splines including 95% pointwise confidence bands. The additive model (blue) gives translations of the same nonlinear relation for all four trade groups whereas the interaction model (red) gives a separate nonlinear relation for all four trade groups.

Figure 7.5 shows the fitted models stratified according to trade group. The fitted values were computed using the `predict()` function in R, which can also give upper and lower confidence bounds on the fitted values. The fitted value for the i -th observation is $X_i^T \hat{\beta}$. By Theorem 2.3 a confidence interval for the fitted value can be computed using Equation 2.10 with $a = X_i$. The confidence intervals reported by `predict()` for

lm objects are computed using this formula, and these are the pointwise confidence bands shown in Figure 7.5. From this figure any gain of the interaction model over the additive model is questionable. The increase of R^2 , say, appears to be mostly driven by spurious fluctuations. For trade group 4 the capped claims seem overly influential, and because trade group 3 has relatively few observations the nonlinear fit is poorly determined for this group.

```
claims_lm_add <- lm(
  log(claims) ~ log(sum) + grp,
  data = claims
)
```

```
anova(
  claims_lm_add,
  claims_lm_spline_add,
  claims_lm_spline_int
) |> knitr::kable()
```

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
4031	14902.71	NA	NA	NA	NA
4027	14822.42	4	80.28605	5.484225	0.0002115
4012	14683.37	15	139.04900	2.532863	0.0009413

Table 7.3: Analysis of variance table comparing the two models based on spline expansions and the additive linear model.

In the final comparison above using F -tests we first test the hypothesis that the relation is linear in the additive model, and then we test the additive hypothesis when the nonlinear relation is included. The tests formally reject both hypotheses, but the graphical summary in Figure 7.5 suggests that the nonlinear additive model is preferable anyway.

Introducing a penalty can dampen the spurious fluctuations the arose from the nonlinear basis expansion. We conclude the analysis of the insurance data by illustrating how the penalized least squares fit can be computed, and what effect the penalty has on the least squares solution. First recall that the least squares estimator could have been computed by solving the normal equation directly.

```

X <- model.matrix(claims_lm_spline_int)
y <- model.response(model.frame(claims_lm_spline_int))
XtX <- crossprod(X)      # Faster than but equivalent to t(X) %*% X
Xty <- crossprod(X, y)   # Faster than but equivalent to t(X) %*% y
coef_hat <- solve(XtX, Xty)
# Comparison
range(coef_hat - coefficients(claims_lm_spline_int))

```

```
[1] -8.441248e-11  1.365330e-11
```

The result obtained by solving the normal equation directly is up to numerical errors identical to the solution computed using `lm()`. The R function `solve()` calls the Fortran routine DGESV from the LAPACK library for solution of linear equations using LU decomposition with partial pivoting (Gaussian elimination with row permutations).

Likewise, we can compute the penalized estimator by solving the corresponding linear equation. We will do so where we only add a penalty on the coefficients corresponding to the interaction terms in the nonlinear basis expansion. We will use penalty matrices of the form $\lambda\Omega$ for a fixed matrix Ω so that the amount of penalization is controlled by the parameter $\lambda > 0$.

```

Omega <- diag(c(rep(0, 9), rep(1, 15)))
coef_hat <- cbind(
  solve(XtX + 0.1 * Omega, Xty), # Small penalty
  solve(XtX + Omega, Xty),      # Medium penalty
  solve(XtX + 10 * Omega, Xty)  # Large penalty
)

```

Figure 7.6 shows the resulting penalized model compared to the unpenalized interaction model ($\lambda = 0$) and the additive model ($\lambda = \infty$). A minimal penalization is enough to dampen the most pronounced spurious fluctuation for the fourth trade group considerably.

```

tmp <- cbind(
  coefficients(claims_lm_spline_int),
  coef_hat,
  c(coefficients(claims_lm_spline_add), rep(0, 15))
)
pred <- cbind(claims[, c("sum", "grp")], exp(X %*% tmp))
pred <- tidyr::pivot_longer(pred, cols = !c(grp, sum), names_to = "lambda")
p0 + geom_line(

```

```

aes(y = value, color = lambda),
data = pred,
linewidth = 1) +
theme(legend.position = "top") +
scale_color_discrete(
  expression(~lambda),
  labels = c(0, 0.1, 1, 10, expression(infinity~~(additive~model)))
)

```

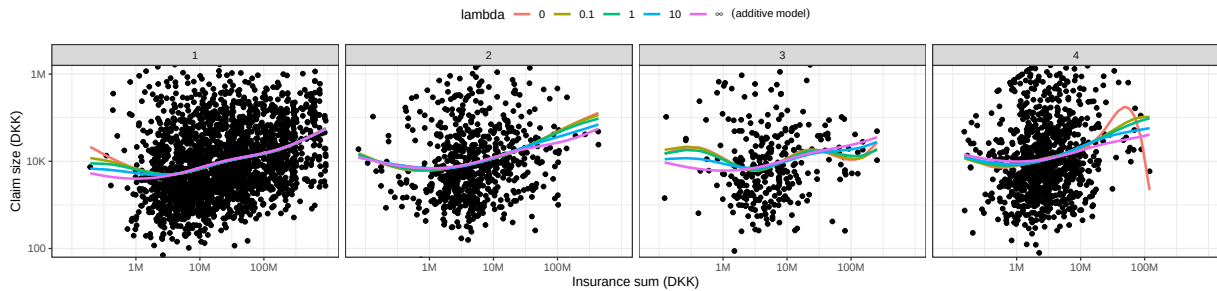


Figure 7.6: Scatter plots and fitted values for the claim size models where the nonlinear interaction terms have been penalized. The unpenalized interaction and additive model fits are also added.

The penalized least squares estimates can also be computed using the QR decomposition via the function `lm.fit()`. To achieve this we need to augment the model matrix and the response data suitably.

```

fit_lamb1 <- lm.fit(rbind(X, sqrt(0.1) * Omega), c(y, rep(0, ncol(Omega))))
fit_lamb2 <- lm.fit(rbind(X, sqrt(10) * Omega), c(y, rep(0, ncol(Omega))))
## Comparisons
range(coef_hat[, 1] - coefficients(fit_lamb1))

```

```
[1] -5.108247e-12  4.279577e-12
```

```
range(coef_hat[, 3] - coefficients(fit_lamb2))
```

```
[1] -1.565414e-13  2.406964e-13
```

The results are again (numerically) identical to using `solve()` above. If several penalized estimates using matrices $\lambda\Omega$ for different choices of λ are to be computed, a third option using a diagonalization of $\mathbf{X}^T\mathbf{X}$ is beneficial. This is implemented in, e.g., the `lm.ridge()` function in the package MASS, which

unfortunately doesn't let you choose your own penalty matrix – it only implements $\Omega = \mathbf{I}$.

In conclusion, we have found evidence in the data for an overall nonlinear relation between insurance sum and claim size plus an additive effect of trade group. For small insurance sums (less than DKK 1M) there is almost no relation, while for larger insurance sums there is roughly a linear relation on a log-log scale. The model fits the data well with a non-normal and slightly right skewed residual distribution, but as a predictive model it is weak, and insurance sum can together with trade group only explain 6-7% of the variation in the (log) claim size distribution.

7.3.2 Nonlinear birth weight models

Basis expansions can be tried if we expect to get something out of it, that is, if we expect that there are some nonlinear relations in the data. On the other hand, if we construct models tailor-made to capture nonlinear relations we have spotted by eye-balling residual plots we run the risk of overfitting random fluctuations. In particular, formal downstream justifications using statistical tests are invalidated. The eye-balling process is a model selection procedure with statistical implications, which are difficult to account for.

We decided to expand `gestationalAge` using natural cubic splines with three knots in 38, 40, and 42 weeks. The boundary knots were determined by the range of the data set, and were thus 25 and 47. We also expanded `age`, but we let the `ns()` function determine the knots automatically for that predictor. The last continuous predictor, `alcohol` has a very skewed marginal distribution, and it was not judged to be suitable for a standard basis expansion. We also present a test of the nonlinear effect.

```
data("birth_weight")
birth_weight <- filter(
  birth_weight,
  weight > 32,
  length > 10 & length < 99,
  gestationalAge > 18
) |>
na.omit() |>
select(~ c(interviewWeek, fetalDeath))
```

```

form <- weight ~ gestationalAge +
                age +
                children +
                coffee +
                alcohol +
                smoking +
                abortions +
                feverEpisodes

birth_weight_lm <- lm(form, data = birth_weight)

binScale <- scale_fill_continuous(
  breaks = c(1, 10, 100, 1000),
  low = "gray80",
  high = "black",
  trans = "log",
  guide = "none"
)

```

Nonlinear main effects model.

```

nsg <- function(x) ns(x, knots = c(38, 40, 42), Boundary.knots = c(25, 47))

form <- weight ~ nsg(gestationalAge) +
                ns(age, df = 3) +
                children +
                coffee +
                alcohol +
                smoking +
                abortions +
                feverEpisodes

birth_weight_lm_spline <- lm(form, data = birth_weight)

anova(birth_weight_lm, birth_weight_lm_spline) |> knitr::kable()

```

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
11139	2537626346	NA	NA	NA	NA
11134	2466413420	5	71212926	64.29455	0

Table 7.4: Test of the model including a spline expansion of `gestationalAge` against the main effects model.

The main effects model is a submodel of the nonlinear main effects model. This is not obvious. The nonlinear expansion will result in 4 columns in the model matrix $\mathbf{X}$, none of which being `gestationalAge`. However, the linear function

is definitely a natural cubic spline, and it is thus in the span of the 4 basis functions. With $\mathbf{X}'$ the model matrix for the main effects model, it follows that there is a C such that Equation 2.12 holds. This justifies the use of the F -test. The conclusion from Table 7.4 is that the nonlinear model is highly significant.

```
birth_weight_diag <- augment(birth_weight_lm_spline, data = birth_weight)
p1 <- ggplot(birth_weight_diag, aes(.fitted, .std.resid)) +
  stat_binhex(bins = 20) + binScale + geom_smooth(linewidth = 1, fill = "blue") +
  xlab("fitted values") + ylab("standardized residuals")
p2 <- ggplot(birth_weight_diag, aes(gestationalAge, .std.resid)) +
  stat_binhex(bins = 20) + binScale + geom_smooth(linewidth = 1, fill = "blue") +
  xlab("gestationalAge") + ylab("")
p3 <- ggplot(birth_weight_diag, aes(sample = .std.resid)) +
  geom_abline(intercept = 0, slope = 1, color = "blue", linewidth = 1) +
  geom_qq() +
  xlab("theoretical quantiles") + ylab("")
gridExtra::grid.arrange(p1, p2, p3, ncol = 3)
```

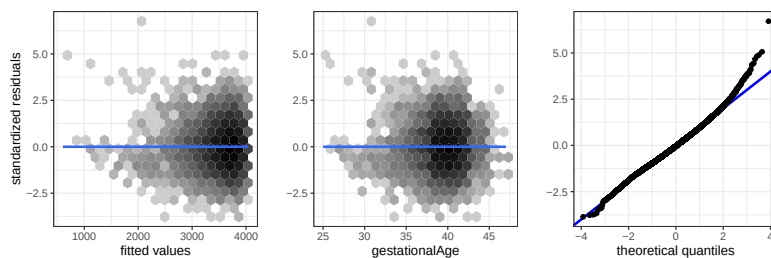


Figure 7.7: Diagnostic plots for the model with `gestationalAge` expanded using splines.

The positioning of the knots for spline expansion is selected based on the marginal distribution of the predictor²². The knots should be placed reasonably relative to the distribution of the predictor, so that we learn about nonlinearities where there is data to learn from. In this case we placed knots at the median and the 10% and 90% quantiles of the distribution of `gestationalAge`. The `ns()` function makes a similar automatic selection of knots based on the marginal distribution of the predictor variables when it is applied in the formula. One just have to specify the number of basis functions using the `df` argument.

An eye-ball decision based on Figure 3.8 would be that a single knot around 41 would have done the job, but we refrained from making such a decision. A subsequent test of the nonlinear effect with 1 degrees of freedom would not appropriately take into account how the placement of the knot was made.

²² Letting the knots be parameters to be estimated is not a statistically viable idea. The resulting estimation problem becomes computationally much more difficult, and it is not straightforward how to adjust subsequent statistical analyses to account for the data adaptive choice of knots.

Figure 7.7 shows diagnostic plots for the nonlinear main effects model. They show that the inclusion of the nonlinear effect removed the previously observed problem with assumption **A1**. The error distribution is still not normal, but right skewed with a fatter right tail than the normal distribution. There is a group of extreme residuals for preterm born children, which should be given more attention than they will be given here.

Reduced nonlinear main effects model.

```
form <- weight ~ nsg(gestationalAge) +
  children +
  coffee +
  smoking

birth_weight_lm_spline_small <- lm(form, data = birth_weight)

anova(birth_weight_lm_spline_small, birth_weight_lm_spline) |>
  knitr::kable()
```

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
11142	2469351552	NA	NA	NA	NA
11134	2466413420	8	2938131	1.657931	0.1032394

Table 7.5: Test of weak predictors in the nonlinear main effects model.

Table 7.5 shows that dropping the four weak predictors from the nonlinear main effects model is again borderline and not really significant.

Figure 7.8 shows examples of the fit for the reduced nonlinear main effects model. The figure illustrates the general nonlinear relation between **weight** and **gestationalAge**. Differences due to other variables are purely additive in this model, which amounts to translations up or down of the curve. The figure shows a couple of extreme cases; the majority group who have had children before and who do not smoke or drink coffee, and a minority group who have had children before and smoke and drink coffee the most. What we should notice is the wider confidence band on the latter (**smoke** = 3, **coffee** = 3) compared to the former, which is explained by the skewness of the predictor distributions. Table 7.6 gives confidence intervals for the remaining parameters based on the nonlinear main effects model.


```

pred_frame <- expand.grid(
  children = factor(1),
  smoking = factor(c(1, 3)),
  coffee = factor(c(1, 3)),
  gestationalAge = seq(25, 47, 0.1),
  alcohol = 0,
  age = median(birth_weight$age),
  feverEpisodes = 0,
  abortions = factor(0)
)

pred <- predict(
  birth_weight_lm_spline,
  newdata = pred_frame,
  interval = "confidence"
)

pred_frame <- cbind(pred_frame, pred)

ggplot(pred_frame, aes(gestationalAge, fit, color = coffee)) +
  geom_line() + ylab("weight") +
  geom_ribbon(
    aes(ymin = lwr, ymax = upr, fill = coffee),
    alpha = 0.3
  ) + facet_grid(. ~ smoking, label = label_both)

```

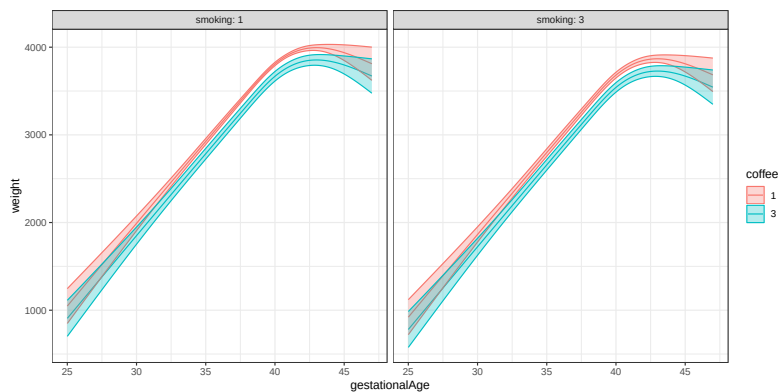


Figure 7.8: Main effects model with basis expansion of `gestationalAge`. Here illustrations of the fitted mean and 95% confidence bands for `children=1`.

term	conf.low	conf.high
children1	155.470	194.087
coffee2	-82.608	-42.431
coffee3	-193.020	-87.630
alcohol	-13.167	6.509
smoking2	-125.758	-75.259
smoking3	-155.765	-97.975

term	conf.low	conf.high

Table 7.6: Confidence intervals.

To conclude the analysis, we will again include interactions. However, the variable `gestationalAge` is in fact only taking the integer values 25 to 47, and the result of a nonlinear effect coupled with a third order interaction, say, results in an almost saturated model. That is, the interaction model has more or less a separate mean for all observed combinations of the predictors. We choose to consider a less complex model with all second order interactions between `gestationalAge` and the three factors that we have judged to be strong predictors.

Nonlinear interaction model.

```
form <- weight ~ (smoking + coffee + children) * nsg(gestationalAge) +
  ns(age, df = 3) +
  alcohol +
  abortions +
  feverEpisodes

birth_weight_lm_spline_int <- lm(form, data = birth_weight)
```

```
pred_frame <- expand.grid(
  children = factor(c(0, 1)),
  smoking = factor(c(1, 2, 3)),
  coffee = factor(c(1, 2, 3)),
  gestationalAge = 25:47,
  alcohol = 0,
  age = median(birth_weight$age),
  feverEpisodes = 0,
  abortions = factor(0)
)

pred <- predict(birth_weight_lm_spline_int, newdata = pred_frame, interval = "confidence")

pred_frame <- cbind(pred_frame, pred)
pred_frame$fit_spline <- predict(birth_weight_lm_spline, newdata = pred_frame)

ggplot(pred_frame, aes(gestationalAge, fit)) +
  geom_point(data = birth_weight, aes(y = weight), size = 0.3, alpha = 0.1) +
  facet_grid(coffee ~ children + smoking, label = label_both) +
  geom_ribbon(aes(ymin = lwr, ymax = upr), fill = gray(0.85)) +
  geom_line(color = "red") + coord_cartesian(ylim = c(0, 5000)) +
  geom_line(aes(y = fit_spline), color = "blue") + ylab("weight") +
  scale_y_continuous(breaks = c(1000, 2000, 3000, 4000))
```

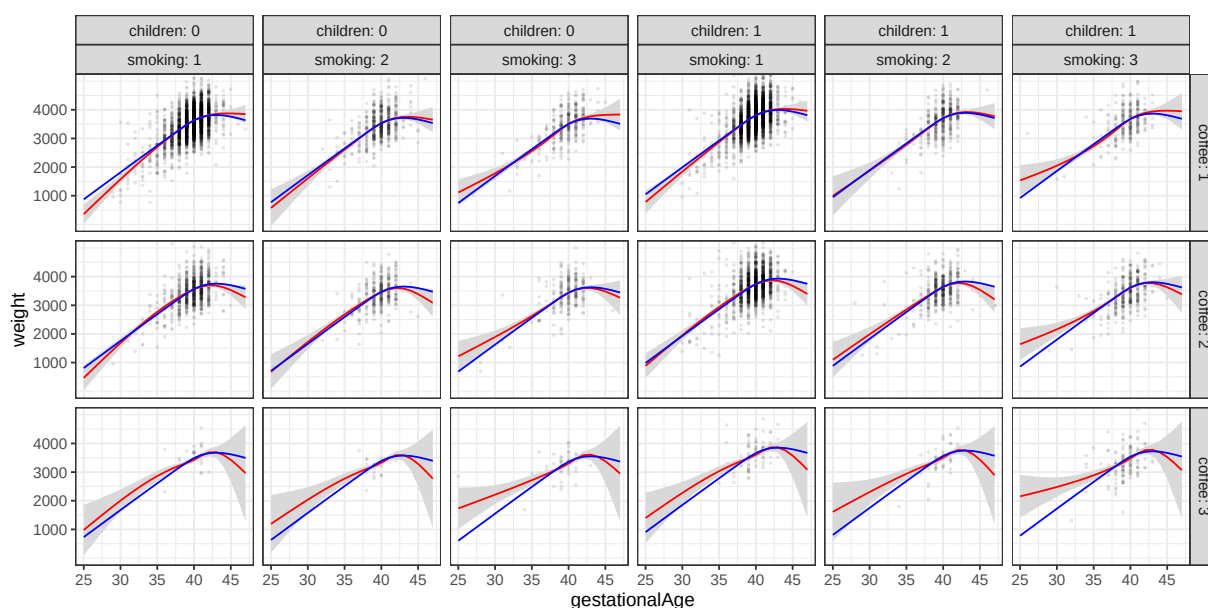


Figure 7.9: Comparison of the interaction model (red, 95% gray confidence bands) with the nonlinear main effects model (blue).

Figure 7.9 shows the predicted values for the reduced nonlinear main effects model for all combinations of **children**, **smoking** and **coffee** (blue curves). These curves are all just translations of each other. In addition, the figure shows the predicted values and a 95% confidence band as estimated with the nonlinear interaction model. We observe minor deviations for the reduced main effects model, which can explain the significance of the test, but the deviations appear unsystematic and mostly related to extreme values of **gestationalAge**. The conclusion is that even though the inclusion of interaction effects is significant, there is little to gain over the reduced nonlinear main effects model.

```
anova(birth_weight_lm_spline, birth_weight_lm_spline_int) |>
  knitr::kable(digits = 10)
```

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
11134	2466413420	NA	NA	NA	NA
11114	2451152208	20	15261213	3.459865	2.612e-07

Table 7.7: Test of nonlinear interaction model.

Table 7.7 shows that inclusion of interaction terms is still significant by a formal statistical test. It is, however, always a good idea to visualize how the nonlinear interaction model differs from the model with only main effects as above instead of just focusing on the formal test. The test may, on the other hand, support the visualization as a quantification of whether the differences seen in a figure are actually statistically significant.

7.4 Practical data analysis

7.5 Causal interpretations

Exercises

Exercise 7.1. This is a continuation of Exercise Exercise 2.1.

```
set.seed(04022016)
factor_labels <- rep(c("A", "B", "C"), each = 50)
formula_data <- tibble(
  x = rnorm(150, 0),
  z = factor(sample(factor_labels)),
  mu = (z == "A") + cos(x) + 0.5 * (z == "B") * sin(x),
  y = rnorm(150, mu, 0.5)
)
```

Expansions can be achieved by applying an R function that returns a matrix of basis function evaluations when applied to a vector of predictor values. We consider a spline basis expansion.

```
library(splines)
model.matrix(y ~ ns(x, df = 4), formula_data) |>
  head()
```

```
(Intercept) ns(x, df = 4)1 ns(x, df = 4)2 ns(x, df = 4)3 ns(x, df = 4)4
1          1      0.75361809   0.169569207   0.08775628   -
0.04862859
2          1      0.07603605  -0.191091633   0.44504011   -
0.25394848
```

3	1	0.77817043	0.005536704	0.13653515	-
0.07790959					
4	1	0.14168625	-0.207697339	0.48371373	-
0.27601639					
5	1	0.15040919	0.515081509	0.26234462	0.07216468
6	1	0.20011297	0.527404800	0.24317864	0.02930359

Before you run the R expression below, answer the following question on the basis of what you have learned.

1. Determine the number of columns for the model matrix below and explain and interpret what the columns mean?

```
model.matrix(y ~ ns(x, df = 4) * z, formula_data) |>
  head()
```

Figure 7.10 was obtained by fitting the linear model using the model matrix, or rather the formula, from above. The figure illustrates the model fit by comparing the fitted model directly to the true means and via the residuals. Fitted values and residuals were computed using the `augment()` function from the broom package.

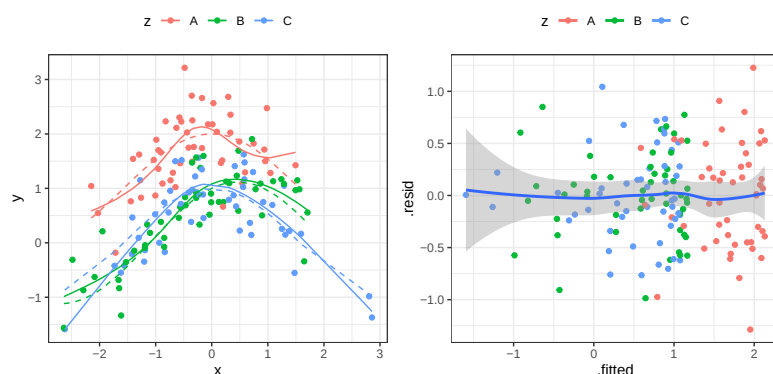


Figure 7.10: Left: The fitted values plotted as lines together with the data. The dashed lines are the true means. Right: a residual plot.

2. Reproduce Figure 7.10 and discuss the model fit.

Exercise 7.2. Consider the following matrices

```
X <- splines::bs(
  seq(0, 1, 0.01),
  knots = c(0.3, 0.5, 0.7)
)
X_prime <- seq(0, 1, 0.01)
```

of B -spline basis functions evaluated on a grid in the interval $[0, 1]$. Use Exercise 2.3 to verify that `X_prime` is in the column space of `X`. You can use the formula from the exercise to compute `C`. Can you also use `lm.fit()`?

8 Model Assessment

Warning

This chapter is still a draft and minor changes are made without notice. Some changes to Section 9.6 are to be expected.

This chapter deals with general statistical methodology for assessment of a regression model's predictive strength. Quantification of predictive strength can be useful in itself for a single model but is most often used for comparing models and for selecting among several models.

It is important to be very clear about the terminology that is used in relation to model assessment. That a model fits the data means throughout that the model assumptions are fulfilled, and *goodness-of-fit-measures* are quantifications of deviations from the model assumptions. In this chapter the quality of a model is quantified in terms of its predictive strength and not in terms of its fit to data. For a model to be a good predictive model it is neither necessary nor sufficient²³ that it fits the data.

²³ Is this surprising?

8.1 Outline and prerequisites

The chapter first introduces a simple example where temperature is used as a predictor for the number of bikes that a bike sharing company is renting out. Then we present the general theoretical framework of loss and risk for quantification of predictive strength. Subsequently a number of prediction error quantities and corresponding estimators are treated in Section 8.4, where we also return to the bike renting example. The different estimators of prediction error are compared in a simulation study in Section 8.5, and the chapter concludes with some specific results for binary responses.

The chapter primarily relies on the following R packages.

```
library(RwR)
library(tibble)
library(ggplot2)
library(splines)
library(dplyr)
```

8.2 Bike rentals

The `bike` dataset is included in the `RwR` package. It is a tibble with data from 50 days (50 rows) on the number of bikes rented from a bike sharing company in a day and the average temperature of the day.

```
# RwR version >= 0.3.2
data("bike", package = "RwR")
```

Count	Temperature
113	20.555556
40	2.222222
70	28.333333
104	15.555556
129	22.777778
79	10.555556

Table 8.1: First six rows in the `bike` dataset.

Figure 8.1 shows a scatter plot of the data. We will build a sequence of predictive models of `Count` using `Temperature` as the predictor. That is, the response variable Y is the number of bikes rented and X is the temperature. All the models are of the form

$$m(x) = \mathbf{E}(Y \mid X = x) = \beta_0 + \sum_{d=1}^{\text{df}} \beta_d h_d(x)$$

where the h_d -s are spline basis functions. The number `df` specifies the number of basis functions and thus the complexity of the model. The dimension of the corresponding linear model will be `df + 1`, since the model also includes an intercept. The code below fits six models for `df` ranging from 0 to 16.

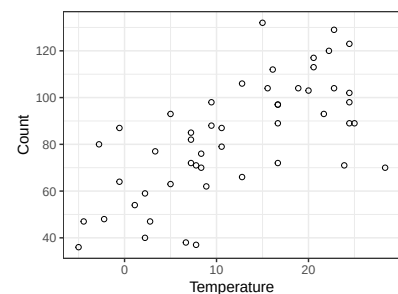


Figure 8.1: Scatter plot of number of rented bikes against temperature.


```

bike_lm <- list()
df <- c(0, 1, 2, 4, 8, 16)
bike_lm[[1]] <- lm(Count ~ 1, data = bike)
for (d in seq_along(df)[-1]) {
  bike_lm[[d]] <- lm(Count ~ ns(Temperature, df = df[d]), data = bike)
}

```

Figure 8.2 shows predictions from the six fitted models. We see that as we increase the complexity of the model, the predictions show more and more fluctuations.

```

p0 <- ggplot(bike, aes(Temperature, Count)) + geom_point(pch = 1)
plot_pred <- function(pred_data) {
  p0 + geom_line(data = pred_data, aes(y = .predict), linewidth = 1)
}
pred_data <- tibble(
  Temperature = seq(min(bike$Temperature), max(bike$Temperature), length = 200)
)
p <- list()
for (d in seq_along(df)) {
  p[[d]] <- mutate(
    pred_data,
    .predict = predict(bike_lm[[d]], newdata = pred_data)
  ) |> plot_pred() + ggtitle(paste("df =", df[d]))
}
gridExtra::grid.arrange(p[[1]], p[[2]], p[[3]], p[[4]], p[[5]], p[[6]], ncol = 3)

```

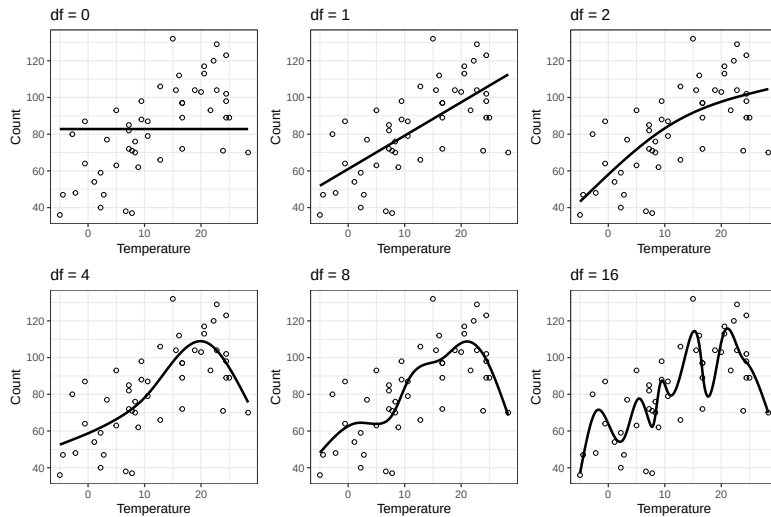


Figure 8.2: Scatter plot of data with model predictions from the six models of bike rentals.

df	MSE	adj r-sq
0	602.1600	0.0000000
1	333.4333	0.4347352

df	MSE	adj r-sq
2	320.3021	0.4454431
4	262.9248	0.5245518
8	251.1411	0.5015540
16	195.7884	0.5172112

Table 8.2: Mean squared error on data used to fit the models for the six models of bike rentals.

Table 8.2 shows the adjusted R^2 together with the mean squared error on the data used to fit the models, which is computed as

$$\frac{1}{n} \sum_{i=1}^n (Y_i - \hat{m}(X_i))^2, \quad (8.1)$$

where $\hat{m} = \hat{\beta}_0 + \sum_{d=1}^{\text{df}} \hat{\beta}_d h_d$. We note that the MSE is monotonely decreasing as a function of the complexity of the model, that is, as a function of df. The adjusted R^2 increases until $\text{df} = 4$, and then it appears to settle just above 0.5 – suggesting that the models can explain around 50% of the variation in the data.

Figure 8.3 shows the residual plots for the six models. We see that the three models with $\text{df} \leq 2$ are misspecified. This is particularly pronounced for the intercept-only model ($\text{df} = 0$). From the residual plots alone, the remaining three models appear to fit the data well.

```
p <- list()
for (d in seq_along(df)) {
  p[[d]] <- broom::augment(bike_lm[[d]], data = bike) |>
    ggplot(aes(Temperature, .resid)) +
    geom_point(pch = 1) + geom_smooth() +
    xlab("") + ylab("") + ggtitle(paste("df =", df[d])) +
    ylim(-50, 50)
}
gridExtra::grid.arrange(p[[1]], p[[2]], p[[3]], p[[4]], p[[5]], p[[6]], ncol = 3)
```

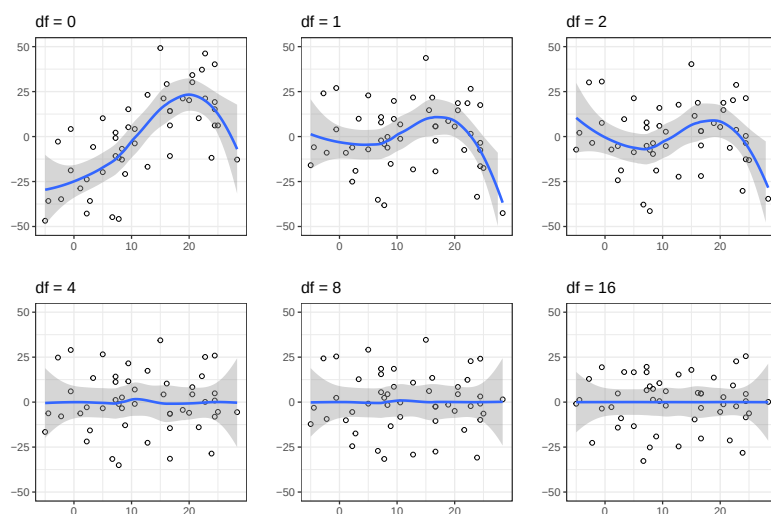


Figure 8.3: Residual plots for the six models of bike rentals.

Based on the MSE in Table 8.2 the most complex model is best. However, the adjusted R^2 suggests that all the three models with $df = 4$, $df = 8$ and $df = 16$ are about equally good, and the residual plots suggest that these three models all fit the data. If we consider the predictions made by the most complex model ($df = 16$) in Figure 8.2, we are probably a little wary about the oscillations of the estimated mean value function. These seem arbitrary and unlikely to generalize to future observations. And by referring to some form of *principle of parsimony* we can argue that the model with $df = 4$ is “best”. It is as complex as it needs to be but not more complex than that. In this chapter we will develop a precise framework for assessing and comparing models, which clarifies what we mean by a model’s predictive performance. We can regard this as a substantiation and clarification of the principle of parsimony that we will adhere to.

8.3 Quantifying model predictions

How do we then quantify the predictive strength of a model? We may already suspect that the empirical MSE, as computed in the bike rental example, is not a good measure. Now MSE is a sensible theoretical quantification, but the empirical MSE given by (8.1) is an example of a *training error*, as defined in Section 8.4, which is generally a bad estimate of MSE. Alternatives include the adjusted R^2 -statistic, as introduced in Section 2.7, and its equivalents for generalized linear models

expressed in terms of either the χ^2 - or deviance statistic as introduced in Section 5.4. As discussed there, these quantities are measures of the descriptive or predictive strength of a model, but they may appear *ad hoc*, and we have not dealt with their theoretical properties.

This section deals more systematically with the question of assessing models in terms of their predictive performance by defining various quantifications of a model's predictive strength. In the subsequent sections we treat methods for correct estimation of these quantities. This section is largely theoretical, and its purpose is to make the conceptual framework for quantifying predictive strength clear. We will, in particular, clarify how the likelihood (or deviance) for exponential dispersion distributions is related to measures of predictive strength, and how the deviance can be seen as a natural generalization of the squared error.

8.3.1 Loss and risk

The predictive strength of a model for predicting Y is a quantification of the concordance between Y and the prediction of the model. Before introducing a general approach for making such a quantification a motivating example is considered.

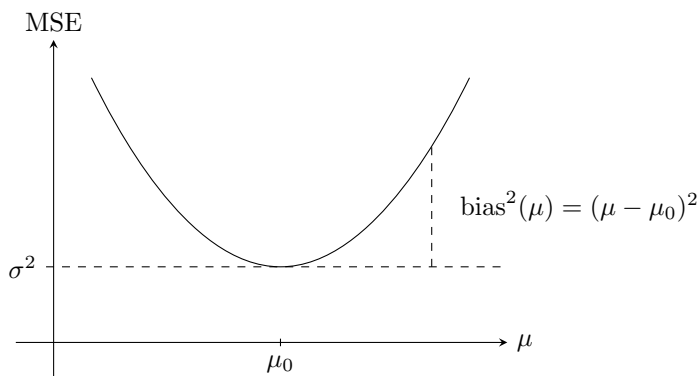


Figure 8.4: The mean squared error is a quadratic function of μ , which can be decomposed as a sum of the variance and the squared bias.

Example 8.1. Assume that Y is a real valued random variable and that we use the point prediction²⁴ $\mu \in \mathbb{R}$ of Y . Just as the squared error can be used for fitting the model to data it can be used for quantifying predictive strength. Thus

$$\mu \mapsto (Y - \mu)^2$$

²⁴ A point prediction is a single element in the sample space for Y . Alternatives include interval predictions – and probabilistic predictions as introduced below.

is a quantification of how good a prediction μ is of Y . Its expectation is the mean squared error

$$\text{MSE}(\mu) = \mathbf{E}(Y - \mu)^2.$$

If $\mu_0 = \mathbf{E}(Y)$ denotes the mean of Y and $\sigma^2 = \mathbf{V}(Y)$ its variance then

$$\text{MSE}(\mu) = (\mu_0 - \mu)^2 + \mathbf{E}(Y - \mu_0)^2 = \text{bias}^2(\mu) + \sigma^2. \quad (8.2)$$

It is clear from (8.2) that the mean squared error is bounded below by σ^2 – that does not depend upon μ – and that the optimal predictor from a mean squared error perspective minimizes the squared bias term and is equal to μ_0 .

The size of the squared bias can best be interpreted relative to the variance σ^2 , and the *unitless* quantity

$$\frac{\text{bias}^2(\mu)}{\sigma^2}$$

is therefore a useful quantification of the predictive strength of μ .

A general method for quantifying predictive strength of a probability model is introduced below by replacing the squared error with an arbitrary loss function. The mean squared error is then replaced by a general quantity called the risk. The convention is that a loss function quantifies how *discordant* predictions and observations are, whereas a reward function quantifies concordance. The negative of a reward function can always be used as a loss function and vice versa, so we will only treat loss functions. The focus in this chapter is mostly on the negative log-likelihood loss, for which the risk is also known as (twice) the cross-entropy, and where it has a decomposition similar to the bias-variance decomposition above.

A point prediction as in Example 8.1 may not in all cases be the most relevant kind of prediction. If Y is a binary variable, say, it is more informative to make a prediction of the probability that $Y = 1$ than just predicting that Y will be 0 or 1. This is particularly true if $\mathbf{P}(Y = 1)$ is small, in which case the only sensible point prediction of Y is 0. A *probabilistic prediction* of Y is simply a probability measure P , and to quantify how well P predicts Y we need to quantify

In a regression context the prediction μ would depend on X , and the mean μ_0 and the variance σ^2 would be conditional on X .

the discordance between Y and P – or perhaps rather the discordance between the distribution of Y and P .

For the quantification of the discordance we choose a (sensible) *loss function*

$$(y, P) \mapsto \text{Loss}(y, P) \in \mathbb{R}.$$

Then $\text{Loss}(Y, P)$ quantifies how well P predicts the random variable Y . It is common to also call a loss function used for probabilistic predictions a *scoring rule*.

Letting P_0 denote the distribution of Y , the *risk*²⁵ of P is the expectation of the loss,

$$\text{Risk}(P) = E(\text{Loss}(Y, P)) = \int \text{Loss}(y, P) P_0(dy),$$

provided that the expectation exists. The larger the value of the loss is, the greater is the discordance, but there is no convention on the scale. Low risk is better than high risk, but the risk of a model can only be interpreted relative to the risk of other models.

Definition 8.1. If $P = h \cdot \nu$ has density h w.r.t. ν the negative log-likelihood²⁶ loss is

$$\text{Loss}(y, P) = -2 \log h(y). \quad (8.3)$$

There is an arbitrariness in the choice of ν that makes the absolute value of the negative log-likelihood impossible to interpret, but it is suitable for comparisons between two or more probability distributions that all have densities w.r.t. the same measure ν . This is particularly clear if we compare P to the distribution of Y – if this distribution has density w.r.t. ν as well. To this end we introduce the *entropy* of the distribution of $Y \sim P_0 = h_0 \cdot \nu$ as

$$H(P_0) = -\mathbf{E}(\log h_0(Y)) = - \int \log h_0(y) P_0(dy),$$

provided that $\log h_0$ is P_0 -integrable.

Lemma 8.1. Assume that $Y \sim P_0 = h_0 \cdot \nu$ with entropy $H(P_0)$. Then the risk for the negative log-likelihood loss fulfills

$$\text{Risk}(P) \geq 2H(P_0) \quad (8.4)$$

for any probability measure $P = h \cdot \nu$ with $\log h$ being P_0 -integrable. Moreover, the inequality in (8.4) is sharp unless $P = P_0$.

²⁵ The loss and risk terminology relates to economic decision theory, but there is no direct monetary interpretation of these quantities in this statistical context. For probabilistic predictions the risk is also known as the *expected score*.

²⁶ The convention is that $\log(0) = -\infty$.

The factor “2” in (8.3) is an arbitrary convention, but it aligns with the scaling of the deviance and results in an unscaled squared error loss for the standard normal distribution.

The entropy introduced here is related to Shannon’s entropy and information theory rather than thermodynamics.

Proof. First note that $-\mathbf{E}(\log h(Y)) = -\int \log h(y)P_0(dy)$. Introducing the set $A = \{y \mid h_0(y) > 0\}$ with $P_0(A) = 1$ and the function

$$r(y) = \begin{cases} \frac{h(y)}{h_0(y)} & \text{for } y \in A \\ 1 & \text{for } y \notin A \end{cases}$$

we find that

$$\begin{aligned} \frac{1}{2}\text{Risk}(P) - H(P_0) &= -\int \log h(y)P_0(dy) + \int \log h_0(y)P_0(dy) \\ &= \int_A -(\log h(y) - \log h_0(y))P_0(dy) \\ &= \int_A -\log r(y)P_0(dy) \\ &= \int -\log r(y)P_0(dy). \end{aligned}$$

Since the negative logarithm on $[0, \infty)$ is convex it follows from Jensen's inequality that

$$\begin{aligned} \frac{1}{2}\text{Risk}(P) - H(P_0) &\geq -\log \int r(y)P_0(dy) \\ &= -\log \int_A r(y)P_0(dy) \\ &= -\log \int_A h(y)\nu(dy) \\ &= -\log P(A) \geq 0 \end{aligned}$$

since $P(A) \leq 1$. The negative logarithm is, in fact, strictly convex on $(0, \infty)$, so Jensen's inequality implies that there is equality only if $r(y) = 1$ for P_0 -almost all y , in which case $P = P_0$. $\square$

Lemma 8.1 provides some insight on the use of the negative log-likelihood as a loss function. The distribution P_0 should by any reasonable standards be in greatest concordance with Y (have lowest risk) – and this is indeed the case as inequality (8.4) shows. Moreover, all other probability measures have strictly larger risk.

Definition 8.2. The Kullback-Leibler divergence of $P = h \cdot \nu$ from $P_0 = h_0 \cdot \nu$ is defined as

$$K(P_0 \| P) = \int \log \frac{h_0(y)}{h(y)} P_0(dy) \quad (8.5)$$

The negative log-likelihood loss is a so-called *strictly proper scoring rule*. The cross-entropy of P_0 and P is

$$H(P_0, P) = -\int \log h(y)P_0(dy),$$

and

$$K(P_0 \| P) = H(P_0, P) - H(P_0).$$

The risk for the negative log-likelihood loss is equal to twice the cross-entropy.

provided that the risk and the entropy are finite.

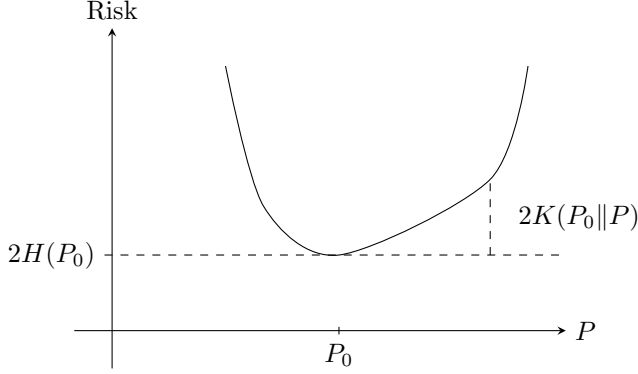


Figure 8.5: Schematic illustration of the risk for the negative log-likelihood loss as a function of the distribution P . The risk is a convex function of P , which can be decomposed as a sum of twice the entropy and twice the Kullback-Leibler divergence.

From the definition of the Kullback-Leibler divergence we have that for the negative log-likelihood loss,

$$\text{Risk}(P) = 2H(P_0, P) = 2K(P_0 \| P) + 2H(P_0). \quad (8.6)$$

This decomposition has a similar interpretation as the bias-variance decomposition for the mean squared error. The entropy term corresponds to the variance term while the Kullback-Leibler term corresponds to the squared bias term. Minimizing the risk is equivalent to minimizing the Kullback-Leibler divergence.

As a direct consequence of Lemma 8.1 we see that the Kullback-Leibler divergence takes values in $[0, \infty)$ and equals 0 if and only if $P = P_0$. It also follows directly from its integral representation above that its value does not depend upon the reference measure ν . The arbitrariness in the risk as well as the entropy cancels out in the Kullback-Leibler divergence. The Kullback-Leibler divergence can be interpreted as a distance from P_0 to P , but it is not a metric since it is neither symmetric nor does it fulfill the triangle inequality.

Example 8.2. If $P = \mathcal{N}(\mu, \sigma^2)$ then P has a density w.r.t. Lebesgue measure, which is proportional to $e^{-(y-\mu)^2/(2\sigma^2)}$, hence

$$-2 \log h_\mu(y) = \frac{1}{\sigma^2} (y - \mu)^2.$$

If $Y \sim \mathcal{N}(\mu_0, \sigma^2)$ we find that

$$\text{Risk}(P) = \frac{1}{\sigma^2} \mathbf{E}(Y - \mu)^2 = \frac{\text{MSE}(\mu)}{\sigma^2}.$$

The reference measure is taken to be

$$\frac{1}{\sqrt{2\pi}\sigma} \cdot m,$$

and the example does therefore not cover the comparison of two normal distributions with different variances.

The entropy²⁷ is seen to be $H(P_0) = 1/2$. The Kullback-Leibler divergence of P from P_0 is therefore

$$K(P_0\|P) = \frac{\text{MSE}(\mu)}{2\sigma^2} - \frac{1}{2} = \frac{\text{bias}^2(\mu)}{2\sigma^2},$$

which is precisely the squared bias relative to the variance as discussed in Example 8.1.

The example above shows the relation between the squared error loss and the negative log-likelihood loss for the normal distribution. For a semiparametric model of the mean – where a distributional assumption is not made – the squared error loss is still a sensible loss under a constant variance assumption.

Example 8.3. If $P_\theta = \mathcal{E}(\theta, \nu_\varphi)$ is an exponential dispersion distribution then by (4.2)

$$-2 \log h_\theta(y) = 2 \frac{\kappa(\theta) - \theta y}{\varphi},$$

so if $Y \sim \mathcal{E}(\theta_0, \nu_\varphi)$ the risk is given as

$$\text{Risk}(P) = 2 \frac{\kappa(\theta) - \theta \mu(\theta_0)}{\varphi},$$

where μ is the mean value function.

The entropy is similarly

$$H(P_0) = \frac{\kappa(\theta_0) - \theta_0 \mu(\theta_0)}{\varphi},$$

and the Kullback-Leibler divergence is

$$\begin{aligned} K(P_0\|P_\theta) &= \frac{(\mu(\theta_0)(\theta_0 - \theta) - \kappa(\theta_0) + \kappa(\theta))}{\varphi} \\ &= \frac{d(\mu(\theta_0), \mu(\theta))}{2\varphi} \end{aligned}$$

with d the unit deviance.

Note that in the example above the deviance $d(y, \mu(\theta))$ is equal to $-2 \log h_\theta(y)$ up to an additive constant that does not depend upon θ . We can therefore just as well replace the negative log-likelihood loss by the deviance in the computation

²⁷ The entropy of the normal distribution is typically reported to be

$$\log(\sigma\sqrt{2\pi e}) = \log(\sigma\sqrt{2\pi}) + \frac{1}{2},$$

which is the *differential entropy* for which the reference measure is Lebesgue measure.

The reference measure is taken as ν_φ , and this example does therefore not cover the comparison of distributions with different dispersion parameters.

of the risk. For comparison purposes, the constant will cancel out.

For any given unit deviance d the deviance loss, $d(Y, \mu)$, can be computed for any probability distribution P on $\mathbb{R}$, provided it has finite mean

$$\mu = \int yP(dy).$$

Since the deviance loss depends only on the mean, it is useful for semiparametric models just as the squared error loss is, and it may be more appropriate than the squared error loss if the variance depends on the mean.

If $\mathcal{V}$ denotes a variance function, an alternative semiparametric loss function is the Pearson loss

$$\frac{(Y - \mu)^2}{\mathcal{V}(\mu)}.$$

For a mean-variance relation that does not correspond to an exponential dispersion model, the Pearson loss can always be used.

Example 8.4. From Example 4.10 it follows that the deviance for the Bernoulli distribution is

$$d(y, p) = 2y \log \left(\frac{y}{p} \right) + 2(1 - y) \log \left(\frac{1 - y}{1 - p} \right)$$

for $y \in [0, 1]$ and $p \in (0, 1)$.

If $\mathbf{P}(Y = 1) = p_0$ we get that

$$\text{Risk}(p) = -2p_0 \log p - 2(1 - p_0) \log(1 - p).$$

The entropy is

$$H(p_0) = p_0 \log p_0 + (1 - p_0) \log(1 - p_0),$$

and the Kullback-Leibler divergence is given by

$$\begin{aligned} 2K(p_0 \| p) &= \text{Risk}(p) - 2H(p_0) \\ &= 2p_0 \log \left(\frac{p_0}{p} \right) + 2(1 - p_0) \log \left(\frac{1 - p_0}{1 - p} \right) \\ &= d(p_0, p). \end{aligned}$$

Though a Bernoulli variable will only take the values 0 or 1, the deviance is defined for y in the closed interval $[0, 1]$.

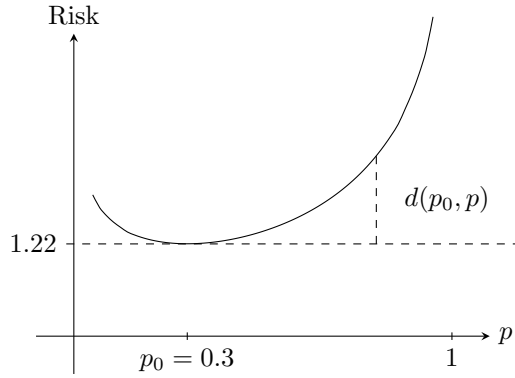


Figure 8.6: Risk for the Bernoulli deviance is a convex function of the probability parameter p . The true probability is $p_0 = 0.3$ with entropy $1.22/2 = 0.61$. The deviance $d(p_0, p)$ equals twice the Kullback-Leibler divergence of p from p_0 .

8.3.2 Regression risk

For a regression model of Y conditionally on the predictor variable X there are two different ways of assessing the predictive strength of the model. Either conditionally on $X = x$ or integrated over the distribution of X . With $P(x)$ denoting the probability model of Y given $X = x$ it is thus possible to consider the conditional risk

$$\text{Risk}(P(x)) = E(\text{Loss}(Y, P(x)) \mid X = x),$$

as well as the unconditional risk

$$\text{Risk}(P) = E(\text{Loss}(Y, P(X))) = E(\text{Risk}(P(X))),$$

where the tower property of conditional expectations is used to get the last identity. The unconditional risk is also called the *generalization error* of the model P . We denote it $\text{Risk}(P)$, but remember that P is now a map, $x \mapsto P(x)$, that gives a model of the conditional distribution of Y given $X = x$.

With m predictors $X_1, \dots, X_m$ and the conditional risk for the i -th response denoted $\text{Risk}_i(P(X_i))$, we can summarize the conditional risks as the average risk

$$\text{Risk}_{\text{avg}}(P) = \frac{1}{m} \sum_{i=1}^m \text{Risk}_i(P(X_i)).$$

When the empirical distribution of the observed predictors resembles the marginal distribution²⁸ of X , the average risk is a sensible estimate of $\text{Risk}(P)$. However, if the empirical distribution of the X_i -s does not match the distribution of X , $\text{Risk}_{\text{avg}}(P)$ can be a biased estimate of the unconditional risk. This is particularly problematic if the X_i -s are selected

The convention is that an expectation is always over all random variables inside the parentheses. That is, it is an integration out over the joint distribution of all variables. Conditional expectations are used when one or more variables are held fixed.

²⁸ Which is the case if $X_1, \dots, X_m$ are i.i.d. observations with the same distribution as X by the law of large numbers.

to overrepresent situations where the model gives very good predictions – whether or not the selection was deliberate to give a favorable validation or just due to ignorance.

8.4 Error estimates

Practical model assessment requires that we can compute the generalization error $\text{Risk}(\hat{P})$ for the model, $\hat{P}$, that has been fitted to data. Since P_0 is unknown the theoretical formulas in the preceding section are of little value. The generalization error has to be estimated somehow. This section deals with the estimation of the generalization error.

Risk estimation is not straightforward because the data is already used once for fitting the model. If $\hat{P}$ denotes the fitted model based on the data $(X_1, Y_1), \dots, (X_n, Y_n)$, its average loss on the dataset is

$$\text{err} = \frac{1}{n} \sum_{i=1}^n \text{Loss}(Y_i, \hat{P}(X_i)). \quad (8.7)$$

This number is also known as the *training error*. It is **not** a good estimate of neither the average risk nor the generalization error of $\hat{P}$ because (Y_i, X_i) and $\hat{P}$ are dependent. It will typically have a downward bias, that is, it underestimates the generalization error and is thus too optimistic about the predictive strength of the model. As a rule of thumb, the more complex and flexible the model is, the more will err underestimate the generalization error of $\hat{P}$.

It is possible that a complex model with a small training error is both a better fit to data and a better predictive model than a simple model. But it is also possible that the complex model *overfits* the data and is worse as a predictive model. For the linear model used in the bike rent example, the inclusion of a more flexible nonlinear expansion of a predictor variable appears to improve the model fit – up to some point – but eventually the model appears to pick up specific fluctuations in the data.

An overfitted model captures properties of the data that are specific to the given dataset, and which do not generalize to other data. An overfitted model may appear to fit the data very well, but it will have less predictive strength than

For the squared error loss

$$\text{err} = \frac{\text{RSS}}{n},$$

for the deviance loss

$$\text{err} = \frac{D}{n},$$

where D is the total deviance and for the Pearson loss

$$\text{err} = \frac{\chi^2}{n},$$

where χ^2 is the χ^2 -statistic.

a model that is not overfitted. For a given dataset there is always a limit to how complex a model we can fit without overfitting. The main challenge when assessing a model is to estimate its predictive strength correctly so that overfitted models are adjusted for their optimism.

8.4.1 Error estimation via validation data

To obtain an unbiased estimate of the risk of $\hat{P}$ an independent *validation dataset* is required. That is, in addition to the data used for fitting $\hat{P}$ we need an independent dataset

$$(X_1^{\text{new}}, Y_1^{\text{new}}), \dots, (X_m^{\text{new}}, Y_m^{\text{new}})$$

with the Y_i^{new} -s conditionally independent given the predictors and with Y_i^{new} having conditional distribution $P_0(X_i^{\text{new}})$. In this case we define the estimator

$$\widehat{\text{Risk}} = \frac{1}{m} \sum_{i=1}^m \text{Loss}(Y_i^{\text{new}}, \hat{P}(X_i^{\text{new}})).$$

It follows by the independence between $\hat{P}$ and the validation data that $\widehat{\text{Risk}}$ is an unbiased estimate of the average risk, in the sense that

$$\begin{aligned} \mathbf{E}(\widehat{\text{Risk}} \mid \hat{P}, \mathbf{X}^{\text{new}}) &= \frac{1}{m} \sum_{i=1}^m \mathbf{E}(\text{Loss}(Y_i^{\text{new}}, \hat{P}(X_i^{\text{new}})) \mid \hat{P}, X_i^{\text{new}}) \\ &= \frac{1}{m} \sum_{i=1}^m \text{Risk}_i(\hat{P}(X_i^{\text{new}})) \\ &= \text{Risk}_{\text{avg}}(\hat{P}). \end{aligned}$$

Whether $\widehat{\text{Risk}}$ can be regarded as an unbiased estimate of the generalization error depends – as in the discussion above – on whether the empirical distribution of $X_1^{\text{new}}, \dots, X_m^{\text{new}}$ is representative of the distribution of future X -s.

Error estimation using validation data is fairly straightforward. It is, however, important not to make the mistake of peeping at the validation data while estimating P . We should, for example, avoid preprocessing steps, such as data standardization, that use the validation data. While the inclusion of the validation data for standardization is a small blunder, it is a major mistake to include the validation data in a preprocessing variable selection step, say.

In practice, the validation dataset is obtained by setting aside a randomly selected subset containing 10-25% of the data available. The remaining 75-90% of the data is used for model fitting. If the dataset is small, setting aside a part of the data to be used for model assessment only is not a viable option. And we can in all cases argue that we could fit a better model if we used all data for the model fitting. The following section treats data splitting techniques that allow us to use all data for model fitting while still giving good estimates of predictive strength – with some extra computational costs.

8.4.2 Error estimation via data splitting

Data splitting techniques comprise a collection of procedures that mimic setting aside a validation dataset while not actually doing so. Hence the full dataset is used for fitting the model while the model assessment is based on the data splitting procedure. The data splitting procedures discussed in this section are all based on Assumption A6, that is, we assume that the observations $(X_1, Y_1), \dots, (X_n, Y_n)$ are i.i.d.

Expected generalization error

A slightly tricky aspect of using data splitting techniques is that we will not directly estimate the generalization error of the model $\hat{P}$ that is fitted to the entire dataset. We will instead estimate the *expected* generalization error.

Definition 8.3. With $(Y^{\text{new}}, X^{\text{new}})$ independent of $\hat{P}$ the expected generalization error is defined as

$$\text{Err} = E(\text{Loss}(Y^{\text{new}}, \hat{P}(X^{\text{new}}))).$$

It follows from the definition of the generalization error and the tower property of conditional expectations that

$$\text{Err} = E\left(E(\text{Loss}(Y^{\text{new}}, \hat{P}(X^{\text{new}})) \mid \hat{P})\right) = E(\text{Risk}(\hat{P})).$$

The hierarchy of error measures and estimates can be quite mind-boggling, but think about Err as an assessment of the *estimation method* used for computing $\hat{P}$ rather than an assessment of $\hat{P}$ *per se*.

What we call “validation data” is also known as “test data”. Sometimes, data is split in three parts with “training data” used to fit models, “test data” used to select among models and “validation data” used to assess the selected model.

Observe that $\hat{P}$ is random as a function of the data, thus by convention the expectation defining the generalization error is an expectation over the data as well.

Example 8.5 (Squared error loss). For the squared error loss and any estimator $\hat{m}$ of the regression function, we first condition on $X^{\text{new}} = x$. Then with

$$\begin{aligned} m(x) &:= \mathbf{E}(\hat{m}(x)) \\ m_0(x) &:= \mathbf{E}(Y^{\text{new}} \mid X^{\text{new}} = x) \\ \sigma^2(x) &:= \mathbf{V}(Y^{\text{new}} \mid X^{\text{new}} = x) \end{aligned}$$

we have

$$\text{Risk}(\hat{m}(x)) = \text{MSE}(\hat{m}(x)) = (m_0(x) - \hat{m}(x))^2 + \sigma^2(x),$$

and we get the decomposition

$$\begin{aligned} \mathbf{E}(\text{MSE}(\hat{m}(x))) &= \mathbf{E}((m_0(x) - \hat{m}(x))^2) + \sigma^2(x) \\ &= (m_0(x) - m(x))^2 + \mathbf{E}((m(x) - \hat{m}(x))^2) \\ &\quad + 2(\mu_0(x) - m(x))\mathbf{E}(m(x) - \hat{m}(x)) + \sigma^2(x) \\ &= (m_0(x) - m(x))^2 + \mathbf{V}(\hat{m}(x)) + \sigma^2(x) \\ &= \text{bias}(x)^2 + \text{variance}(x) + \sigma^2(x). \end{aligned}$$

where we have used that $\mathbf{E}(m(x) - \hat{m}(x)) = 0$ for the third identity. From this it follows that

$$\text{Err} = \text{bias}^2 + \text{variance} + \sigma^2 \quad (8.8)$$

where

$$\begin{aligned} \text{bias}^2 &= \mathbf{E}(\text{bias}^2(X)) \\ \text{variance} &= \mathbf{E}(\text{variance}(X)) \\ \sigma^2 &= \mathbf{E}(\sigma^2(X)). \end{aligned}$$

Thus the expected generalization error decomposes into a sum of three non-negative terms. The term σ^2 is called the *irreducible error*, sometimes also known as the *Bayes risk*. It does not depend on the model or the estimator and lower bounds the expected generalization error for all models and estimators. The two other terms are the expected variance and the expected squared bias of the estimator – the expectation being over the marginal distribution of the predictor X .

The decomposition given by Equation 8.8 is conceptually useful. It provides some insights about what the expected generalization error measures, and it tells us that to minimize the expected generalization error we need to minimize both

the squared bias and the variance of the model/estimator. We generally expect that the more complex a model is, the smaller is the bias, but the larger is the variance of the estimator. Thus there is a *bias-variance* tradeoff. In practice it can be difficult to directly estimate the two terms bias² and variance, and the decomposition is therefore less useful for estimating the expected generalization error.

Data splitting procedures

To estimate Err we turn to *data splitting*. Besides leading to estimators of the expected generalization error, these procedures also work for any choice of loss function – not just the squared error loss, say.

The three main data splitting procedures we describe below work by the same general principle. They consist of B rounds of data splitting, and in each round one subset of the dataset is used for fitting the model and the other works as an independent validation dataset. In round b for $b = 1, \dots, B$ the index set $\{1, \dots, n\}$ is divided into two disjoint subsets $I(b)$ and $J(b)$ followed by two steps:

1. The model $\hat{P}_b$ is fitted based on data with indices in $I(b)$.
2. The losses $L(b, i) = \text{Loss}(Y_i, \hat{P}_b(X_i))$ are computed for $i \in J(b)$.

After the B rounds of model fitting and loss computation the average loss is computed as

$$\widehat{\text{Err}} = \frac{1}{N} \sum_{b=1}^B \sum_{i \in J(b)} L(b, i) \quad (8.9)$$

with $N = \sum_{b=1}^B |J(b)|$. It is essential that the model $\hat{P}_b$ is fitted to the data given by the index set $I(b)$ in *exactly* the same way as $\hat{P}$ is fitted to the full dataset. This includes any kind of variable selection, parameter tuning or the like that might be carried out as part of the model fitting. Otherwise $\widehat{\text{Err}}$ could be overly optimistic.

A complete specification of a data splitting procedure requires that the division of the index set is specified. The following three are the most used:

Subsampling: Subset I of size $m < n$ is sampled *without replacement* from $\{1, \dots, n\}$ and J is the complement of I .

Cross-validation (K -fold): The index set $\{1, \dots, n\}$ is randomly divided into K disjoint sets $H_1, \dots, H_K$ of roughly equal size $\approx n/K$. This defines K rounds of data splitting with $I(k) = \{1, \dots, n\} \setminus H_k$ and $J(k) = H_k$ for $k = 1, \dots, K$.

Bootstrapping: Subset I of size n (replicates counted with multiplicities) is sampled *with replacement* from $\{1, \dots, n\}$ and J is the complement of I .

Remember that the estimate $\hat{P}$ is computed using all n observations in the dataset, but that the data splitting procedures result in an average of losses computed for different estimates based on different subsets of the full dataset. We can therefore not regard $\widehat{\text{Err}}$ as an estimate of the generalization error²⁹ $\text{Risk}(\hat{P})$. It is, however, fairly clear that the data splitting procedures are empirically mimicking the definition of the *expected* generalization error Err : the model is fitted on one part of the data and the loss is computed on an independent part, and the losses are the averaged. Intuitively, $\widehat{\text{Err}}$ is therefore an estimate of Err .

Since Err is a quantification of the estimation procedure rather than the specific estimate $\hat{P}$, it is essential that the *entire* model fitting process is properly accounted for. To avoid overly optimistic error estimates, the estimation procedure should preferably be completely specified prior to seeing data, and $\hat{P}$ should then be computed according to that procedure. The estimates computed in the data splitting process can then be computed in exactly the same way. This can be difficult to do in practice, since elements of human judgement, such as studying residual plots, are often part of the model fitting process in real world applications.

An important step towards more valid error estimates is to represent the entire model fitting process programmatically to the greatest extent possible³⁰. This is one very good reason for documenting your data analysis in R scripts or Quarto documents and for wrapping up entire methods as new R functions instead of working interactively.

Example 8.6 (Bike rental revisited). We return to the bike rental example and implement K -fold cross-validation for the

For cross-validation, $B = MK$ is always an integer multiple $M \in \mathbb{N}$ of K .

²⁹ To estimate $\text{Risk}(\hat{P})$ we really need an independent validation dataset.

³⁰ Hardliners would accept only programmatic model fitting procedures. This goes against seeing data analysis as a fundamentally interactive discipline. Both viewpoints have merits, and the concrete application should determine which considerations are given the biggest weight. It requires human judgement!

squared error loss and for linear models that can be specified via a formula.

```
cv <- function(form, data, K = 10, M = 20) {
  n <- nrow(data)
  response <- all.vars(form)[1]
  loss <- vector("list", M)
  mu_hat <- numeric(n)
  for(b in 1:M) {
    # Generating the random subdivision into groups
    group <- sample(rep(1:K, length.out = n))
    for(i in 1:K) {
      model <- lm(form, data = data[group != i, ])
      mu_hat[group == i] <- predict(model, newdata = data[group == i, ])
    }
    loss[[b]] <- (data[, response] - mu_hat)^2
  }
  unlist(loss) |> mean()
}
```

We then run cross-validation for the sequence of models with df ranging from 1 to 12, and we also compute the training MSE. We use the default settings with $K = 10$ and $M = 20$ in the implementation of `cv()` above. That is, we do 10-fold cross-validation for $20 \cdot 10 = 200$ rounds.

```
set.seed(42)
cv_bike <- tibble(
  df = 1:12,
  cv = Vectorize(\(d) cv(Count ~ ns(Temperature, df = df[d]),
  mse = Vectorize(\(d) lm(Count ~ ns(Temperature, df = df[d])
    (\(x) residuals(x)^2)() |> mean())(df)
)
```

Figure 8.7 shows the resulting training MSE and cross-validated MSE. We make the following three observations:

1. The training MSE (red) is monotonely decreasing with model complexity. This is in accordance with our previous observations.
2. The cross-validated MSE (blue) is always larger than the training MSE. This is expected and reflects the optimism of the training error. We note that this optimism increases with model complexity.
3. The cross-validated MSE (blue) attains a minimum for $df = 4$. This shows that the model with optimal (expected) predictive performance is the model with $df = 4$.

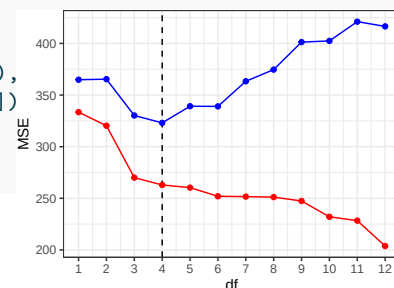


Figure 8.7: Training MSE (red) and cross-validated MSE (blue).

Properties of data splitting estimators

A formal justification that $\widehat{\text{Err}}$ is actually a good estimate of the expected generalization error is a bit involved, and we will only give some superficial arguments.

We should first observe that Err actually depends on the size of the dataset n used for fitting the model. To emphasize this dependence we write $\text{Err}(n)$ whenever needed. The graph of $\text{Err}(n)$ as a function of n is known as the error curve³¹. It will typically be monotonely decreasing and bounded below. If the sets $I(b)$ in the data splitting procedure are all of roughly equal size, $|I(b)| \approx m$ for all b , say, it is appropriate³² to regard $\widehat{\text{Err}}$ as an estimate of $\text{Err}(m)$ rather than $\text{Err}(n)$. If $m < n$ and the error curve is steep around m , $\widehat{\text{Err}}$ will overestimate $\text{Err}(n)$.

For subsampling it follows directly that

$$E(\widehat{\text{Err}}) = \text{Err}(m) \quad (\text{subsampling})$$

and for K -fold cross-validation

$$E(\widehat{\text{Err}}) \approx \text{Err}((1 - 1/K)n) \quad (\text{cross-validation}).$$

For small datasets the error curve may be rather steep in a neighborhood of n , and it is recommended to take $K = 10$ to obtain an estimate of $\text{Err}(0.9n)$. For large datasets the error curve may be less steep and taking $K = 5$ to obtain an estimate of $\text{Err}(0.8n)$ will suffice.

The situation for bootstrapping is more complicated. Even though $|I(b)| = n$, some of the data points appear multiple times due to the sampling with replacement. Efron (1983) showed that $E(\widehat{\text{Err}})$ agrees with $\text{Err}(n/2)$ up to second order, which makes the bootstrap method comparable with 2-fold cross-validation. This gives a potential bias of the method. Several corrections exist but they will not be pursued here, see Efron and Tibshirani (1997).

There is some instability of $\widehat{\text{Err}}$ caused by the random splitting. This can be reduced by replicating the data splitting procedure to give as many rounds as is computationally feasible. For K -fold cross-validation the number of rounds are integer multiples M of K . Cross-validation can sensibly be used with $M = 1$ giving the minimal number of K rounds, which makes

³¹ A close relative to the *learning curve*, which is given by $c - \text{Err}(n)$ for some suitable constant c .

³² For subsampling and cross-validation but not bootstrapping.

Efron, Bradley. 1983. "Estimating the Error Rate of a Prediction Rule: Improvement on Cross-Validation." *Journal of the American Statistical Association* 78 (382): 316–31.

Efron, Bradley, and Robert Tibshirani. 1997. "Improvements on Cross-Validation: The 632+ Bootstrap Method." *Journal of the American Statistical Association* 92 (438): 548–60.

it only moderately computationally demanding. Among the three data splitting procedures, cross-validation with $K = 5$ or $K = 10$ is the most widely used and recommended as it usually strikes a good balance between computation time and stability of $\widehat{\text{Err}}$. Adding extra rounds will increase the stability of the estimate $\widehat{\text{Err}}$ but also increase the computational burden. Acceptable stability of $\widehat{\text{Err}}$ usually requires more than 5–10 rounds for subsampling or bootstrapping.

The special case of n -fold cross-validation is also known as leave-one-out cross-validation. There is no instability in this procedure, just n rounds of model fitting to $n - 1$ data points and loss computation for the last omitted data point. The computation time scales in principle linearly with the size of the dataset, which makes it computationally intractable for large datasets. However, for a class of estimators that are linear in the response there exists an analytic shortcut, which allows for the computation of leave-one-out estimate of $\widehat{\text{Err}}$ directly from the original model fit. In this case leave-one-out cross-validation can be a computationally cheap alternative to the other data splitting procedures.

Irrespective of the data splitting procedure used, it is difficult to quantify the uncertainty of $\widehat{\text{Err}}$ as an estimator of Err . The estimator is an average of dependent variables and we cannot easily estimate its variance, say. It is common for cross-validation with $B = K$ (that is, $M = 1$) to pretend that the terms in (8.9) are i.i.d. and compute standard errors under this assumption. This is an acceptable heuristic, but generally we don't have any good finite sample approximations of the distribution of $\widehat{\text{Err}}$. The distributions of the different error estimators are explored empirically in a simulation study in Section 8.5.

8.4.3 Analytic error estimates

In the previous section the training error, err , was discarded as a valid error estimate. Alternatives were described based on the idea of using independent validation data for error estimation. This section gives analytic error estimates that can be understood as bias corrections of err . These analytic estimates do not rely on independent validation or repeated data splitting and refitting of the model, and they can typically be computed efficiently. This comes at the price of added

model assumptions, and the analytic error estimates are unfortunately estimates of yet another error quantity closely related to – but not exactly the same as – the average risk.

Definition 8.4. With $Y_1^{\text{new}}, \dots, Y_n^{\text{new}}$ conditionally independent of $\hat{P}$ given $\mathbf{X}$ and with the same conditional distribution as $Y_1, \dots, Y_n$ the in-sample error is defined as

$$\text{Err}_{\text{in}} = \frac{1}{n} \sum_{i=1}^n E(\text{Loss}(Y_i^{\text{new}}, \hat{P}(X_i)) \mid \mathbf{X}).$$

The general definition above of the in-sample error is based on Assumption A5 of conditionally independence of the response variables given the predictors, which is weaker than Assumption A6. This is sensible as the in-sample error fixes the predictors to be the observed predictions, and it is therefore also a weaker quantification of predictive strength than the (expected) generalization error. It can, nevertheless, still be useful for model comparisons.

The main reason we consider the in-sample error is that we can derive analytic representations of it that links it to the training error, and which can be used for easy estimation without using independent validation data or data splitting. For the deviance loss the following theorem gives such a representation of the in-sample error where any reference to independent data is eliminated.

Theorem 8.1. *If the loss is the unit deviance for an exponential dispersion model with corresponding canonical link function g then*

$$\text{Err}_{\text{in}} = \mathbf{E}(\text{err} \mid \mathbf{X}) + \frac{2}{n} \sum_{i=1}^n \text{cov}(g(\hat{\mu}_i), Y_i \mid \mathbf{X}). \quad (8.10)$$

Proof. From Definition 4.5 of the deviance, or rather Equation 4.5,

$$d_i^{\text{new}} := d(Y_i^{\text{new}}, \hat{\mu}_i) = r(Y_i^{\text{new}}) - 2g(\hat{\mu}_i)Y_i^{\text{new}} + 2\kappa(g(\hat{\mu}_i))$$

Observe that $\mathbf{E}(\text{Err}_{\text{in}})$ is *not* equal to Err in general. There is a difference between reusing the X_i -s or using independent X_i^{new} -s for the prediction.

Recall that for a generalized linear model,

$$\hat{\mu}_i = \mu(X_i^T \hat{\beta}),$$

The proof is based on the deviance formula in Equation 4.5, which not only holds for $Y_i \in \mathcal{J}$ but also for $Y_i \in \mathcal{J} \setminus \mathcal{J}$ for some specific cases, including the binomial and Poisson models.

and

$$d_i := d(Y_i, \hat{\mu}_i) = r(Y_i) - 2g(\hat{\mu}_i)Y_i + 2\kappa(g(\hat{\mu}_i))$$

for some function r . Subtracting these two quantities and taking conditional expectation given $\mathbf{X}$ yields

$$\mathbf{E}(d_i^{\text{new}} - d_i \mid \mathbf{X}) = 2\mathbf{E}(g(\hat{\mu}_i)(Y_i - Y_i^{\text{new}}) \mid \mathbf{X})$$

since Y_i and Y_i^{new} are assumed to have the same conditional distribution. This implies, additionally, that $Y_i - Y_i^{\text{new}}$ has conditional mean 0, and the right hand side above is a covariance. Thus

$$\begin{aligned} \mathbf{E}(d_i^{\text{new}} - d_i \mid \mathbf{X}) &= 2\text{cov}(g(\hat{\mu}_i), Y_i - Y_i^{\text{new}} \mid \mathbf{X}) \\ &= 2\text{cov}(g(\hat{\mu}_i), Y_i \mid \mathbf{X}) \end{aligned}$$

since Y_i^{new} is conditionally independent of $\hat{\mu}_i$. This gives that

$$\begin{aligned} \text{Err}_{\text{in}} &= \frac{1}{n} \sum_{i=1}^n \mathbf{E}(d_i^{\text{new}} \mid \mathbf{X}) \\ &= \frac{1}{n} \sum_{i=1}^n \mathbf{E}(d_i \mid \mathbf{X}) + \frac{1}{n} \sum_{i=1}^n 2\text{cov}(g(\hat{\mu}_i), Y_i \mid \mathbf{X}) \\ &= \mathbf{E}(\text{err} \mid \mathbf{X}) + \frac{2}{n} \sum_{i=1}^n \text{cov}(g(\hat{\mu}_i), Y_i \mid \mathbf{X}). \end{aligned}$$

□

Equation 8.10 may appear intimidating at first, but it is really quite interpretable. Clearly, err is an unbiased estimate of $\mathbf{E}(\text{err} \mid \mathbf{X})$, but (8.10) shows that it is biased as an estimate of the in-sample error. The bias is expressed in terms of the covariances between $g(\hat{\mu}_i)$ and Y_i , which makes good sense. A large positive correlation between the response Y_i and the value $g(\hat{\mu}_i)$ for $i = 1, \dots, n$ indicates that the $\hat{\mu}_i$ -s will be somewhat worse at predicting independent Y_i^{new} -s than err suggests.

Definition 8.5. For a canonical link function g the corresponding *effective degrees of freedom* of the estimator $\hat{\mu}_i$, $i = 1, \dots, n$, is defined as

$$\text{df} = \frac{1}{\varphi} \sum_{i=1}^n \text{cov}(g(\hat{\mu}_i), Y_i \mid \mathbf{X}).$$

The effective degrees of freedom only makes sense when using the deviance corresponding to the canonical link g as loss. However, it makes sense for any choice of estimator $\hat{\mu}_i$.

If $\widehat{\text{df}}$ denotes a suitable estimator of the effective degrees of freedom we can estimate the in-sample error as

$$\widehat{\text{Err}}_{\text{in}} = \text{err} + 2\frac{\widehat{\varphi}}{n}\widehat{\text{df}}, \quad (8.11)$$

where $\widehat{\varphi}$ denotes an estimator of the dispersion parameter (if it needs to be estimated). For a generalized linear model we can use the estimator

$$\widehat{\varphi} = \frac{1}{n-p}\mathcal{X}^2$$

of the dispersion parameter, where $\mathcal{X}^2$ is the Pearson $\mathcal{X}^2$ -statistic for the model with p parameters. If $\widehat{\text{Err}}_{\text{in}}$ is used for comparing multiple models, we should use a single model with a small bias (large p) to estimate φ . A closely related quantity is Mallows's C_p -statistic, which is defined as

$$C_p = n\widehat{\text{Err}}_{\text{in}} = n\text{err} + 2\widehat{\varphi}\widehat{\text{df}}.$$

From a predictive point of view it is sensible to use $\widehat{\text{Err}}_{\text{in}}$ or C_p for comparison of two models whether they are nested or not. The smaller $\widehat{\text{Err}}_{\text{in}}$ is the better is the model.

For (8.11) to be of practical use we need to be able to compute or estimate the effective degrees of freedom. The following examples show that it is, indeed, possible to compute this number in several different situations.

Example 8.7 (Squared error loss and linear estimators). Suppose that $\widehat{\mu}_i = (\mathbf{S}\mathbf{Y})_i$ for an $n \times n$ matrix $\mathbf{S}$ (which depends on $\mathbf{X}$ but *not* $\mathbf{Y}$). Such an estimator is called linear. The projection estimator derived in Section 2.5 for the linear model is a linear estimator given by

$$\mathbf{S} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T.$$

Suppose also that the distributional assumptions A2 and A4 are fulfilled³³. We can then observe that for the *squared error loss* with corresponding canonical link being the identity, the

Many so-called *smoothers* are, in fact, linear estimators for a suitable choice of matrix $\mathbf{S}$ – hence the notation $\mathbf{S}$ for smoother.

³³ Note that there is no assumption about the conditional expectation of the Y_i -s. It is only assumed that they are conditionally uncorrelated and have homogeneous variance.

effective degrees of freedom equals

$$\begin{aligned}
df &= \frac{1}{\sigma^2} \sum_{i=1}^n \text{cov}(\hat{\mu}_i, Y_i \mid \mathbf{X}) \\
&= \frac{1}{\sigma^2} \sum_{i,j=1}^n \text{cov}(S_{ij}Y_j, Y_i \mid \mathbf{X}) \\
&= \frac{1}{\sigma^2} \sum_{i=1}^n S_{ii}\sigma^2 \\
&= \text{trace}(\mathbf{S}).
\end{aligned}$$

Thus

$$\text{Err}_{\text{in}} = \mathbf{E}(\text{err} \mid \mathbf{X}) + \frac{2\sigma^2}{n} \text{trace}(\mathbf{S}).$$

and

$$\widehat{\text{Err}}_{\text{in}} = \text{err} + \frac{2\hat{\sigma}^2}{n} \text{trace}(\mathbf{S}).$$

When $\mathbf{S}$ is an orthogonal projection its trace is equal to the dimension p of its image, in which case

$$\widehat{\text{Err}}_{\text{in}} = \text{err} + \frac{2\hat{\sigma}^2 p}{n}. \quad (8.12)$$

Example 8.8 (Bike rentals revisited). All the models considered for the bike rental example are linear models, and we can use (8.12) to estimate the in-sample error. Doing so, we also add cross-validation errors estimated using either 5-fold or 20-fold cross-validation – in addition to 10-fold cross-validation already considered in Example 8.6.

```

n <- nrow(bike)
sigma_sq_hat <- cv_bike$mse[12] * n / (n - 13)
cv_bike <- mutate(
  cv_bike,
  mse_in = mse + 2 * sigma_sq_hat * (df + 1) / n,
  cv_5 = Vectorize(\(d) cv(Count ~ ns(Temperature, df = df[d]), bike, K = 5))(df),
  cv_25 = Vectorize(\(d) cv(Count ~ ns(Temperature, df = df[d]), bike, K = 25))(df),
)

```

Figure 8.8 shows that even though there are differences both between the in-sample error and the cross-validated errors and within the cross-validated errors depending on the folds, all the error estimates have a similar shape and agree that models with $df = 3$ or $df = 4$ are best.

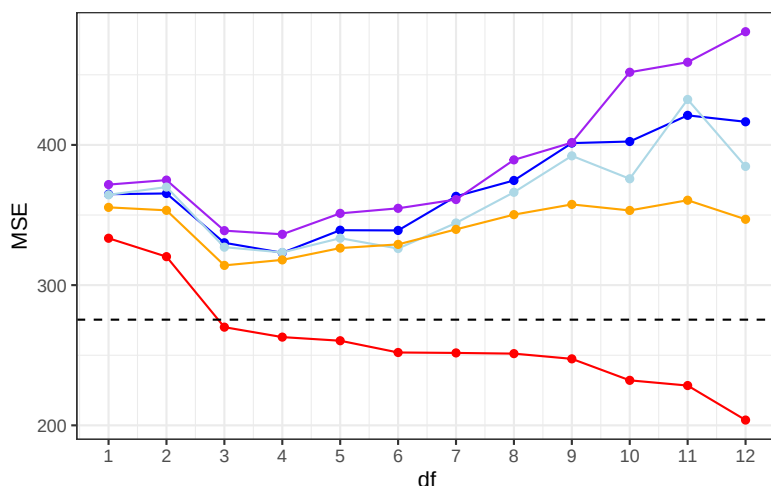


Figure 8.8: Training MSE (red), cross-validated MSE (purple: $K = 5$; blue: $K = 10$; light blue: $K = 25$) and in-sample MSE (orange). The horizontal dashed line is $\hat{\sigma}^2$ (an estimate of the lower bound on MSE).

As discussed in Section 8.4, 5-fold cross-validation is in this case effectively estimating $\text{Err}(40)$, corresponding to a sample size of $n = 40$. Since 10-fold and 25-fold cross-validation correspond to estimating $\text{Err}(45)$ and $\text{Err}(48)$, respectively, and since we expect $\text{Err}(40) > \text{Err}(45) > \text{Err}(48) > \text{Err}(50)$, the ordering of the cross-validated error estimates is unsurprising. We note that the in-sample error is generally smaller than the cross-validated errors, but it resembles 25-fold cross-validation more than 5-fold cross-validation.

It is possible to justify a generalization of Equation 8.11 to generalized linear models when the estimator is the MLE and the link function is canonical.

Example 8.9. Assume that $\hat{\mu}_i = \mu(X_i^T \hat{\beta}^{\text{ideal}})$ with $\hat{\beta}^{\text{ideal}}$ the idealized least squares estimator introduced in Section 5.3.1, and the $\mu = g^{-1}$ is the inverse of the canonical link g . Then $g(\hat{\mu}_i) = (\mathbf{S}\mathbf{Z})_i$ for

$$\mathbf{S} = \mathbf{X}(\mathbf{X}^T \mathbf{W} \mathbf{X})^{-1} \mathbf{X}^T \mathbf{W}.$$

Under the distributional assumptions GA2 and A4, Z_i is conditionally uncorrelated with Y_j for $i \neq j$ and

$$\text{cov}(Z_i, Y_i | \mathbf{X}) = \frac{\varphi \mathcal{V}(\mu_i)}{\mu'_i} = \varphi,$$

since $\mu = g^{-1}$. As in Example 8.7 we conclude that

$$\sum_{i=1}^n \text{cov}(\hat{\mu}_i, Y_i | \mathbf{X}) = \varphi \text{trace}(\mathbf{S}).$$

Now

$$\begin{aligned}\text{trace}(\mathbf{S}) &= \text{trace}\left(\mathbf{X}(\mathbf{X}^T\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{W}\right) \\ &= \text{trace}\left((\mathbf{X}^T\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{W}\mathbf{X}\right) \\ &= \text{trace}(\mathbf{I}_p) = p,\end{aligned}$$

where $\mathbf{I}_p$ denotes the $p \times p$ identity matrix.

AIC

In addition to the estimate of the in-sample error or Mallows's C_p above, AIC must be mentioned as another classical analytic way to quantify a model's predictive strength. As stated below, AIC is up to scaling by the dispersion parameter identical to Mallows's C_p if the deviance loss is used.

Definition 8.6. Assume that $P_i = P_i(\beta)$ for $\beta \in \mathbb{R}^p$ a p -dimensional parameter β . With ℓ the negative log-likelihood loss we define Akaike's information criteria as

$$\text{AIC} = 2 \sum_{i=1}^n \ell(\hat{P}_i) + 2p.$$

In the context of generalized linear models³⁴ we observe that

$$\text{AIC} = \frac{1}{\varphi} D + 2p$$

where $D = \sum_{i=1}^n d(Y_i, \hat{\mu}_i)$ is the deviance in Definition 5.2. The log-likelihood and hence AIC is only determined up to an additive constant. The estimator of Risk derived from AIC is

$$\frac{1}{2n} \text{AIC}.$$

The justification of AIC hinges on a number of assumptions, and its justification is asymptotic for $n \rightarrow \infty$. We will not pursue the arguments in any detail, but just mention three important assumptions for AIC to be justifiable:

1. The estimator must be $\hat{P}_i = P_i(\hat{\beta})$ with $\hat{\beta}$ the *maximum likelihood estimator*.

The factor “2” in the definition of AIC is a somewhat arbitrary convention.

³⁴ Either assuming that we know the dispersion parameter or that we just plainly ignore that it is estimated too. If it is estimated we should use a single value for all models as discussed above in relation to Mallows's C_p .

2. The data generating model must be of the form $P(\beta)$ for some β , thus AIC is invalid if the model is misspecified.
3. For comparability of AIC between two models it is essential that the same reference measure is used for the definition of the likelihood.

One has to pay particular attention to point 3 above for generalized linear models. For example, the Poisson and the binomial distributions have different structure measures as exponential families³⁵. If we do not account for this, the models are incomparable in terms of their AIC. This is unsurprising from a risk perspective as different models give different deviance loss functions. When comparing models we must use the same loss function throughout.

³⁵ One has the factor $\frac{1}{Y!}$ and the other $\binom{m}{Y}$ on the counting measure.

8.5 Data splitting estimators compared

This section illustrates aspects of the practical implementation of data splitting for quantification of predictive strength. This is done via a simulation study using the squared error loss, which also results in a comparison of the different data splitting procedures and the empirical distribution of their resulting estimators.

8.5.1 Implementations in R

We first implement a couple of general functions for computing training error, validation error and estimates of expected generalization error by subsampling or bootstrapping. We have already implemented cross-validation in Section 8.4.2, which we reuse.

```
train_err <- function(form, data) {
  model <- lm(form, data = data)
  residuals(model)^2 |> mean()
}

val_err <- function(form, data, newdata) {
  response <- all.vars(form)[1]
  model <- lm(form, data = data)
  (newdata[, response] - predict(model, newdata = newdata))^2 |> unlist() |> mean()
}
```

The two functions implemented below compute estimates of expected generalization error using subsampling and bootstrapping. Their implementations are quite similar and similar to the implementation of `cv()` in Example 8.6.

```
subsamp <- function(form, data, B = 200, q = 0.9) {
  n <- nrow(data)
  response <- all.vars(form)[1]
  loss <- vector("list", B)
  for(b in 1:B) {
    # Subsampling a data set of size q * n
    i <- sample(n, floor(q * n))
    model <- lm(form, data = data[i, ])
    mu_hat <- predict(model, newdata = data[- i, ])
    loss[[b]] <- (data[-i, response] - mu_hat)^2
  }
  unlist(loss) |> mean()
}
```

```
boot <- function(form, data, B = 200) {
  n <- nrow(data)
  response <- all.vars(form)[1]
  loss <- vector("list", B)
  for(b in 1:B) {
    i <- sample(n, n, replace = TRUE)
    model <- lm(form, data = data[i, ])
    mu_hat <- predict(model, newdata = data[- i, ])
    loss[[b]] <- (data[-i, response] - mu_hat)^2
  }
  unlist(loss) |> mean()
}
```

8.5.2 Simulation study

First we apply the different functions on a single data set for a simple linear regression model with p_0 predictors. None of the predictors are related to the normally distributed response, whence we simulate from the model with

$$Y \mid X \sim \mathcal{N}(0, \sigma^2).$$

In the simulations we fix $p_0 = 20$ and $\sigma^2 = 1$.

```
sim_yx <- function(n, p0) {
  x <- matrix(rnorm(p0 * n), n, p0)
  tibble(
    y = rnorm(n),
```

```

    as_tibble(x, .name_repair = "unique_quiet")
  )
}
n <- 100
p0 <- 20
# p = p0 + 1 = 21 as there will also be an intercept
set.seed(21)
yx <- sim_yx(n, p0)
form <- y ~ . # Includes all 20 variables as predictors

```

train	subsamp.	cv	bootstrap
0.73	1.24	1.21	1.47

Table 8.3: Four error estimates from running the simulation once.

Table 8.3 shows four error estimates for the single simulation above. With reference to Example 8.5 we know that the lower bound on the expected generalization error is $\sigma^2 = 1$ (the Bayes risk). The training error in Table 8.3 is smaller, illustrating the optimism of the training error. The other three error estimates in Table 8.3 are larger, and the bootstrap estimate is notably larger than the two others.

In this example we can actually compute the expected training error³⁶. Since $\text{err} = \hat{\sigma}^2 \frac{n-p}{n}$, it follows from Theorem 2.2 that

$$\mathbf{E}(\text{err}) = \mathbf{E}(\text{err} \mid \mathbf{X}) = \sigma^2 \left(1 - \frac{p}{n}\right) = 1 - 21/100 = 0.79.$$

The in-sample error can then also be computed as

$$\text{Err}_{\text{in}} = \mathbf{E}(\text{err} \mid \mathbf{X}) + 2\frac{p}{n} = 1 + 21/100 = 1.21.$$

For subsampling with $q = 0.9$ and for 10-fold cross-validation, the error estimates have mean $\text{Err}(90)$ and the bootstrapping error estimate has approximately mean $\text{Err}(50)$. Though these numbers are not exactly the same as the corresponding in-sample errors for $n = 90$ and $n = 50$, which are 1.23 and 1.42, respectively, they turn out to be relatively close.

In the simulation above a large number of rounds ($B = 200$) were used. In real applications this may be computationally prohibitive as this requires a large number of refits of the

³⁶ Unconditionally as well as conditionally on $\mathbf{X}$.

model. Below we investigate by simulations the distribution of the error estimators, and we study the effect of the number of rounds B on these distributions.

First we take $B = 10$ rounds, which for 10-fold cross-validation corresponds to a single outer loop ($M = 1$).

```
err_hat <- vector("list", 200)
for(m in seq_along(err_hat)) {
  yx <- sim_yx(n, p0)
  yx_new <- sim_yx(n, p0)
  err_hat[[m]] <- tibble(
    train = train_err(form, yx),
    validation = val_err(form, yx, yx_new),
    subsamp. = subsamp(form, yx, B = 10),
    cv = cv(form, yx, K = 10, M = 1),
    bootstrap = boot(form, yx, B = 10)
  )
}
```

```
err_hat <- do.call(rbind, err_hat) |>
  tidyr::pivot_longer(
    cols = everything(),
    names_to = "Method",
    values_to = "Squared error"
  ) |>
  mutate(
    Method = factor(Method, c("train", "validation", "subsamp.", "cv", "bootstrap"))
  )

ggplot(err_hat, aes(x = Method, y = `Squared error`)) +
  geom_violin(fill = gray(0.8)) +
  geom_hline(yintercept = 1 + (p0 + 1) / n, color = "red", alpha = 0.5, linewidth = 1) +
  geom_hline(yintercept = 1 - (p0 + 1) / n, color = "blue", alpha = 0.5, linewidth = 1) +
  stat_summary(fun = mean, color = "blue", geom = "point") +
  coord_cartesian(ylim = c(0, 2.5))
```

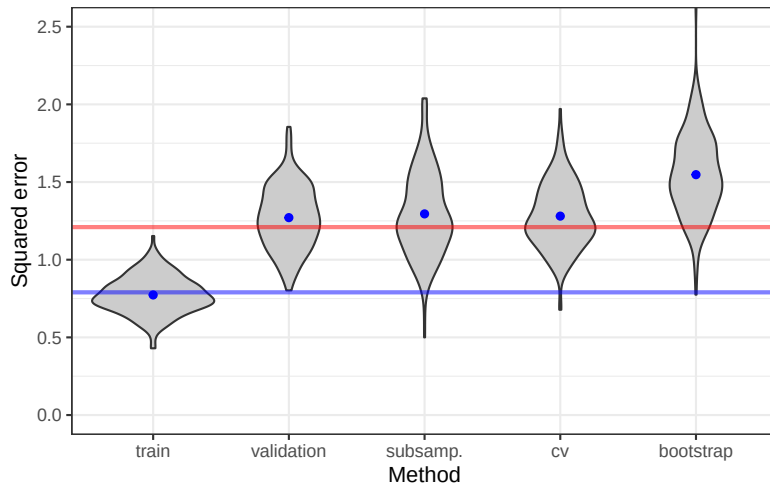


Figure 8.9: Distributions of training error and risk estimators using validation data, subsampling, cross-validation and bootstrapping. All three data splitting methods used $B = 10$ rounds. The blue line is the expected training error and the red line is the in-sample error.

Method	Mean	SE
train	0.77	0.12
validation	1.27	0.20
subsamp.	1.29	0.27
cv	1.28	0.21
bootstrap	1.55	0.26

Table 8.4: Estimates of mean and standard error for the five different error estimators with $B = 10$ rounds.

Figure 8.9 shows the distribution of the different error estimators together with the training error. We observe that the training error generally underestimates the expected generalization error as well as the in-sample error. Note that the C_p -statistic corresponds exactly to the training error plus $2p/n = 42/100 = 0.42$ in this case.

The different estimates of risk show a considerable variation. A fraction of this variation is due to a random data splitting. By increasing B we can decrease the variance. Figure 8.10 shows the result of increasing the number of rounds by a factor 5 to $B = 50$.

```
err_hat <- vector("list", 200)
for(m in seq_along(err_hat)) {
  yx <- sim_yx(n, p0)
  yx_new <- sim_yx(n, p0)
  err_hat[[m]] <- tibble(
    train = train_err(form, yx),
```

```

validation = val_err(form, yx, yx_new),
subsamp. = subsamp(form, yx, B = 50),
cv = cv(form, yx, K = 10, M = 5),
bootstrap = boot(form, yx, B = 50)
)
}

```

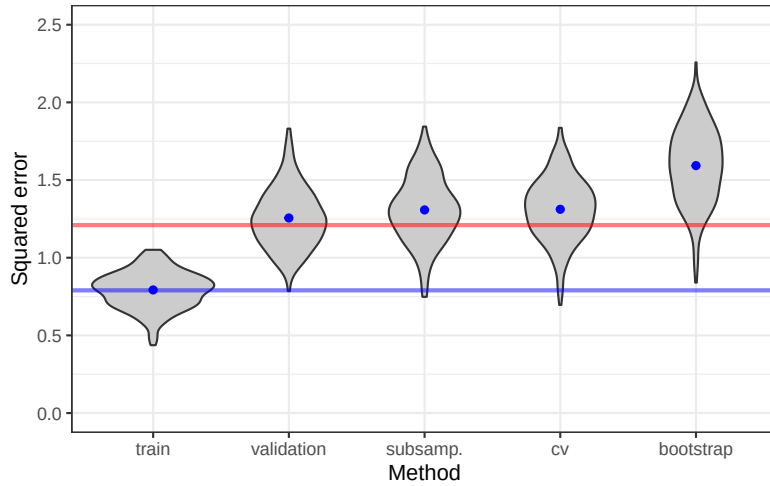


Figure 8.10: Distributions of training error and risk estimators using validation data, subsampling, cross-validation and bootstrapping. All three data splitting methods used $B = 50$ rounds. The blue line is the expected training error and the red line is the in-sample error.

Table 8.4 and Table 8.5 show the means and standard errors for the error estimators. The conclusion is that the increase of B does decrease the spread of the error estimators' distributions. This is most pronounced for subsampling and bootstrapping methods. For cross-validation the decrease is detectable, but smaller. We also observe that bootstrapping, as expected, results in a larger error estimate.

Method	Mean	SE
train	0.79	0.12
validation	1.26	0.19
subsamp.	1.31	0.21
cv	1.31	0.20
bootstrap	1.59	0.25

Table 8.5: Estimates of mean and standard error for the five different error estimators with $B = 50$ rounds.

8.6 Binary model assessment

The framework of risk, generalization error, in-sample error etc. was introduced for a general loss function in the previous sections. The deviance loss – or negative log-likelihood loss – was treated in detail. With a binary response variable there are alternative and widely used methods for assessing the predictive strength of a regression model. This section treats some of the most important ones.

8.6.1 Concordance parameters

A probability model of a binary variable is completely characterized by the probability $p = \mathbf{P}(Y = 1)$. For a regression model, this probability will be a function of the predictor,

$$p(x) = \mathbf{P}(Y = 1 \mid X = x) = \mathbf{E}(Y \mid X = x).$$

This is simply the regression function for a binary response.

To give a prediction in the sample space $\{0, 1\}$ we can, for instance, choose a threshold like 0.5 and predict that $Y = 1$ if $p(x) > 0.5$. It is, however, more informative to know $p(x)$ than just to know if $p(x) > 0.5$. The focus is therefore typically on modeling $p(x)$ rather than just if $p(x) > 0.5$. It is, moreover, of considerable interest to study predictions generated by a probability model for a range of different values of the threshold instead of just choosing a fixed threshold such as 0.5.

For a generalized linear model, $p(x) = \mu(x^T\beta)$ where μ is the mean value function and $x^T\beta$ is the linear predictor. In this section we will study general prediction models being real valued functions, $x \mapsto \eta(x) \in \mathbb{R}$, of the predictor. In combination with a choice of mean value function $\mu : \mathbb{R} \rightarrow [0, 1]$ a prediction model gives rise to a probability model

$$p(x) = \mu(\eta(x)).$$

If the mean value function is strictly monotonely increasing with inverse, g , being the corresponding link function, we observe that

$$p(x) > t \quad \Leftrightarrow \quad \eta(x) > g(t).$$

Definition 8.7. For η_0 and η_1 independent real valued random variables

$$\text{AUC} = \mathbf{P}(\eta_0 < \eta_1) + \frac{1}{2}\mathbf{P}(\eta_0 = \eta_1).$$

The parameter AUC is a *concordance parameter*, which measures concordance between the distributions of the variables η_0 and η_1 . For AUC around 0.5 the two distributions are overlapping or in concordance. For AUC close to 1 or 0 the two distributions are well separated.

If Y is a binary variable and $\eta(X)$ is a predictor we will be interested in AUC for

$$\eta_0 = \eta(X) \mid Y = 0 \quad \text{and} \quad \eta_1 = \eta(X) \mid Y = 1.$$

One interpretation of AUC is as a quantification of how separated the conditional distributions of $\eta(X) \mid Y = 0$ and $\eta(X) \mid Y = 1$ are. A predictor model with $\text{AUC} \approx 0.5$ will not give accurate predictions of Y by thresholding $\eta(X)$. If $\text{AUC} \approx 1$ it is possible to choose a threshold t that almost separates the two distributions, so that Y can be accurately predicted as $1(\eta(X) > t)$.

Changing the threshold will always change the tradeoff between the type of errors we will make, but the specific choice of threshold does not affect AUC. This is one of the attractive properties of AUC. If $\text{AUC} \approx 0$, the threshold predictor $1(\eta(X) < t)$ will be an accurate predictor of Y , but we prefer in this case to replace $\eta(X)$ by $-\eta(X)$. We will thus always assume that the sign of η is chosen so that larger values of $\eta(X)$ indicate that $Y = 1$.

If the distribution functions for η_0 and η_1 are denoted F_0 and F_1 , respectively, then

$$\begin{aligned} \text{AUC} &= \int F_0(t) dF_1(t) - \frac{1}{2} \sum_t \Delta F_0(t) \Delta F_1(t) \\ &= \int 1 - F_1(t) dF_0(t) + \frac{1}{2} \sum_t \Delta F_0(t) \Delta F_1(t). \end{aligned}$$

If the distribution of η_0 or η_1 is continuous,

$$\begin{aligned} \text{AUC} &= \mathbf{P}(\eta_0 < \eta_1) \\ &= \mathbf{P}(\eta_0 \leq \eta_1). \end{aligned}$$

Exercises

Exercise 8.1. This exercise is based on the gapminder dataset included in the Rwr package. It contains the gross domestic product (GDP) per capita and the average life expectancy at birth (Life) for 139 countries. Figure 8.11 shows a scatter plot of the data.

```
# Rwr version >= 0.3.3
data(gapminder, package = "Rwr")
head(gapminder)
```

```
# A tibble: 6 x 2
  Life  GDP
<dbl> <dbl>
1  75.3 12314
2  58.3  7103
3  75.5 14646
4  72.5  7383
5  81.5 41312
6  80.4 43952
```

We will consider a linear model of the form

$$m(x) = \mathbf{E}(\text{Life} \mid \text{GDP} = x) = \beta_0 + \sum_{d=1}^{\text{df}} \beta_d h_d(x)$$

where m is a natural cubic spline. When answering the questions below, use the squared error loss function.

1. Fit the above model with 1, 2, 4, 8, and 16 degrees of freedom to the gapminder data. Compute the training error for each of the fitted models. Based on a visual inspection of the fitted models, is there any indication of underfitting and overfitting?
2. Compute an estimate of the expected generalisation error using 10-fold cross-validation (with $M = 1$). Which model has the best predictive performance?
3. For the model selected above, repeat 10-fold cross-validation 100 times and visualize the distribution of the error estimator. How robust are the estimates with respect to the randomness in the data splitting?

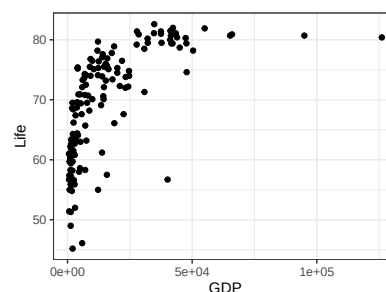


Figure 8.11: Scatter plot of average life expectancy at birth against gross domestic product (GDP).

4. Redo question 3 above for K -fold cross-validation with $K = 5, 10, 20, 50, 100$. How does the robustness of the estimates change as K changes? What are the advantages and disadvantages associated with choosing small and large values for K ?

Exercise 8.2. Recall from Example 8.1 that a linear estimator is of the form

$$\hat{\mu} = \mathbf{S}\mathbf{Y}$$

for an $\mathbf{X}$ -dependent matrix $\mathbf{S}$. Here $\mu = \mathbf{E}(\mathbf{Y} \mid \mathbf{X}) \in \mathbb{R}^n$ is the vector of conditional expectations of the response given the predictors.

1. Show that the penalized and weighted least squares estimator, solving the normal equation in Theorem 2.1, is linear and identify the matrix $\mathbf{S}$.

For the rest of this exercise we assume that the weight matrix is $\mathbf{W} = \mathbf{I}$ and the penalty matrix is diagonal and of the form

$$\Omega = \lambda \begin{pmatrix} \omega_1 & 0 & \dots & 0 \\ 0 & \omega_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \omega_p \end{pmatrix}$$

where $\lambda > 0$. You should think about λ as a *tuning parameter* that we want to optimize over, while $\omega_1, \dots, \omega_p \geq 0$ are fixed penalty weights determining the relative distribution of penalization on the different parameters. We also let $\hat{\beta}$ and $\hat{\mu}$ denote the (unpenalized) least squares estimator of β and μ , respectively.

2. Show that if the model matrix $\mathbf{X}$ has orthogonal columns³⁷, then with

$$\gamma_j = \sum_{i=1}^n X_{ij}^2 = \|\mathbf{X}_{\cdot,j}\|^2$$

the squared 2-norm of the j -th column of $\mathbf{X}$, the penalized least squares estimator, $\hat{\beta}_\lambda$, is given by

$$\hat{\beta}_{\lambda,j} = \frac{\gamma_j}{\gamma_j + \lambda\omega_j} \hat{\beta}_j$$

³⁷ An $n \times p$ matrix $\mathbf{X}$ with *orthonormal* columns satisfies that $\mathbf{X}^T \mathbf{X} = \mathbf{I}$, and we may call $\mathbf{X}$ *orthogonal* even though it is typically not square ($n > p$). Here we only assume that the columns are orthogonal implying that $\mathbf{X}^T \mathbf{X}$ is diagonal.

and $\hat{\mu}_\lambda = \mathbf{X}\hat{\beta}_\lambda$. Show also that the effective degrees of freedom is

$$\text{df}_\lambda = \sum_{j=1}^p \frac{\gamma_j}{\gamma_j + \lambda \omega_j}.$$

3. Show that if $\omega_1 = \dots = \omega_p = 1$ then, irrespectively of whether $\mathbf{X}$ has orthogonal columns, the effective degrees of freedom is

$$\text{df}_\lambda = \sum_{j=1}^p \frac{\gamma_j}{\gamma_j + \lambda}.$$

where $\gamma_1, \dots, \gamma_p \geq 0$ are the eigenvalues of $\mathbf{X}^T \mathbf{X}$.

For the remaining part of this exercise we consider the bike rental data introduced in Section 8.2. Using the `poly()` function we construct a model matrix with $p = 7$ orthogonal columns, corresponding to expanding the regression function using orthogonal polynomials of degree 6.

```
data(bike, package = "RwR")
X <- model.matrix(~ poly(Temperature, degrees = 6), data = bike)
beta_hat <- lm.fit(X, bike$Count) |> coefficients()
```

The least squares estimate is computed above as `beta_hat`, and the corresponding fitted values are shown in Figure 8.12. For the following questions we take $\omega_1 = \omega_2 = 0$ and $\omega_3 = \dots = \omega_7 = 1$.

4. Check numerically that $\mathbf{X}$ (approximately) has *orthogonal* columns.
5. Compute an estimate of σ^2 based on the least squares estimate $\hat{\beta}$ of β .
6. Implement the computation of $\hat{\mu}_\lambda$ and df_λ for a sequence³⁸ of values of λ .
7. Compute err for the sequence of λ -values and compute then a sequence of estimates of the in-sample error. Plot the in-sample error against the effective degrees of freedom. What is the best model according to the in-sample error?
8. Implement cross-validation to estimate the expected generalization error. Compare graphically err , $\widehat{\text{Err}}_{\text{in}}$ and $\widehat{\text{Err}}$ as a function of the effective degrees of freedom.

Note: There is a subtle difference between the following two procedures: i) splitting the data and recomputing the model

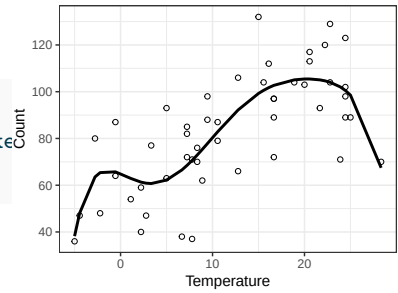


Figure 8.12: Least squares fit of a degree 6 polynomial to the bike rental data.

³⁸ It can be a little tricky to find a suitable sequence. Note that $\lambda = 0$ gives the least squares estimate and a large value of λ effectively makes $\hat{\beta}_\lambda = (\hat{\beta}_0, \hat{\beta}_1, 0, \dots, 0)^T$.

matrix using `poly()` on a subset of the data; and ii) subsetting the model matrix X that is computed from all data. You may want to explore this difference and think about which of the two procedures is most appropriate.

Exercise 8.3. Let η_0 and η_1 be independent real valued random variables fulfilling that

$$\eta_1 \stackrel{\mathcal{D}}{=} \eta_0 + \xi$$

for $\xi \in \mathbb{R}$. That is, the distribution of η_1 is a translation by ξ of the distribution of η_0 . Assume also that the distribution function, F_0 , for η_0 is continuous.

1. Show that

$$\text{AUC} = G(\xi)$$

where G is

$$G(z) = \int F_0(z + x) dF_0(x).$$

2. Argue that G is the distribution function for the distribution of $\eta_0 - \tilde{\eta}_0$, where $\tilde{\eta}_0$ is independent of η_0 with the same distribution.
3. Show that if $\eta_0 \sim \mathcal{N}(\mu, \sigma^2)$ then

$$\text{AUC} = \Phi\left(\frac{\xi}{\sqrt{2}\sigma}\right).$$

This exercise shows that for Gaussian distributions with the same variance, AUC is a monotonely increasing function of the *signal-to-noise ratio* ξ/σ .

9 Small for gestational age – a case study

! Important

This chapter is still under development and certain parts are still a bit rough. Notable changes will be made without notice, but the core structure of the chapter and the analyses and code are in place.

This chapter is a continuation of the case study on birth weight in Chapter 3. In that chapter the focus is on models of the *expected* birth weight. In this chapter the focus is on models of quantiles and more extreme cases, and on regression models with a binary response. Specifically, we will develop a model of a binary variable that we construct from the data and call *small for gestational age*. It will represent cases where the birth weight is relatively extreme among births for the same gestational age.

The construction of the binary variable will, in itself, be an instructive application of regression modeling of a quantile in a conditional distribution. Subsequently, we will use the constructed binary variable as our response and demonstrate various regression modeling techniques for binary responses using the framework of generalized linear models. One of these, the *logistic regression model*, deserves special attention, since it is both the historical antecedent of generalized linear models and the most widely applied generalized linear model.

9.1 Outline and prerequisites

In Section 9.2 we revisit the regression modeling of birth weight with only gestational age as predictor. We develop several different models of the quantiles in the conditional

distributions of birth weight given gestational age. We are particularly interested in the conditional 10% quantile, and based on one such model we define the variable *small for gestational age* as the indicator that birth weight is smaller than the 10% quantile. In Section 9.3 we cover a number of binary regression models, most notably logistic regression. The section covers both standard applications of the general theory and methodology for generalized linear models to this particular response distribution, and it covers some specific methods for binary responses, in particular in terms of measuring predictive performance and special model diagnostic tools such as calibration plots.

The chapter primarily relies on the following R packages.

```
library(RwR)
library(ggplot2)
library(tibble)
library(broom)
library(splines)
library(dplyr)
library(glmnet)
```

9.2 Small for gestational age

We first load the data from the RwR package and clean data as in Chapter 3.

```
data(birth_weight, package = "RwR")
birth_weight <- filter(
  birth_weight,
  weight > 32,
  length > 10 & length < 99,
  gestationalAge > 18
) |>
na.omit() |>
select(- c(interviewWeek, fetalDeath))
```

We then revisit the weight as a function of gestational age, see also Figure 3.6. We are now less interested in the mean or median weight but rather in the quantiles, specifically the lower quantiles of the distribution of weight as a function of gestational age. We compute a range of empirical quantiles stratified according to gestational age.


```
weight_quant <-
  group_by(birth_weight, gestationalAge) |>
  summarize(
    q05 = quantile(weight, 0.05),
    q10 = quantile(weight, 0.1),
    q50 = quantile(weight, 0.5),
    q90 = quantile(weight, 0.9),
    q95 = quantile(weight, 0.95)
  )
```

```
p_box <- ggplot(birth_weight, aes(x = as.factor(gestationalAge), y = weight)) +
  geom_boxplot(fill = gray(0.8), color = gray(0.5), alpha = 0.4, outlier.alpha = 0.1) + xlab("Gestational age")
p_box +
  geom_line(data = weight_quant, aes(y = q05, group = 1), color = "red") +
  geom_line(data = weight_quant, aes(y = q10, group = 1), color = "blue") +
  geom_line(data = weight_quant, aes(y = q50, group = 1), color = "purple") +
  geom_line(data = weight_quant, aes(y = q90, group = 1), color = "blue") +
  geom_line(data = weight_quant, aes(y = q95, group = 1), color = "red")
```

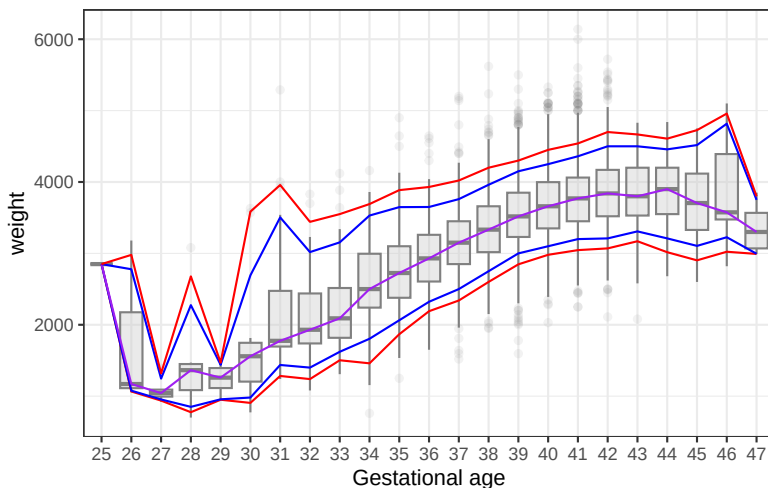


Figure 9.1: Boxplots of weight distributions as a function of gestational age, including the median (purple), the upper and lower deciles (blue) and the 5% and 95% quantiles (red).

It is clear from Figure 9.1 that the empirical quantiles are rather poorly determined for early and late births – due to the smaller number of cases. We therefore turn to a spline based regression model of the conditional mean, and then based on the model we can compute quantiles using two different methods. Either as if the residuals were normally distributed, or by computing the empirical quantiles³⁹ for the distribution of the residuals.

³⁹ This then avoids assuming that the residuals follow a normal distribution, but it does assume that the residual distribution, or at least the relevant quantile, is independent of the predictors. This is a slightly stronger assumption than A2.

```

nsg <- function(x) ns(x, knots = c(30, 35, 38, 40, 42, 45), Boundary.knots = c(25, 47))
small_lm_nsg <- lm(weight ~ nsg(gestationalAge), data = birth_weight)
df_res <- glance(small_lm_nsg)$df.residual
res <- residuals(small_lm_nsg)
sigma_hat <- (res^2 |> sum() / df_res) |> sqrt()

weight_quant <- mutate(
  weight_quant,
  q50_norm = predict(small_lm_nsg, newdata = weight_quant),
  q05_norm = q50_norm + sigma_hat * qnorm(0.05),
  q10_norm = q50_norm + sigma_hat * qnorm(0.10),
  q90_norm = q50_norm + sigma_hat * qnorm(0.90),
  q95_norm = q50_norm + sigma_hat * qnorm(0.95),
  q05_emp = q50_norm + quantile(res, 0.05),
  q10_emp = q50_norm + quantile(res, 0.10),
  q90_emp = q50_norm + quantile(res, 0.90),
  q95_emp = q50_norm + quantile(res, 0.95)
)

```

```

p1 <- ggplot(weight_quant, aes(gestationalAge)) +
  geom_line(aes(y = q05_norm), color = "red", linetype = 2) +
  geom_line(aes(y = q10_norm), color = "blue", linetype = 2) +
  geom_line(aes(y = q50_norm), color = "purple", linetype = 2) +
  geom_line(aes(y = q90_norm), color = "blue", linetype = 2) +
  geom_line(aes(y = q95_norm), color = "red", linetype = 2) +
  geom_line(aes(y = q05), color = "red") +
  geom_line(aes(y = q10), color = "blue") +
  geom_line(aes(y = q50), color = "purple") +
  geom_line(aes(y = q90), color = "blue") +
  geom_line(aes(y = q95), color = "red") +
  ylab("weight") + ylim(0, 5000) +
  ggtitle("A")

p2 <- ggplot(weight_quant, aes(gestationalAge)) +
  geom_line(aes(y = q05_norm), color = "red", linetype = 2) +
  geom_line(aes(y = q10_norm), color = "blue", linetype = 2) +
  geom_line(aes(y = q50_norm), color = "purple", linetype = 2) +
  geom_line(aes(y = q90_norm), color = "blue", linetype = 2) +
  geom_line(aes(y = q95_norm), color = "red", linetype = 2) +
  geom_line(aes(y = q05_emp), color = "red") +
  geom_line(aes(y = q10_emp), color = "blue") +
  geom_line(aes(y = q90_emp), color = "blue") +
  geom_line(aes(y = q95_emp), color = "red") +
  ylab("weight") + ylim(0, 5000) +
  ggtitle("B")

gridExtra::grid.arrange(p1, p2, ncol = 2)

```

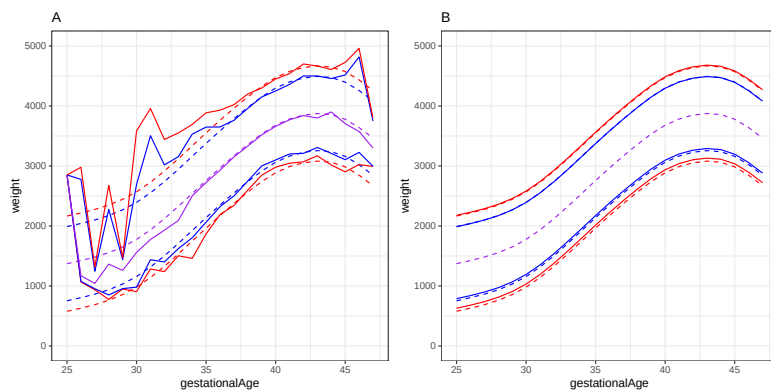


Figure 9.2: Medians (purple), upper and lower deciles (blue) and 5% and 95% quantiles (red). Plot A shows the quantiles based on normally distributed residuals (dashed) compared to the empirical quantiles. Plot B compares the quantiles based on normally distributed residuals (dashed) to the quantiles based on the empirical distribution of the residuals.

There are several other possibilities for computing an estimate of the lower decile (the 10% quantile), say, than the three methods presented above. We mention a couple here.

- Quantile regression via the check-loss.
- Spline expansions with optimized smoothness, e.g., via `smooth.spline()`.

Based on the regression model of the mean weight conditionally on the gestational age and the empirical distribution of the residuals, we construct a binary variable we call `small`, and which is an indicator of whether the child's weight is below the 10% quantile for the gestational age. The variable is thus indicating that the child is *small for gestational age*.

```
birth_weight <- mutate(
  birth_weight,
  q10 = predict(small_lm_nsg) + quantile(res, 0.10),
  small = as.numeric(weight < q10)
)
```

9.3 Binary regression

When the response variable can only take two possible values we refer to a corresponding regression model as a binary regression model. Usually we encode the two possible outcomes of a binary response variable as 0 (zero) and 1 (one)⁴⁰. This is convenient since for $Y \in \{0, 1\}$ it holds that the (conditional) probability that $Y = 1$ given $X = x$ can be written as

$$p(x) = \mathbf{P}(Y = 1 \mid X = x) = \mathbf{E}(Y \mid X = x).$$

⁴⁰ It is also common to encode the binary outcomes as ± 1 . If the two outcomes represent asymmetric numerical values, e.g., a gain and loss in a game, we can also consider using that encoding.

Thus $p(x)$ is the regression function, and we usually denote it by $p(x)$ rather than using the general notation $m(x)$. This is to indicate that $p(x)$ is a probability.

In the remaining part of this chapter, the response variable will be the binary indicator `small` that was constructed in the section above. The following two questions will drive our analysis:

- How strong is the association between the two lifestyle factors `coffee` and `smoking` and the indicator `small`?
- How good a predictive model of `small` can we fit to the data?

Seeking answers to these two questions will get us around many aspects of modeling a binary response variable using a regression model.

We note that we will not redo any exploratory data analysis in this chapter. All the predictor variables in the dataset are the same as in Chapter 3 and we refer to that chapter for an exploratory analysis of the distributions of the predictors.

9.3.1 Logistic regression

We will use all variables in the dataset except `weight`, `length` and `gestationalAge` as predictors of the indicator variable `small`. The first model we fit is a logistic regression model with all variables entering additively and with the numerical variables `age`, `alcohol` and `feverEpisodes` entering linearly (on the link scale). The logistic regression model is the model with a binomial response distribution and logit link function (the canonical link for the binomial model).

```
form <- small ~ age + children + coffee + alcohol +  
  smoking + abortions + feverEpisodes  
small_glm <- glm(form, data = birth_weight, family = "binomial")
```

We then consider various model diagnostic plots to investigate if the logistic regression model fits the data. Figure 9.3 shows four plots based on the Pearson residuals. Three plots show the residuals against the fitted values and two of the numerical predictors, and they do not suggest any problems with the mean value specification. In all three plots the smoothed curve

falls closely around zero, which indicates that the residuals approximately have mean zero.

```
small_glm_aug <- augment(small_glm, type.residuals = "pearson") |>
  mutate(.fitted_group = cut(.fitted, quantile(.fitted, probs = seq(5, 95, 1) / 100)))

p1 <- small_glm_aug |>
  ggplot(aes(.fitted, .resid)) +
  geom_point() +
  geom_smooth() +
  xlab("fitted values") +
  ylab("Pearson residuals")

p2 <- small_glm_aug |>
  ggplot(aes(age, .resid)) +
  geom_point() +
  geom_smooth() +
  ylab("Pearson residuals")

p3 <- small_glm_aug |>
  ggplot(aes(feverEpisodes, .resid)) +
  geom_point() +
  geom_smooth(method = "lm") +
  ylab("Pearson residuals")

p4 <- small_glm_aug |>
  group_by(.fitted_group) |>
  summarize(.fitted_local = mean(.fitted), .var_local = var(.resid)) |>
  na.omit() |>
  ggplot(aes(.fitted_local, .var_local)) +
  geom_point() +
  geom_hline(yintercept = 1, linetype = 2) +
  geom_smooth() +
  xlab("fitted values") +
  ylab("variance")

gridExtra::grid.arrange(p1, p2, p3, p4, ncol = 4)
```

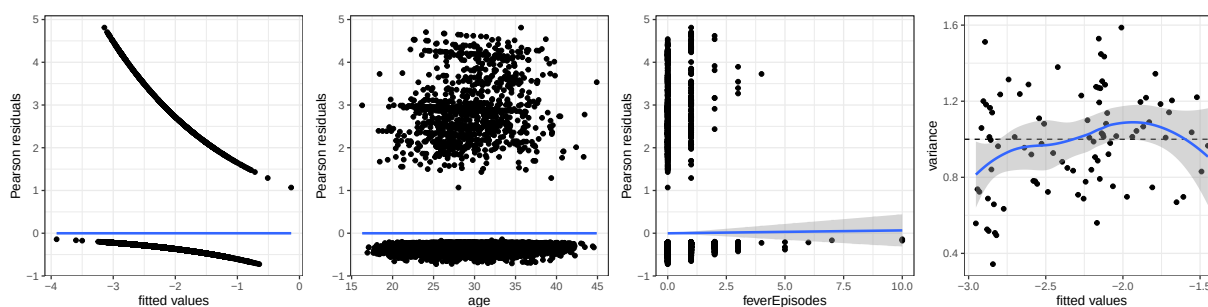


Figure 9.3: Diagnostic plots for the logistic regression model. The three leftmost plots shows Pearson residuals against fitted values and the two numeric predictors `age` and `feverEpisodes`. The rightmost plot shows local estimates of residual variance against fitted values.

What about the variance function and the dispersion parameter? Note first that since the response is binary, its (conditional) distribution is completely determined by $p(x)$ and its (conditional) variance equals

$$\mathbf{V}(Y \mid X = x) = p(x)(1 - p(x)).$$

Thus if the mean value model is correctly specified, the variance function for the binomial model,

$$\mathcal{V}(\mu) = \mu(1 - \mu),$$

is *automatically* correct, and the dispersion parameter equals $\varphi = 1$. For binary response variables there is no such phenomenon as “overdispersion” – but the mean value model can still be misspecified, which will then also lead to a misspecification of the variance.

To assess the variance specification, the rightmost plot in Figure 9.3 is constructed by locally estimating the variance of the Pearson residuals for groups of observations with similar fitted values. The plot shows that these variances all fall around one and there is no systematic dependence on the fitted values. Thus there is no indication of a model misspecification in terms of the variance either. Following the arguments above, this is not an independent assessment of whether the variance function is correct. It is rather a supplementary assessment of whether the mean value model is correct.

```
drop1(small_glm, test = "LRT") |>
  tidy() |>
  arrange(p.value, desc(LRT)) |>
```

```
filter(term != "<none>") |>
select(-AIC) |>
knitr::kable()
```

term	df	deviance	LRT	p.value
children	1	7099.384	110.6780196	0.0000000
smoking	2	7050.455	61.7488036	0.0000000
coffee	2	7014.290	25.5838028	0.0000028
alcohol	1	6991.293	2.5866851	0.1077657
abortions	3	6994.769	6.0627452	0.1085971
feverEpisodes	1	6990.928	2.2219953	0.1360571
age	1	6989.276	0.5697652	0.4503522

Table 9.1: Tests of models excluding each term from the full model one at a time using the deviance test (likelihood-ratio test).

To assess the significance of the included predictors, Table 9.1 shows the results from testing the parallel hypotheses where each variable is dropped from the model while all others remain in the model. The tests are based on the deviance test statistic, see Definition 5.2, and the p -values are computed using a χ^2 -approximation. We see from the table that the three predictors **children**, **smoking** and **coffee** cannot be removed from the model, while neither of the hypotheses for the remaining four variables can be rejected.

We refit the model below with a reordering of the variables according to their p -values in Table 9.1. This does not change the model but results in a more useful ordering of the output in various cases.

```
form <- small ~ children + smoking + coffee + alcohol +
  + abortions + feverEpisodes + age
small_glm <- glm(form, data = birth_weight, family = "binomial")
```

```
tidy(small_glm)[-1, "estimate"] |>
cbind(confint.lm(small_glm)[-1, ]) |>
knitr::kable(digits = 2)
```

	estimate	2.5 %	97.5 %
children1	-0.74	-0.87	-0.60
smoking2	0.46	0.29	0.62
smoking3	0.65	0.47	0.82
coffee2	0.28	0.14	0.42
coffee3	0.66	0.35	0.96
alcohol	0.05	-0.01	0.12
abortions1	-0.09	-0.28	0.10
abortions2	-0.20	-0.60	0.19
abortions3	0.58	0.05	1.11
feverEpisodes	-0.11	-0.25	0.04
age	0.01	-0.01	0.02

Table 9.2: Estimates and standard 0.95-confidence intervals for all parameters, except the intercept, in the logistic regression model.

Table 9.2 shows standard 0.95-confidence intervals for all parameters in the model. We note that these are generally in concordance with the conclusions from the tests above. That is, for they do not contain 0 for any of the parameters related to the variables **children**, **smoking** and **coffee**, while they do contain 0 for the remaining variables – except **abortions3** where the interval just misses 0 to the left.

What is perhaps more important is that Table 9.2 allows us to read of the signs and magnitudes of the associations. For the two variables we are primarily interested in, **smoking** and **coffee**, we see that the corresponding parameters are all positive. This means that compared to the baseline groups (not smoking and not drinking coffee) there is a *higher* risk that the child is small for gestational age if the mother smokes or drinks coffee. There is even an monotone relation because the parameters for groups **smoking3** and **coffee3** are larger than for **smoking2** and **coffee2**. The magnitudes of the associations are about the same for both variables.

Contrary to **smoking** and **coffee**, the variable **children** is negatively associated with the reponse, which means that if the mother had children before, there is a *lower* risk that the child will be small for gestational age.

It is notoriously difficult to interpret the magnitudes of the parameter estimates correctly in a logistic regression analysis.

The parameters are log-odds-ratios, which mean that their exponentials are odds-ratios. If we take an estimate like $\hat{\beta}_{\text{smoking2}} = 0.45$ and exponentiate it we get $e^{0.45} = 1.56$. Now suppose that the risk for a mother that does not smoke is p , then if the smoking status were 2 instead, the probability would change to $\tilde{p}$ satisfying

$$\frac{\tilde{p}}{1 - \tilde{p}} = 1.56 \cdot \frac{p}{1 - p}.$$

We can rewrite this relation as

$$\tilde{p} = \frac{1.56 \cdot p}{1 - (1 - 1.56) \cdot p} = \frac{1.56 \cdot p}{1 + 0.56 \cdot p}.$$

If p is small, the odds-ratio is roughly equal to the *risk ratio* ($\tilde{p}/p$). If $p = 0.1$, say, then $\tilde{p} = 0.1477$, which is roughly a factor 1.56 larger than p but not quite.

If we now compare a mother that neither smokes nor drinks coffee to one with both smoking and coffee status 3 (all other variables being equal) then the estimated odds-ratio equals $e^{0.64+0.62} = 3.53$ and the formula for $\tilde{p}$ becomes

$$\tilde{p} = \frac{3.53 \cdot p}{1 + 2.53 \cdot p}.$$

This changes $p = 0.1$ to $\tilde{p} = 0.28$ and is a notable increase of the risk. It is not quite a factor 3.5 but still about a factor 3.

Mothers that had children before has, on the other hand, a smaller risk of having a small child compared to first-time mothers. The odds-ratio is ($e^{-0.74} = 0.4771$), and the risk is thus reduced by roughly a factor 0.5 (for small probabilities).

9.3.2 Other binary regression models

The logistic regression model makes a specific choice about the link function. Setting the argument `family = "binomial"` when calling `glm()` specifies implicitly that the link function is the canonical link for the binomial distribution. That is, the link function is the logit function. There are a couple of standard alternative link functions used with the binomial model

$$\begin{aligned}\text{logit}(p) &= \log \frac{p}{1-p} \\ \text{probit}(p) &= \Phi^{-1}(p) \\ \text{cloglog}(p) &= \log(-\log(1-p))\end{aligned}$$

Here Φ denotes the distribution function for the standard normal distribution, and the cloglog function is known as the *complementary log-log* link. The corresponding mean value functions are

$$\begin{aligned}\mu(\eta) &= \frac{e^\eta}{1 + e^\eta} && (\text{logit}) \\ \mu(\eta) &= \Phi(\eta) && (\text{probit}) \\ \mu(\eta) &= 1 - e^{-e^\eta} && (\text{cloglog}).\end{aligned}$$

These three mean value functions are also distribution functions⁴¹, and one could, in principle, use any distribution function as a mean value function. The logit and cloglog links are particularly convenient because the link function and its inverse have simple analytic expressions. We note that when p is small, the logit and cloglog links are very similar, and both behave like the logarithm, which we also consider below as a possible link function.

The logarithm and the identity are occasionally also used as link functions. They do not map $\mathbb{R}$ into $(0, 1)$, which can lead to various issues – both numerical issues and issues with interpretations and applications. But they may sometimes be better choices if the effects of the predictors are better captured this way. If we want to use the identity link, we can consider using `lm()` instead of `glm(family = binomial(link = "identity"))`. The former will fit the model using least squares (and will ignore if probabilities are estimated to be negative or greater than one). The latter will fit the model using IWLS based on the variance function $\mathcal{V}(p) = p(1-p)$, and it will result in an error if the estimated probabilities fall outside of $(0, 1)$.

In addition to the logistic regression model above, we fit four models for different link functions. Note that for fitting the model with the identity link we use `lm()`, which thus uses a

⁴¹ The distribution function corresponding to the cloglog link is a Gumbel distribution reflected at 0.

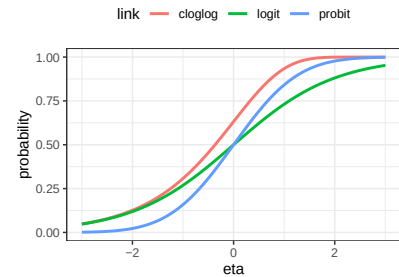


Figure 9.4: The mean value functions for three standard link functions.

misspecified variance function. However, it can potentially still fit a sensible mean value model, but since Assumption A2 does not hold, we should not trust results that hinge on this assumption, e.g., standard errors and p -values.

```
small_glm_prob <- glm(
  form,
  data = birth_weight,
  family = binomial(link = "probit")
)
small_glm_cloglog <- glm(
  form,
  data = birth_weight,
  family = binomial(link = "cloglog")
)
small_glm_log <- glm(
  form,
  data = birth_weight,
  family = binomial(link = "log")
)
small_lm <- lm(form, data = birth_weight)
```

```
tidy(small_glm_prob)[-1, "estimate"] |>
  cbind(confint.lm(small_glm_prob)[-1, ]) |>
  knitr::kable(digits = 2)
```

	estimate	2.5 %	97.5 %
children1	-0.38	-0.45	-0.31
smoking2	0.24	0.16	0.33
smoking3	0.35	0.25	0.44
coffee2	0.15	0.07	0.22
coffee3	0.34	0.18	0.51
alcohol	0.03	-0.01	0.06
abortions1	-0.04	-0.13	0.06
abortions2	-0.11	-0.30	0.09
abortions3	0.31	0.02	0.60
feverEpisodes	-0.05	-0.12	0.02
age	0.00	-0.01	0.01

Table 9.3: Estimates and standard 0.95-confidence intervals for all parameters, except the intercept, in the binary regression model using the probit link.

Table 9.3 shows the estimates and standard 0.95-confidence intervals for the model using the probit link. We observe that

the numerical values of the parameter estimates are different from those in Table 9.2, but the signs and magnitudes as well as the statistical conclusions we can draw from the confidence intervals are the same as for the logistic regression model. The picture is similar for the other models, though the parameter estimates are somewhat different for the identity link.

The interpretation of the magnitudes of the parameter estimates on these alternative link scales can be even more challenging than on the logit scale. For the probit scale there is no clear interpretation, but if we use the log link the parameters are log-relative-risks. As for the logit link, the complementary log-log link also behaves like the log link for small probabilities, in which case the parameters are also approximately log-relative-risks. If we use the identity link, the parameter directly represent additive changes of the probabilities.

To get a better idea about how much the fitted models differ for different choices of link function, we plot the fitted probabilities from all models against each other. Figure 9.5 shows the resulting scatter plot matrix. We observe that most points fall close to the diagonal all models agree to a very large extent on the fitted probabilities. This is also reflected by the correlations between the fitted probabilities that are all very close to one. The only model that differs a bit from the others is the one using the identity link. Compared to the other models it overestimates both the large and the small probabilities.

```
comp_fit <- function(data, mapping, ...) {
  ggplot(data, mapping, ...) +
    geom_abline(slope = 1, intercept = 0, color = "blue") +
    geom_point(..., alpha = 0.1, pch = 20) +
    coord_cartesian(xlim = c(0, 0.5), ylim = c(0, 0.5))
}

tibble(
  logit = fitted(small_glm),
  probit = fitted(small_glm_prob),
  cloglog = fitted(small_glm_cloglog),
  log = fitted(small_glm_log),
  identity = fitted(small_lm)
) |>
  GGally::ggpairs(
    lower = list(continuous = comp_fit),
    diag = list(continuous = "blankDiag")
  )
```

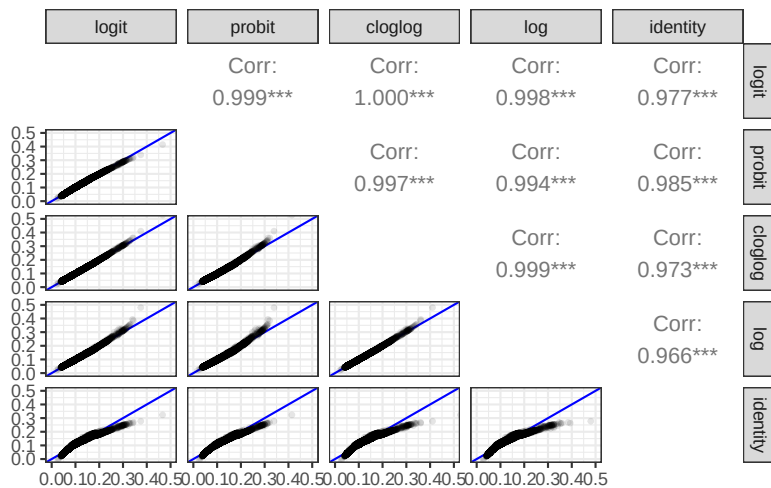


Figure 9.5: Comparisons of the fitted probabilities for the five different choices of link function. The model with the identity link is fitted using `lm()`.

In conclusion, the different choices of link function have limited influence on the probabilities that the models predict, though it will affect how the models are parametrized and the interpretations of the magnitudes of the individual parameters.

9.3.3 Calibration

When we make probabilistic predictions, a central question is whether events we predict to have probability p also happen with (roughly) probability p . We say that the predictions are *well calibrated* if that is true. For a binary regression model this means that $Y = 1$ happens in about $p(x)$ of the cases where $X = x$.

Figure 9.6 shows a plot of predicted probabilities of rain against observed probabilities for three different kinds of weather forecasting service. We see that on local TV the risk of rain is exaggerated for most of the scale of actual probabilities. The Weather Channel exaggerates the risk when the probabilities are small. For example, when they report a 20% risk of rain, it actually only rains in about 5% of the cases. However, from 30% and up, the The Weather Channel's predictions are well calibrated, and The National Weather Service makes well calibrated probabilistic predictions on the entire scale.

It is not obvious how to produce a calibration plot for a regression model, since we typically have at most one observation

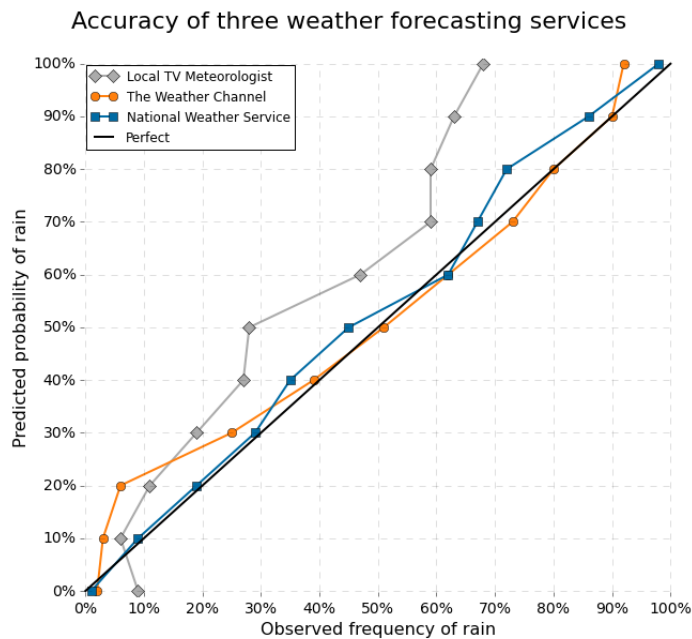


Figure 9.6: Calibration plot (flipped) showing how well (or poorly) calibrated probabilistic predictions of rain are. Source: “The Signal and the Noise” by Nate Silver.

with exactly $X = x$ for any particular x . The standard technique is therefore to group together observations that have about the same predicted probability and then compute the relative frequency of the outcome $Y = 1$ within each group. We can then make a *calibration plot* where we plot these observed probabilities against the predicted probabilities. The points should fall roughly on the diagonal for the predictions to be well calibrated. Figure 9.6 is flipped compared to how calibration plots are usually made.

To construct a calibration plot we first implement a small function for computing the predicted and observed probabilities based on a model. The function includes the computation of standard errors on the observed probabilities, reflecting the number of observations in each group.

```
group_p <- function(y, p, n_probs = 100) {
  qt <- quantile(p, probs = seq(0, n_probs, 1) / n_probs)
  tibble(
    y = y,
    p = p,
    p_group = cut(p, qt, include.lowest = TRUE)
  ) |>
  group_by(p_group) |>
  summarize(
    p_local = mean(p),
```

```

      obs_local = mean(y),
      se_local = sqrt(p_local * (1 - p_local) / n()),
    )
  }

```

We then implement the generic ggplot code for making the calibration plot.

```

cal_plot <- ggplot(mapping = aes(p_local, obs_local)) +
  geom_ribbon(aes(
    ymin = p_local - 1.96 * se_local,
    ymax = p_local + 1.96 * se_local
  ), fill = "orange", alpha = 0.1) +
  geom_point() +
  geom_abline(slope = 1, intercept = 0, linetype = 2, color = "orange") +
  geom_smooth() +
  xlab("predicted probability") +
  ylab("observed probability")

```

```

gridExtra::grid.arrange(
  cal_plot %>% group_p(birth_weight$small, fitted(small_glm), n_probs = 50) +
    coord_cartesian(xlim = c(0, 0.35), ylim = c(0, 0.35)),
  cal_plot %>% group_p(birth_weight$small, fitted(small_glm), n_probs = 100) +
    coord_cartesian(xlim = c(0, 0.35), ylim = c(0, 0.35)),
  ncol = 2
)

```

Warning: <ggplot> %>% x was deprecated in ggplot2 4.0.0.
i Please use <ggplot> + x instead.

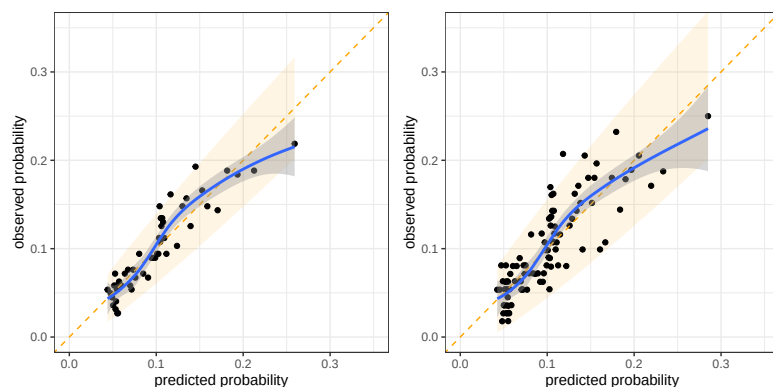


Figure 9.7: Calibration plots for the logistic regression model using 50 (left) and 100 (right) quantiles for grouping.

Figure 9.7 shows the calibration plot for the logistic regression model. The predictions are not poorly calibrated, but they are not perfect either. The predicted probabilities slightly

underpredicts the observed probabilities in the range between 0.1 and 0.2.

One way to potentially improve the model is to include interactions. The model below includes all first order interactions among all variables.

```
form_int <- small ~ (age + children + coffee + alcohol +  
  smoking + abortions + feverEpisodes)^2  
small_glm_int <- glm(form_int, data = birth_weight, family = "binomial")
```

```
cal_plot %>% group_p(birth_weight$small, fitted(small_glm_int)) +  
  coord_cartesian(xlim = c(0, 0.35), ylim = c(0, 0.35))
```

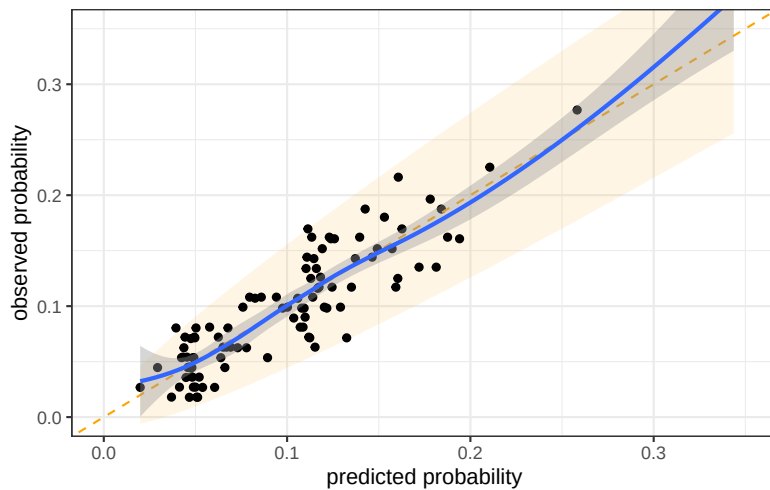


Figure 9.8: Calibration plot for the logistic regression model including all first order interaction terms.

The calibration plot for the interaction model, Figure 9.8, shows that the interaction model appears well calibrated. But there is a caveat here. The interaction model is a more complex model, and the model is fitted by maximizing the likelihood, which is an attempt to make the model well calibrated. Thus we should expect a more complex model to *appear* better calibrated on the training data than a less complex one – just as the training error underestimates the generalization error for a complex model.

The solution to the optimism of the calibration plot on training data is again to use data splitting. Here we implement a simple K -fold cross-validation procedure with just K rounds. It returns the predicted values on the probability scale.


```

cv <- function(form, data, K = 10) {
  n <- nrow(data)
  response <- all.vars(form)[1]
  mu_hat <- numeric(n)
  group <- sample(rep(1:K, length.out = n))
  for(i in 1:K) {
    model <- glm(form, data = data[group != i, ], family = "binomial")
    mu_hat[group == i] <-
      predict(model, newdata = data[group == i, ], type = "response")
  }
  mu_hat
}

```

```

p1 <- cal_plot %>% group_p(birth_weight$small, cv(form, birth_weight)) +
  coord_cartesian(xlim = c(0, 0.35), ylim = c(0, 0.35)) + ggtitle("Additive model")
p2 <- cal_plot %>% group_p(birth_weight$small, cv(form_int, birth_weight)) +
  coord_cartesian(xlim = c(0, 0.35), ylim = c(0, 0.35)) + ggtitle("Interaction model")
gridExtra::grid.arrange(p1, p2, ncol = 2)

```

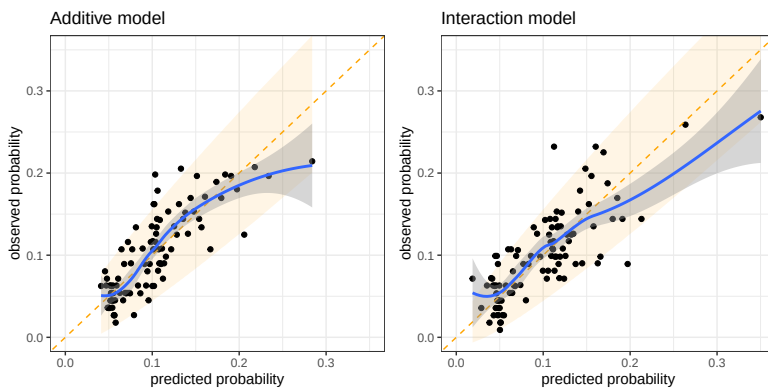


Figure 9.9: Cross-validated calibration plots.

Figure 9.9 does not change our perspective on the calibration of the additive model (without interaction terms). The cross-validated calibration plot for that model resembles Figure 9.7. The cross-validated calibration plot for the interaction model in Figure 9.9 is, however, not as convincing as Figure 9.8. It seems that the interaction model underpredicts the small probabilities and overpredicts the large probabilities. There are random fluctuations in the plot due to the random splits in the cross-validation procedure, but repeated runs result in calibration plots with similar overall structure.

9.3.4 Measuring and optimizing predictive performance

In this section we investigate if the interaction model actually gives better predictions than the additive model. Before doing that, we will also formerly test the interaction model against the additive model using the deviance test statistic.

```
anova(
  small_glm,
  small_glm_int,
  test = "LRT"
) |> knitr::kable()
```

Resid. Df	Resid. Dev	Df	Deviance	Pr(>Chi)
11140	6988.706	NA	NA	NA
11090	6907.355	50	81.35137	0.0033427

Table 9.4: Likelihood-ratio test of the interaction model against the additive model.

Table 9.4 shows that we can reject the hypothesis of the additive model. Thus there is evidence for some interactions. Table 9.5 shows which of the different terms are significant.

```
drop1(small_glm_int, test = "LRT") |>
  tidy() |>
  arrange(p.value, desc(LRT)) |>
  filter(term != "<none>") |>
  select(-c(deviance, AIC)) |>
  knitr::kable(digits = 3)
```

term	df	LRT	p.value
children:smoking	2	18.402	0.000
children:alcohol	1	10.128	0.001
coffee:feverEpisodes	2	10.854	0.004
abortions:feverEpisodes	3	6.662	0.084
alcohol:smoking	2	4.674	0.097
coffee:smoking	4	5.571	0.234
alcohol:abortions	3	3.090	0.378
children:abortions	3	2.931	0.402
coffee:abortions	6	5.896	0.435

term	df	LRT	p.value
children:coffee	2	0.992	0.609
age:feverEpisodes	1	0.198	0.656
coffee:alcohol	2	0.722	0.697
children:feverEpisodes	1	0.118	0.731
smoking:abortions	6	3.558	0.736
age:smoking	2	0.440	0.803
age:children	1	0.055	0.814
alcohol:feverEpisodes	1	0.052	0.820
age:abortions	3	0.435	0.933
age:alcohol	1	0.001	0.979
age:coffee	2	0.025	0.988
smoking:feverEpisodes	2	0.019	0.991

Table 9.5: Tests of models excluding each interaction term from the full interaction model one at a time using the deviance test (likelihood-ratio test).

We will now turn to measuring the predictive performance of the models using the Pearson and deviance loss functions and the AUC concordance parameter. First we implement a function for computing and AUC estimate using the `wilcox.test()` function.

```
auc <- function(y, eta) {
  eta1 <- eta[y == 1]
  eta0 <- eta[y == 0]
  wilcox.test(eta1, eta0)$statistic / (length(eta1) * length(eta0))
}
```

Then we compute the training errors for the two models.

```
err <- tibble(
  model = c("Additive (train)", "Interaction (train)"),
  Pearson = c(
    residuals(small_glm, type = "pearson")^2 |> mean(),
    residuals(small_glm_int, type = "pearson")^2 |> mean()
  ),
  Deviance = c(
    residuals(small_glm)^2 |> mean(),
    residuals(small_glm_int)^2 |> mean()
  ),
  AUC = c(
    auc(birth_weight$small, predict(small_glm)),
    auc(birth_weight$small, predict(small_glm_int))
  )
)
```

```
)
)
err
```

```
# A tibble: 2 x 4
  model      Pearson Deviance  AUC
  <chr>      <dbl>    <dbl> <dbl>
1 Additive (train)  0.992    0.627 0.644
2 Interaction (train) 1.00    0.619 0.656
```

Unsurprisingly, the more complex interaction model has (slightly) smaller training deviance and (slightly) larger training AUC. Recall that the AUC reflects how much the conditional distributions of the linear predictors given $Y = 1$ and $Y = 0$, respectively, agree. Figure 9.10 shows these distributions, and it is clear that there is quite a large overlap, which is also reflected in the rather small value of (training) AUC for both models.

```
p1 <- augment(small_glm) |>
  ggplot(aes(.fitted, fill = factor(small))) +
  geom_density(alpha = 0.6, color = NA) +
  scale_fill_brewer("small", palette = "Set1") +
  ggtitle("Additive model") +
  coord_cartesian(xlim = c(-5, 0), ylim = c(0, 1.4))

p2 <- augment(small_glm_int) |>
  ggplot(aes(.fitted, fill = factor(small))) +
  geom_density(alpha = 0.6, color = NA) +
  scale_fill_brewer("small", palette = "Set1") +
  ggtitle("Interaction model") +
  coord_cartesian(xlim = c(-5, 0), ylim = c(0, 1.4))

gridExtra::grid.arrange(p1, p2, ncol = 1)
```

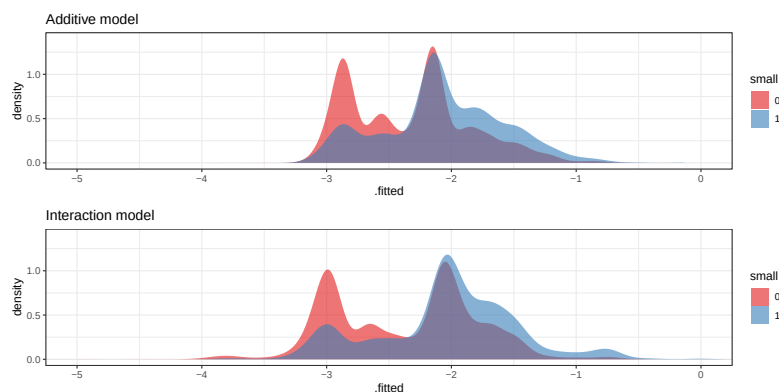


Figure 9.10: Distribution of linear predictors for the additive model and the interaction model stratified according to the value of the response variable.

```

pearson <- function(y, p) (y - p)^2 / (p * (1 - p))
deviance <- function(y, p) ifelse(y == 1, -2 * log(p), -2 * log(1 - p))
Err <- tibble(
  model = c("Additive (CV)", "Interaction (CV)"),
  Pearson = c(
    pearson(birth_weight$small, cv(form, birth_weight)) |> mean(),
    pearson(birth_weight$small, cv(form_int, birth_weight)) |> mean()
  ),
  Deviance = c(
    deviance(birth_weight$small, cv(form, birth_weight)) |> mean(),
    deviance(birth_weight$small, cv(form_int, birth_weight)) |> mean()
  ),
  AUC = c(
    auc(birth_weight$small, cv(form, birth_weight)),
    auc(birth_weight$small, cv(form_int, birth_weight))
  )
)
rbind(err, Err)

```

```

# A tibble: 4 x 4
  model          Pearson Deviance  AUC
  <chr>          <dbl>    <dbl> <dbl>
1 Additive (train)  0.992    0.627 0.644
2 Interaction (train) 1.00    0.619 0.656
3 Additive (CV)    1.00    0.629 0.637
4 Interaction (CV)  1.09    0.646 0.623

```

The expected generalization errors, estimated above using cross-validation, show that the training errors were all optimistic. The two models become almost identical in terms of both deviance and in terms of AUC.

9.3.4.1 GLMNET

Explanations of what glmnet does are outstanding.

```

X <- model.matrix(small_glm_int)[, -1]
small_glmnet <- cv.glmnet(
  x = X,
  y = birth_weight$small,
  family = "binomial"
)

```

```

plot(small_glmnet)
abline(h = 0.644, col = "blue")

```

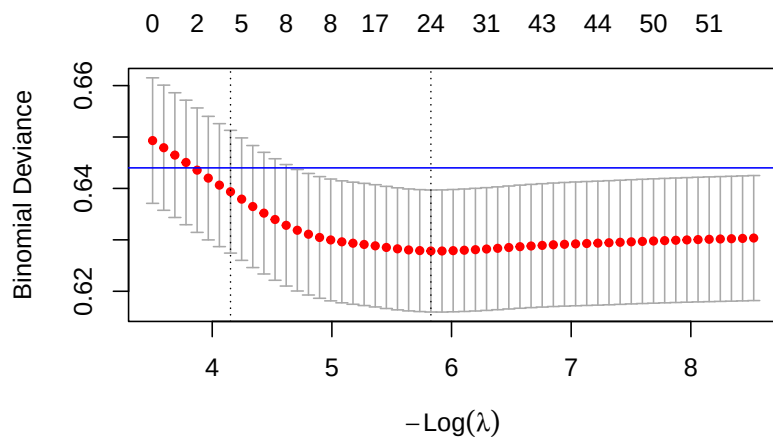


Figure 9.11: Cross-validated deviance loss for a range of penalty parameters

```
tibble(
  glm_add = predict(small_glm, type = "response"),
  glm_int = predict(small_glm_int, type = "response"),
  glmnet = drop(predict(small_glmnet, newx = X, s = "lambda.min", type = "response"))
) |>
  GGally::ggpairs(
    lower = list(continuous = comp_fit),
    diag = list(continuous = "blankDiag")
  )
```

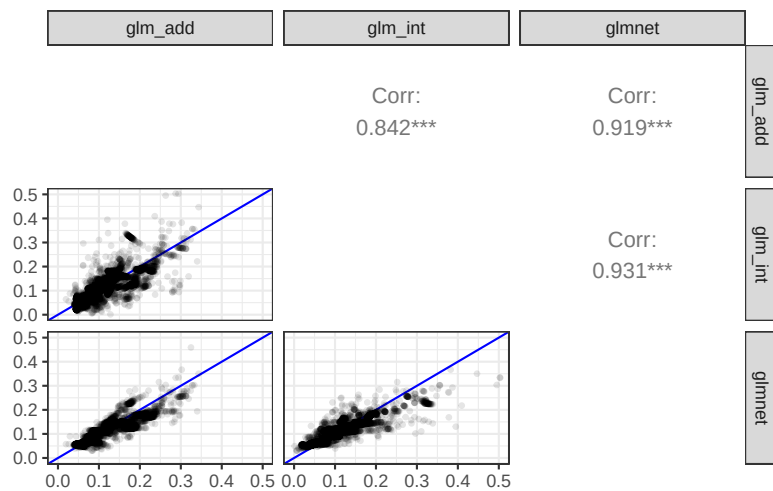


Figure 9.12: Comparisons of the predicted probabilities for the additive model, the interaction model and the model fitted with glmnet that minimizes the cross-validated deviance loss.

Exercises

10 Advertising – a case study

Warning

This chapter is still a draft and minor changes are made without notice.

In this chapter we consider data extracted from a large database from a major supermarket chain. The primary goal of the supermarket is to predict the number of items sold in specific stores in those weeks where the chain runs an advertising campaign, which consists of a combination of advertisements and discounts. A secondary goal is to assess the effect of the discount and the advertising campaign.

In this chapter we consider the sales of one particular item: frozen vegetables. In reality, the objective would be to build a model for all the items in each store. We focus here on building a model for the sale of frozen vegetables based on data from a few weeks and a selection of stores in the supermarket chain in Sweden.

10.1 Outline and prerequisites

The chapter primarily relies on the following R packages.

```
library(ggplot2)
library(tibble)
library(broom)
library(splines)
library(readr)
library(skimr)
library(dplyr)
```

10.2 Exploratory data analysis

We load the data from the `RwR` package.

```
data(vegetables, package = "RwR")
```

The 7 variables and their encoding in the `vegetables` data set:

- **sale**: Total number of sold items.
- **normalsale**: A proxy of the normal sale in the same week.
- **store**: The id-number of the store.
- **ad**: Advertising (0 = no advertising, 1 = advertising).
- **discount**: Discount in percent.
- **discountSEK**: Discount in Swedish kroner.
- **week**: Week number (2, 4, 5, 7, 8, 9).

```
skim(vegetables)
```

Variable type: factor										
skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts					
store	0	1	FALSE	352	1: 4, 102: 4, 106: 4, 107: 4					
ad	0	1	FALSE	2	0: 687, 1: 379					
Variable type: numeric										
skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
sale	0	1.00	40.29	58.44	1.00	12.00	21.00	40.00	571.00	
normalSale	0	1.00	11.72	13.83	0.20	4.20	7.25	12.25	102.00	
discount	26	0.98	37.47	8.39	12.06	29.86	37.05	46.04	46.04	
discountSEK	26	0.98	9.94	2.81	2.40	7.50	10.30	12.80	12.80	
week	0	1.00	6.95	1.76	2.00	7.00	7.00	8.00	9.00	

Table 10.1: Quick data summary using the `skim()` function from the `skimr` package.

From the summary of the data in Table 10.1 we note that there are 26 missing values of **discount** and **discountSEK**. A further investigation shows that it is the same 26 cases for which values are missing for both variables. Moreover, 25 of these cases are the 25 cases from week 2. The last case is from week 4. The cross tabulation of **week** and **ad** in Table 10.2 shows that for all other cases, advertising is either 1 or 0 for all stores within each week except for one case in week 4.


```
table(vegetables[, c("week", "ad")]) |> knitr::kable()
```

	0	1
2	25	0
4	1	164
5	0	44
7	344	0
8	317	0
9	0	171

Table 10.2: Cross tabulation of week number and advertising indicator.

The only case in week 4 registered as no advertising is also the one where the value of `discount` and `discountSEK` is missing. We believe this to be an error in the database.

We have to decide what we do with the 26 cases with missing values. For the 25 cases from week 2 we have no data to support an imputation of the missing values. We could attempt to impute the discount (and correct `ad`, since we believe it to be registered wrongly) in the last case. We choose, however, to remove all 26 cases with missing values.

We will also reorder the levels for the factor `store`. The default ordering is the lexicographical order of the factor levels, which in this case amounts to a meaningless and arbitrary ordering. Instead, we order the levels according to the mean normal sale of the stores over the observed weeks.

```
# Data cleaning
vegetables <- filter(vegetables, !is.na(discount), week != 2)
normal_sale <-
  group_by(vegetables, store) |>
    summarize(
      meanNormSale = mean(normalSale)
    ) |>
    arrange(meanNormSale)

store_order <- as.character(normal_sale$store)

vegetables <- left_join(vegetables, normal_sale) |>
  mutate(
    store = factor(store, levels = store_order)
  )
```

number of weeks	frequency
1	28
2	113
3	58
4	153

Table 10.3: Frequency of weeks that a store is included.

The categorical variable **store** represents the total of 352 stores. As Table 10.3 show, not all stores are represented each week. Some stores (28) only appear once, while 153 stores appear in a total of four of the five weeks.

The marginal distributions of **sale** and **normalSale** are seen in Figure 10.1. It shows that the number of items sold is larger than the normal sale in a distributional sense, which is not surprising given that we are considering the sale in weeks with a discount. Note that we have log-transformed the variables because their distributions are quite right skewed.

```
p1 <- ggplot(vegetables, aes(sale)) +
  geom_histogram(fill = gray(0.5), bins = 40) +
  scale_x_log10() +
  coord_cartesian(xlim = c(1, 600)) +
  xlab("") + ylab("") + ggtitle("sale")
p2 <- ggplot(vegetables, aes(normalSale)) +
  geom_density(fill = gray(0.5)) +
  scale_x_log10() +
  coord_cartesian(xlim = c(1, 600)) +
  xlab("") + ylab("") + ggtitle("normalSale")
gridExtra::grid.arrange(p1, p2)
```

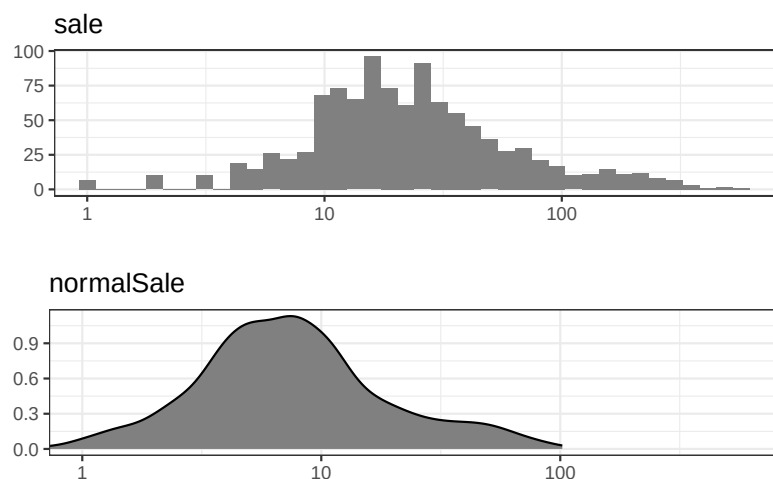


Figure 10.1: Histogram of **sale** and density estimate of **normalSale**. Note the log-scale on the x -axis.

The distributions of the two categorical variables **ad** and **week** are given in the summary above, and the distributions of the remaining two variables **discount** and **discountSEK** are given as barplots in Figure 10.2. We note a quite skewed distribution of discounts with around one-third of the observations corresponding to the largest discount.

```
p1 <- ggplot(vegetables, aes(discount)) +
  geom_histogram(fill = gray(0.5), bins = 40) +
  xlab("") + ylab("") + ggtitle("Discount")
p2 <- ggplot(vegetables, aes(discountSEK)) +
  geom_histogram(fill = gray(0.5), bins = 40) +
  xlab("") + ylab("") + ggtitle("DiscountSEK")
gridExtra::grid.arrange(p1, p2)
```

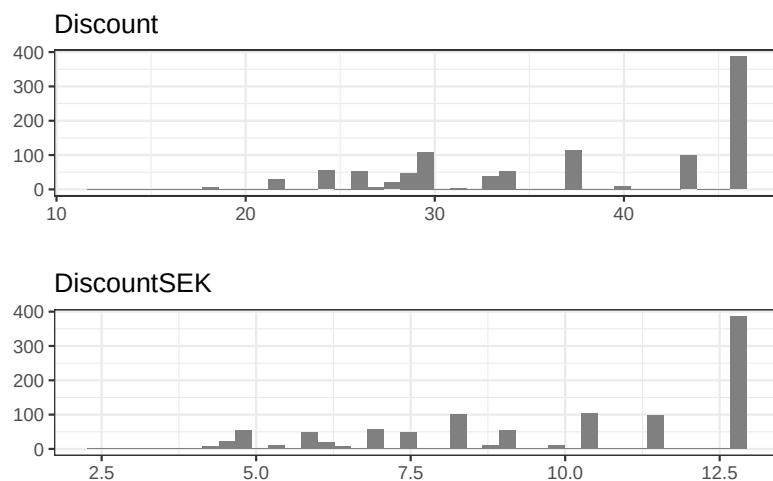


Figure 10.2: Barplots of discount variables.

10.2.1 Pairwise associations

We consider scatter plots and Pearson correlations between the 4 continuous variables. As for the histograms we log-transform `sale` and `normalSale`.

```
transform(vegetables, log_sale = log10(sale), log_normalSale = log10(normalSale)) |>
GGally::ggpairs(
  columns = c("log_sale", "log_normalSale", "discount", "discountSEK"),
  lower = list(continuous = GGally::wrap("points", shape = 16, size = 2, alpha = 0.1)),
  diag = list(continuous = "blankDiag")
)
```

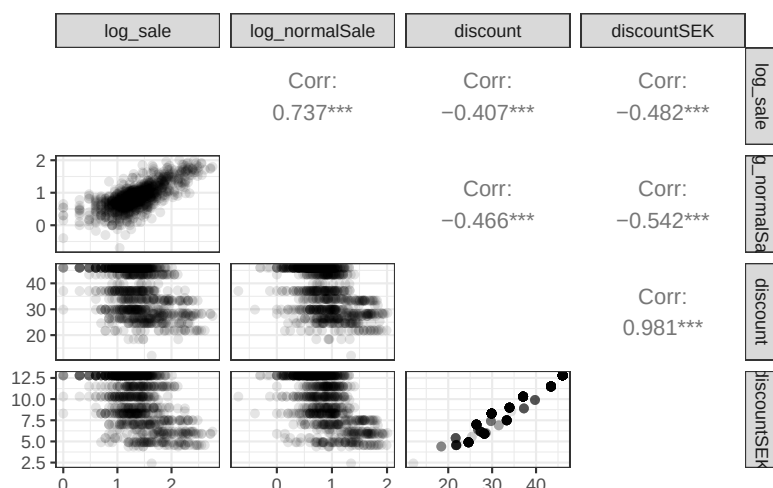


Figure 10.3: Scatter plot matrix of the 4 continuous variables and the corresponding Pearson correlations.

The scatter plot matrix, Figure 10.3, shows that `normalsale` and `sale` are quite strongly correlated. As one should expect, `discount` and `discountSEK` are also extremely correlated and close to being collinear. It should also be observed that the discount variables and the total sale are strongly negatively correlated. However, as the figure shows, the `discount` variables are negatively correlated with the normal sale as well. Thus the negative marginal correlation may simply reflect that stores with a larger sale is only present with a smaller discount. Figure 10.4 further shows how the `discount` and `normalSale` distributions change depending upon the week number and the advertising variable.

```
deciles <- function(x) {
  quan <- quantile(x, c(0.1, 0.5, 0.9))
  data.frame(ymin = quan[1], y = quan[2], ymax = quan[3])
}

vegetables_long <- select(vegetables, c(discount, normalSale, ad, week)) |>
  transform(week = as.factor(week)) |>
  tidyr::pivot_longer(cols = c(ad, week))

p1 <- ggplot(vegetables_long, aes(x = value, y = normalSale)) +
  geom_violin(scale = 'width', fill = I(gray(0.8))) +
  scale_y_log10() +
  stat_summary(fun.data = deciles, color = "blue") + xlab("") +
  facet_wrap(~ name, scale = "free_x")
```

```
p2 <- ggplot(vegetables_long, aes(x = value, y = discount)) +
  geom_violin(scale = 'width', adjust = 2, fill = I(gray(0.8))) +
  stat_summary(fun.data = deciles, color = "blue") + xlab("") +
  facet_wrap(~ name, scale = "free_x")
gridExtra::grid.arrange(p1, p2, ncol = 1)
```

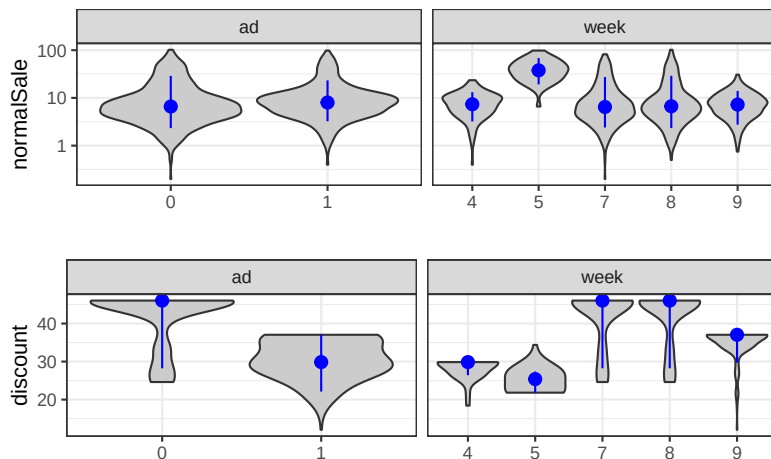


Figure 10.4: Violin plots, medians and interdecile ranges for the distribution of `normalSale` and `discount` stratified according to `ad` and `week`. A violin plot is a rotated kernel density estimate and can be seen as an alternative to a boxplot.

The data consist of observations from different stores (of different sizes) in different weeks, and as mentioned above we don't have observations from the same stores for all weeks. Since the advertising campaigns run for a given week, we don't have observations from all stores for all combinations of campaigns. In fact, Figure 10.5 shows that stores with a large normal sale are only included in weeks 5, 7 and 8, and that a considerable number of stores are only included in weeks 7 and 8.

```
ggplot(vegetables, aes(
  x = store,
  ymin = week - 0.3,
  ymax = week + 0.3,
  group = store,
  color = meanNormSale
)) +
  geom_linerange() +
  coord_flip() +
  scale_x_discrete(breaks = c()) +
  scale_y_continuous("week", breaks = c(4, 5, 6, 7, 8, 9)) +
  theme(legend.position = "top") +
  scale_color_continuous(
    "Mean normal sale",
    guide = guide_colorbar(title.position = "top"),
```

```
trans = "sqrt"
)
```

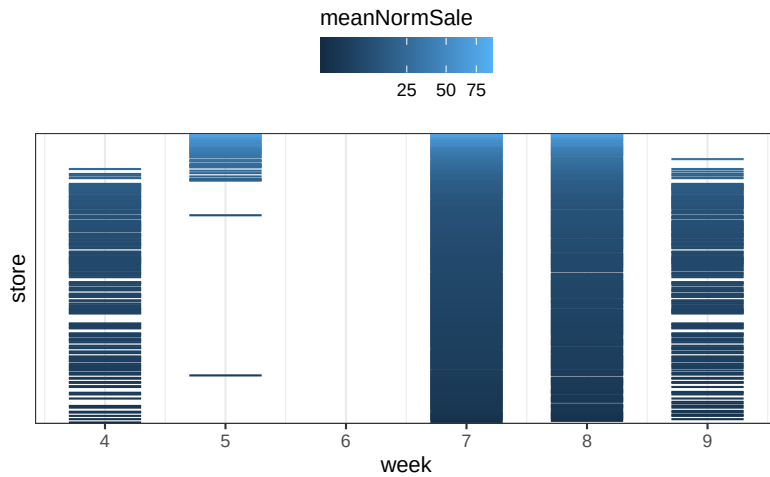


Figure 10.5: Illustration of which stores we have observations from for each of the weeks. The stores are ordered according to the mean normal sale.

To further illustrate this point we consider the relation between the stores and the discount. Figure 10.6 shows that for stores with a large normal sale we only have observations with moderate discounts. This can explain why we observe a negative marginal correlation between the discount and the sale as well as the normal sale.

```
ggplot(vegetables, aes(discount, store, group = store, color = meanNormSale)) +
  geom_point() +
  scale_y_discrete(breaks = c()) +
  theme(legend.position = "top") +
  scale_color_continuous(guide = "none", trans = "sqrt")
```

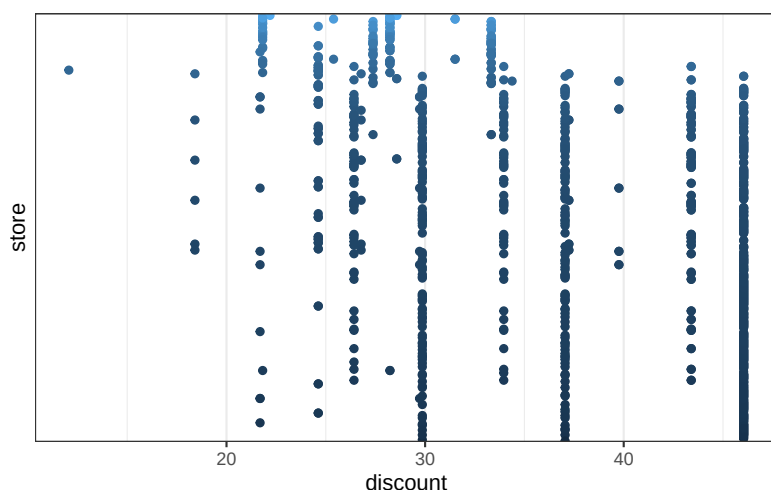


Figure 10.6: Illustration of the relation between the stores and the discount.

10.3 A Poisson regression model

The supermarket chain has an interest in a predictive model on several levels. We will focus on predictions for the individual store of how many items the store can expect to sell in a week. The predictive model will be based on the store's normal sale in the week in combination with campaign variables. All observations are from weeks with a discount on the item, but the discounts differ between the weeks.

The response variable Y is the number of sold items in a given week and in a given store. The proxy variable `normalSale` is based on historical data and provides, for the given store and week, an estimate of the expected number of sold items without taking advertising campaigns into account. We let N denote the normal sale. The normal sale is clearly predictive for the number of sold items, as shown by the pairwise association we found in the exploratory analysis. We will attempt to build a model of how to adjust the normal sale with the information on the advertising campaign.

We will consider multiplicative models of the form

$$\mathbf{E}(Y \mid N, X) = Ne^{X^T\beta} = e^{\log(N)+X^T\beta} \quad (10.1)$$

where X is the vector of additional predictors besides the normal sale. This is a log-linear model, that is, a model

with an exponential mean value function (a logarithmic link function). This is the default (and canonical) link function for the Poisson response distribution when using the `glm()` function in R. Since `sale` is an integer count, it may also be plausible that we can model it with a Poisson distribution.

In the log-linear model above the predictor $\log(N)$ has a fixed coefficient. Fixing the coefficient of a term in the linear predictor is achieved using the `offset()` function in the formula specification of the model.

As the very first thing we investigate the marginal association of each of the predictors `ad`, `discount`, `discountSEK` and `store` using the Poisson regression model that includes the normal sale as an offset. The significance of each predictor is assessed formally by a likelihood ratio test (LRT), that is, a test based on the deviance test statistic, see Definition 5.2.

```
form <- sale ~ ad + discount + discountSEK + store
null_model <- glm(
  sale ~ offset(log(normalSale)),
  family = poisson,
  data = vegetables
)
one_term_models <- add1(null_model, form, test = "LRT")
```

term	df	LRT	p.value
store	351	8137.699090	0.0000000
discountSEK	1	8.873470	0.0028934
ad	1	7.276940	0.0069845
discount	1	3.252817	0.0713008

Table 10.4: Marginal association tests sorted according to the p -value.

Table 10.4 shows the results of testing each of the predictors individually with a fixed control for the normal sale. As we see from the table, the predictors are marginally significantly related to the number of sold items, though `discount` is borderline. This conclusion rests on some model assumptions, in particular the relation between mean and variance that is determined by the Poisson model. Below, we find evidence against this relation in a more general model, and we should

not trust the absolute magnitude of the p -values for these marginal association tests.

Note that `week` was not included as a predictor. First of all, it would be of limited use in a predictive model, if we can only predict for the weeks included in the data set. Second, the value of `ad` is completely determined by `week`, and including both would result in perfect collinearity of the `ad` column in the design matrix with the columns representing `week`. The `week` variable is, however, useful for subsequent model diagnostics.

The next thing we consider is a main effects model including the four predictors linearly and additively. We should remember that `discount` and `discountSEK` are highly correlated, and we should expect the collinearity phenomenon that neither of them are significantly related to the response when the other is included.

A main effects model. We could, of course, in this and in subsequent

```
form <- sale ~ offset(log(normalSale)) + store + ad + discount + discountSEK
vegetables_glm <- glm(
  form,
  family = poisson,
  data = vegetables
)
```

We do not want to consider all the estimated parameters for the individual stores – only the parameters for the other variables. Table 10.5 shows that in this model the relations between `sale` and all three variables `ad`, `discount` and `discountSEK` appear to be significant, but somewhat surprisingly, the estimated coefficient of `discount` is negative. This counter intuitive result makes us suspicious about the model. The collinearity might explain a negative estimate, but it should not be significant then.

```
vegetables_glm |>
  tidy() |>
  tail(3) |>
  knitr::kable()
```

term	estimate	std.error	statistic	p.value
ad1	0.3173135	0.0321609	9.866444	0.00e+00
discount	-0.0769348	0.0174726	-4.403175	1.07e-05

term	estimate	std.error	statistic	p.value
discountSEK	0.4164590	0.0598074	6.963339	0.00e+00

Table 10.5: Summary table of parameter estimates, standard errors and z -tests for the poisson model of sale with 4 predictors.

10.4 Improvements of the Poisson model

The first analysis of the advertising data above resulted in a log-linear Poisson regression model. We should be somewhat suspicious of this model, and before we consider any extensions we will investigate the model fit. For this purpose we first extract the residuals (and other quantities) using the `augment()` function from the broom package. It will by default extract the deviance residuals, but here we first extract the Pearson residuals, which we immediately use to estimate the dispersion parameter, cf. (5.6).

```
vegetables_glm_diag <- augment(
  vegetables_glm,
  type.residuals = "pearson"
)
# Estimated dispersion parameter
vegetables_glm_diag$.resid^2 |> sum() / 685
```

```
[1] 11.77101
```

We see that $\hat{\varphi}$ is 11.77, which is notably greater than 1. This is indicating that the Poisson regression model does not fit the data, since for the Poisson model the dispersion parameter is fixed and equal to 1. When modeling count data it is a common phenomenon, known as *over dispersion*, that the estimated dispersion parameter is greater than 1. This could be due to model misspecification. Either the log-linear model of the mean is wrong, or the variance function $\mathcal{V}(\mu) = \mu$ entailed by the Poisson model is wrong. However, it can happen that the log-linear model fits the data reasonable well but that there is overdispersion for other reasons, e.g., unobserved heterogeneity or clustering, see Exercise 4.9.

It is possible to adjust the statistical analysis for overdispersion via the quasi Poisson family. The quasi Poisson family does not correspond to an actual exponential dispersion model on the integers – there is no exponential dispersion model on the integers with mean μ and variance $\varphi\mu$ for all dispersion parameters $\varphi > 0$, see Exercise 4.8. The quasi family is just a way to specify a mean-variance relation that allows for a dispersion parameter. We fit the same main effects Poisson regression model as above, but using the quasi Poisson family. We then extract the estimated dispersion parameter via the `summary()` function – an estimate that is identical to the one we computed above via the Pearson residuals.

```
vegetables_quasi_glm <- glm(
  form,
  family = quasipoisson,
  data = vegetables
)
summary(vegetables_quasi_glm)$dispersion
```

```
[1] 11.77102
```

Table 10.6 shows the parameter estimates using the quasi Poisson model for the same three parameters as in Table 10.5. We note that the actual estimates are identical⁴², but standard errors and p -values are larger – reflecting that the estimated dispersion parameter is greater than 1. In fact, the `discount` and `discountSEK` parameters no longer appear to be significantly different from 0. This aligns with our previous observation that these two variables are almost collinear.

⁴² This is unsurprising as the dispersion parameter does not affect the score equation for generalized linear models.

```
tidy(vegetables_quasi_glm) |> tail(3) |> knitr::kable()
```

Table 10.6: xxx

term	estimate	std.error	statistic	p.value
ad1	0.3173135	0.1103405	2.875766	0.0041557
discount	-0.0769348	0.0599465	-1.283391	0.1997891
discountSEK	0.4164590	0.2051926	2.029600	0.0427833

To further investigate the model fit we turn to residual plots.

```
p1 <- ggplot(vegetables_glm_diag, aes(.fitted, .resid)) +
  geom_point(alpha = 0.3) +
  geom_smooth(linewidth = 1, fill = "blue") +
  xlab("fitted values") + ylab("Pearson residuals")
p2 <- ggplot(vegetables_glm_diag, aes(.fitted, sqrt(abs(.resid)))) +
  geom_point(alpha = 0.3) +
  geom_smooth(linewidth = 1, fill = "blue") +
  xlab("fitted values") + ylab("sqrt(|Pearson residuals|)")
gridExtra::grid.arrange(p1, p2, ncol = 2)
```

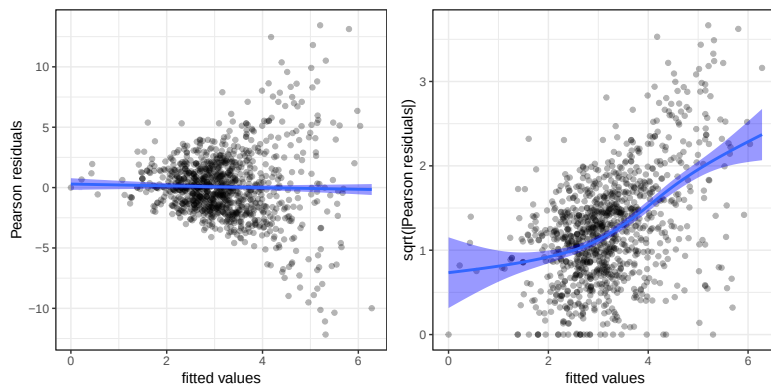


Figure 10.7: Diagnostic plots. Pearson residuals and the square root of the absolute value of the Pearson residuals plotted against fitted values.

Figure 10.7 shows three things. First, the residuals are generally centered around zero, and there is no clear indication that the mean value is misspecified. Second, as we already know, there is a clear overdispersion in the model corresponding to the Poisson distribution (the residuals are too large). Third, the linear relation between mean and variance, that the Poisson model dictates, is not appropriate. This is revealed both in the funnel shape in Figure 10.7 of the ordinary residual plot (left) and in the non-constant relation between the square root of the absolute value of the residuals as a function of the fitted values (right).

We should note that the “fitted values” in Figure 10.7 are the linear predictors. This is the default scale used by `augment()`, and it is known as the *link scale*. If we need the fitted mean values we can either transform the fitted values from the link scale by the mean value function. In this example by the exponential function. We can alternatively set argument `type.predict = "response"` in `augment()`, which will give the fitted values on the *response scale*. For making residual plots it is typically best to use the link scale.

The residuals plot above were based on the Pearson residuals.

We can also consider the deviance residuals.

```
vegetables_glm_diag2 <- augment(vegetables_glm)
p1 <- ggplot(vegetables_glm_diag2, aes(.fitted, .resid)) +
  geom_point(alpha = 0.3) +
  geom_smooth(linewidth = 1, fill = "blue") +
  xlab("fitted values") + ylab("deviance residuals")
p2 <- ggplot(vegetables_glm_diag2, aes(.fitted, sqrt(abs(.resid)))) +
  geom_point(alpha = 0.3) +
  geom_smooth(linewidth = 1, fill = "blue") +
  xlab("fitted values") + ylab("sqrt(|deviance residuals|)")
gridExtra::grid.arrange(p1, p2, ncol = 2)
```

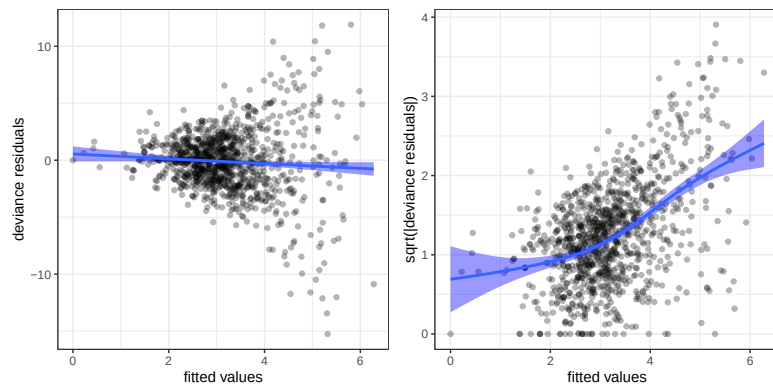


Figure 10.8: Diagnostic plots. Deviance residuals and the square root of the absolute value of the deviance residuals plotted against fitted values.

Figure 10.8 is quite similar to Figure 10.7, and the conclusions are the same. The mean-variance relation given by the variance function $\mathcal{V}(\mu) = \mu$ does not fit the data, and we need a variance function that increases more rapidly.

The Gamma model has variance function $\mathcal{V}(\mu) = \mu^2$, and we will therefore try that model, but still with a log-link function. For the subsequent models we will also exclude `discountSEK` due to its collinearity with `discount`. We fit two models. One where the predictors enter linearly and additively and an additive model with a spline basis expansion of `discount` using natural cubic splines with 3 internal knots.

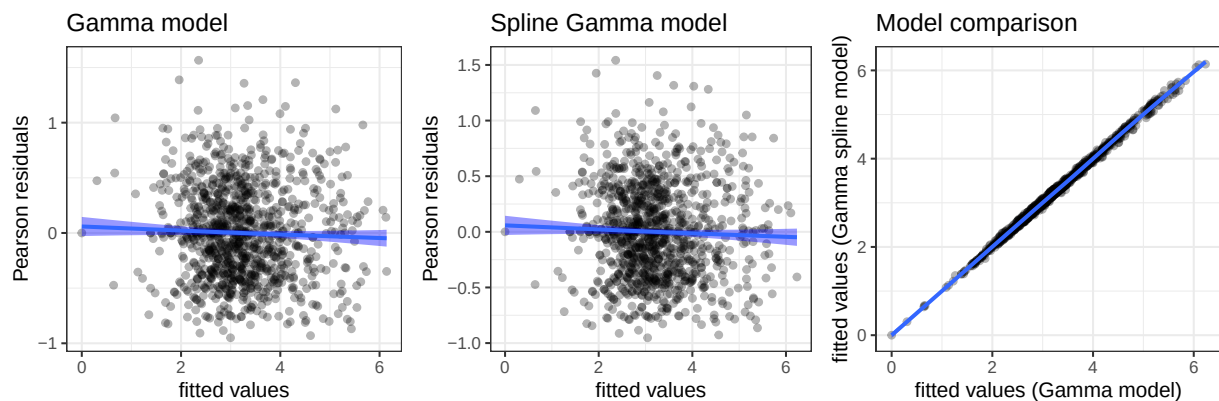
```
form <- sale ~ offset(log(normalSale)) + store + ad + discount
vegetables_gamma_glm <- glm(
  form,
  family = Gamma("log"),
  data = vegetables
)
form <- sale ~ offset(log(normalSale)) + store + ad +
  ns(discount, knots = c(20, 30, 40), Boundary.knots = c(0, 50))
```

```

vegetables_gamma_glm_spline <- glm(
  form,
  family = Gamma("log"),
  data = vegetables
)

```

Figure 10.9 shows several diagnostic plots. It is clear from these plots that both Γ -models fit the data well – also in terms of the mean-variance relation. The figure additionally shows the fitted values from the two models plotted against each other. The points all fall very close to the diagonal, which shows that the two models are almost identical in terms of predictions. That is, the inclusion of the nonlinear spline expansion does not change the predicted values by very much.



Now that we have developed models that fits the data reasonably well, it makes more sense to consider formal tests. The decision to include the nonlinear expansion of `discount` can, for instance, be justified by a likelihood ratio test. In the glm jargon the likelihood ratio test statistic is known as the deviance, and it is computed as the difference in deviance between the two (nested) models.

```

anova(
  vegetables_gamma_glm,
  vegetables_gamma_glm_spline,
  test = "LRT"
) |> tidy() |> knitr::kable()

```

term	df.residual	deviance	df	deviance	p.value
sale ~ offset(log(normalSale)) + store + ad + discount	686	229.6964	NA	NA	NA
sale ~ offset(log(normalSale)) + store + ad + ns(discount, knots = c(20, 30, 40), Boundary.knots = c(0, 50))	683	227.2337	3	2.4627	0.293166

Table 10.7: Test of the Gamma model including a nonlinear expansion of `discount` against a linear effects model.

To conclude the analysis illustrate predictions from the model. Since there are many stores we need to summarize the model somehow. We do this by choosing a few stores representing different coefficients. One of these, store 206, has regression coefficient -3.1, which is the median coefficient, and we refer to this store as the *typical store*. For this store we present predictions and confidence bands corresponding to a normal sale of one item.

```

pred_frame <- expand_grid(
  normalSale = 1,
  store = factor(c(91, 84, 66, 206, 342, 256, 357)),
  ad = factor(c(0, 1)),
  discount = seq(10, 50, 1)
)

pred_sale <- predict(
  vegetables_gamma_glm_spline,
  newdata = pred_frame,
  se.fit = TRUE
)

pred_sale0 <- predict(
  vegetables_gamma_glm,
  newdata = pred_frame,
  se.fit = TRUE
)

pred_frame0 <- cbind(pred_frame, as_tibble(pred_sale0))
pred_frame <- cbind(pred_frame, as_tibble(pred_sale))

p1 <- filter(pred_frame, store == 206) |>

```



```

ggplot(aes(discount, exp(fit))) +
  geom_line() +
  ylab("sale") +
  geom_ribbon(
    aes(
      ymin = exp(fit - 2 * se.fit),
      ymax = exp(fit + 2 * se.fit)
    ),
    alpha = 0.3
  ) +
  geom_line(data = filter(pred_frame0, store == 206), col = "blue", linetype = 2) +
  facet_grid(cols = vars(ad), label = label_both) +
  coord_cartesian(ylim = c(0, 10)) +
  scale_y_continuous(breaks = c(1, 3, 5, 7, 9))

p2 <- ggplot(pred_frame, aes(discount, fit, color = store)) +
  geom_line() +
  geom_line(data = pred_frame0, linetype = 2) +
  ylab("sale") +
  facet_grid(cols = vars(ad), label = label_both) +
  scale_y_continuous("log-sale")

gridExtra::grid.arrange(p1, p2, ncol = 2)

```

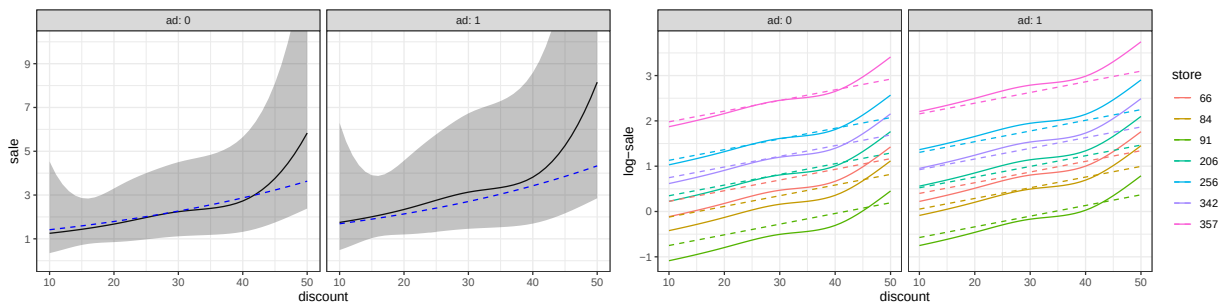


Figure 10.10: Predictions and confidence bands for the spline model of sale per normal sale for one typical store (206), and predictions for 7 stores spanning the range of the different stores. Dashed lines show predictions from the linear model with no spline expansion.

As Figure 10.10 shows, the model predicts that the sale will increase by a factor larger than 1 (larger than 0 on a log-scale) for many stores. The factor is increased if the campaign includes advertising. The model predicts a nonlinear relation between the discount (in percent) and the increase in sale. However, the confidence bands (for the median store) show that the predictions are estimated with a large uncertainty. The differences to the predictions from the linear model (dashed lines in Figure 10.10) are small, and even if we could justify the nonlinear model by the formal likelihood

ratio test, the actual differences are small and uncertain.

Exercises

Exercise 10.1. This exercise uses the data set `carInsurance` from the R package `RwR`. The data set contains data on insurance claims cross-tabulated according to the three variables `Age`, `District` and `Motor`:

- `Age` is an ordered categorical variable specifying the age range in years of the policy holder. The possible values are 25 (less than 25 years), 25–29, 30–35 and 35 (more than 35 years).
- `District` is categorical taking the values A, B, C and D. (D is major cities).
- `Motor` is an ordered categorical variable specifying the size of the motor in liters. The possible values are 1 (less than a 1l motor), 1–1.5, 1.5–2 and 2 (more than a 2l motor).

The data set consists of the number of claims Y_i and the total number N_i of policy holders for the i -th group in addition to the three categorical variables.

We will consider models of the form $Y_i \sim \text{Pois}(\mu_i)$ with

$$\mu_i = N_i e^{X_i^T \beta},$$

where X_i is a vector of predictors for the i -th group derived from the categorical variables. In this model,

$$p_i = \frac{\mu_i}{N_i} = e^{X_i^T \beta}$$

is the *proportion* of claims in the i -th group. We are interested in a model of these proportions.

The second order interaction model, which has the symbolic description

`Age * District * Motor`

is the saturated model where each group has its own mean, and thus its own proportion.

1. Plot the estimates of the proportions of claims in the saturated model against `Motor` stratified according to `Age` and `District`. Add confidence intervals to the point estimates.

Hint: Such a plot is most easily produced using `ggplot()`. Confidence intervals can be added with `geom_errorbar()`. Note that the estimates and confidence intervals can be computed using elementary computations as well as using `glm`.

2. Fit the model with additive main effects of the three categorical variables. The additive main effects model has the symbolic description

`Age + District + Motor`.

Investigate the model fit and compare it with the saturated model.

Hint: The investigation can include, but is not limited to, a test of the additive main effects model against the saturated model. In the next question you are asked to investigate the distribution of the deviance test statistic, but for this question, you can rely on standard approximations.

3. Investigate using simulations under the fitted additive model if the deviance test statistic for the additive main effects model against the saturated model follows the approximate χ^2 -distribution that the theory suggests.

Exercise 10.2. This exercise is on the relation between mean and variance for counts of small water organisms (Polychaeta). The dataset `polychaeta` from the R package `RwR` contains counts from samples taken at different stations in Odense Fjord.

The data set contains the integer response variable `Count` (which is, in fact, inferred from the weight of the total number of organisms) and the variable `Station` (numbered 1 to 45, and with between 7 and 48 observations per station).

It can be assumed throughout that the response is given as

$$Y_i = \mu_i + \varepsilon_i$$

with $\mathbf{E}(\varepsilon_i) = 0$, $\mathbf{V}(\varepsilon_i) = \varphi\mathcal{V}(\mu_i)$ and

$$\mu_i = \beta_{\text{Station}_i}$$

being a station specific mean value. Note that even if the response is integer valued, the observed counts are relatively large, hence it does make sense to use, e.g., a normal distribution as response distribution.

1. Estimate the station specific mean values and investigate if the variance can be assumed constant. That is, investigate if the variance function can be assumed to be $\mathcal{V}(\mu) = 1$.

Taylor's Law is an empirical law established in ecology. It states that there is a power law relationship between mean and variance when counting organisms. Specifically, that the variance function is given as

$$\mathcal{V}(\mu) = \mu^b$$

for some power $b \geq 0$. The objective of the following is to estimate b . To this end introduce $\xi_i = \mathbf{E}(\varepsilon_i^2)$.

2. Show that if Taylor's law holds then

$$\log(\xi_i) = \log(\varphi) + b \log \mu_i.$$

Show that if ε_i is normally distributed then $\mathbf{V}(\varepsilon_i^2) = 2\xi_i^2$.

3. Show that Taylor's law in general implies that

$$\mathbf{V}(\varepsilon_i^2) = \xi_i^2(\text{kurt}_i - 1)$$

provided that ε_i has finite fourth moment, and with kurt_i denoting the kurtosis⁴³ of ε_i .

4. Still assuming that Taylor's law holds use an appropriate generalized linear model to estimate the parameter b for the Polychaeta counts by using the squared residuals

$$\hat{\varepsilon}_i^2 = (Y_i - \hat{\mu}_i)^2$$

as responses. Check if Taylor's law holds for the Polychaeta counts and compute a 95% confidence interval for b .

Exercise 10.3. Consider the generalization of (10.1) where the logarithm of the normal sale does not have a fixed coefficient:

$$\log(\mathbf{E}(Y \mid N, X)) = \gamma \log(N) + X^T \beta.$$

⁴³ The convention is that kurtosis, denoted by kurt , is the *fourth standardized moment*, while $\text{kurt} - 3$ is *excess kurtosis*, which is 0 for the normal distribution. This convention concurs with Wikipedia.

The residuals are used as approximations of the ε_i -s since the latter are not observed.

That is, it does not enter as an offset in the formula specifications of the generalized linear models. Fit such models to the vegetables data and compare the resulting models to the Poisson or Gamma models considered throughout the chapter.

11 Survival analysis

Warning

This chapter is still a draft and minor changes are made without notice. In particular, a few practical examples will be added in Sections 11.4 and 11.5.

This chapter deals with regression models for time-to-event data with possible right censoring, which is common for survival data but also in other applications of statistics⁴⁴ such as reliability theory and credit risk.

Some of the notation in this chapter will deviate a bit from the notation used for other regression models, which is deliberate and to follow some of the standard conventions in the survival literature. The response variable will, for example, be denoted T instead of Y , which is also a natural notation for a time-to-event observation. The regression models are, however, still of the form, where a linear predictor $\eta = X^T\beta$ is formed from the vector of predictors $X \in \mathbb{R}^p$ and a parameter $\beta \in \mathbb{R}^p$, and the effect of the predictors on the survival distribution is then only through η .

⁴⁴ Time to failure of an electronic component, say, or time default of a debtor.

11.1 Outline and prerequisites

Section 11.2 is covers the basic definitions specific to survival analysis and motivates why survival (regression) analysis is special. It also covers the basic nonparametric estimators of survival distributions. Section 11.3 covers parametric survival models and likelihood based methods for estimation and statistical inference. The accelerated failure time models and the proportional hazard model are introduced, which show two different ways that the linear predictor can affect the survival distribution. Section 11.4 primarily treats Cox's proportional

hazard model, which is the most widely used survival regression model. As the name suggests, it is a proportional hazard model, but the methodology is *semiparametric*, meaning in this case that a baseline hazard function is left unconstrained. Section 11.5 introduces survival residuals, which differ a bit from residuals considered for linear and generalized linear models. These differences reflect that survival models don't target conditional expectations, but otherwise the purposes of the residuals for model diagnostics are the same.

The chapter primarily relies on the following R packages.

```
library(RwR)
library(ggplot2)
library(tibble)
library(dplyr)
library(tidyr)
library(skimr)
library(broom)
library(lubridate)
library(survival)
library(splines)
```

11.2 Introduction to survival analysis

A typical application of survival analysis is to the study of survival of patients after an initial diagnosis. In such a study, patients are enrolled whenever they are diagnosed with a given (serious, life threatening) disease. We refer to the time of enrollment⁴⁵ as the *baseline*, and we note that the baseline for different patients can be different calendar times, and that the survival time is the time from baseline to death. The baseline is thus time zero on the time axis of the survival time.

The patients are followed for some period of time, and if they die during this followup period, their survival time is registered. At a planned calendar time the study ends and the status of the patients are registered, that is, we register if they died, in which case we know their survival time, or if they are still alive. All patients alive at the end of the study have *right censored* survival times. This form of censoring is sometimes called *administrative censoring*. Some patients might also have dropped out of the study earlier, because they

⁴⁵ For some applications, all individuals or units can have the same baseline, e.g., in an experiment on the longevity of electronic components, but in many applications the baseline calendar times differ among units. In credit risk models of time to default, say, the baseline could be the date of the loan application.

moved or didn't want to participate anymore, say. At the time they drop out, we say that they are *lost-to-followup*, and this also results in right censoring of their survival time.

In addition to the, possibly right censored, survival times of the patients, we may collect additional data on the patients. We can then use some or all of such additional variables to construct a regression model of the survival distributions. But it is not clear that we want to model *expected* survival times, and the conditional expectation is often not⁴⁶ the correct target of a survival regression model.

⁴⁶ One reason is technical. Expectations rely on the tails of distributions, and with right censoring we don't get that information.

Regression models for survival data thus differ from the linear and generalized linear regression models in at least the following two ways:

- There is almost always a *censoring mechanism*, and certain aspects of the data are consequently missing. We need to deal with this in the modeling.
- Survival distributions are skewed distributions on the positive half line. It is often the *shape* of the distribution rather than the location of the distribution that is of interest.

The overall objective of a survival regression model is the same as for any regression model: how are the different predictors associated with the survival time. We might then have different more specific questions we want to answer. For example, to determine if the survival distributions depends on a choice of treatment (testing). Or we might want to quantify how strongly a predictor is associated with survival response.

We might, of course, also be interested in prediction, which we often call *survival prognosis*. It is rarely of interest to make a point prediction of a survival time for an individual, but we might report a prediction interval (the interquartile range, say), or we might report probabilistic predictions in terms of an entire (conditional) survival distribution.

Example 11.1. We will throughout this chapter use a dataset on prostate cancer survival, which is included in the `RwR` package as the dataset `prostate`. The data is from a randomized clinical trial on prostate cancer patients with four different treatments (one placebo and three different dosages of estrogen). There are 502 patients in the trial, and we will in this chapter focus on four of the variables.

- **rx**: Treatment encoded as a factor.
- **status**: Either the cause of death or indication that the patient was alive at the end of study.
- **dtime**: The censored survival time since baseline in months.
- **age**: The baseline age of the patient in years.

There are several other variables included in the dataset, and among these are **sdate**: the date that the patient entered the study. We will not use **sdate** for modeling, but it is used in Figure 11.1 to visualize the data. For later purposes, the data is converted to a tibble.

```
data(prostate, package = "RwR") # version >= 0.3.5
prostate <- prostate |> as_tibble()
```

```
# A tibble: 502 x 5
  rx          status      dtime  age sdate
<fct>      <fct>      <dbl> <dbl> <date>
1 0.2 mg estrogen alive      72   75 1977-08-10
2 0.2 mg estrogen dead - other ca      1   54 1977-09-21
3 5.0 mg estrogen dead - cerebrovascular 40   69 1978-01-12
4 0.2 mg estrogen dead - cerebrovascular 20   75 1978-03-19
5 placebo      alive      65   67 1978-03-22
6 0.2 mg estrogen dead - prostatic ca  24   71 1978-06-14
7 placebo      dead - heart or vascular 46   75 1978-06-27
8 placebo      alive      62   73 1978-06-28
9 1.0 mg estrogen alive      61   60 1978-08-01
10 1.0 mg estrogen alive      60   78 1978-08-07
# i 492 more rows
```

Figure 11.1 illustrates the data for the first 100 patients that enter the study. The figure shows their entry date (baseline), their survival time from that date and the status indicator when follow-up ends, indicating if they are dead or if the survival time is censored.

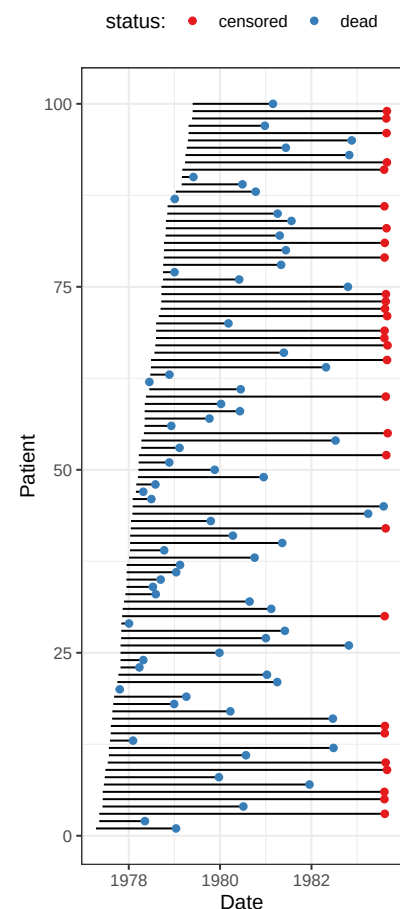


Figure 11.1: The entry date of the 100 first prostate cancer patients into the study, their follow-up time and their status at end of follow-up.

Since the treatments were randomly assigned, differences in the survival distributions between the four treatment groups can be ascribed to the treatment alone. That is, the treatments can be interpreted as the *causes* of any such differences. We will first of all be interested in testing if there are effects of treatment as well as estimating the magnitudes of such effects.

There are other variables included in the dataset, such as age, and we could use these variables for two different purposes. We could use the variables for prognostic purposes, that is, for fitting regression models that give more accurate survival prognoses for individual patients. Another purpose could be to investigate *heterogeneity* of treatment effects, that is, to investigate if the treatment works differently on different subgroups of the patients.

11.2.1 Survival distributions and censoring

A survival time T^* is simply a non-negative random variable. We will denote its distribution function by F , and if the distribution has density f w.r.t. Lebesgue measure, we have

$$F(t) = \mathbf{P}(T^* \leq t) = \int_0^t f(s)ds.$$

The corresponding *survival function* of T^* is

$$S(t) := 1 - F(t) = \mathbf{P}(T^* > t) = \int_t^\infty f(s)ds.$$

In terms of S we define the *cumulative hazard function* as

$$\Lambda(t) := -\log(S(t)), \quad (11.1)$$

so that

$$S(t) = \exp(-\Lambda(t)). \quad (11.2)$$

If S is continuously differentiable, $-S'(t) = f(t)$ is the density of the survival distribution, and the *hazard function* is defined as

$$\lambda(t) := -(\log S(t))' = \frac{f(t)}{S(t)}. \quad (11.3)$$

There are one-to-one relations between the distribution function, the survival function and the cumulative hazard function.

We can observe that

$$\begin{aligned}\lambda(t) &= \lim_{\varepsilon \rightarrow 0+} \frac{1}{\varepsilon} \frac{F(t+\varepsilon) - F(t)}{S(t)} \\ &= \lim_{\varepsilon \rightarrow 0+} \frac{1}{\varepsilon} \mathbf{P}(T^* \in (t, t+\varepsilon] | T^* > t),\end{aligned}$$

and $\lambda(t)$ thus has the interpretation as the instantaneous rate of death at time t given survival beyond time t . We also note that

$$\Lambda(t) = \int_0^t \lambda(s) ds = -\log S(t)$$

which explains why Λ is called the *cumulative* hazard function.

We will not observe T^* but only

$$T = \min\{T^*, C\} \quad \text{and} \quad \delta = 1(T^* \leq C)$$

for a *censoring variable* C . We call T the *right censored* survival time and δ the *status indicator*. The status indicator is 1 if the T is uncensored and 0 otherwise. To distinguish the distribution of C clearly from the distribution of the survival time, we will generally denote its survival function by H and its density by h (if it exists). That is,

$$H(t) = \mathbf{P}(C > t) = \int_t^\infty h(s) ds.$$

11.2.2 Model assumptions

We will generally consider n observations and let $T_1^*, \dots, T_n^*$ denote the survival times and $C_1, \dots, C_n$ the censoring times. We then observe the right censored survival times

$$T_i = \min\{T_i^*, C_i\}$$

and the status indicators

$$\delta_i = 1(T_i^* \leq C_i)$$

for $i = 1, \dots, n$.

The vector of predictor variables is as elsewhere denoted $X_i \in \mathbb{R}^p$, and we are generally interested in models of the conditional distributions⁴⁷

⁴⁷ Note that the target is the conditional distribution of the unobserved and uncensored T_i^* , not the observed and censored T_i .

$$T_i^* \mid X_i = x.$$

An issue of particular concern for survival analysis is *when* a predictor variable is observed. In some studies, some potential predictor variables are monitored throughout the study. Such variables as BMI and blood pressure might be recorded on a monthly basis, say. It is possible to develop a general time-to-event theory where predictors also change with time, but it is both technically beyond this chapter, and it is also conceptually challenging to understand and interpret such models correctly.

We will throughout assume that all predictors are observed⁴⁸ at or before baseline, and we refer to such predictors as *baseline predictors*.

⁴⁸ Or, as a minimum, observable at time zero.

The modeling assumptions across the different individuals are now straightforward modifications of A5 or A6 to the survival setup.

SA5: The conditional distribution of (T_i^*, C_i) given $\mathbf{X}$ depends only on X_i and $(T_1^*, C_1), \dots, (T_n^*, C_n)$ are conditionally independent given $\mathbf{X}$.

SA6: The triples $(T_1^*, C_1, X_1), \dots, (T_n^*, C_n, X_n)$ are i.i.d.

Since (T_i, δ_i) is a function of (T_i^*, C_i) , then either assumption implies the same distributional statement about the (T_i, δ_i) -s as about the (T_i^*, C_i) -s.

An additional delicate survival modeling question is what to assume about the joint distribution of T_i^* and C_i (possibly, conditionally on X_i). Throughout, we will make the following assumption.

Independent censoring: T_i^* and C_i are conditionally independent given X_i .

Even though the assumption above is a *conditional* independence statement, it is known simply as “independent censoring”. In the literature, “independent censoring” may refer to a slightly more permissive assumption (Definition III.2.1 in Andersen et al. (1993)), stating that information about censoring status at any given timepoint is not allowed to affect the residual survival distribution beyond that timepoint. For this chapter we will make the stronger, but simpler, independent censoring assumption as stated above.

Andersen, Per Kragh, Ørnulf Borgan, Richard D. Gill, and Niels Keiding. 1993. *Statistical Models Based on Counting Processes*. Springer Series in Statistics. Springer-Verlag.

11.2.3 The Kaplan-Meier and Nelson-Aalen estimators

If $T_1^*, \dots, T_n^*$ are i.i.d. and if they were observed, we could define the empirical survival function

$$\hat{S}_n^*(t) = \frac{1}{n} \sum_{i=1}^n 1(T_i^* > t)$$

as a nonparametric estimator of S . Clearly, we cannot compute $\hat{S}_n^*$ if we only have censored observations. The purpose of this section is to introduce two nonparametric estimators of a survival distribution when the survival times are i.i.d. and independently censored⁴⁹.

To handle the censoring we introduce the *individuals at risk* at time t :

$$Y(t) = \sum_{i=1}^n 1(t \leq T_i).$$

It can be computed based on the censored survival times, and it counts how many individuals have survived to time t and are uncensored at that time.

Definition 11.1 below gives the general definition of the *Kaplan-Meier* estimator of the survival function, but to motivate this definition, we first suppose that there are no ties among the observed survival times, that is, where we have strict inequalities among the ordered survival times

$$T_{(1)} < T_{(2)} < \dots < T_{(n-1)} < T_{(n)}.$$

Here $T_{(k)}$ denotes the survival time with rank k , and we will denote the corresponding status indicator by $\delta_{(i)}$.

When there are no ties, and $T_{(k)} \leq t < T_{(k+1)}$, we can factorize the survival function as

$$\begin{aligned} S(t) &= \mathbf{P}(T^* > t) \\ &= \mathbf{P}(T^* > t \mid T^* \geq T_{(k)}) \times \mathbf{P}(T^* \geq T_{(k)} \mid T^* \geq T_{(k-1)}) \\ &\quad \dots \times \mathbf{P}(T^* \geq T_{(2)} \mid T^* \geq T_{(1)}) \times \mathbf{P}(T^* \geq T_{(1)}) \end{aligned}$$

where $\mathbf{P}(T^* \geq T_{(i+1)} \mid T^* \geq T_{(i)})$ is the conditional probability of surviving from $T_{(i)}$ to $T_{(i+1)}$ given survival beyond time $T_{(i)}$.

⁴⁹ In this section there are no predictors, and we effectively assume that $(T_1^*, C_1), \dots, (T_n^*, C_n)$ are i.i.d. and that T_i^* and C_i are independent.

Without censoring, the Kaplan-Meier estimator equals $\hat{S}^*$, cf. Exercise 11.2

Since $Y(T_{(i)})$ are the number of individuals at risk⁵⁰ at time $T_{(i)}$ and $Y(T_{(i)}) - \delta_{(i)}$ are the number of individuals that die in the interval $[T_{(i)}, T_{(i+1)})$, we can regard

⁵⁰ Alive and *uncensored*.

$$\frac{Y(T_{(i)}) - \delta_{(i)}}{Y(T_{(i)})} = 1 - \frac{\delta_{(i)}}{Y(T_{(i)})}$$

as an estimator of the conditional probability $\mathbf{P}(T^* \geq T_{(i+1)} \mid T^* \geq T_{(i)})$. This gives the *Kaplan-Meier* estimator for $t \in [T_{(k)}, T_{(k+1)})$

$$\hat{S}(t) = \left(1 - \frac{\delta_{(k)}}{Y(T_{(k)})}\right) \cdots \left(1 - \frac{\delta_{(2)}}{Y(T_{(2)})}\right) \left(1 - \frac{\delta_{(1)}}{Y(T_{(1)})}\right).$$

We can write the Kaplan-Meier estimator in a slightly more abstract notation as

$$\hat{S}(t) = \prod_{i: T_i \leq t} \left(1 - \frac{\delta_i}{Y(T_i)}\right). \quad (11.4)$$

To define the Kaplan-Meier estimator that allows for multiple deaths at the same time (ties), we introduce the *counting process* of deaths (non-censored events)

$$N(t) = \sum_{i=1}^n 1(T_i \leq t, \delta_i = 1),$$

and the jumps for the counting process given as

$$\Delta N(t) = N(t) - N(t-) = N(t) - \lim_{\varepsilon \rightarrow 0+} N(t - \varepsilon),$$

Definition 11.1. With the above notation, the Kaplan-Meier estimator is defined as

$$\hat{S}(t) = \prod_{s \leq t} \left(1 - \frac{\Delta N(s)}{Y(s)}\right). \quad (11.5)$$

In the infinite product above, all factors where $\Delta N(s) = 0$ are equal to 1. There is thus only a finite number of factors different from 1. When $\Delta N(s) \in \{0, 1\}$ we see that Equation 11.5 coincides with Equation 11.4. The definition allows for $\Delta N(s) > 1$ to correctly account for multiple deaths at the same time and thus ties among the observed survival times.

The function `survfit()` from the R package `survival` can be used to compute the Kaplan-Meier estimator.

Since the cumulative hazard function is $\Lambda = -\log(S)$, we can estimate it nonparametrically as

$$-\log(\hat{S}(t)) = -\sum_{s \leq t} \log \left(1 - \frac{\Delta N(s)}{Y(s)} \right). \quad (11.6)$$

But there is also a different, and direct, nonparametric estimator of the cumulative hazard function. The logic behind this estimator is that for a *differentiable* survival function S and $t > s$ (but $t - s$ small),

$$\begin{aligned} \mathbf{P}(T^* \in [s, t] \mid T^* \geq s) &= \frac{S(s) - S(t)}{S(s)} \\ &\approx -\frac{S'(s)}{S(s)}(t - s) = \lambda(s)(t - s). \end{aligned}$$

By a Riemann sum approximation we see that

$$\Lambda(t) \approx \sum_{i: T_{(i)} \leq t} \mathbf{P}(T^* \in [T_{(i)}, T_{(i+1)}] \mid T^* \geq T_{(i)}),$$

and if there are no ties, the conditional probabilities in the sum above can be estimated⁵¹ as $\delta_{(i)}/Y(T_{(i)})$. This leads to the *Nelson-Aalen* estimator

$$\hat{\Lambda}(t) = \sum_{i: T_{(i)} \leq t} \frac{\delta_{(i)}}{Y(T_{(i)})}. \quad (11.7)$$

The general definition below allows for multiple deaths at the same time.

Definition 11.2. The Nelson-Aalen estimator of the cumulative hazard function is

$$\hat{\Lambda}(t) = \sum_{s \leq t} \frac{\Delta N(s)}{Y(s)}. \quad (11.8)$$

Since the survival function is given by $S(t) = \exp(-\Lambda(t))$, we can estimate the survival function by plugging the Nelson-Aalen estimator of the cumulative hazard, which gives

$$\exp(-\hat{\Lambda}(t)) = \prod_{s \leq t} \exp \left(-\frac{\Delta N(s)}{Y(s)} \right). \quad (11.9)$$

⁵¹ Either $\delta_{(i)} = 1$ and one individual out of $Y(T_{(i)})$ dies in the interval (at time $T_{(i)}$), or $\delta_{(i)} = 0$ and no individuals die in the interval.

We note that $\exp(-\hat{\Lambda})$ is not the Kaplan-Meier estimator and nor is $-\log(\hat{S})$ the Nelson-Aalen estimator.

The mismatch between the Kaplan-Meier and the Nelson-Aalen estimators is a small nuisance, but we may note that

$$\exp\left(-\frac{\Delta N(s)}{Y(s)}\right) \approx \left(1 - \frac{\Delta N(s)}{Y(s)}\right)$$

for large $Y(s)$ by using the Taylor expansion $\exp(x) = 1 + x + o(x^2)$. Thus differences between the factors are small unless s is so large that the number of individuals at risk, $Y(s)$, is small.

One can argue that the way ties are handled above presumes that survival times can be exactly equal, which is at odds with any continuous survival distribution. If we regard ties as a consequence of lack of precision in our registration or due to rounding of the survival times, it makes sense to replace $\frac{\Delta N(s)}{Y(s)}$ by

$$\frac{1}{Y(s)} + \frac{1}{Y(s) - 1} + \dots + \frac{1}{Y(s) - \Delta N(s) + 1}.$$

This is known as the *Fleming-Harrington* correction for ties. It often doesn't make much of a practical difference though.

Example 11.2. We continue Example 11.1 by first considering a summary of the marginal distributions of the four variables.

The function `survfit()` with argument `stype = 2` estimates the survival function in terms of the Nelson-Aalen estimator via Equation 11.9.

The function `survfit()` with argument `ctype = 2` will use the Fleming-Harrington correction for ties.

```
select(prostate, c(rx, status, dtime, age)) |> skim()
```

Variable type: factor										
skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts					
rx	0	1	FALSE	4	pla: 127, 1.0: 126, 5.0: 125, 0.2: 124					
status	0	1	FALSE	10	ali: 148, dea: 130, dea: 96, dea: 31					
Variable type: numeric										
skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
dtime	0	1	36.13	23.32	0	14.25	34	57.75	76	
age	1	1	71.46	7.08	48	70.00	73	76.00	89	

Table 11.1: Data summary of the four variables in the `prostate` dataset.

Table 11.1 shows 148 patients (out of 502) were alive at the end of the study. Thus 148 of the survival times are right censored. Table 11.2 shows a more detailed tabulation of the values of the `status` variable, which specifies one out of nine different causes of death – in addition to specifying if the patient is still alive. We may also note that the survival time (`mtime`) is registered in months since baseline, which is a very coarse scale. Consequently there are many ties in the data.

status	Freq
alive	148
dead - cerebrovascular	31
dead - heart or vascular	96
dead - other ca	25
dead - other specific non-ca	28
dead - prostatic ca	130
dead - pulmonary embolus	14
dead - respiratory disease	16
dead - unknown cause	7
dead - unspecified non-ca	7

Table 11.2: Tabulation of different causes of death.

We will fit a survival distribution, using either of the non-parametric estimators above, where we ignore the information about different causes of death. Thus the `status` variable will only be used to determine the censoring status.

We fit survival functions for all patients using the `survfit()` function from the survival package. See `?survfit.formula` for details about how to specify a model using the formula interface. Note that the left hand side of the formula, that specifies the response, uses the special `Surv()` function syntax to include information about the censoring status. We fit the survival functions below using both the Kaplan-Meier and the Nelson-Aalen, and we also fit the survival function using the Nelson-Aalen estimator with the Fleming-Harrington correction for ties.

```
prostate_KM <- survfit(
  Surv(mtime, status != "alive") ~ 1,
  data = prostate,
)
```

```

prostate_NA <- survfit(
  Surv(dtime, status != "alive") ~ 1,
  data = prostate,
  stype = 2
)

prostate_NA_FH <- survfit(
  Surv(dtime, status != "alive") ~ 1,
  data = prostate,
  stype = 2,
  ctype = 2
)

```

Figure 11.2 shows the fitted survival curves – including a plot that zooms in on a part of the curves. The differences are small and barely visible, and they are most pronounced in the tail of the distribution.

```

p1 <- summary(prostate_KM, data.frame = TRUE) |>
  ggplot(aes(time, surv)) +
  geom_step(data = summary(prostate_NA, data.frame = TRUE), color = "blue") +
  geom_step(data = summary(prostate_NA_FH, data.frame = TRUE), color = "red") +
  geom_step() +
  ylim(0, 1) +
  xlab("time (months)") +
  ylab("survival probability")

p2 <- p1 + coord_cartesian(xlim = c(50, 75), ylim = c(0.20, 0.40))

gridExtra::grid.arrange(p1, p2, ncol = 2)

```

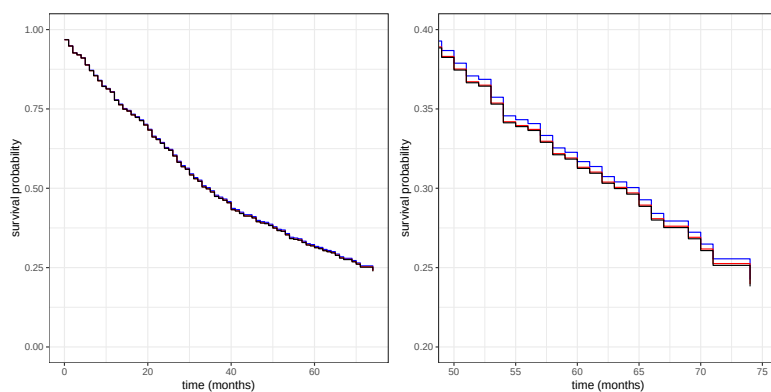


Figure 11.2: Left: estimated survival function for patients with prostate cancer using the Kaplan-Meier estimator (black), the Nelson-Aalen estimator (blue) and the Nelson-Aalen estimator with the Fleming-Harrington correction for ties (red). Right: the same as the left figure but zoomed in.

We then fit a separate survival function for each treatment group. We can think about this as a simple regression model with a single categorical predictor variable (the treatment variable `rx`), and the formula specification below is similar

to other regression model specifications. The main difference is that using `survfit()` we fit a survival function for each treatment group instead of a mean value, say.

```
prostate_KM_rx <- survfit(  
  Surv(dtime, status != "alive") ~ rx,  
  data = prostate  
)
```

```
summary(prostate_KM_rx, data.frame = TRUE) |>  
  ggplot(aes(time, surv, color = strata)) +  
  geom_step() +  
  ylim(0, 1) +  
  xlab("time (months)") +  
  ylab("survival probability") +  
  theme(legend.position = "top")
```

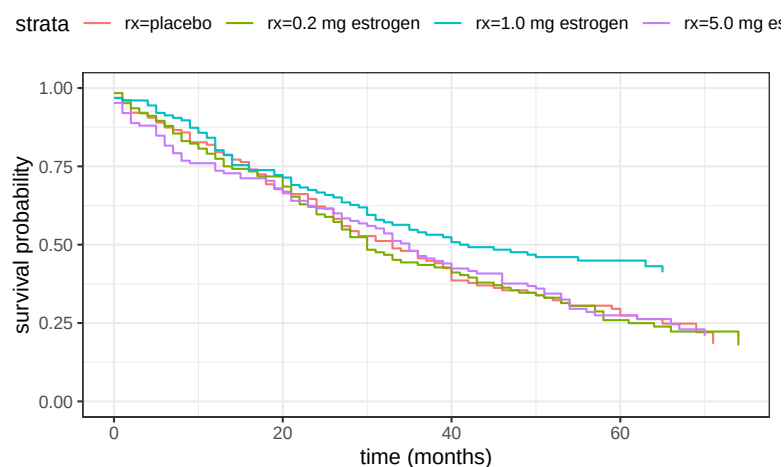


Figure 11.3: Kaplan-Meier estimators of the survival function for each of the four treatment groups.

11.3 Parametric survival models

In this section we focus on parametric models of the survival distribution, and we will first derive a generic expression for the likelihood function. We then turn to regression models and introduce the accelerated failure time models and the proportional hazards models as two different ways for the predictors to affect the response distribution.

11.3.1 The likelihood

Recall that in terms of T^* and C we have $T = \min\{T^*, C\}$ and $\delta = 1(T^* \leq C)$. Our first result gives the density of the joint distribution of (T, δ) in terms of the marginal distributions of T^* and C when they are independent.

Theorem 11.1. *If T^* and C are independent the joint distribution of (T, δ) has density*

$$g(t, \delta) = f(t)^\delta S(t)^{1-\delta} h(t)^{1-\delta} H(t)^\delta$$

w.r.t. the product measure $m \otimes \tau$ (Lebesgue measure times the counting measure).

Proof. First note that

$$\begin{aligned} \mathbf{P}(T \leq t, \delta = 1) &= \mathbf{P}(T^* \leq t, \delta = 1) \\ &= \mathbf{P}(T^* \leq t, C \geq T^*) \\ &= \int_0^t f(s) \int_s^\infty h(u) du ds \\ &= \int_0^t f(s) H(s) ds. \end{aligned}$$

Likewise,

$$\mathbf{P}(T \leq t, \delta = 0) = \int_0^t h(s) S(s) ds,$$

and we conclude that the density is

$$g(t, \delta) = \begin{cases} f(t)H(t) & \text{if } \delta = 1 \\ h(t)S(t) & \text{if } \delta = 0. \end{cases}$$

□

Based on the density result above we can introduce the full likelihood for right censored observations. With $(T_1, \delta_1), \dots, (T_n, \delta_n)$ independent the full likelihood is

$$L = \prod_{i=1}^n f_i(T_i)^{\delta_i} S_i(T_i)^{1-\delta_i} h_i(T_i)^{1-\delta_i} H_i(T_i)^{\delta_i}.$$

For a parametrized model of the distributions of $T_1^*, \dots, T_n^*$ we have that $f_i = f_{i,\beta}$ for some parameter β . The censoring distributions, given by the densities g_i , usually hold no information about the parameter β . This implies that

$$L(\beta) = \prod_{i=1}^n f_{i,\beta}(T_i)^{\delta_i} S_{i,\beta}(T_i)^{1-\delta_i} K_i$$

with K_i depending on the observations but not the parameter β .

Note how the derivation of the likelihood makes it explicit what the dominating measure is, and thus that the joint distribution in fact has a density. The derivation also makes it clear how the distribution of the censoring mechanism enters, and why it can be ignored when it does not depend on the unknown parameter β . The argument for ignorance is not independent censoring⁵² but that the g_i -s do not depend upon β . Technically, the censoring distribution and the parameter β are *variation independent*.

⁵² Though independent censoring does play a role here for our particular derivation and form of the full likelihood.

By dropping the factors K_i , that do not depend upon the parameter β , in the expression above for the likelihood, we arrive at the following definition of the survival likelihood.

Definition 11.3. With the above setup, the survival likelihood is

$$L(\beta) = \prod_{i=1}^n f_{i,\beta}(T_i)^{\delta_i} S_{i,\beta}(T_i)^{1-\delta_i}. \quad (11.10)$$

It is possible to take a slightly different point of view and condition on the censoring variables instead. Then the dominating measure for the i -th censored observation, T_i , becomes $m + \delta_{C_i}$ (δ_{C_i} is the Dirac measure in C_i). One arrives at the same likelihood as above though.

We can also express the likelihood in terms of the hazard and cumulative hazard functions.

Proposition 11.1. If $f_{i,\beta}$ has hazard function $\lambda_{i,\beta}$ and cumulative hazard function $\Lambda_{i,\beta}$, the survival likelihood equals

$$L(\beta) = \prod_{i=1}^n \lambda_{i,\beta}(T_i)^{\delta_i} e^{-\Lambda_{i,\beta}(T_i)} \quad (11.11)$$

Proof. Recall that

$$\lambda_{i,\beta}(t) = \frac{f_{i,\beta}(t)}{S_{i,\beta}(t)}$$

and that

$$S_{i,\beta}(t) = e^{-\Lambda_{i,\beta}(t)}.$$

Hence

$$\begin{aligned} L(\beta) &= \prod_{i=1}^n f_{i,\beta}(T_i)^{\delta_i} S_{i,\beta}(T_i)^{1-\delta_i} \\ &= \prod_{i=1}^n \lambda_{i,\beta}(T_i)^{\delta_i} S_{i,\beta}(T_i)^{\delta_i} S_{i,\beta}(T_i)^{1-\delta_i} \\ &= \prod_{i=1}^n \lambda_{i,\beta}(T_i)^{\delta_i} S_{i,\beta}(T_i) \\ &= \prod_{i=1}^n \lambda_{i,\beta}(T_i)^{\delta_i} e^{-\Lambda_{i,\beta}(T_i)}. \end{aligned}$$

□

Example 11.3. The *Weibull distribution* has hazard function

$$\lambda(t) = \alpha\gamma t^{\gamma-1}$$

for $\alpha, \gamma > 0$ two parameters. The special case $\gamma = 1$ gives us the exponential distribution with constant hazard function. The Weibull distribution has increasing hazard functions over time when $\gamma > 1$ and decreasing hazard function when $\gamma < 1$.

The cumulative hazard function is

$$\Lambda(t) = \alpha \int_0^t \gamma s^{\gamma-1} ds = \alpha t^\gamma.$$

For right censored i.i.d. observations from the Weibull model, the likelihood becomes

$$L(\alpha, \gamma) = \prod_{i=1}^n (\alpha\gamma)^{\delta_i} T_i^{\delta_i(\gamma-1)} e^{-\alpha T_i^\gamma},$$

and the log-likelihood is

$$\begin{aligned} \ell(\alpha, \gamma) &= \log L(\alpha, \gamma) \\ &= \sum_{i=1}^n \delta_i (\log(\alpha) + \log(\gamma) + (\gamma - 1) \log(T_i)) - \alpha T_i^\gamma. \end{aligned}$$

Example 11.4 (MLE for the exponential distribution). Continuing Example 11.3 and fixing $\gamma = 1$, corresponding to an exponential distribution with hazard rate α , we find that the log-likelihood equals

$$\ell(\alpha) = \sum_{i=1}^n \delta_i \log(\alpha) - \alpha T_i$$

up to additive terms that do not depend on α . This log-likelihood has a unique stationary point

$$\hat{\alpha} = \frac{\sum_{i=1}^n \delta_i}{\sum_{i=1}^n T_i} = \frac{n_u}{\sum_{i=1}^n T_i}$$

where $n_u = \sum_{i=1}^n \delta_i$ are the number of deaths (the number of uncensored events). The second derivative of ℓ is negative, so ℓ is concave and $\hat{\alpha}$ is the MLE.

If we had treated the observations as if there were no censoring, we would get the MLE

$$\frac{n}{\sum_{i=1}^n T_i} > \hat{\alpha},$$

which overestimates the rate.

If we, on the other hand, discard the censored observations, we would get the MLE

$$\frac{n_u}{\sum_{i=1}^n \delta_i T_i} > \hat{\alpha},$$

which likewise overestimates the rate.

11.3.2 Regression models

We introduce two ways that predictor variables can affect the survival response distribution. The accelerated failure time models exemplified by the log-logistic model, and the proportional hazards models exemplified by the Weibull model. In the latter case, it turns out that the likelihood can be computed and optimized using methods for Poisson generalized linear models.

Accelerated failure time models

Example 11.5 (Log-logistic distribution). Suppose Z has a logistic distribution with distribution function

$$G(z) = \frac{e^{\lambda z}}{1 + e^{\lambda z}}.$$

The distribution of $T^* = e^Z$ is then the *log-logistic* distribution, whose distribution function is

$$F(t) = G(\log(t)) = \frac{t^\lambda}{1 + t^\lambda},$$

with corresponding density

$$f(t) = F'(t) = \frac{\lambda t^{\lambda-1}}{(1 + t^\lambda)^2}$$

for $t > 0$.

Definition 11.4. Let G be a distribution function (on $\mathbb{R}$). The corresponding accelerated failure time (AFT) model has survival function given as

$$S_\eta(t) = 1 - G((\log(t) - \eta)/\sigma)$$

with $\eta = X^T\beta$ the linear predictor and $\sigma > 0$ a scale parameter.

The AFT model is a scale-location model of the log-survival time.

The interpretation of the AFT model is that a unit change of the j -th predictor increases – or *accelerates* – the failure time by a factor e^{β_j} .

Example 11.6 (Log-logistic AFT model). If G is the logistic distribution with scale parameter is $\sigma = \lambda^{-1}$, the survival function for the corresponding AFT model is

$$\begin{aligned} S_\eta(t) &= \frac{1}{1 + e^{\lambda(\log(t) - \eta)}} \\ &= \frac{1}{1 + e^{-\lambda\eta} t^\lambda}. \end{aligned} \tag{11.12}$$

where $\eta = X^T\beta$ is the linear predictor. This is a log-logistic distribution as introduced in Example 11.5. The effect of η is⁵³ a *scale transformation* of the log-logistic distribution with

⁵³ If T has log-logistic distribution with density f_0 , then $e^\eta T$ is log-logistic distributed with density f_η , cf. Exercise 11.1.

scale parameter e^η . The density is

$$\begin{aligned} f_\eta(t) &= \frac{\lambda e^{-\lambda\eta} t^{\lambda-1}}{(1 + e^{-\lambda\eta} t^\lambda)^2} \\ &= \frac{\lambda e^{\lambda(\log(t)-\eta)} t^{-1}}{(1 + e^{\lambda(\log(t)-\eta)})^2}. \end{aligned} \quad (11.13)$$

For maximum likelihood estimation we need to compute and optimize the likelihood in Equation 11.10. Efficient computation requires access to the density as well as the survival function, which are both explicitly available for the log-logistic distribution, which makes this distribution attractive for AFT models.

Example 11.7. We implement the negative log-likelihood (`nll()`) for the AFT model with a log-logistic survival distribution below. The implementation is based on Equation 11.10 using the expressions (11.12) and (11.13) for the survival function and density, respectively, of the log-logistic distribution.

The `nll()` function takes four arguments. Besides a vector of parameters (`par`), the censored survival times (`time`) and the status indicator (`status`), it takes a single additional argument `x` representing a categorical variable with k values. The parameter vector `par` is of the form

$$(\sigma, \beta_0, \dots, \beta_{k-1}) = (\lambda^{-1}, \beta_0, \dots, \beta_{k-1})$$

with σ the scale parameter, β_0 the intercept and $\beta_1, \dots, \beta_{k-1}$ are contrasts. The survival function is also implemented below for later usage.

```
# Argument `x` is an integer representing the k levels of a single
# categorical predictor variable. `par[2:(k+1)]` are the k
# beta-parameters corresponding to the intercept and contrasts.
# Its default value corresponds to no predictor.
nll <- function(par, time, status, x = rep(1, length(time))) {
  lambda <- par[1]^{-1}
  eta <- par[2] + c(0, par[-c(1, 2)])[x] # beta_x
  uncens <- status == 1
  lt <- log(time)
  logf <- (log(lambda) - lambda * eta[uncens] + (lambda - 1) * lt[uncens] -
    2 * log(1 + exp(lambda * (lt[uncens] - eta[uncens])))) |> sum()
  logS <- log(1 + exp(lambda * (lt[!uncens] - eta[!uncens]))) |> sum()
  - logf + logS
}
```

We use `optim()` with method L-BFGS-B to compute the maximum-likelihood estimate of a single survival distribution. Since some of the patients have survived less than one month, some survival times are registered as 0, which is first of all numerically problematic (due to $\log(0)$). For this reason we offset all times by 0.5 corresponding to shifting all death times by half a months.

```
prostate_aft <- optim(
  c(2, 2),
  nll,
  method = "L-BFGS-B",
  lower = c(0.01, -Inf),
  time = prostate$ddtime + 0.5, # to avoid log(0)
  status = prostate$status != "alive"
)
prostate_aft[1:2]
```

```
$par
[1] 0.8219375 3.5378103
```

```
$value
[1] 1758.922
```

We see from the `optim()` result that the parameter estimates are $\hat{\sigma} = 0.82$ and $\hat{\beta}_0 = 3.54$, and we can also see that $-\ell(\hat{\sigma}, \hat{\beta}_0) = 1758.92$. The corresponding fitted survival function is compared to the Kaplan-Meier estimator in Figure 11.4.

We can also use our implementation to fit the regression model where the survival distribution depends on treatment.

```
prostate_aft_rx <- optim(
  c(2, 2, 0, 0, 0),
  nll,
  method = "L-BFGS-B",
  lower = c(0.1, -Inf, -Inf, -Inf, -Inf),
  time = prostate$ddtime + 0.5, # to avoid log(0)
  status = prostate$status != "alive",
  x = as.numeric(prostate$rx)
)
prostate_aft_rx[1:2]
```

```
$par
[1] 0.81992315 3.48305068 -0.04002854 0.33958359 -
0.05824063
```

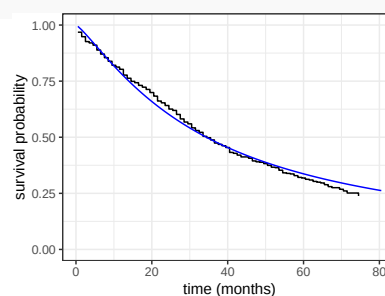


Figure 11.4: The survival function for the fitted log-logistic distribution (blue) and the Kaplan-Meier estimator (black).

```
$value
[1] 1755.851
```

```
# The log-logistic survival function in the scale-location parametrization.
time0 <- 0:80
fit <- tibble(
  time = rep(time0, 4),
  surv = c(
    S(time0 + 0.5, prostate_aft_rx$par[1], prostate_aft_rx$par[2]),
    S(time0 + 0.5, prostate_aft_rx$par[1], prostate_aft_rx$par[2] + prostate_aft_rx$par[3]),
    S(time0 + 0.5, prostate_aft_rx$par[1], prostate_aft_rx$par[2] + prostate_aft_rx$par[4]),
    S(time0 + 0.5, prostate_aft_rx$par[1], prostate_aft_rx$par[2] + prostate_aft_rx$par[5])
  ),
  strata = rep(paste("rx=", levels(prostate$rx), sep = ""), each = 81)
)

summary(prostate_KM_rx, data.frame = TRUE) |>
  ggplot(aes(time + 0.5, surv, color = strata)) +
  geom_step(linetype = 2) +
  geom_line(data = fit) +
  ylim(0, 1) +
  xlab("time (months)") +
  ylab("survival probability") +
  theme(legend.position = "top")
```

strata — rx=placebo — rx=0.2 mg estrogen — rx=1.0 mg estrogen — rx=5.0 mg estrogen

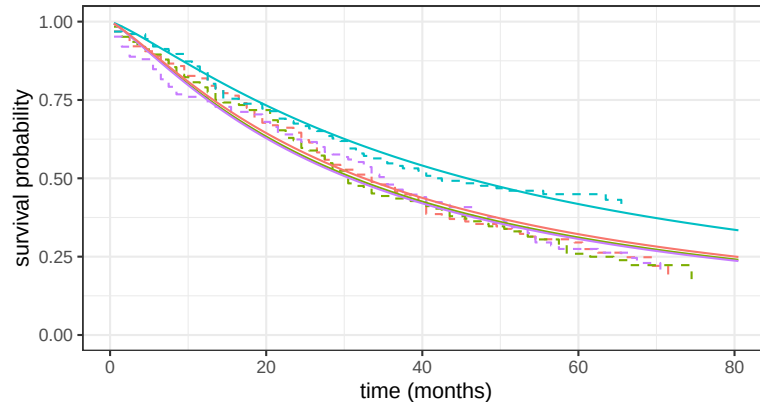


Figure 11.5: The AFT survival functions for the fitted log-logistic distributions (full) and the Kaplan-Meier estimators (dashed) for each of the four treatment groups.

We can fit AFT models, and a range of other parametric regression models, via the `survreg()` function from the `survival` package.

```
survreg(
  Surv(dtime + 0.5, status != "alive") ~ rx,
  data = prostate,
```

```

    dist = "loglogistic"
)

```

Call:

```

survreg(formula = Surv(dtime + 0.5, status != "alive") ~ rx,
        data = prostate, dist = "loglogistic")

```

Coefficients:

```

      (Intercept) rx0.2 mg estrogen rx1.0 mg estrogen rx5.0 mg estrogen
      3.48297715      -0.03983952      0.33947524      -
0.05802741

```

Scale= 0.8199158

Loglik(model)= -1755.9 Loglik(intercept only)= -1758.9

Chisq= 6.14 on 3 degrees of freedom, p= 0.105

n= 502

Note that the parametrization used by `survreg()` is exactly the same as we implemented, and that the estimated parameter values and computed log-likelihood are the same as well (up to small numerical differences).

The `survreg()` function even gives a likelihood-ratio test statistic (`Chisq`) of the hypothesis that there are no differences between the treatments. The corresponding p -value is computed using an approximating χ^2 -distribution with 3 degrees of freedom. It is above 0.1 and we cannot reject that there is no treatment effect.

As for `lm` and `glm` objects there are methods for `survreg` objects implemented in the `broom` package for the `glance()` and `tidy()` functions and in the `survival` package for `anova()`. We will illustrate some of their usages in subsequent examples.

Proportional hazards models

A different model class is given by letting the linear predictor have a multiplicative effect on the hazard function. The resulting proportional hazards models are among the most widely used survival models.

Definition 11.5. With λ_0 a baseline hazard function, the proportional hazards model is given by the hazard function

$$\lambda(t) = \lambda_0(t)e^\eta$$

with η the linear predictor.

It follows that for the cumulative hazard function,

$$\Lambda(t) = \Lambda_0(t)e^\eta,$$

and the proportionality holds for the cumulative hazard as well. The factor e^{β_j} is the *hazard ratio* between two models corresponding to a unit change of the j -th predictor.

The proportional hazards model is sometimes called the *Cox model*. Cox (1972) discovered that for the proportional hazards model, the parameters that enter into the linear predictor can be estimated without specifying the baseline hazard function. This will be covered in Section 11.4. Here we will instead show how the parameters can be estimated via maximum-likelihood when the baseline is assumed to be a Weibull distribution.

Cox, D. R. 1972. “Regression Models and Life-Tables.” *Journal of the Royal Statistical Society. Series B (Methodological)* 34 (2): pp. 187–220.

Example 11.8. The Weibull baseline hazard function and cumulative hazard function are

$$\lambda_0(t) = \gamma t^{\gamma-1} \quad \text{and} \quad \Lambda_0(t) = t^\gamma.$$

The log-likelihood is

$$\begin{aligned} \ell &= \sum_{i=1}^n \delta_i \log(\gamma T_i^{\gamma-1} e^{\eta_i}) - T_i^\gamma e^{\eta_i} \\ &= \underbrace{\sum_{i=1}^n \delta_i \log(T_i^\gamma e^{\eta_i}) - T_i^\gamma e^{\eta_i}}_{\text{Poisson log-likelihood}} + \sum_{i=1}^n \delta_i \log(\gamma T_i^{-1}). \end{aligned}$$

This is (surprisingly) up to a constant the log-likelihood for a Poisson model of the δ_i -s with logarithmic link function and mean value $T_i^\gamma e^{\eta_i}$ for fixed γ . Note that the α parameter appearing in Example 11.3 is captured by an intercept in the linear predictor.

For a fixed value of the linear predictor we also find that

$$\partial_\gamma \ell = \sum_{i=1}^n (\delta_i - T_i^\gamma e^{\eta_i}) \log T_i + \frac{\delta_i}{\gamma}.$$

Thus γ solves the equation

$$\gamma = \frac{n_u}{\sum_{i=1}^n (T_i^\gamma e^{\eta_i} - \delta_i) \log T_i} \quad (11.14)$$

with n_u the number of uncensored observations.

A practical algorithm can alternate between: i) estimate the linear predictor as $\hat{\eta}^{(k)}$ by optimizing the Poisson log-likelihood (for fixed $\gamma^{(k-1)}$); and ii) update

$$\gamma^{(k)} = \frac{n_u}{\sum_{i=1}^n (T_i^{\gamma^{(k-1)}} e^{\hat{\eta}_i^{(k)}} - \delta_i) \log T_i}.$$

The example above shows how implementations of generalized linear models – and Poisson regression models, in particular – can be used to fit the Weibull proportional hazards model (for fixed γ) with the survival times entering as an offset term. This was of historical importance as it made computations practical early on using some standard algorithms available. Today, the connection between the survival likelihood for Weibull distributions and the Poisson likelihood is mostly of theoretical and historical interest.

To fit the model we will instead show that the proportional hazards model with a Weibull baseline hazard function is also an accelerated failure time model, which makes it possible to use `survreg()` to fit the model.

Example 11.9. If we take G in Definition 11.4 to be the extreme value distribution

$$G(z) = 1 - \exp(-e^z),$$

then the corresponding AFT model has survival function

$$S_\eta(t) = \exp\left(-(e^{-\eta}t)^{\frac{1}{\sigma}}\right) = \exp(-e^{-\gamma\eta}t^\gamma)$$

where $\gamma = \frac{1}{\sigma}$. That is, the cumulative hazard function is , which is the proportional hazards model with a Weibull baseline, but reparametrized in terms of the scale parameter $\sigma = \frac{1}{\gamma}$, and the linear predictor is multiplied by $-\gamma$.

We can use the `survreg()` function to fit the model.

We can compute the estimated survival functions for the Weibull model much in the same way as we did above for the log-logistic model, where we used the MLE of the parameters and the explicit formula for the survival function. However, this is a bit cumbersome and does not generalize well. We can instead use the `predict()` method for `survreg` objects to predict quantiles for the survival functions, which implicitly also gives the estimated survival functions.

```
fit <- tibble(rx = factor(1:4, labels = levels(prostate$rx)))
p_seq <- seq(0.01, 0.90, 0.01)
pred <- predict(prostate_survreg, newdata = fit, type = "quantile", p = p_seq)
fit <- cbind(fit, as_tibble(pred, .name_repair = "minimal")) |>
  pivot_longer(cols = !rx, values_to = "time") |>
  mutate(
    surv = 1 - rep(p_seq, 4),
```

```
prostate_survreg <- survreg(
  Surv(dtime + 0.5, status != "alive") ~ rx,
  data = prostate,
  dist = "weibull"
)
prostate_survreg |> tidy()
```

```
# A tibble: 5 x 5
  term          estimate std.error statistic    p.value
<chr>          <dbl>    <dbl>    <dbl>    <dbl>
1 (Intercept)    3.87      0.105     36.9 1.05e-297
2 rx0.2 mg estrogen -0.0314    0.148    -0.212 8.32e- 1
3 rx1.0 mg estrogen  0.392     0.161     2.43 1.49e- 2
4 rx5.0 mg estrogen -0.00534    0.149    -0.0358 9.71e- 1
5 Log(scale)    0.0222     0.0470     0.472 6.37e- 1
```

Table 11.3: Parameter estimates for the Weibull accelerated failure time model fitted to the prostate cancer survival data.

```
strata = paste("rx=", rx, sep = "")
) |>
filter(time <= 80)

summary(prostate_KM_rx, data.frame = TRUE) |>
  ggplot(aes(time + 0.5, surv, color = strata)) +
  geom_step(linetype = 2) +
  geom_line(data = fit, aes(time)) +
  ylim(0, 1) +
  xlab("time (months)") +
  ylab("survival probability") +
  theme(legend.position = "top") +
  xlim(0, 82)
```

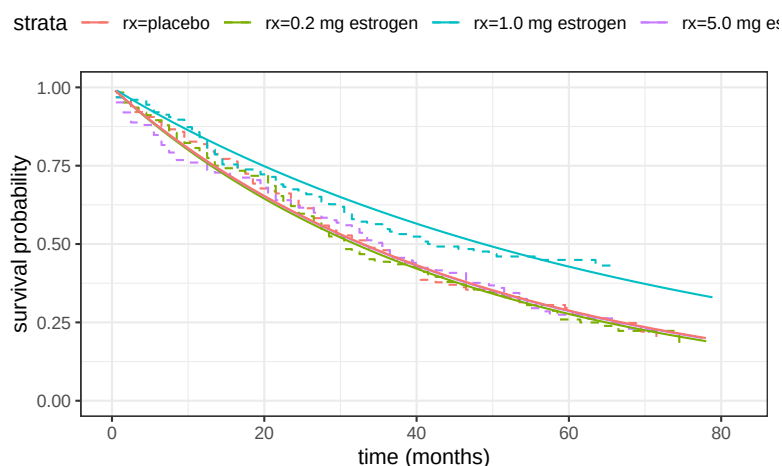


Figure 11.6: The survival functions for the fitted Weibull distributions (full) and the Kaplan-Meier estimators (dashed) for each of the four treatment groups.

Figure 11.6 compares the fitted Weibull distributions to the Kaplan-Meier estimators for the four treatment groups, and Table 11.4 reports the test statistic and p -value for testing if there are no differences between the four treatment groups. The p -value is a bit smaller in this case than when using the log-logistic distribution, and this time just below 5%.

```
prostate_survreg |>
  anova() |>
  select(-4) |>
  tidy() |>
  filter(term == "rx") |>
  knitr::kable(digits = 3)
```

term	df	deviance	df.residual	p.value
rx	3	9.619	497	0.022

Table 11.4: Deviance (likelihood-ratio) test of the hypothesis that there are no difference between the three treatment groups.

To corroborate the technical relation between the log-linear Poisson regression model and the proportional hazards model with a Weibull baseline, we refit the β -parameters using `glm()` (fixing the scale parameter at its estimated value).

To conclude our analyses of the prostate survival times, we have shown how one treatment group (1.0 mg estrogen) has a slightly different survival distribution than the other three


```
sigma <- prostate_survreg$scale
glm((status != "alive") ~ offset(log(dtime + 0.5) / sigma) + rx,
    data = prostate,
    family = "poisson"

) |> coefficients() * (- sigma)
```

```
(Intercept) rx0.2 mg estrogen rx1.0 mg estrogen rx5.0 mg estrogen
3.870053562      -0.031447138      0.392042914      -
0.005335812
```

Table 11.5: Regression parameters estimated using `glm()` and identical to those in Table 11.3.

with a better survival prognosis. The statistical tests of any treatment effect, based on two parametric models, are at best borderline significant though, and we cannot draw any clear conclusion.

There do not seem to be any detectable difference between the other three treatment groups. It is particularly noteworthy that the treatment group with the highest dose (5.0 mg estrogen) have a survival distribution similar to the placebo group. If there were an effect, we would often expect it to be monotone in dosage, and it would be peculiar if the effect is only there for the medium dose.

11.4 Semiparametric survival models

For the parametric proportional hazards models in Definition 11.5, the baseline hazard function λ_0 is either known or parametrized itself. There is a risk that we have misspecified this baseline model, and we would like to avoid a model of the baseline hazard if possible. This leads to the semiparametric proportional hazards model, where an individual's hazard is proportional to a common baseline hazard λ_0 with an individual weight factor, but where we do not model⁵⁴ the baseline λ_0 .

The general proportional hazards model assumes that the dependence of the predictors is via a multiplicative factor on the

⁵⁴ This is similar to *not* making distributional assumptions such as A3 or GA3 for the linear and generalized models.

baseline, and the i -th individual thus has hazard function

$$\lambda_i(t) = w_i \lambda_0(t)$$

for $w_i > 0$ a weight⁵⁵. The cumulative hazard function is consequently $\Lambda_i(t) = w_i \Lambda_0(t)$.

Definition 11.6. In terms of the observations $T_1, \dots, T_n$ and weights $w_1, \dots, w_n \geq 0$ we define the cumulative weight process

$$W(t) = \sum_{j:t \leq T_j} w_j$$

and the cumulative weights $W_i = W(T_i)$.

Proposition 11.2. When $\Lambda_i(t) = w_i \Lambda_0(t)$ the likelihood equals

$$L = \prod_{i:\delta_i=1} \frac{w_i}{W_i} \left(\prod_i (W_i \lambda_0(T_i))^{\delta_i} \right) e^{-\int_0^\infty W(t) \lambda_0(t) dt}.$$

Proof. First note that with $W(t) = \sum_{j:t \leq T_j} w_j$,

$$\sum_i \Lambda_i(T_i) = \int_0^\infty W(t) \lambda_0(t) dt.$$

The likelihood can therefore be factorized as

$$\begin{aligned} L &= \prod_i \lambda_i(T_i)^{\delta_i} e^{-\Lambda_i(T_i)} \\ &= \left(\prod_{i:\delta_i=1} (w_i \lambda_0(T_i))^{\delta_i} \right) e^{-\int_0^\infty W(t) \lambda_0(t) dt} \\ &= \prod_{i:\delta_i=1} \frac{w_i}{W_i} \left(\prod_i (W_i \lambda_0(T_i))^{\delta_i} \right) e^{-\int_0^\infty W(t) \lambda_0(t) dt}. \end{aligned}$$

□

⁵⁵ The i -th weight could be e^{η_i} with

$$\eta_i = X_i^T \beta$$

the linear predictor, but it does not need to be for the general arguments.

To ease notation we leave out limits on the generic index i in products and sums. It will always run through observations from 1 to n .

The purpose of Proposition 11.2 is to decompose the likelihood into two factors, where the second factor

$$\tilde{L}(\lambda_0) = \left(\prod_i (W_i \lambda_0(T_i))^{\delta_i} \right) e^{-\int_0^\infty W(t) \lambda_0(t) dt}.$$

is the only one depending on λ_0 . This factor can be made arbitrarily large if we try to maximize over λ_0 , and there is thus no unconstrained “maximum-likelihood” estimator of λ_0 in a classical sense.

It is possible to formalize, via a certain renormalization, that for fixed weights the “maximizer” is a generalized function of the form

$$\hat{\lambda}_0 = \sum_i \frac{\delta_i}{W_i} \delta_{T_i}$$

where δ_{T_i} is the Dirac measure in T_i . Since

$$\int_0^\infty W(t) \hat{\lambda}_0 dt = \sum_i W_i \frac{\delta_i}{W_i} = \sum_i \delta_i = n_u,$$

the number of uncensored observations, this leads to

$$\tilde{L}(\hat{\lambda}_0) = e^{-n_u},$$

which is independent of the weights. We pursue a similar perspective in Section 11.4.2.

11.4.1 Cox’s partial likelihood

Proposition 11.2 and the subsequent discussion suggest that for estimation of the parameters that enter into the weights⁵⁶, and not into λ_0 , we can ignore the factor $\tilde{L}$ in the likelihood. This motivates our definition of Cox’s *partial* likelihood.

⁵⁶ And are variation independent with λ_0 .

Definition 11.7. Cox’s partial likelihood for the proportional hazards model is

$$L_{\text{par}} = \prod_{i:\delta_i=1} \frac{w_i}{W_i}$$

where the cumulative weights W_i are given by Definition 11.6.

In practice, the predictors are mostly linked to the weights via an exponential function and are thus of the form

$$w_i = e^{X_i^T \beta} = e^{\eta_i}. \quad (11.15)$$

In this case we find that the partial log-likelihood equals

$$\ell_{\text{par}}(\beta) = \sum_{i:\delta_i=1} X_i^T \beta - \log \left(\underbrace{\sum_{j:T_i \leq T_j} w_j}_{W_i} \right), \quad (11.16)$$

with corresponding score function

$$\mathcal{U}_{\text{par}}(\beta) = D\ell_{\text{par}}(\beta) = \sum_{i:\delta_i=1} X_i^T - \frac{\sum_{j:T_i \leq T_j} X_j^T w_j}{W_i} \quad (11.17)$$

and information function

$$\begin{aligned} \mathcal{J}_{\text{par}}(\beta) &= -D^2 \ell_{\text{par}}(\beta) \\ &= \sum_{i:\delta_i=1} \frac{\sum_{j:T_i \leq T_j} X_j X_j^T w_j}{W_i} - \frac{(\sum_{j:T_i \leq T_j} X_j w_j)(\sum_{j:T_i \leq T_j} X_j w_j)^T}{W_i^2}. \end{aligned} \quad (11.18)$$

The estimate, $\hat{\beta}$, is computed by maximizing ℓ_{par} , and the observed information, $\mathcal{J}(\hat{\beta})$, is used for quadratic approximations of ℓ_{par} , for standard error estimates and Wald tests. In practice, the computations proceed just as if ℓ_{par} were a log-likelihood function.

The particular score test

$$\mathcal{U}(0)^T \mathcal{J}(0)^{-1} \mathcal{U}(0)$$

for testing $\beta = 0$ reduces to a classical test known as the *log-rank test* in the special case with only a single categorical predictor.

Example 11.10. Cox's proportional hazards model can be fitted to data via the function `coxph()` from the survival R package.

```
anova(prostate_cox) |>
  tidy() |>
  filter(term == "rx") |>
```

```

prostate_cox <- coxph(
  Surv(dtime + 0.5, status != "alive") ~ rx,
  data = prostate,
)

prostate_cox |> tidy()

```

```

# A tibble: 3 x 5
  term                estimate std.error statistic p.value
<chr>                <dbl>    <dbl>    <dbl>    <dbl>
1 rx0.2 mg estrogen  0.0321      0.145     0.221     0.825
2 rx1.0 mg estrogen -0.386      0.157    -2.46     0.0140
3 rx5.0 mg estrogen  0.00291     0.146     0.0200    0.984

```

Table 11.6: Estimated regression parameters for Cox’s proportional hazards model for the prostate cancer survival data.

```
knitr::kable(digits = 3)
```

term	logLik	statistic	df	p.value
rx	-2005.899	9.709	3	0.021

Table 11.7: Likelihood-ratio test of the hypothesis of no treatment effect.

The parameter estimates are comparable to those obtained by `survreg()` with the Weibull distribution, by appropriately rescaling, since that model is parametrized as an AFT model. In particular, they Cox model regression parameters have the opposite sign. This is noteworthy, since a negative value of a Cox model regression parameter means that the corresponding predictor has a positive association with the survival time. That is, if $\beta_i < 0$ then a unit increase of X_i decreases the hazard by a factor $e^{-\beta_i} < 1$, which results in a survival distribution with longer survival times.

Note that the (partial) likelihood-ratio test reported in Table 11.7 results in a very similar conclusion as using the Weibull baseline.

11.4.2 Poisson empirical likelihood

To provide an additional perspective on Cox's partial likelihood, we will show how it can also be derived from a rigorous profiling argument of the Poisson empirical likelihood we define below. Our main purpose of giving this argument is that we then also get an explicit formula for a nonparametric estimator of the cumulative baseline hazard function.

For $\lambda_1, \dots, \lambda_n \geq 0$ we define the corresponding *jump cumulative hazard function* as

$$\Lambda(t) = \sum_{i: T_i \leq t} \lambda_i.$$

We can regard this cumulative hazard function as having the hazard function

$$\lambda(t) = \sum_i \lambda_i \delta_{T_i},$$

which is, of course, not a function but a measure, and $\Lambda(t)$ is the measure of the interval $[0, t]$. The measure has point mass λ_i in the jump point T_i .

Definition 11.8. The Poisson empirical likelihood of a jump cumulative hazard function is

$$L^*(\Lambda) = \prod_i \lambda_i^{\delta_i} e^{-\Lambda_i}$$

where $\Lambda_i = \Lambda(T_i)$.

In this section we think about L^* as simply a formal extension of the likelihood (11.11) from continuously differentiable cumulative hazard functions to jump functions, but see also Section 11.6 on discrete time survival distributions, and see Johansen (1983) for a justification of L^* as a real likelihood.

The next lemma gives an expression for the Poisson empirical log-likelihood for jump cumulative hazard functions of the form

$$\Lambda_i(t) = w_i \sum_{j: T_j \leq t} e^{h_j}.$$

Lemma 11.1. Suppose that Λ_0 has jumps $\lambda_i = e^{h_i}$ for $h_i \in \mathbb{R}$. For the proportional hazards model with baseline Λ_0 the Poisson empirical log-likelihood is

$$\ell^* = \sum_i \delta_i h_i + \delta_i \log(w_i) - e^{h_i} W_i. \quad (11.19)$$

Johansen, Søren. 1983. "An Extension of Cox's Regression Model." *International Statistical Review* 51 (2): 165–74.

Proof. The Poisson log-likelihood for the proportional hazards model is

$$\begin{aligned}\ell^* &= \sum_i \delta_i \log(\lambda_0(T_i)) + \delta_i \log(w_i) - \Lambda_0(T_i)w_i \\ &= \sum_i \delta_i h_i + \delta_i \log(w_i) - w_i \sum_{j: T_j \leq T_i} e^{h_j},\end{aligned}$$

where ℓ^* is expressed in terms of the weights and the exponents h_j . Interchanging the sums in the log-likelihood we get

$$\ell^* = \sum_i \delta_i h_i + \delta_i \log(w_i) - e^{h_i} W_i.$$

□

Lemma 11.2. *The maximizer of ℓ^* in (11.19) over $h_1, \dots, h_n$ is given by*

$$e^{h_i} = \frac{\delta_i}{W_i}$$

for $i = 1, \dots, n$.

With all weights being 1 in Lemma 11.2, the estimator is recognized as the Nelson-Aalen estimator.

Proof. The parameters h_i are variation independent, and we can thus maximize the function by maximizing each term. Differentiating we find that

$$\partial_{h_i} \ell^* = \delta_i - e^{h_i} W_i.$$

If $\delta_i = 0$ the function is monotonely decreasing and maximized in $-\infty$ corresponding to $e^{h_i} = 0$. If $\delta_i = 1$ we equate the derivative equal to 0 and find the claimed solution to be the unique stationary point. An additional differentiation shows that the second derivative is

$$-e^{h_i} W_i < 0,$$

and this implies that the solution is a global maximum. □

As an immediate consequence of plugging the maximizer into ℓ^* we get the following corollary.

Corollary 11.1. *The profile log-likelihood for the weights of the Poisson empirical log-likelihood is*

$$\ell_{\text{profile}}^* = \underbrace{\sum_{i: \delta_i=1} \log\left(\frac{w_i}{W_i}\right)}_{\text{Cox's partial log-likelihood}} - n_u$$

where $n_u = \sum_i \delta_i$ is the total number of uncensored observations (the number of deaths).

Corollary Corollary 11.1 shows how Cox's partial likelihood appears as a profile likelihood function from the Poisson empirical likelihood for the proportional hazards model.

When $w_i = e^{X_j^T \beta}$, Lemma 11.2 results in the estimator

$$\hat{\Lambda}_0(t) = \sum_{i: T_i \leq t} \frac{\delta_i}{\sum_{j: T_j \leq T_j} e^{X_j^T \hat{\beta}}}$$

of the cumulative hazard function in terms of the survival times and the estimate $\hat{\beta}$. This estimator is known as the *Breslow estimator*, which can be seen as an extension of the Nelson-Aalen estimator of the baseline cumulative hazard function.

Example 11.11. We continue Example 11.10 and compute the Breslow estimator of the baseline cumulative hazard using `survfit()`, see `?survfit.coxph` for details. In fact, `survfit()` returns an estimate of the baseline survival function as

$$\hat{S}_0(t) = e^{-\hat{\Lambda}_0}$$

where $\hat{\Lambda}_0$ is the Breslow estimator.

```
survfit(prostate_cox) |>
  summary(data.frame = TRUE) |>
  ggplot(aes(time + 0.5, surv)) +
    geom_ribbon(aes(ymin = lower, ymax = upper), fill = "gray") +
    geom_step() +
    ylim(0, 1) +
    xlab("time (months)") +
    ylab("survival probability")
```

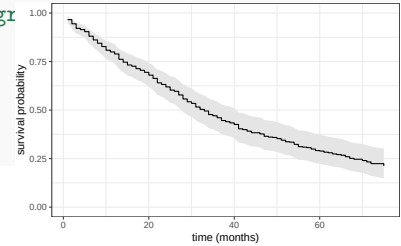


Figure 11.7: The Breslow estimate of the baseline survival function.

Figure 11.7 shows the Breslow estimate including a point-wise 95% confidence band, which is also computed by the `survfit()` function. Figure 11.8 shows the fitted survival functions for the four different treatment groups, using the Breslow estimator of the baseline cumulative hazard function, compared to using a Weibull parametric model as baseline. The survival curves are based on the Cox model and in principle computed as

$$\hat{S}_k = e^{-e^{\hat{\beta}_k} \hat{\Lambda}_0}, \quad k = 0, 1, 2, 3. \quad (11.20)$$

Note that for Cox's proportional hazards model, $\beta_0 = 0$ always. That is, there is no intercept parameter. Since the baseline hazard is nonparametric, we cannot identify a separate intercept parameter in this model, and the intercept is effectively absorbed into the nonparametric baseline. As pointed out in the `survfit()` documentation, this also means that there is no such thing as *the* baseline hazard. What the baseline hazard is depends on a choice of baseline value of X , and thus also on how the predictors are encoded. In this example, the baseline computed by `survfit()` corresponds to the placebo treatment group, but in more complicated setups the baseline corresponds to a pseudo individual – see `?survfit.coxph`.

While it can be instructive to extract the estimated regression parameters and use (11.20) explicitly, the code below used for actually computing the estimated survival functions shown in Figure 11.8 use `survfit()` with a new dataset, which will give a survival function for each row of the new data. This is both a safer and more convenient way of computing such estimates.

```
pred_frame <- tibble(
  data = 1:4,
  rx = factor(1:4, labels = levels(prostate$rx)),
  strata = factor(1:4, labels = paste("rx=", rx, sep = ""))
)
survfit(prostate_cox, newdata = pred_frame) |>
  summary(data.frame = TRUE) |>
  left_join(pred_frame) |>
  ggplot(aes(time, surv, color = strata)) +
    geom_line(data = fit, linetype = 2) +
    geom_step() +
    ylim(0, 1) +
    xlab("time (months)") +
    ylab("survival probability") +
    theme(legend.position = "top")
```

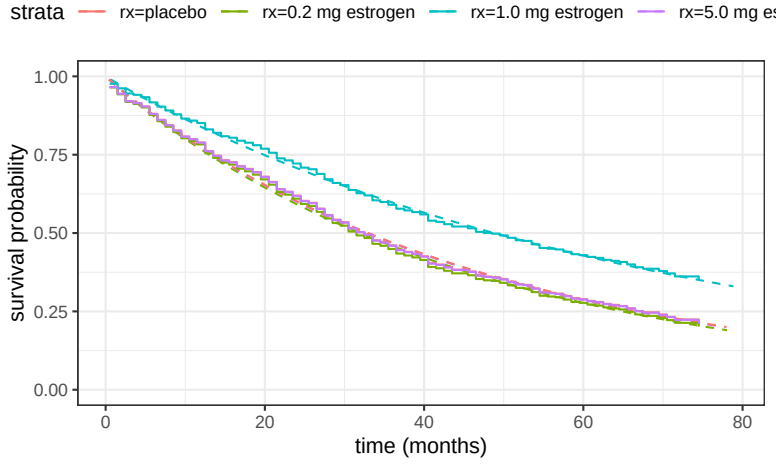


Figure 11.8: The fitted survival functions for proportional hazards models using the Cox proportional hazards model with the Breslow estimator of the baseline (full, step) and the parametric Weibull survival functions (dashed) for each of the four treatment groups.

11.4.3 Ties

We did not deal with ties explicitly in any of the derivations in previous sections. If there are ties among the survival times, the summation over $j : T_i \leq T_j$ in the definition of the cumulative weights is still well defined. If it happens that $T_i = T_k$ the corresponding cumulative weights $W_i = W_k$ also become identical. And if both observations are uncensored, they enter into the partial likelihood as the product

$$\frac{w_i}{W_i} \frac{w_k}{W_k} = \frac{w_i w_k}{W_i^2}.$$

This implicit way of handling ties is known as Breslow's method. It treats ties as if they were really due to events that happend at the exact same time.

If the tie $T_i = T_k$ was due to rounding or grouping, the contribution to the partial likelihood from those two observations should have been one of the two factors:

$$\frac{w_i w_k}{W_i(W_i - w_i)} \quad \text{or} \quad \frac{w_i w_k}{W_k(W_k - w_k)}. \quad (11.21)$$

We could try to break the tie, e.g., by letting the R function `order()` break ties in whatever arbitrary way it chooses depending on the sorting algorithm and the original order of the data. This naive solution will break all ties but in a unsystematic way, choosing one of the two factors above arbitrarily. Breaking ties in this way appears difficult to justify, and we should probably only break ties if we actually know and use the correct ordering prior to the rounding.

Efron's method⁵⁷ represents instead the “correct but unknown” factor by an approximation. Efron's approximation for two ties is

$$\frac{w_i w_k}{W_i(W_i - (w_i + w_k)/2)} = \frac{w_i w_k}{W_k(W_k - (w_i + w_k)/2)}$$

with similar formulas for more than two ties. The approximation is close to the geometric mean of the factors (11.21).

Results rarely depend much on whether Efron's or Breslow's method (or even naive tie breaking) is used, but Efron's method is usually preferred if the ties are due to a lack of precision and not to a true discrete nature of the survival times.

⁵⁷ The default in `coxph()` for handling ties is Efron's method.

11.5 Survival residuals

Model fit can be assessed for survival regression models just as for generalized linear models by means of residuals. There are, of course, several different possible choices.

11.5.1 Cox-Snell residuals

Cox-Snell residuals are not really residuals in a classical sense, but a transformation of the observed survival times to a set of random variables whose distribution should be exponential if the model is correct. They are based on the following result.

Lemma 11.3. *If T has continuous cumulative hazards function Λ then $\Lambda(T)$ is exponentially distributed with rate 1.*

Proof. The survival function $S(t) = \exp(-\Lambda(t))$ is continuous, whence

$$S(T) \sim \text{unif}([0, 1]).$$

From this it follows that

$$\mathbf{P}(\Lambda(T) \leq t) = \mathbf{P}(S(t) > \exp(-t)) = 1 - \exp(-t)$$

for $t \geq 0$, or $\Lambda(T) \sim \text{Exp}(1)$. □

Definition 11.9. With $\hat{\Lambda}_i$ an estimator of the cumulative hazard function for T_i the Cox-Snell residuals are defined as

$$\hat{\Lambda}_i(T_i) \quad (11.22)$$

for $i = 1, \dots, n$.

From the lemma above we conclude that if $(T_1, \delta_1), \dots, (T_n, \delta_n)$ are independently right censored survival times with corresponding hazard functions $\Lambda_1, \dots, \Lambda_n$, then

$$(\Lambda_1(T_1), \delta_1), \dots, (\Lambda_n(T_n), \delta_n)$$

are independently right censored i.i.d. exponentially distributed variables.

If there were no censoring, we would expect the Cox-Snell residuals to follow approximately an exponential distribution. With censoring the Cox-Snell residuals are expected to be approximately independently censored exponentially distributed. We use this in practice by estimating the cumulative hazards function for the Cox-Snell residuals using, e.g., the Nelson-Aalen estimator or the negative log of the Kaplan-Meier estimator. The resulting estimate is then compared to the cumulative hazard for the exponential distribution, which is $\Lambda(t) = t$. In practice we plot the estimated cumulative hazards against the Cox-Snell residuals, which should fall on a straight line with slope 1.

11.5.2 Martingale residuals

The martingale residuals are defined as

$$\delta_i - \hat{\Lambda}_i(T_i),$$

and are thus the status indicators minus the Cox-Snell residuals. Their usefulness is partly justified by the following result.

Lemma 11.4. *If $T = \min\{T^*, C\}$ with T^* and C having continuously differentiable survival functions it holds that*

$$\mathbf{E}(\Lambda(T)) = \mathbf{P}(T^* \leq C).$$

Proof. Let S and H denote the survival functions of T^* and C , respectively. By integration by parts

$$\begin{aligned}\mathbf{E}(\Lambda(T)) &= - \int_0^\infty \Lambda(t)(SH)'(t)dt = \int_0^\infty \Lambda'(t)S(t)H(t)dt \\ &= \int_0^\infty f(t)H(t)dt = \mathbf{P}(T^* \leq C).\end{aligned}$$

□

It follows from Lemma 11.4 that with $\delta = 1(T^* \leq C)$ then

$$\mathbf{E}(\delta - \Lambda(T)) = 0. \quad (11.23)$$

Thus if $\hat{\Lambda}_i$ were the true cumulative hazards function, the martingale residuals would have mean zero.

Now $\hat{\Lambda}_i$ is only an estimate, but if the model fits the data, the martingale residuals should have approximately mean 0. In the more advanced theory of survival analysis they also enjoy a martingale property, which explains their name. Here we will not pursue this, but just observe that we can investigate the model fit by investigating if the martingale residuals approximately have mean 0. The residuals can be used much as regression residuals to judge overall fit as well as the fit of how the individual predictor variables enter into the model. It should be noted, though, that they by definition have a left skewed distribution⁵⁸ concentrated on $(-\infty, 1)$.

The survival function for the minimum of the two independent random variables T^* and C is the product SH , whence the density of their distribution is the negative derivative $-(SH)'$.

Example 11.12. As for other regression models, we can use `augment()` from the broom package to extract fitted values (the linear predictor) and residuals. Figure 11.9 shows the martingale residuals plotted against the treatment (which is the only predictor in the model at this point). Since the dataset is rather small, it makes sense here to use `geom_beeswarm()` from the ggbeeswarm package to visualize the full distribution of the residuals as well as the individual points. Note the upper cap at 1 of all martingale residual distributions.

⁵⁸ The martingale residuals are *never* approximately normally distributed!

```
augment(prostate_cox, data = prostate) |>
  ggplot(aes(rx, .resid)) +
  ggbeeswarm::geom_beeswarm(alpha = 0.4) +
  geom_point(
    stat = "summary",
    fun = mean,
```

```

    color = "blue",
    size = 2.5
  ) +
  geom_errorbar(
    stat = "summary",
    fun.data = mean_se,
    fun.args = list(mult = 1.96),
    color = "blue",
    width = 0.15,
    linewidth = 0.7
  )

```

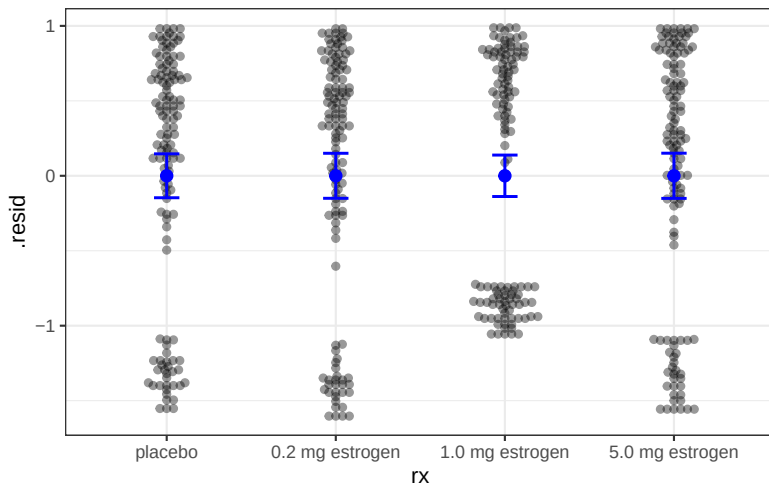


Figure 11.9: Martingale residuals for the Cox model for each of the four treatment groups. The blue points and errorbars are means and 95% confidence intervals.

Figure 11.9 includes means and confidence intervals for each group of residuals, which should include zero for all groups as in the figure.

We can also plot the residuals against another potential predictor, such as **age**, as in Figure 11.10. We see a clear association between age and the residuals suggesting that **age** (unsurprisingly) holds predictive information about the survival times.

We want to include **age** in the model, and we will first try to include it as an additive term.

```

prostate_cox_add <- coxph(
  Surv(dtime + 0.5, status != "alive") ~ age + rx,
  data = prostate,
)

```

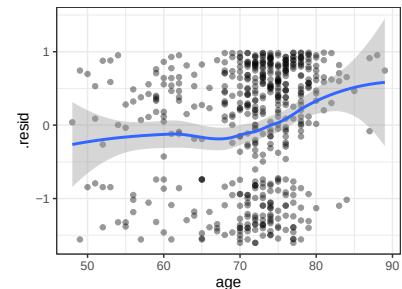


Figure 11.10: Martingale residuals

Figure 11.11 shows two residual plots. One based on comparing the Cox-Snell residuals to the exponential distribution and

standard type of a residual plot, this time of the martingale residuals against the fitted values. The fitted values are the linear predictors. For the computation of the Cox-Snell residuals, we used that `type.predict = "expected"` in the call of `augment()` gives $\hat{\Lambda}(T_i)$ as the predicted values (`.fitted` column in the output).

```
p1 <- augment(prostate_cox_add, data = prostate, type.predict = "expected") |>
  survfit(Surv(.fitted, status != "alive") ~ 1, data = _) |>
  summary(data.frame = TRUE) |>
  ggplot(aes(time, cumhaz)) +
  geom_abline(slope = 1, color = "blue") +
  geom_point() +
  xlab("Cox-Snell residuals") +
  ylab("Cumulative hazard")

p2 <- augment(prostate_cox_add) |>
  ggplot(aes(.fitted, .resid)) +
  geom_point(alpha = 0.4) +
  geom_smooth() +
  xlab("Fitted values") +
  ylab("Martingale residuals")

gridExtra::grid.arrange(p1, p2, ncol = 2)
```

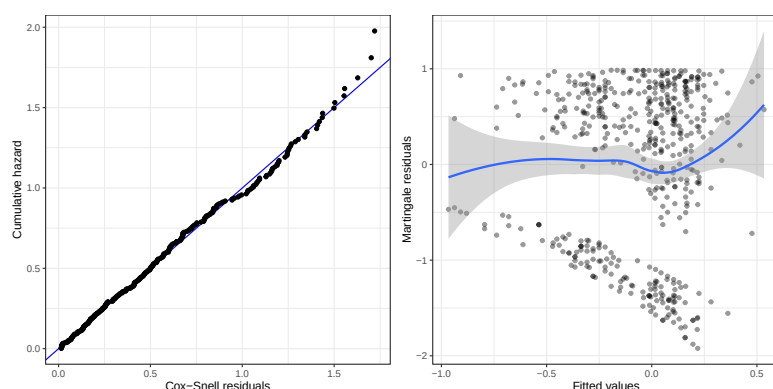


Figure 11.11: The Cox-Snell residual plot (left) and the martingale residuals plotted against fitted values (right) for the additive model.

We can see minor model misspecifications from the residual plots in Figure 11.11. First, there are deviations from a straight line in the Cox-Snell plot, suggesting some form of model misspecification. The smoothed curve in the martingale residual plot also suggests some minor model misspecification. We could suspect that `age` is not correctly modeled.

11.5.3 Deviance residuals

Alternative residuals include the deviance residuals and the Schoenfeld residuals. We will not treat the latter, but the former are based on the following observation. The Poisson nonparametric log-likelihood as well as the parametric log-likelihood with a continuous baseline can be written as

$$\ell^* = \sum_{i=1}^n \underbrace{\delta_i \log(w_i \Lambda_{0,i}) - w_i \Lambda_{0,i}}_{\text{Poisson log-likelihood term}} + \sum_{i=1}^n \delta_i \log\left(\frac{\lambda_{0,i}}{\Lambda_{0,i}}\right)$$

The first term is identical to the Poisson log-likelihood, and the second term does not depend on parameters that enter into the weights. Based on this we define the deviance residuals in terms of the Poisson deviances

$$d(\delta_i, \hat{\Lambda}_i) = 2 \left(\delta_i \log(\delta_i / \hat{\Lambda}_i) - \delta_i + \hat{\Lambda}_i \right),$$

where $\hat{\Lambda}_i = \hat{w}_i \hat{\Lambda}_{0,i}$.

The result is the *deviance residuals*

$$\text{sign}(\delta_i - \hat{\Lambda}_i) \sqrt{d(\delta_i, \hat{\Lambda}_i)} = \begin{cases} -\sqrt{2\hat{\Lambda}_i} & \text{if } \delta_i = 0 \\ \text{sign}(1 - \hat{\Lambda}_i) \sqrt{2(\hat{\Lambda}_i - 1 - \log \hat{\Lambda}_i)} & \text{if } \delta_i = 1 \end{cases}$$

Observe that the deviance residuals can be expressed as

$$\text{sign}(\hat{r}_i) \sqrt{-2(\hat{r}_i + \delta_i \log(\delta_i - \hat{r}_i))}$$

with $\hat{r}_i = \delta_i - \hat{\Lambda}_i$ the martingale residuals. Compared to the martingale residuals, the distribution of the deviance residuals is less skewed.

11.5.4 Prostate cancer continued

We conclude this section by a final analysis of the prostate cancer data based on the insights we have obtained from the previous sections. We include both **age** and treatment **rx** in the model. Now with **age** nonlinearly expanded using natural cubic splines and **rx** interacting (linearly) with **age**.

```
prostate_cox_int <- coxph(
  Surv(dtime + 0.5, status != "alive") ~ ns(age, df = 4) + age * rx,
```



```
data = prostate,
)
```

We first reconsider the residual plots in Figure 11.12. In contrast to Figure 11.11 there are no indications in these residual plots of any model misspecification.

```
p1 <- augment(prostate_cox_int, data = prostate, type.predict = "expected") |>
  survfit(Surv(.fitted, status != "alive") ~ 1, data = _) |>
  summary(data.frame = TRUE) |>
  ggplot(aes(time, cumhaz)) +
  geom_abline(slope = 1, color = "blue") +
  geom_point() +
  xlab("Cox-Snell residuals") +
  ylab("Cumulative hazard")

p2 <- augment(prostate_cox_int) |>
  ggplot(aes(.fitted, .resid)) +
  geom_point() +
  geom_smooth() +
  xlab("Fitted values") +
  ylab("Martingale residuals")

gridExtra::grid.arrange(p1, p2, ncol = 2)
```

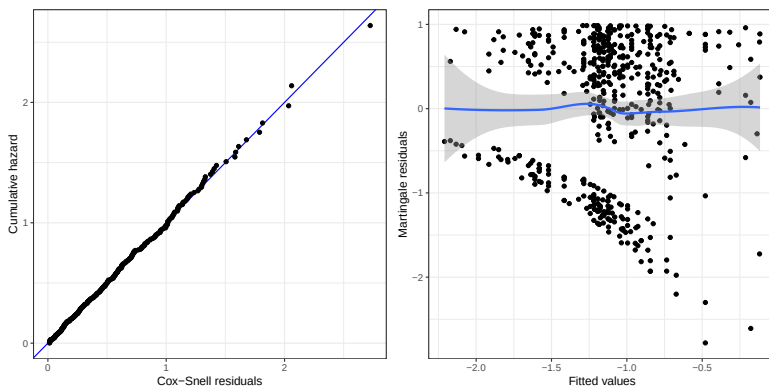


Figure 11.12: The Cox-Snell residual plot (left) and the martingale residuals plotted against fitted values (right) for the interaction model.

To better understand the interaction model we will compute and visualize the estimated risk factors

$$e^{\hat{\eta}}$$

for some combinations of treatment and age as well as the corresponding fitted survival curves.

```

pred_frame <- expand.grid(
  age = c(55, 62.5, 70, 77.5),
  rx = factor(1:4, labels = levels(prostate$rx))
)

pred_frame <- mutate(
  pred_frame,
  risk = predict(
    prostate_cox_int,
    newdata = pred_frame,
    type = "risk"
  ),
  risk = risk / risk[1],
  data = 1:16
)

```

age	rx	risk	data
55.0	placebo	1.000	1
62.5	placebo	0.989	2
70.0	placebo	0.894	3
77.5	placebo	1.313	4
55.0	0.2 mg estrogen	1.438	5
62.5	0.2 mg estrogen	1.231	6
70.0	0.2 mg estrogen	0.963	7
77.5	0.2 mg estrogen	1.224	8
55.0	1.0 mg estrogen	0.391	9
62.5	1.0 mg estrogen	0.499	10
70.0	1.0 mg estrogen	0.582	11
77.5	1.0 mg estrogen	1.102	12
55.0	5.0 mg estrogen	0.494	13
62.5	5.0 mg estrogen	0.679	14
70.0	5.0 mg estrogen	0.852	15
77.5	5.0 mg estrogen	1.738	16

Table 11.8: Estimated risk factors for the interaction model.

The computations above give a data frame with estimated risk factors for 16 combinations of age and treatment. Since these factors are always relative risk factors, we have normalized them so that the placebo treatment for age 55 is the reference. Table 11.8 shows the table of the resulting risk factors. Recall that the smaller the risk factor is, the smaller is the risk of dying and the longer is the survival time.

Figure 11.13 shows a plot of the estimated risk factors, from

which the nonlinear and non-additive nature of the model is clear. For the high treatment dosage there is a notable change from reducing the risk relative to placebo for patients younger than about 70 years while it increases the risk for the older patients.

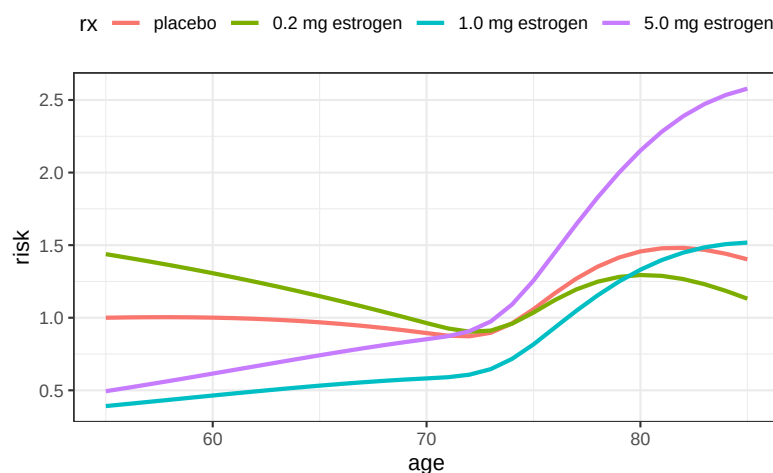


Figure 11.13: Estimated risk factors as a function of age and treatment relative to placebo treatment for patient of age 55.

Figure 11.14 shows the fitted survival functions corresponding to the risk factors in Table 11.8. It gives a different perspective on how the estimated survival curves change with age, and how the interaction effect makes the treatment with the high dosage appear beneficial for younger patients while harmful for older patients.

```
survfit(prostate_cox_int, newdata = pred_frame) |>
  summary(data.frame = TRUE) |>
  left_join(pred_frame) |>
  ggplot(aes(time, surv, color = rx)) +
  geom_step(
    data = summary(prostate_KM, data.frame = TRUE),
    aes(color = NULL),
    linetype = 2
  ) +
  geom_step() +
  facet_wrap("age", ncol = 4) +
  theme(legend.position = "top")
```

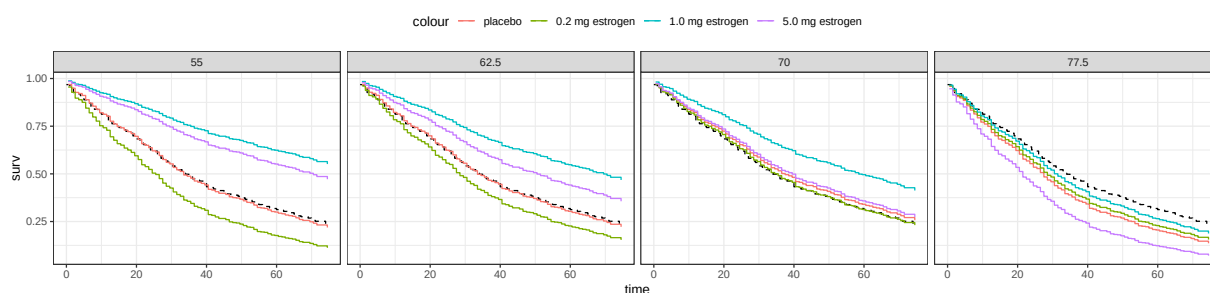


Figure 11.14: Estimated survival functions for the Cox interaction model. The dashed curve is the Kaplan-Meier estimator for the entire data.

We have not quantified the uncertainty in the above figures. One way to globally test if there is a treatment effect in this interaction model is via a likelihood-ratio test (using the partial likelihood). Table 11.9 shows that the p -value is about 0.003 and thus fairly small. Based on this test we therefore reject the hypothesis that there is no treatment effect.

```
prostate_cox_age <- coxph(
  Surv(dtime + 0.5, status != "alive") ~ ns(age, df = 4),
  data = prostate,
)
anova(prostate_cox_age, prostate_cox_int) |>
  tidy() |>
  knitr::kable()
```

term	logLik	statistic	df	p.value
~ ns(age, df = 4)	-1999.203	NA	NA	NA
~ ns(age, df = 4) + age * rx	-1989.345	19.7167	6	0.0031098

Table 11.9: Likelihood-ratio test of the hypothesis that there is no treatment effect.

Today, estrogen treatment of prostate cancer has largely been replaced by better alternatives. Though it is documented to be effective against the cancer, a serious known sideeffect is increased risk of cardiovascular disease. This knowledge is in line with our findings for this particular study.

11.6 Discrete time survival models

We have in previous sections mostly considered survival distributions that are continuous, that is, distributions on $[0, \infty)$ with a continuous survival function S . It was typically even differentiable with $-S'$ the density of the distribution, in which case the hazard function is well defined. In this chapter we will briefly touch on discrete survival distributions, and almost exclusively on distributions with point masses in the observations $T_1, \dots, T_n$. We assume throughout this section that there are no ties.

Definition 11.10. With observations $T_1, \dots, T_n$ and P a probability measure on $[0, \infty)$ with survival function S , the empirical likelihood with right censoring is

$$L(P) = \prod_i P(T_i)^{\delta_i} S(T_i)^{1-\delta_i}.$$

The factors in the empirical likelihood are simply the probabilities

$$P(T_i)^{\delta_i} S(T_i)^{1-\delta_i} = \mathbf{P}(T^* = T_i, \delta = \delta_i) = \begin{cases} \mathbf{P}(T^* = T_i) & \text{if } \delta_i = 1 \\ \mathbf{P}(T^* > T_i) & \text{if } \delta_i = 0 \end{cases}$$

Recall also that for the discrete probability measure P , the survival function is

$$S(t) = \mathbf{P}(T^* > t) = \sum_{i: T_i > t} P(T_i).$$

We will show in this section that the Kaplan-Meier estimator can be derived as the maximizer of the empirical likelihood.

11.6.1 Discrete hazards

In the previous section we only introduced hazards for continuous distributions. The *discrete hazards* are by definition the conditional probabilities

$$\lambda_i := \frac{P(T_i)}{S(T_i-)} = \mathbf{P}(T^* = T_i \mid T^* \geq T_i).$$

Note that the discrete hazards are probabilities and thus constrained to lie in the interval $[0, 1]$.

Lemma 11.5. *For a discrete probability measure with point masses in $T_1, \dots, T_n$ and corresponding hazards λ_i the survival function is*

$$S(t) = \prod_{i: T_i \leq t} (1 - \lambda_i).$$

Proof. Note that $S(T_{(j)}-) = S(T_{(j-1)})$ for $j \geq 2$ and that it is sufficient to show the identity for $t = T_{(j)}$, $j = 1, \dots, n$.

The proof is by induction. For $T_{(1)}$ we have $S(T_{(1)}-) = 1$ and

$$S(T_{(1)}) = 1 - P(T_{(1)}) = 1 - \lambda_{(1)},$$

which gives the induction start.

For the induction step, observe that the definition of the hazard implies that

$$\begin{aligned} S(T_{(j)}) &= S(T_{(j)}-) - P(T_{(j)}) \\ &= S(T_{(j)}-) - S(T_{(j)}-)\lambda_{(j)} \\ &= S(T_{(j-1)})(1 - \lambda_{(j)}), \end{aligned}$$

which gives the induction step. $\square$

Lemma 11.5 shows that the discrete hazards determine the survival function and thus the distribution, but in a slightly different way than for continuous distributions. As a direct corollary we can express the empirical likelihood in terms of the discrete hazards.

Corollary 11.2. *With $Y_i = Y(T_i) = \sum_j 1(T_i \leq T_j)$ the number of individuals at risk at time T_i , the empirical likelihood equals*

$$L(P) = \prod_i \lambda_i^{\delta_i} (1 - \lambda_i)^{Y_i - \delta_i}. \quad (11.24)$$

Proof. From the definition of the hazards and Lemma 11.5

$$P(T_i) = \lambda_i S(T_i-) = \prod_{j: T_j \leq T_i} (1 - \lambda_j).$$

We then plug this expression into the empirical likelihood function and interchange the product order to get

$$\begin{aligned} L(P) &= \prod_i \lambda_i^{\delta_i} \prod_{j: T_j < T_i} (1 - \lambda_j)^{\delta_i} \prod_{j: T_j \leq T_i} (1 - \lambda_j)^{1 - \delta_i} \\ &= \prod_i \lambda_i^{\delta_i} (1 - \lambda_i)^{Y_i - \delta_i}. \end{aligned}$$

□

11.6.2 Empirical likelihood estimators

In terms of the empirical likelihood we define the nonparametric maximum-likelihood estimator (NPMLE) of the survival distribution as the maximizer of L . To compute it we maximize L as a function of the discrete hazards. To this end we need the following lemma.

Lemma 11.6. *With $y \geq 1$ the function*

$$\lambda \mapsto \lambda(1 - \lambda)^{y-1}$$

attains a unique maximum over $[0, 1]$ in $1/y$.

Proof. With $f(\lambda) = \lambda(1 - \lambda)^{y-1}$ we have

$$\begin{aligned} f'(\lambda) &= (1 - \lambda)^{y-1} - \lambda(y - 1)(1 - \lambda)^{y-2} \\ &= (1 - \lambda)^{y-2}(1 - \lambda - \lambda(y - 1)) \\ &= (1 - \lambda)^{y-2}(1 - \lambda y). \end{aligned}$$

For $y = 1$, the function is the identity, which attains its maximum in $1 = 1/y$. For $y > 1$ we have $f(\lambda) \geq 0$ for $\lambda \in [0, 1]$, $f(0) = f(1) = 0$ and the function has a unique stationary point in $(0, 1)$ in $1/y$ with $f(1/y) > 0$. Thus $1/y$ is the unique maximizer in $[0, 1]$ for $y \geq 1$. □

Theorem 11.2. *The NPMLE of the hazards – the hazards for the probability measure that maximizes the empirical likelihood – is given by*

$$\hat{\lambda}_i = \frac{\delta_i}{Y_i}.$$

Proof. The factors of L in (11.24) equal

$$\lambda_i^{\delta_i} (1 - \lambda_i)^{Y_i - \delta_i} = \begin{cases} (1 - \lambda_i)^{Y_i} & \text{if } \delta_i = 0 \\ \lambda_i (1 - \lambda_i)^{Y_i - 1} & \text{if } \delta_i = 1 \end{cases}$$

In the first case, when $\delta_i = 0$, the factor is maximized in 0, and in the second case, when $\delta_i = 1$, the factor is maximized in $1/Y_i$. □

By plugging in the NPMLE of the hazards in the formula in Lemma 11.5 for the survival function we find that the NPMLE of the survival function is

$$\hat{S}(t) = \prod_{i:T_i \leq t} \left(1 - \frac{\delta_i}{Y_i}\right).$$

We recognize this as the Kaplan-Meier estimator.

In terms of the discrete hazards the corresponding jump cumulative hazard function is

$$\Lambda(t) = \sum_{i:T_i \leq t} \lambda_i.$$

If we plug in the NPMLE of the hazards in this formula, we get

$$\hat{\Lambda}(t) = \sum_{i:T_i \leq t} \frac{\delta_i}{Y_i},$$

which we recognize as the Nelson-Aalen estimator. The previously observed

discrepancy between the Kaplan-Meier and Nelson-Aalen estimators was due to the relation in (11.9) assuming a continuous distribution.

The two main points from this section are:

- The Kaplan-Meier estimator has a justification as a nonparametric maximum-likelihood estimator
- When we use the *correct* relation between hazards and the survival function for discrete distributions, as given by Lemma 11.5, there is no discrepancy between the Kaplan-Meier and Nelson-Aalen estimators.

11.6.3 The discrete proportional hazards model

It is legitimate to consider the proportional hazards model for discrete probability distributions, and use the empirical likelihood function L for profiling. Contrary to the derivations in Section 11.4.2, that use the Poisson empirical likelihood, the maximizer is not explicit. Moreover, the proportionality assumption for the discrete model does not really provide an approximation of the proportionality assumption for the continuous model.

The appropriate generalization of the proportional hazards model to the discrete case is actually the model

$$\lambda_i = 1 - (1 - \lambda_{0,i})^{w_i}. \quad (11.25)$$

The empirical likelihood becomes

$$L = \prod_i (1 - (1 - \lambda_{0,i})^{w_i})^{\delta_i} (1 - \lambda_{0,i})^{W_i - w_i \delta_i},$$

and similar arguments as in the proof of Theorem 11.2 will show that the NPMLE of $\lambda_{0,i}$ for given weights is

$$\hat{\lambda}_{0,i} = \delta_i \left(1 - \left(1 - \frac{w_i}{W_i} \right)^{1/w_i} \right).$$

The corresponding profile likelihood is

$$\prod_{i:\delta_i=1} \frac{w_i}{W_i} \left(1 - \frac{w_i}{W_i} \right)^{W_i/w_i - \delta_i}. \quad (11.26)$$

The discrete proportional hazards model (11.25) is arguably the “correct” generalization of the continuous proportional hazards model, see Jacobsen (1984), and using the empirical likelihood yields the profile likelihood (11.26). The resulting profile likelihood is not Cox’s partial likelihood, but it is for most practical purposes very close. In the literature, Cox’s partial likelihood has caught on as the default choice for semiparametric estimation of parameters that enter the weights.

Jacobsen, Martin. 1984. “Maximum Likelihood Estimation in the Multiplicative Intensity Model: A Survey.” *International Statistical Review* 52 (2): 193–207.

Exercises

Exercise 11.1. Show that if T has a log-logistic distribution with density $f = f_0$ then

$$e^{\eta T}$$

has density f_η . Show that if T has a log-logistic distribution with density f_η then

$$\log T - \eta$$

has a logistic distribution.

Exercise 11.2. Show that without censoring then the Kaplan-Meier estimator equals the empirical survival function

$$\hat{S}(t) = \frac{1}{n} \sum_{i=1}^n 1(T_i^* > t).$$

Exercise 11.3. In this exercise you should consider the prostate dataset from the `RwR` package and this time include the variable `hx` in the analysis. The variable `hx` indicates if the patient has a history of cardiovascular disease. You should ignore the treatment variable `rx`.

1. Fit a proportional hazards model using the two variables `age` and `hx` as predictors.
2. Make one or more residual plots for the model you have fitted. Does it fit the data?
3. Test the hypothesis that survival is not associated with `hx` given `age`.

Exercise 11.4. Let $T_1^*, \dots, T_n^*$ be independent survival times for n patients and let $X_i \in \{0, 1\}$ indicate if the i -th patient is treated ($X_i = 1$) or untreated ($X_i = 0$). We are interested in estimating the treatment effect, which we summarize in the following *parameter of interest*

$$\rho := \frac{\text{median}(T_1^* \mid X_1 = 1)}{\text{median}(T_1^* \mid X_1 = 0)},$$

that is, ρ is the ratio of the median survival time for treated vs. untreated patients.

In this exercise it is assumed that T_i^* follows a Weibull distribution with shape parameter $\gamma > 0$ and scale parameter α_{x_i} , that is, the hazard rate is

$$\lambda_i(t) = \alpha_{x_i} \gamma t^{\gamma-1}.$$

1. Show that

$$\rho = \sqrt[\gamma]{\frac{\alpha_0}{\alpha_1}}$$

in terms of the Weibull parameters.

2. Show that the Weibull model can be formulated as a proportional hazard model in terms of the linear predictor

$$\eta_i = \beta_0 + \beta_1 X_i$$

and relate the parameters β_0 and β_1 to α_0 and α_1 . Express ρ in terms of the β_0 and β_1 parameters.

In the remaining part of this exercise we will study two different estimators of ρ on simulated data. To this end introduce i.i.d. censoring times $C_1, \dots, C_n$ and the censored observed survival times

$$T_i = \min\{T_i^*, C_i\}.$$

The data set `testWei` available from the R package `RwR` is a simulated data set with $n = 200$, $X_1 = \dots = X_{100} = 0$ and $X_{101} = \dots = X_{200} = 1$, with $\gamma = 2$, $\beta_0 = 0$ and $\beta_1 = -1$ (so that $\rho \approx 1.649$). The censoring times are uniformly distributed on $[0, 4]$. The `time` column contains the T_i -s and the `status` column is the indicator of censoring (1 is uncensored, 0 is censored).

3. Compute and plot the Kaplan-Meier estimates of the survival function within each of the two treatment groups for the simulated `testWei` data set.
4. Use the Weibull model with fixed $\gamma = 2$ to estimate β_0 and β_1 . Compare the resulting estimated survival functions from the Weibull model with the Kaplan-Meier estimates.
5. Compute the MLE $\hat{\rho}_{\text{MLE}}$ of ρ based on the Weibull model (with fixed $\gamma = 2$) and compute a 95% confidence interval for ρ . Compute also, based on the Kaplan-Meier estimates, a direct estimate, $\hat{\rho}_{\text{KM}}$, of ρ .
6. Write an R program that simulates a data set (including the censoring) using the same setup as for the `testWei` data set described above. Simulate $B = 200$ data sets and compute for each data set the estimates $\hat{\rho}_{\text{KM}}$ and $\hat{\rho}_{\text{MLE}}$ as above and compare their distributions.

Hint: Use the R function `rweibull()` to simulate from the Weibull distribution, but pay attention to the parametrization.

Exercise 11.5. For this exercise we consider the data set `lung` on lung cancer survival available from the R package `RwR`. It contains four variables:

Variable	Description
time	survival time in days
status	censored = 1, dead = 2
age	age in years
sex	male = 1, female = 2

1. Plot the Kaplan-Meier estimates of the survival functions for males and females separately.
2. Use the Cox proportional hazard model to investigate if the survival time depends upon age.
3. Check if the Cox proportional hazard model fits the data.

Exercise 11.6. This exercise is about the proportional hazard model with cumulative hazard function

$$\Lambda_i(t) = e^{\eta_i} \Lambda_0(t)$$

for the i -th individual and linear predictor

$$\eta_i = \beta_0 + \beta_{\text{age}} X_{i,\text{age}} + \beta_{\text{sex}} X_{i,\text{sex}}.$$

Thus the two predictor variables are the age and sex of the individual.

You will need that data set `simSurv` from the R package `RwR` that contains the four variables, `surv`, `status`, `age` and `sex` being the censored survival times, the censoring indicator, the age and the indicator of gender (1 is male), respectively, of the individuals. There is a total of 200 individuals.

1. Estimate, using Cox's proportional hazard model, the parameters β_{age} and β_{sex} . Compute a 95% confidence interval for β_{sex} and determine whether males have a higher risk of dying than females.

For the remaining questions the baseline cumulative hazard function is assumed to be

$$\Lambda_0(t) = e^{\gamma t} - 1, \quad t \geq 0.$$

The cumulative hazard function for the i -th individual is thus

$$\Lambda_i(t) = e^{\eta_i} (e^{\gamma t} - 1).$$

We assume that we have independent censored observations $(T_1, \delta_1), \dots, (T_n, \delta_n)$ with $T_i \geq 0$ the i -th censored survival time and δ_i the indicator of whether the i -th observation is censored.

2. Compute the baseline hazard rate $\lambda_0(t)$ and show that the log-likelihood function is

$$\sum_i \delta_i \eta_i + \delta_i \log(e^{\gamma T_i} - 1) - e^{\eta_i} (e^{\gamma T_i} - 1) + \delta_i \log \frac{\gamma e^{\gamma T_i}}{e^{\gamma T_i} - 1}.$$

3. Plot the profile log-likelihood for the γ -parameter. Compute the MLE of γ , β_0 , β_{age} and β_{sex} .

Hint: Identify the first two terms of the log-likelihood as the log-likelihood for a Poisson generalized linear model. Use the `glm()`-function for the profiling.

4. Use the glm framework to compute a 95% confidence interval for β_{sex} by plugging in the MLE, $\hat{\gamma}$, of γ and pretending that it is fixed and known. Compute next a 95% confidence interval using non-parametric bootstrapping. Compare the results.

Exercise 11.7. This exercise deals with the numerical algorithms associated with the optimization of Cox's partial likelihood. Consider censored survival data (T_i, δ_i, X_i) for $i = 1, \dots, n$ with T_i the possibly censored survival time, δ_i the indicator of whether T_i is uncensored, and $X_i \in \mathbb{R}$ a univariate real valued predictor. The proportional hazard model is given by the individual weights

$$w_i = e^{\beta X_i}$$

for an unknown parameter β . In the following we will make the assumption that the survival times are ordered and with no ties, that is

$$T_1 < T_2 < \dots < T_n.$$

1. Show that Cox's partial log-likelihood equals

$$\ell(\beta) = \sum_{i: \delta_i=1} \beta X_i - \log \left(\sum_{j \geq i} e^{\beta X_j} \right),$$

and compute the first and second derivative of ℓ w.r.t. β .

The estimator $\hat{\beta}$ is the maximizer of ℓ provided that ℓ attains a maximum. It then solves the equation

$$\ell'(\beta) = 0.$$

2. Show that ℓ is concave. Find a condition on the X_i -s that ensure that ℓ has a unique maximum.

For the remaining part of this problem you first need to simulate a data set from the model above. You may experiment with different choices of β , but try also $\beta = 0$.

3. Implement Newton's algorithm for computing $\hat{\beta}$. Apply the algorithm with start value $\beta_1 = 0$ to the simulated data set and investigate the values of β_m , $\ell(\beta_m)$, $\ell'(\beta_m)$ and $\ell''(\beta_m)$ in the iterations of the algorithm.

4. Compute and plot an estimate of the baseline cumulative hazard function.
5. Compute $\hat{\beta}$ using the `coxph()` function in R. Compute also a standard 95% confidence interval for β . Compare this interval with a (partial) likelihood interval based on ℓ .

A naive implementation of the computation of e.g. ℓ scales like n^2 . Make sure that your implementation scales like n .

Exercise 11.8. This exercise is on theoretical properties of the survival analysis martingale residuals. Let T^* and C be independent random variables on $[0, \infty)$, whose distributions have continuous densities f and g , respectively. The corresponding survival functions are denoted S and H . Note that $f = -S'$ and $g = -H'$. Then $T = \min\{T^*, C\}$ denotes the observed (censored) survival time and $\delta = 1(T^* \leq C)$ is the status indicator.

1. Show that for any function $K : [0, \infty) \times \{0, 1\} \rightarrow \mathbb{R}$, continuously differentiable in the first variable, it holds that

$$\begin{aligned} \mathbf{E}(K(T, \delta)) &= \int_0^\infty \left(K(0, 1) + (K(s, 0) - K(s, 1))S(s) \right. \\ &\quad \left. + \int_0^s \partial_t K(t, 1)S(t)dt \right) g(s)ds \end{aligned}$$

provided that the expectation is well defined. Use this to show that⁵⁹

⁵⁹ That the expectation is 0 is derived in the main text, but a slightly different derivation is provided here. The variance formula is new.

$$\mathbf{E}(\delta - \Lambda(T)) = 0 \quad \text{and} \quad \mathbf{V}(\delta - \Lambda(T)) = \mathbf{E}(\Lambda(T))$$

where Λ denotes the cumulative hazard function for the distribution given by f .

Recall that for the proportional hazard model, the i -th individual has cumulative hazard function $w_i \Lambda_0$ where w_i is an individual weight and Λ_0 is the baseline hazard.

Assume that $\hat{w}_i$ is the estimate of the i -th weight and that $\hat{\Lambda}_0$ is the estimate of the baseline. With $\hat{\Lambda}_i = \hat{w}_i \hat{\Lambda}_0(T_i)$ the fitted cumulative hazard function for the i -th individual evaluated in T_i ,

$$\hat{r}_i = \delta_i - \hat{\Lambda}_i$$

is the i -th martingale residual. Let $\hat{d}_i$ denote the i -th deviance residual⁶⁰.

2. Give an argument for considering the following standardization of the martingale residuals:

$$\tilde{r}_i = \frac{\hat{r}_i}{\sqrt{\hat{\Lambda}_i}}.$$

3. Show that if $\delta_i = 1$ then

$$\hat{d}_i^2 = \tilde{r}_i^2 + o((1 - \hat{\Lambda}_i)^2).$$

4. Explore graphically how good the approximation

$$\hat{d}_i \approx \tilde{r}_i$$

is as a function of $\hat{\Lambda}_i$ when $\delta_i = 1$. Does a similar approximation hold if $\delta_i = 0$?

⁶⁰ The deviance residuals are sometimes regarded as distribution stabilized transforms of the martingale residuals. This exercise gives some insight on this perspective.

12 Company default – a case study

! Important

This chapter is still a rough draft of a case study on modeling time to company default using survival modeling techniques. For now it will not be substantially revised.

12.1 Outline and prerequisites

This chapter is based on a dataset collected by the winners of a competition (Business Analytics Challenge, 2017). The data considered in this chapter is a subset of the full dataset. It contains some key financial variables from the annual reports of Danish companies from the period 2013 to 2016. The dataset also contains information about whether the companies defaulted during the time period. Our objective is to build a predictive model of the time until default of a company and use this model to explore the associations between the various financial predictors and this survival distribution of the company.

The chapter primarily relies on the following R packages.

```
library(RwR)
library(ggplot2)
library(tibble)
library(tidyr)
library(skimr)
library(lubridate)
library(broom)
library(splines)
library(dplyr)
library(survival)
```


12.2 Exploratory analysis

In this section we load the data and present some aspects of the dataset. We reorganize and clean the data to bring the data on a form where it is suitable for modeling the survival distribution of companies.

```
data(default, package = "RwR")
```

The first nine rows of the dataset.

fiscal_start	fiscal_end	default	equity	profit	assets	return_on_assets	CVR	start
2013-01-01	2013-12-31	0	39217	-76713	56399	-1.3601837	10000025	2013-01-01
2014-01-01	2014-12-31	0	-798210	-47427	357316	-0.1327313	10000025	2014-01-01
2015-01-01	2015-12-31	0	-972885	-174675	389334	-0.4486508	10000025	2015-01-01
2013-01-01	2013-12-31	0	67255000	20058000	194020000	0.1033811	10000157	1993-01-01
2014-01-01	2014-12-31	0	96773000	24198000	226974000	0.1066113	10000157	1994-01-01
2015-01-01	2015-12-31	0	112820000	16185000	200219000	0.0808365	10000157	1995-01-01
2014-01-01	2014-12-31	0	694457	128263	739690	0.1734010	10000211	1996-01-01
2013-01-01	2013-12-31	0	566194	58620	942710	0.0621824	10000211	1997-01-01
2015-01-01	2015-12-31	0	764365	69908	804631	0.0868821	10000211	1998-01-01

Each row in the dataset corresponds to an annual report from a Danish company. The dataset was constructed in May, 2017, by extracting annual reports for all Danish companies from the period 2013–2016.

Variables:

- **fiscal_start**: Start date of fiscal year.
- **fiscal_end**: End date of fiscal year.
- **default**: Indicator of whether the company defaulted.
- **equity**: Equity in DKK from the annual report that fiscal year.
- **profit**: Profit in DKK from the annual report that fiscal year.
- **assets**: Assests in DKK from the annual report that fiscal year.
- **return_on_assets**: The profit / assets (ROA) ratio.
- **CVR**: The unique Danish company identification number as character.
- **start**: Registered start date of company.

- **end**: Registered end date of company. Missing if the company is not registered as closed.
- **industry_type**: A categorical variable.
- **econ_activity**: A numerical variable indicating the economic activity in spatial proximity of the company.
- **competition_density**: A numerical variable indicating the competition in spatial proximity of the company.

We reorganize the data so that each row corresponds to a single company and contains the data from the earliest annual report in the dataset for that company. The **end** variable is missing if the company was not closed when the data was gathered, and we choose to replace these missing values by the administrative censoring date “2017-05-01”.

```
default <- group_by(default, CVR) |>
  summarize(fiscal_end = min(fiscal_end, na.rm = TRUE)) |>
  inner_join(default, y = _) |>
  mutate(end = replace_na(end, as_date("2017-05-01")))
```

Since the annual report is not publicly available immediately after the end of the fiscal year we decide that the baseline date for the company is five months after the end of the fiscal year. We clean the data by removing all rows where:

- The fiscal year starts after it ends (data error).
- The company is closed within five months after the end of the fiscal year.
- The company has a start date later than the end of the fiscal year.

```
default <- filter(
  default,
  fiscal_start < fiscal_end,
  fiscal_end + months(5) < end,
  start <= fiscal_end
)
```

```
skim(default)
```

We introduce two new variables.

- **age**: The age in days of the company at baseline.
- **followup**: The time from baseline in days and until the company is either closed or administratively censored.

Variable type: character							
skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
CVR	0	1	8	8	0	227204	0
industry_type	37	1	6	29	0	20	0
Variable type: Date							
skim_variable	n_missing	complete_rate	min	max	median	n_unique	
fiscal_start	0	1	2011-11-29	2016-11-25	2013-03-01	1398	
fiscal_end	0	1	2012-12-31	2016-11-30	2013-12-31	194	
start	0	1	1735-06-10	2016-11-25	2007-11-15	14351	
end	0	1	2014-01-28	2017-05-17	2017-05-01	910	
Variable type: factor							
skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts		
default	0	1	FALSE	2	0: 217881, 1: 9323		
Variable type: numeric							
skim_variable	n_missing	complete_rate	mean	sd	p0	p	
equity	1	1.00	25336557.89	1.873027e+09	-7.217768e+10	45314.	
profit	4	1.00	1481978.49	1.698615e+08	-4.249600e+10	-28084.	
assets	5503	0.98	55588303.44	5.377255e+09	-3.106368e+06	278214.	
return_on_assets	14586	0.94	70.20	2.526159e+05	-8.531832e+07	-0.	
econ_activity	10175	0.96	61761.28	4.513554e+04	1.159632e+04	28498.	
competition_density	5234	0.98	117.41	2.076000e+02	0.000000e+00	11.	

Table 12.1: Data summary of the default data using the `skim()` function.

```

default <- mutate(
  default,
  age = as.numeric(fiscal_end + months(5) - start),
  followup = as.numeric(end - (fiscal_end + months(5)))
)

```

We will model `followup` as the censored survival time until default with the status indicator being `default == 1`. The baseline is the end of the company's first fiscal year plus five months. The predictor variables we will use are all available at baseline, and we do, e.g., not use information from subsequent annual reports. Though such information might be informative about the trajectories taken by companies that end up defaulting, we cannot meaningfully use the information as predictors at baseline.

A fraction of companies have missing values of `assets` and a larger fraction has missing values of `return_on_assets`. Upon inspection it turns out that `return_on_assets` is missing if either `assets` is missing or equal to zero. Below we impute both values as zero if they are missing.

We also note from the summary statistics above that all the financial variables span multiple orders of magnitude. We therefore choose to nonlinearly normalize these variables to bring them on a common scale. This can be done in many ways, e.g., by using the marginal empirical distributions to transform each variable to the unit interval. Below we choose to first standardize variables by the inter-quartile range and then to transform them to the interval $(-1, 1)$ by the (scaled) arctangent function. This handles the heavy tails well and the zero remains interpretable. We finally log-transform age.

```

# norm <- function(x) rank(x) / length(x) # quantile normalization
# norm <- function(x) atan((x - median(x, na.rm = TRUE)) / IQR(x, na.rm = TRUE))
norm <- function(x) 2 * atan( x / IQR(x, na.rm = TRUE)) / pi
default <- mutate(default,
  assets = replace_na(assets, 0),
  return_on_assets = replace_na(return_on_assets, 0),
  equity_norm = norm(equity),
  assets_norm = norm(assets),
  profit_norm = norm(profit),
  return_norm = norm(return_on_assets),
  log_age = log10(age)
)

```

After the normalize and transformation it is meaningful to explore the pairwise associations between the predictor variables, e.g., in terms of the scatter plot matrix in Figure 12.1. The association patterns are complicated for several different reasons. For once, because there are certain logical constraints among the variables (`equity` and `assets`, `profit` and `return`).

```
ggally_hexbin <- function(data, mapping, ...)  
  ggplot(data, mapping) + stat_binhex(...)  
ggally_spear <- function(...) GGally::ggally_cor(..., method = "spearman", stars = FALSE)  
  
GGally::ggpairs(  
  na.omit(default),  
  columns = c("log_age", "equity_norm", "assets_norm", "profit_norm", "return_norm"),  
  upper = list(continuous = "spear"),  
  lower = list(continuous = GGally::wrap("hexbin", bins = 150)),  
  ) + scale_fill_continuous(trans = "log")
```

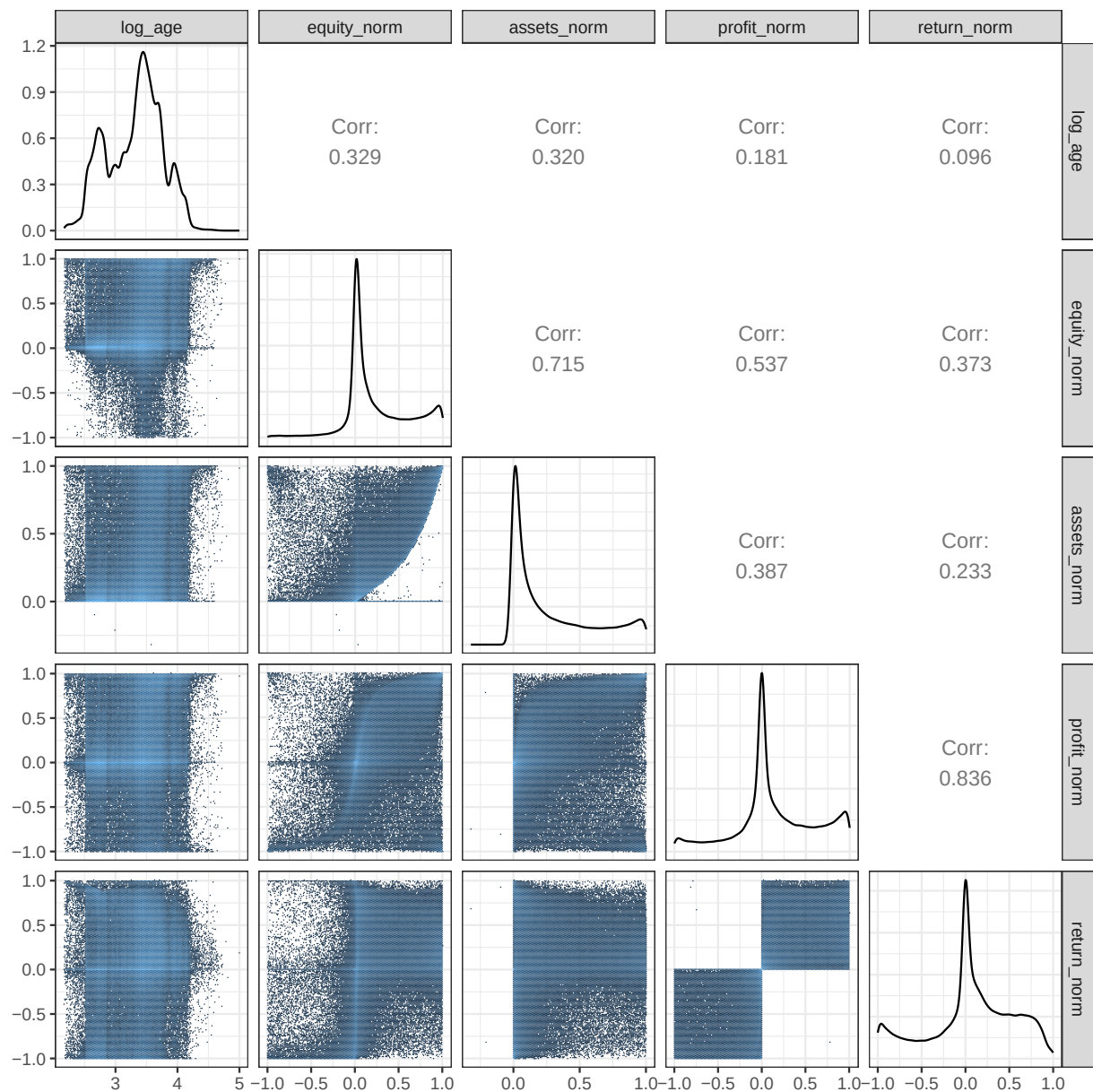


Figure 12.1: Scatter plot matrix of the normalized numerical variables and corresponding Spearman correlations.

12.3 A Cox model

We illustrate the modeling of company default by fitting a Cox model using an additive model on the log hazard ratio scale using nonlinear spline expansions of the five numerical

predictors. We choose to only consider one of the industry types corresponding to about 30000 of the companies.

```
form <- Surv(followup, default == 1) ~
  ns(log_age, df = 5) + ns(equity_norm, df = 5) + ns(assets_norm, df = 5) +
  ns(profit_norm, df = 5) + ns(return_norm, df = 5)
default_trade <- filter(default, industry_type == "Handel")
default_cox <- coxph(form, data = default_trade)
```

The baseline estimate of the survival function for the Cox model is then computed.

```
baseline <- survfit(default_cox) |>
  summary(data.frame = TRUE)

ggplot(baseline, aes(time, surv)) +
  geom_ribbon(aes(ymin = lower, ymax = upper), fill = "gray60", alpha = 0.4) +
  geom_step(color = "#3366FF") +
  ylim(0.8, 1) +
  scale_x_continuous(
    "followup time",
    breaks = c(0, 365, 2 * 365, 3 * 365),
    labels = c("baseline", "one year", "two years", "three years")) +
  ylab("survival probability")
```

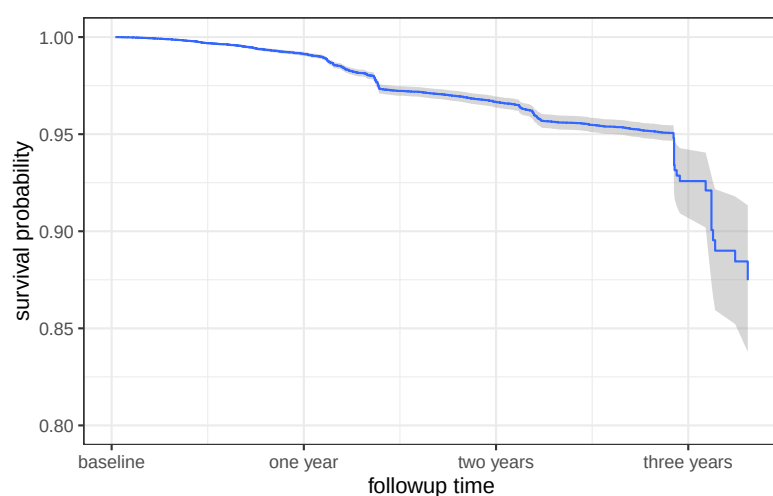


Figure 12.2: Baseline survival function for the Cox model of default.

12.3.1 Model predictions

To understand the model better we first explore model predictions. Figure 12.3 shows the empirical distribution of the log hazard ratios on the dataset. Recall that for the proportional

hazards model, a log hazard ratio of -2.5 corresponds to a multiplicative effect of $e^{-2.5} = 0.082 \approx 1/12$ on the (cumulative hazard), while a log hazard ratio of 2.5 corresponds to a multiplicative effect of $e^{2.5} \approx 12$.

```
default_pred <- predict(default_cox)
```

```
probs <- seq(0.1, 0.9, 0.1)
prob_names <- paste("p =", probs)
qt <- quantile(default_pred, probs = probs)
ggplot(mapping = aes(x = default_pred)) +
  geom_histogram(bins = 100, fill = "gray60") +
  xlab("log hazard ratio (linear predictor)") +
  geom_rug(
    data = tibble(default_pred = qt, decile = prob_names),
    aes(color = decile),
    show.legend = TRUE
  )
```

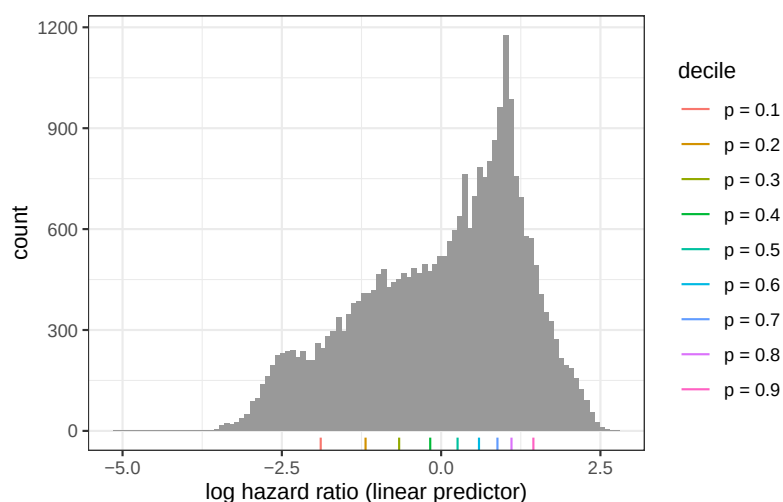


Figure 12.3: Distribution of the log hazard ratio (the linear predictor) on the data. The colored marks are the deciles of the distribution.

To visualize the range of predicted survival functions for different companies, depending on their predictors, Figure 12.4 shows the survival curves for the different deciles of the distribution of the log hazard ratios.

```
baseline_aug <- baseline
for (i in seq_along(probs)) {
  baseline_aug$x = qt[i]
  names(baseline_aug)[ncol(baseline_aug)] <- prob_names[i]
}
surv_range <- pivot_longer(
```



```

baseline_aug,
cols = all_of(prob_names),
names_to = "decile",
values_to = "lhr"
) |>
mutate(
  cumhaz = exp(lhr) * cumhaz,
  surv = exp(-cumhaz)
)

ggplot(surv_range, aes(time, surv, color = decile)) +
  geom_step() +
  ylim(0.6, 1) +
  scale_color_brewer(palette = "Set1") +
  scale_x_continuous(
    "followup time (years)",
    breaks = c(0, 365, 2 * 365, 3 * 365),
    labels = c("baseline", "one year", "two years", "three years")) +
  ylab("survival probability")

```

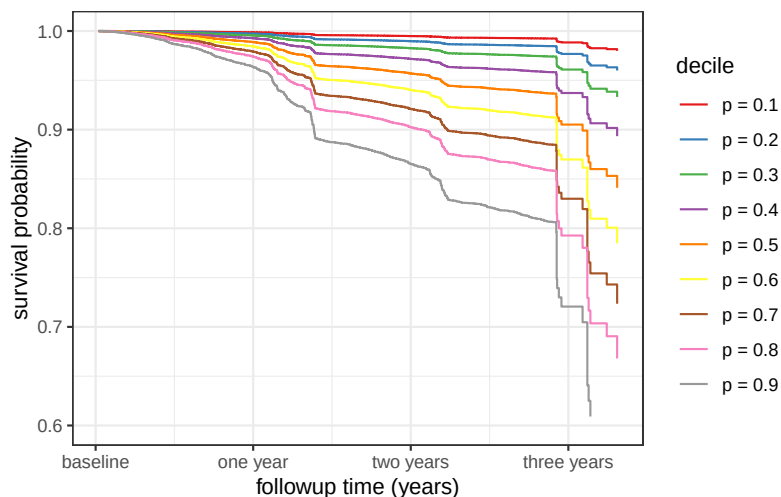


Figure 12.4: Survival functions for the different deciles of the distribution of the log hazard ratio.

We will finally show how each of the individual nonlinear functions in the formula specification are estimated. This can be done using the `predict()` function specifying `type = "terms"` to get a prediction for each term in the formula. Figure 12.5 shows the five different functions including a pointwise 95% confidence band based on the standard errors that are also computed by the `predict()` function.

```

probs <- seq(0.01, 0.99, 0.01)
pred_frame <- reframe(
  default,

```

```

    log_age = quantile(log_age, probs = probs, na.rm = TRUE),
    equity_norm = quantile(equity_norm, probs = probs, na.rm = TRUE),
    assets_norm = quantile(assets_norm, probs = probs, na.rm = TRUE),
    profit_norm = quantile(profit_norm, probs = probs, na.rm = TRUE),
    return_norm = quantile(return_norm, probs = probs, na.rm = TRUE)
  )
pred_default <- predict(
  default_cox,
  newdata = pred_frame,
  type = "terms",
  se.fit = TRUE
)

```

```

p <- ggplot(mapping = aes(x, y, ymin = y - 1.96 * se, ymax = y + 1.96 * se)) +
  geom_hline(yintercept = 0, color = "red", linetype = 2) +
  geom_ribbon(fill = "grey60", alpha = 0.4) +
  geom_line(color = "#3366FF") + coord_cartesian(ylim = c(-2, 1.5)) + ylab("")

p_bar <- ggplot(default) + geom_histogram(bins = 40) + ylab("") +
  scale_y_continuous(label = \(x) rep(" ", length(x))) +
  theme(
    axis.title.x = element_blank(),
    axis.text.x = element_blank(),
  )

gridExtra::grid.arrange(
  p_bar + aes(x = log_age) + coord_cartesian(xlim = range(pred_frame$log_age)),
  p_bar + aes(x = equity_norm) + coord_cartesian(xlim = range(pred_frame$equity_norm)),
  p_bar + aes(x = assets_norm) + coord_cartesian(xlim = range(pred_frame$assets_norm)),
  p_bar + aes(x = profit_norm) + coord_cartesian(xlim = range(pred_frame$profit_norm)),
  p_bar + aes(x = return_norm) + coord_cartesian(xlim = range(pred_frame$return_norm)),
  p %>% tibble(
    x = pred_frame$log_age,
    y = pred_default[[1]][, 1], se = pred_default[[2]][, 1]
  ) + xlab("log_age") + ylab("log hazard ratio"),
  p %>% tibble(
    x = pred_frame$equity_norm,
    y = pred_default[[1]][, 2], se = pred_default[[2]][, 2]
  ) + xlab("equity_norm"),
  p %>% tibble(
    x = pred_frame$assets_norm,
    y = pred_default[[1]][, 3], se = pred_default[[2]][, 3]
  ) + xlab("assets_norm"),
  p %>% tibble(
    x = pred_frame$profit_norm,
    y = pred_default[[1]][, 4], se = pred_default[[2]][, 4]
  ) + xlab("profit_norm"),
  p %>% tibble(
    x = pred_frame$return_norm,
    y = pred_default[[1]][, 5], se = pred_default[[2]][, 5]
  ) + xlab("return_norm"),

```

```

ncol = 5,
heights = c(1, 4)
)

```

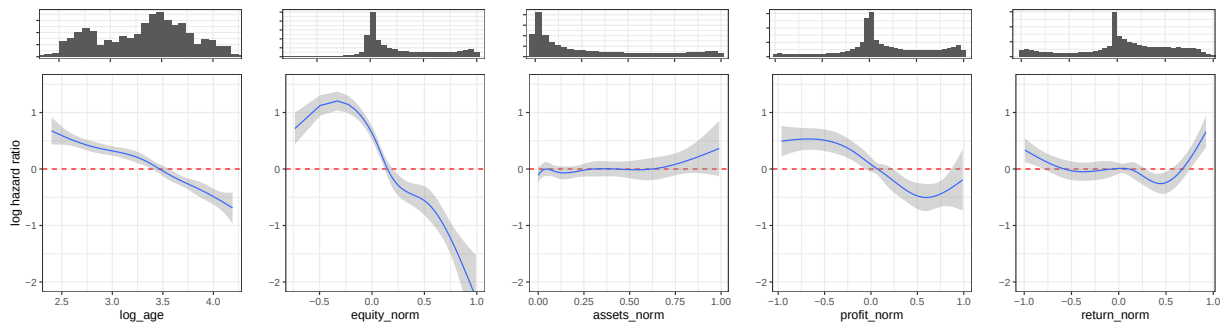


Figure 12.5: Term plot for the five spline terms in the Cox model of company default. The blue band is a pointwise 95% confidence band. The histograms above indicate the marginal distributions of the variables in the data.

12.3.2 Model diagnostic

We investigate the model fit using residual plots.

```

p1 <- augment(default_cox, data = default_trade, type.predict = "expected") |>
  survfit(Surv(.fitted, default == 1) ~ 1, data = _) |>
  summary(data.frame = TRUE) |>
  ggplot(aes(time, cumhaz)) +
  geom_abline(slope = 1, color = "blue") +
  geom_point(alpha = 0.2) +
  xlab("Cox-Snell residuals") +
  ylab("Cumulative hazard")

binScale <- scale_fill_continuous(
  breaks = c(1, 10, 100, 1000, 10000),
  low = "gray80",
  high = "black",
  trans = "log",
  guide = "none"
)

p2 <- augment(default_cox) |>
  ggplot(aes(.fitted, .resid)) +
  geom_hex(bins = 200) +
  binScale +
  geom_smooth() +
  xlab("Fitted values") +
  ylab("Martingale residuals") +

```

```
coord_cartesian(xlim = c(-2, 4), ylim = c(-1, 1))

gridExtra::grid.arrange(p1, p2, ncol = 2)
```

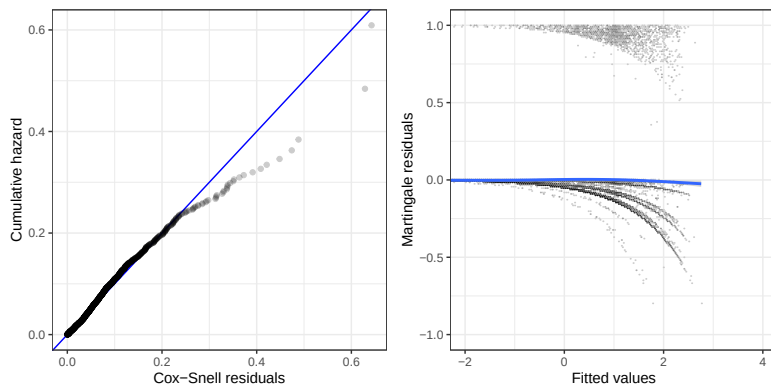


Figure 12.6: Residual diagnostics for the Cox model of default.

Figure 12.6 shows that the Cox model does not fit the data perfectly.

Figure 12.7 investigates the proportional hazards assumption more specifically. It is constructed by stratifying the data according to quantiles of the linear predictor and then fit a nonparametric survival function for each strata. If the proportional hazards assumption holds, the corresponding cumulative hazard functions should be roughly proportional. This is easiest to assess if they are further log-transformed, because on that scale proportionality corresponds to translations.

It is common to also log-transform the time scale (right plot in Figure 12.7). The cumulative hazard function often has the shape of a power function, which on the log-log scale becomes a straight line. Proportionality implies that these lines are roughly vertical translations of each other. Deviations appear in either plot as either direct crossings of the curves or just differences in slope or shape among the curves. In this case there might to be some issues with the proportionality assumption, both when comparing the low risk companies (blue) to the other companies and when comparing the two medium risk groups (green) to each other.

```
probs <- seq(0, 1, length.out = 7)
qt <- quantile(default_pred, probs = probs)
labels <- paste("p =", cumsum(diff(probs)))
```

```

default_trade$eta <- default_pred
default_diag <- survfit(Surv(followup, default == 1) ~ cut(eta, breaks = qt), data = default_trade) |>
  summary(data.frame = TRUE) |>
  mutate(strata = factor(strata, labels = labels))

p1 <- ggplot(default_diag, aes(time, cumhaz, color = strata)) +
  geom_step(linewidth = 0.8) +
  scale_y_log10() +
  scale_color_brewer(palette = "Paired")

p1 <- p1 + theme(legend.position = "none")

gridExtra::grid.arrange(p1, p1 + scale_x_log10(), ncol = 2)

```

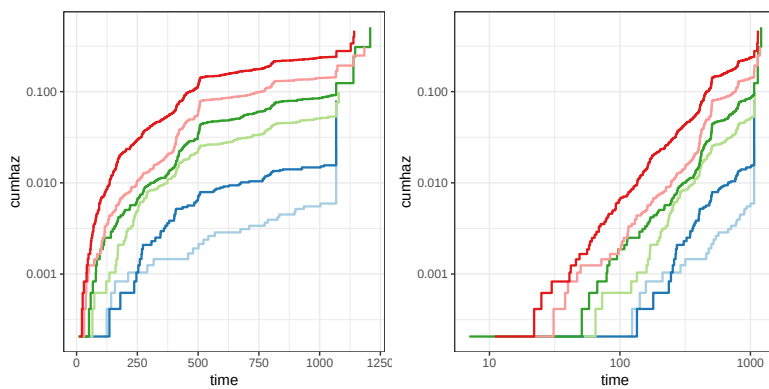


Figure 12.7: The log cumulative hazard function fitted to strata based on the quantiles of the linear predictor. In the right plot the time axis is on a log scale.

13 Interval estimation

Warning

This chapter is still a draft and minor changes are made without notice.

This chapter gives a treatment of the theory and methodology behind interval estimation. While most of the preceeding chapters focus on point estimation of the parameters, the focus of this chapter is on interval estimation of a univariate parameter of interest. The purpose of an interval estimate is to communicate a *range* of plausible parameter values for the given data rather than a single parameter – and in this way also quantify the uncertainty about a parameter estimate.

13.1 Outline and prerequisites

Standard confidence intervals based on Z -scores is one way to construct interval estimates, and this method will be treated here in a general context of interval estimation. Likelihood based interval estimation is a different approach to constructing intervals, which will be treated in some detail, and their connection to the standard confidence intervals will be established.

The bootstrap is covered as a general method for standard error estimation and interval calibration to ensure that the computed interval estimators have desired sampling properties.

Some of the theory covered is not specific to regression models, and in the general presentation there are no explicit predictor variables. The observation vector will in the general case be denoted $\mathbf{Y}$ and consist of n independent random variables.

For regression models this vector then either represents the responses conditionally on the predictors or the joint observation of the responses and predictors.

The chapter primarily relies on the following R packages.

```
library(RwR)
library(ggplot2)
library(tibble)
library(splines)
library(dplyr)
library(sandwich)
library(lmtest)
```

13.2 Interval estimators

The focus of the entire chapter is on *univariate* parameters of interest, and we will denote such a parameter generically by γ . It is, in general, a function of the distribution of $\mathbf{Y}$, which we denote P . We can thus abstractly write $\gamma = \gamma(P)$ as a function of P . For a particular model, the parameter of interest will often be a very explicit function and it may depend on only aspects of the distribution of $\mathbf{Y}$.

Example 13.1 (Generalized linear models). The parameter of interest in a generalized linear regression model can be one of the β -parameters, a linear combination of the β -parameters, the linear predictor $X^T\beta$ or the mean value $\mu(X^T\beta)$ for a given predictor X . All these are examples⁶¹ where $\gamma = \gamma(\beta)$ is actually a function of the regression parameters, in which case a natural estimator of γ is the plug-in estimator

$$\hat{\gamma} = \gamma(\hat{\beta}). \quad (13.1)$$

If $\hat{\gamma}$ denotes an estimate of γ , a simple *interval estimate* based on $\hat{\gamma}$ is of the form $\hat{\gamma} \pm z$, or more generally

$$(\hat{\gamma} - z, \hat{\gamma} - w) \quad (13.2)$$

for some choices of $z \geq w$ (w is typically negative). It is, however, not obvious how we should choose z and w , or what properties such an interval has.

⁶¹ Examples that only rely on the semiparametric mean-variance specification given by GA1 and GA2. If the parameter of interest is a quantile of $Y | X$, say, it is still a function of β , but a plug-in estimator then also relies on GA3.

We may note that (13.2) can be written as

$$(\hat{\gamma} - z, \hat{\gamma} - w) = \{\gamma \in \mathbb{R} \mid \hat{\gamma} - \gamma \in (w, z)\},$$

which shows that

$$\mathbf{P}((\hat{\gamma} - z, \hat{\gamma} - w) \ni \gamma) = \mathbf{P}(\hat{\gamma} - \gamma \in (w, z)).$$

The probability⁶² $\mathbf{P}((\hat{\gamma} - z, \hat{\gamma} - w) \ni \gamma)$ is a *coverage probability* of the random interval containing the parameter γ . As shown above, this probability can be expressed in terms of the distribution of $\hat{\gamma} - \gamma$.

We generalize the simple construction above by introducing the concept of a *combinant statistic* $R(\mathbf{Y}, \gamma) \in \mathbb{R}$, which is a function of the observations and the (unknown) parameter of interest. We could, for instance, take

$$R(\mathbf{Y}, \gamma) = \hat{\gamma} - \gamma$$

as above.

For a fixed $\gamma = \gamma_0$, the random variable $R(\mathbf{Y}, \gamma_0)$ is a statistic, and it is typically a test statistic of the simple hypothesis $H_0 : \gamma = \gamma_0$. We will subsequently refer to R interchangeably as a combinant statistic or just as a statistic, where we use the former when we want to emphasize that R combines the data and the parameter of interest.

Definition 13.1. For a combinant statistic R and $z \geq w$ we define the two-sided interval⁶³ estimate

$$I(\mathbf{Y}) = \{\gamma \in \mathbb{R} \mid R(\mathbf{Y}, \gamma) \in (w, z)\}, \quad (13.3)$$

and for $c \geq 0$ the one-sided interval estimate

$$I(\mathbf{Y}) = \{\gamma \in \mathbb{R} \mid R(\mathbf{Y}, \gamma) < c\}. \quad (13.4)$$

The definitions above are explicit about the intervals $I(\mathbf{Y})$ being functions of data only, and we think about the mapping $\mathbf{Y} \mapsto I(\mathbf{Y})$ as an interval estimator. Specific choices of R , w and z (or c) determine the sampling properties of such estimators. We often aim for constructing intervals with a particular coverage probability.

⁶² This coverage probability is evaluated for a distribution P of $\mathbf{Y}$ with $\gamma(P) = \gamma$, and it depends in general on P , but see Definition 13.3.

⁶³ We use the term “interval” in a liberal sense. The construction will not result in an interval for all combinant statistics, but we want to avoid introducing an abstract term (such as a “set” or “region” estimate) when most practical constructions will result in intervals.

Definition 13.2 (Coverage and confidence). An interval estimator $I(\mathbf{Y})$ of the parameter of interest γ has *coverage* $1 - \alpha$ if

$$\mathbf{P}(I(\mathbf{Y}) \ni \gamma) \geq 1 - \alpha. \quad (13.5)$$

Such an interval is called a *confidence interval* with *confidence level* $1 - \alpha$.

The definition is, in a certain sense, straightforward, but to fully comprehend it we need to unfold the implicit requirements about the distribution, P , of $\mathbf{Y}$. It is implicit that P belongs to some model, that is, a family of probability distributions, and it is also implicit that $\gamma(P) = \gamma$. The coverage statement about an interval estimator is always relative to a model, and it is possible that an interval estimator has coverage $1 - \alpha$ relative to one model but not relative to a (larger) model.

When the interval estimator is constructed from the combinant statistic R , the coverage probability can (in the two-sided case) be expressed as

$$\mathbf{P}(I(\mathbf{Y}) \ni \gamma) = \mathbf{P}(R(\mathbf{Y}, \gamma) \in (w, z)) = R(P, \gamma(P))((w, z))$$

where $R(P, \gamma(P))$ denotes the push-forward measure by R of the distribution P of $\mathbf{Y}$.

Definition 13.3 (Pivot). For a given model of the sampling distribution, P , of $\mathbf{Y}$, and for a parameter of interest $\gamma = \gamma(P)$, the combinant statistic R is said to be a *pivot* if the distribution of $R(\mathbf{Y}, \gamma)$ does not depend on P . That is, if the probability measure $R(P, \gamma(P))$ is independent of P .

If we can find a pivot R then it is particularly easy to construct intervals with guaranteed coverage. If $w = w_\alpha$ and $z = z_\alpha$ are the $\alpha/2$ - and $(1 - \alpha/2)$ -quantiles of the probability distribution $R(P, \gamma(P))$, the two-sided interval (13.3) has coverage⁶⁴

$$\mathbf{P}(I(\mathbf{Y}) \ni \gamma) = R(P, \gamma(P))((w_\alpha, z_\alpha)) = 1 - \alpha/2 - \alpha/2 = 1 - \alpha$$

For the one-sided interval (13.4) we choose $c = c_\alpha$ to be the $(1 - \alpha)$ -quantile of the pivot's distribution, which likewise gives the coverage guarantee

$$\mathbf{P}(I(\mathbf{Y}) \ni \gamma) = R(P, \gamma(P))((-\infty, c_\alpha)) = 1 - \alpha.$$

⁶⁴ To get exact equality here we assume that the distribution of $R(\mathbf{Y}, \gamma)$ is continuous. It is always possible to guarantee coverage but perhaps not with exact equality.

It is, unfortunately, only for a few special models that we can find exact pivots, and in most applications we rely on approximate pivots. Section 13.3.3 deals with several methods for choosing w and z or c to *calibrate* the interval estimates to have (approximately) a desired coverage probability for a given combinant statistic R . In the following sections we focus on constructions and interpretations of the various statistics that are used in practice for interval estimation.

13.2.1 Standard confidence intervals

With $\hat{s}e$ any estimate of the standard error⁶⁵ of $\hat{\gamma}$ we can construct the Z -statistic

$$Z = \frac{\hat{\gamma} - \gamma}{\hat{s}e}.$$

With $z \geq 0$ and $w = -z$ in the two-sided interval based on the Z -statistic we find the corresponding symmetric interval estimate

$$\hat{\gamma} \pm z \cdot \hat{s}e = (\hat{\gamma} - z \cdot \hat{s}e, \hat{\gamma} + z \cdot \hat{s}e).$$

We could also use the statistic Z^2 to construct the one-sided interval. If we choose $c = z^2$ we arrive at the same symmetric interval as above.

Example 13.2. Consider the linear model under the strong modeling assumptions A3 and A5. That is, the joint distribution of the responses given the predictors is a normal distribution. If the parameter of interest is $\gamma = a^T\beta$, the corresponding Z -statistic is

$$Z_a = \frac{\hat{\gamma} - \gamma}{\hat{\sigma} \sqrt{a^T(\mathbf{X}^T\mathbf{X})^{-1}a}}.$$

Theorem 2.3 implies that its distribution is a t_{n-p} -distribution, which does not depend on the distribution of $\mathbf{Y}$ – as long as A3 and A5 are fulfilled. Thus for the model specified by A3 and A5, Z_a is a pivot and we know its exact distribution. Its square,

$$Z_a^2 = \frac{(\hat{\gamma} - \gamma)^2}{\hat{\sigma}^2 a^T(\mathbf{X}^T\mathbf{X})^{-1}a}$$

is likewise a pivot, now with an $F_{1,n-p}$ -distribution.

Quantiles of the t_{n-p} -distribution (or the $F_{1,n-p}$ -distribution) depend on the residual degrees of freedom $df = n - p$ but are

⁶⁵ Either as an analytic formula as in Example 13.2 or computed via bootstrapping as in Example 13.6.

straightforward to compute. As $\text{df} \rightarrow \infty$, the t_{n-p} -distribution will converge to a standard normal distribution $\mathcal{N}(0, 1)$, and the $F_{1, n-p}$ -distribution will converge to a χ_1^2 -distribution.

Beyond the normal model specified by the assumptions A3 and A5 we can rarely find the exact distribution of Z -statistics. However, we typically use the approximation $Z \stackrel{\text{approx}}{\sim} \mathcal{N}(0, 1)$, and we then choose $z = z_\alpha$ as the $1 - \alpha/2$ quantile in the $\mathcal{N}(0, 1)$ -distribution for the construction of the corresponding intervals. This leads to the standard choice of $z = 1.96$ to give approximately⁶⁶ 95% coverage.

Formal justification of the approximation $Z \stackrel{\text{approx}}{\sim} \mathcal{N}(0, 1)$ is beyond the scope of this chapter. The standard asymptotic arguments are in terms of convergence in distribution when $n \rightarrow \infty$. We emphasize the following important points on the use of the standard intervals based on an estimate of the standard error and the approximation of the distribution of Z by a normal distribution.

- The approximation of the distribution of the Z -statistic by a standard normal distribution makes no strong assumptions about the *response* distribution. It is, in particular, not necessary for the responses to follow a normal distribution.
- The approximation may be poor if the sample size is small, but it is virtually impossible to give useful theoretical guarantees on how large n should be for the approximation to be adequate. This is instead investigated through simulation studies, but see also the bootstrap methods in Section 13.3.
- Validity of the standard error estimator, $\hat{\text{se}}$, is crucial for the resulting intervals to have good coverage properties. Poor coverage of a standard interval is often due to model misspecifications that lead to invalid estimators of the standard error, see Section 13.2.3.

⁶⁶ We often specify a *nominal* coverage of $1 - \alpha$, which is the coverage probability we aim for, e.g., by computing quantiles in an *approximate* distribution. The actual coverage probability may then be larger or smaller depending on how good the approximation is.

13.2.2 Likelihood intervals

The standard confidence intervals quantify rather explicitly the sampling uncertainty of the parameter estimate, and they are constructed with a certain nominal coverage probability in mind. In that way they appear relatively intuitive. They

do, however, not really answer the question about which parameters are the most *plausible* for the given data.

This section introduces *likelihood intervals* for a parametrized model as a principled way of constructing interval estimates of the most plausible parameter values. While a confidence interval by definition only has a coverage interpretation, likelihood intervals have intrinsic value that can be justified without reference to sampling properties. They are typically calibrated to have desired coverage properties as well but coverage is not their only virtue. The section concludes by showing that standard confidence intervals based on maximum-likelihood estimation are closely related to likelihood intervals.

For a given dataset, the likelihood ratio quantifies the relative evidence in the data of one probability distribution in our statistical model over another. In terms of the generic log-likelihood function ℓ and generic parameter β , the difference

$$\ell(\beta_1) - \ell(\beta_2)$$

thus quantifies the evidence in the data for β_1 over β_2 . If the difference is positive, there is more evidence for β_1 , and if it is negative, there is more evidence for β_2 . The maximum likelihood estimator (MLE) is the parameter – and corresponding probability distribution in our model – for which there is the largest evidence in the data.

We introduce the supremum of the log-likelihood over the parameter space as

$$\ell_{\max} := \sup_{\beta} \ell(\beta).$$

If the MLE exists and is denoted $\hat{\beta}^{\text{MLE}}$, we have $\ell_{\max} = \ell(\hat{\beta}^{\text{MLE}})$, but $\ell_{\max}$ is well defined even if the MLE is not. To capture abstractly which parameters achieve a likelihood value close to $\ell_{\max}$ we introduce the likelihood regions as sublevel sets of the negative log-likelihood.

Definition 13.4 (Likelihood regions). If $\ell_{\max} < \infty$ we define⁶⁷ the family $\mathcal{L}(c)$ for $c \geq 0$ of likelihood regions for β as

$$\mathcal{L}(c) = \{\beta \mid 2(\ell_{\max} - \ell(\beta)) < c\}.$$

The likelihood regions have the following properties⁶⁸:

⁶⁷ The “2” in the definition is an arbitrary but convenient convention.

⁶⁸ And the interpretation that $\mathcal{L}(c)$ gradually as c increases includes models for which there is less and less evidence in the data.

- If the MLE exists then $\hat{\beta}^{\text{MLE}} \in \mathcal{L}(c)$ for all $c > 0$.
- If $\beta_1 \in \mathcal{L}(c)$ and $\beta_2 \notin \mathcal{L}(c)$ then $\ell(\beta_1) > \ell(\beta_2)$.
- The likelihood regions increase as a function of c : If $c_1 < c_2$ then $\mathcal{L}(c_1) \subseteq \mathcal{L}(c_2)$.

The general (multivariate) likelihood regions are not much easier to comprehend than the likelihood itself, but we can modify the definition to give practically useful interval estimators for a univariate parameter of interest.

Definition 13.5. The profile log-likelihood for a univariate parameter $\gamma = \gamma(\beta)$ is defined as

$$\ell(\gamma) = \sup_{\beta: \gamma(\beta) = \gamma} \ell(\beta).$$

The family of likelihood intervals⁶⁹ for γ is defined as

$$\mathcal{I}(c) = \{\gamma \mid 2(\ell_{\max} - \ell(\gamma)) < c\}.$$

We note that the likelihood intervals are simply the one-sided interval estimates determined by the combinant statistic

$$R(\mathbf{Y}, \gamma) = 2(\ell_{\max} - \ell(\gamma)),$$

where $\ell(\gamma)$ is the profile log-likelihood.

The family of likelihood intervals is, just as the family of abstract likelihood regions, a nested family of sets, but as subsets of $\mathbb{R}$ they are somewhat more comprehensible and easier to communicate. And for any choice of $c > 0$, parameters in $\mathcal{I}(c)$ are more plausible than parameters outside of $\mathcal{I}(c)$.

The obvious question is how the cutoff c should be chosen if we want to report a single likelihood interval? If we want the interval to be a confidence interval, we need to calibrate c to yield the desired nominal coverage, and we discuss this in greater detail in Section 13.3.3. A standard choice is $c = 3.841$ aiming for the interval $\mathcal{I}(3.841)$ to be a nominal 95% confidence interval by supposing that $2(\ell_{\max} - \ell(\gamma))$ approximately follows a χ_1^2 -distribution⁷⁰.

Absolute interpretations of log-likelihood ratios

There is an alternative to coverage for justifying a choice of the cutoff c in the construction of likelihood intervals. It is based on an absolute interpretation of log-likelihood ratios.

⁶⁹ As in other cases, the set $\mathcal{I}(c)$ needs not be an interval, but the terminology becomes less heavy if we accept this slight abuse of the word “interval”.

⁷⁰ The 95% quantile in the χ^2 -distribution with one degrees of freedom is 3.841.

The argument works by constructing a simple analogy to the abstract log-likelihood ratio scale.

The analogy supposes that we draw balls from one of two (infinite) urns, but we don't know which one it is. One urn has only white balls and the other has half black and half white balls. If we at any time draw a black ball we know for sure that we are drawing from the second urn. But if we have drawn b white balls without drawing a black we may ask: how strong is the evidence for the "all white" urn over the alternative? The log-likelihood ratio of the "all white" urn over the alternative is $b \cdot \log(2) \approx 0.69 \cdot b$, and we can regard this number as a quantification of the evidence.

Because the example is simple, we can use it to develop an understanding of the log-likelihood ratio scale. Most would agree that an observation of $b = 10$ white balls is strong evidence in favor of the "all white" urn, while $b = 3$ is not. That is, a log-likelihood ratio of $10 \cdot \log(2) = 6.931$ is "strong evidence", while a log-likelihood ratio of $3 \cdot \log(2) = 2.079$ is not. A log-likelihood ratio of 3.841 corresponds to 5.5 white balls, which seems like moderate evidence for the "all white" urn over the alternative.

The purpose of the simple urn example is to develop a reference by analogy for interpretation of log-likelihood ratios – if we are willing to interpret such ratios absolutely. In that case we can translate a general log-likelihood ratio to an interpretable quantity: the number of white balls drawn. For a thorough treatment of the likelihood principle, promoting this perspective, see the book *Statistical Evidence* by Richard Royall (1997). The urn example is covered in his Section 1.6. We will not push this interpretation too far, but it is an interesting theoretical perspective.

Royall, Richard M. 1997. *Statistical Evidence*. Vol. 71. Monographs on Statistics and Applied Probability. Chapman & Hall, London.

Example 13.3 (Generalized linear models). Suppose $a \in \mathbb{R}^p$ and that the parameter of interest is $\gamma = a^T \beta$. Suppose first that the MLE, $\hat{\beta}$, exists and that $\hat{\gamma} = a^T \hat{\beta}$. The profile log-likelihood over β (ignoring for now the dispersion parameter) is given by

$$\ell(\gamma) = \sup_{\beta: a^T \beta = \gamma} \ell(\beta) = \ell(\hat{\beta}(\gamma))$$

where $\hat{\beta}(\gamma)$ is the MLE under the linear constraint $a^T \hat{\beta}(\gamma) = \gamma$. Since we can reparametrize⁷¹ the model by $(\tilde{\beta}, \gamma)^T = \mathbf{A}\beta$ for

⁷¹ With model matrix $\mathbf{X}$ in the β -parametrization we get that $\mathbf{X}\beta = \mathbf{X}\mathbf{A}^{-1}\mathbf{A}\beta = \mathbf{X}\mathbf{A}^{-1}(\tilde{\beta}, \gamma)^T$, and the reparametrization corresponds to transforming the model matrix to $\tilde{\mathbf{X}} = \mathbf{X}\mathbf{A}^{-1}$.

an invertible matrix $\mathbf{A}$, the practical computation of $\hat{\beta}(\gamma)$ is equivalent to computing the MLE in a glm with a fixed offset parameter.

The likelihood combinant statistic is recognized as the deviance test statistic

$$2(\ell_{\max} - \ell(\gamma)) = D_0(\gamma) - D,$$

where $D_0(\gamma)$ is the deviance corresponding to the affine hypothesis $H_0 : a^T \beta = \gamma$. The computation of the likelihood interval for a fixed c proceed as follows:

- Compute $D_0(\gamma)$ for a small number of γ -values around $\hat{\gamma}$.
- Solve⁷² the equation

$$D_0(\gamma) = \hat{\varphi}c + D$$

using a smooth interpolant of the computed values, e.g., splines.

- Report the two solutions as the interval

$$\mathcal{J}(c) = (\gamma_{\text{lower}}(c), \gamma_{\text{upper}}(c)).$$

The procedure outlined above is implemented via the `confint()` method for glm objects, see the help pages `?confint.glm` and `?profile.glm`. The application of profiling for computing likelihood intervals will be demonstrated in Section 13.4.1. The standard confidence intervals given by (5.7) are based on the Z -statistic

$$Z_a = \frac{\hat{\gamma} - \gamma}{\sqrt{\hat{\varphi}a^T(\mathbf{X}^T\hat{\mathbf{W}}\mathbf{X})^{-1}a}}$$

as introduced in Section 5.3.2 and corresponding to

$$\text{s}\hat{e} = \sqrt{\hat{\varphi}a^T(\mathbf{X}^T\hat{\mathbf{W}}\mathbf{X})^{-1}a}.$$

They can be computed using `confint.lm()`. The standard intervals are faster to compute and often similar to the likelihood intervals, but the likelihood intervals often have better coverage properties, in particular for small samples, where they, for good reason, can be rather asymmetric. The likelihood intervals are actually also well defined, computable and potentially informative even if the MLE does not exist, in

⁷² We define $\gamma_{\text{lower}}(c)$ as the largest solution smaller than $\hat{\gamma}$ and $\gamma_{\text{upper}}(c)$ as the smallest solution larger than $\hat{\gamma}$. If there is no solution smaller than $\hat{\gamma}$ we set $\gamma_{\text{lower}}(c) = -\infty$, and if there is no solution larger than $\hat{\gamma}$ we set $\gamma_{\text{upper}}(c) = \infty$.

which case the left or the right (or both) endpoints equal $-\infty$ or $+\infty$, respectively.

Using the distributional approximation⁷³, cf. also Equation 5.8,

$$\frac{D_0(\gamma) - D}{\hat{\varphi}} \stackrel{\text{approx}}{\sim} \chi_1^2$$

we may choose c according to the quantiles of a χ_1^2 -distribution to achieve the desired nominal coverage. For example, with $c = 3.841$ the nominal coverage of the likelihood interval is 95%.

If the parameter of interest is $\mu = \mu(\gamma)$, where μ is the mean value function, then the likelihood interval for μ (supposing that μ is increasing) equals

$$(\mu(\gamma_{\text{lower}}(c)), \mu(\gamma_{\text{upper}}(c))).$$

That is, transforming a likelihood interval gives the likelihood interval for the transformed parameter. Contrary to standard confidence intervals it therefore does not matter which scale we choose for constructing likelihood intervals.

While it is unproblematic to compute likelihood intervals in practice in the framework of generalized linear models, as outlined above, it is of interest to understand the connection between likelihood intervals and standard confidence intervals for maximum-likelihood estimators. To this end we show the following lemma on profiling a quadratic function.

Lemma 13.1. *Let $\beta = (\alpha^T, \gamma)^T$ with $\alpha \in \mathbb{R}^{p-1}$ and $\gamma \in \mathbb{R}$. Let $h(\beta) = (\hat{\beta} - \beta)^T \mathcal{J}(\hat{\beta} - \beta)$ for a positive definite matrix $\mathcal{J}$. Then*

$$h(\gamma) = \inf_{\alpha} h((\alpha, \gamma)) = \frac{(\hat{\gamma} - \gamma)^2}{(\mathcal{J}^{-1})_{pp}}.$$

Proof. Write the matrix $\mathcal{J}$ as a block matrix

$$\mathcal{J} = \begin{pmatrix} \mathcal{J}_{11} & \mathcal{J}_{1p} \\ \mathcal{J}_{1p}^T & \mathcal{J}_{pp} \end{pmatrix}.$$

Expanding $h(\beta)$ in terms of these blocks gives

$$h(\beta) = (\hat{\alpha} - \alpha)^T \mathcal{J}_{11}(\hat{\alpha} - \alpha) + (\hat{\gamma} - \gamma)^2 \mathcal{J}_{pp} + 2(\hat{\alpha} - \alpha)^T \mathcal{J}_{1p}(\hat{\gamma} - \gamma).$$

⁷³ The theoretical justification is known as *Wilks' theorem* stating that under suitable regularity assumptions then

$$D_0(\gamma) - D \xrightarrow{\mathcal{D}} \varphi \chi_1^2$$

for $n \rightarrow \infty$ if the model is correctly specified and $\gamma = a^T \beta$.

If we transform the standard confidence interval $\hat{\gamma} \pm z \cdot \text{s}\hat{e}$ we get the interval

$$(\mu(\hat{\gamma} - z \cdot \text{s}\hat{e}), \mu(\hat{\gamma} + z \cdot \text{s}\hat{e})).$$

This interval will have the same coverage properties on the transformed scale as the standard interval has on the original scale, but it is not a standard interval on the transformed scale.

Minimization over α for fixed γ gives

$$\hat{\alpha}(\gamma) = \hat{\alpha} + (\hat{\gamma} - \gamma) \mathcal{J}_{11}^{-1} \mathcal{J}_{1p}.$$

Plugging this into $h(\beta)$ gives the profile log-likelihood

$$h(\gamma) = (\hat{\gamma} - \gamma)^2 (\mathcal{J}_{pp} - \mathcal{J}_{1p}^T \mathcal{J}_{11}^{-1} \mathcal{J}_{1p}).$$

By the general formula for inversion of block matrices the lower right corner of $\mathcal{J}^{-1}$ is

$$(\mathcal{J}^{-1})_{pp} = (\mathcal{J}_{pp} - \mathcal{J}_{1p}^T \mathcal{J}_{11}^{-1} \mathcal{J}_{1p})^{-1},$$

and the result follows. $\square$

We can use Lemma 13.1 to connect the log-likelihood intervals to the standard intervals based on the MLE and the observed Fisher information.

For a twice continuously differentiable parametric log-likelihood ℓ and with $\hat{\beta}^{\text{MLE}}$ in the interior of the (Euclidean) parameter space, a second order Taylor expansion of ℓ around $\hat{\beta}^{\text{MLE}}$ gives

$$\ell(\beta) \approx \ell_{\max} - \frac{1}{2} (\hat{\beta}^{\text{MLE}} - \beta)^T \mathcal{J}^{\text{obs}} (\hat{\beta}^{\text{MLE}} - \beta). \quad (13.6)$$

Here $\mathcal{J}^{\text{obs}} = -D^2 \ell(\hat{\beta}^{\text{MLE}})$ is the positive (semi)definite observed Fisher information. From (13.6) we get that

$$2(\ell_{\max} - \ell(\beta)) \approx (\hat{\beta}^{\text{MLE}} - \beta)^T \mathcal{J}^{\text{obs}} (\hat{\beta}^{\text{MLE}} - \beta).$$

Suppose that the j -th coordinate β_j is our parameter of interest and that $\mathcal{J}^{\text{obs}}$ is positive definite, then Lemma 13.1 gives the following quadratic approximation of the likelihood combinant statistic

$$2(\ell_{\max} - \ell(\beta_j)) = \inf_{\beta_{-j}} 2(\ell_{\max} - \ell(\beta)) \approx \frac{(\hat{\beta}_j^{\text{MLE}} - \beta_j)^2}{((\mathcal{J}^{\text{obs}})^{-1})_{jj}}, \quad (13.7)$$

where the infimum over β_{-j} means over all coordinates in β except β_j . We see that the right hand side above equals Z^2 for the Z -statistic

$$Z = \frac{\hat{\beta}_j^{\text{MLE}} - \beta_j}{\sqrt{((\mathcal{J}^{\text{obs}})^{-1})_{jj}}}$$

OBS: For generalized linear models the dispersion parameter can be ignored for the computation of the MLE of the regression parameters, but it has to be included for the computation of likelihood intervals, cf. also Example 13.3. Similarly, the log-likelihood approximations presented here suppose that the dispersion parameter estimate is included, cf. Theorem 5.1

that corresponds to the standard error estimate

$$\widehat{\text{se}} = \sqrt{((\mathcal{J}^{\text{obs}})^{-1})_{jj}}.$$

We emphasize the following key points about likelihood intervals.

- Likelihood intervals have an intrinsic interpretation: the parameter values located centrally in the interval (those that remain in the interval for small c) are more plausible than those close to the boundaries.
- Likelihood intervals are invariant under reparametrizations: we get the same intervals irrespectively of how the model is parametrized.
- The approximation (13.7) allows us to interpret standard confidence intervals based on the MLE as approximate likelihood intervals. For this reason they inherit the interpretations of likelihood intervals, but the quality of the approximation depends on the parametrization.
- The `confint()` method for glm objects compute likelihood intervals by profiling. Standard confidence intervals can be computed by calling `confint.lm()` directly for the glm object.

13.2.3 Robust standard errors

In Chapter 5 on generalized linear models as well as in Chapter 11 on survival models we derived maximum-likelihood estimators of the regression parameters. As discussed in Chapter 5, there is some robustness of the glm estimators toward misspecification of the response distribution. If just the conditional mean and variance are correctly specified, the estimators as well as the statistical inference based on these estimators are still valid. Similarly, the partial likelihood estimator for the proportional hazards model does not require a correct specification of the full response distribution either – the estimators and the statistical inference remain valid as long as the proportional hazards assumption holds and the hazard weights are correctly specified.

In applications we attempt to justify the assumptions we *do* make via model diagnostics, and we correct misspecifications if possible. In Chapter 10, for example, the first Poisson model

of vegetable sales showed a clear but fixable misspecification of the mean-variance relation. It is, however, difficult to eliminate all model misspecifications, either because they are not detectable by the model diagnostics or because they are not easily fixable. Thus we cannot give any guarantees that the model is correctly specified. To partially remedy this we derive in this section analytic methods for standard error estimation that are somewhat robust toward model misspecifications.

We will in this section focus on the regression setup with observations $(X_1, Y_1), \dots, (X_n, Y_n)$ assumed i.i.d. That is, we make assumption A6 on the joint distribution of the observations. The regression parameter is $\beta \in \mathbb{R}^p$, and we consider estimators given as solutions to an *estimating equation*

$$\mathcal{U}(\beta) = \sum_{i=1}^n u(X_i, Y_i, \beta) = 0, \quad (13.8)$$

for some estimating function $u : \mathbb{R}^p \times \mathbb{R} \times \mathbb{R}^p \rightarrow \mathbb{R}^p$. In practice, $\mathcal{U}$ might be a (transposed) score function, but the methods we derive apply to the partial score equation and any other estimating equation.

For a general estimating function we *define* the target value of the parameter β to be the solution to the equation

$$\mathbf{E}(u(X, Y, \beta)) = \int u(x, y, \beta) P(dx, dy) = 0, \quad (13.9)$$

where P denotes the joint distribution of (X, Y) . That is, for *any* joint distribution we define the parameter β in terms of P via (13.9).

Example 13.4. The estimating equation for a generalized linear model is given in terms of the estimating function

$$\begin{aligned} u(X, Y, \beta) &= \theta'(X^T \beta)(Y - \mu(X^T \beta))X \\ &= \frac{\mu'(X^T \beta)}{\mathcal{V}(\mu(X^T \beta))}(Y - \mu(X^T \beta))X, \end{aligned} \quad (13.10)$$

cf. Lemma 5.1 and Theorem 5.1. If the mean is misspecified, the β parameter is determined as the solution to the equation

$$E \left(\frac{\mu'(X^T \beta)}{\mathcal{V}(\mu(X^T \beta))} (\mathbf{E}(Y | X) - \mu(X^T \beta))X \right) = 0.$$

If the mean is correctly specified as $\mathbf{E}(Y | X) = \mu(X^T \beta)$, then this value of β also solves the equation above.

Let $\hat{\beta}$ denote a solution to the estimating equation $\mathcal{U}(\beta) = 0$, then by a Taylor expansion of $\mathcal{U}$ around $\hat{\beta}$ we find

$$\mathcal{U}(\beta) \approx \mathcal{U}(\hat{\beta}) + \hat{J}(\beta - \hat{\beta}) = \hat{J}(\beta - \hat{\beta})$$

where

$$\hat{J} := D_{\beta}\mathcal{U}(\hat{\beta}) = \sum_{i=1}^n D_{\beta}u(X_i, Y_i, \hat{\beta}).$$

Since $\mathbf{E}(u(Y_i, X_i, \beta)) = 0$ the central limit theorem implies that

$$\frac{1}{n}\hat{J}(\hat{\beta} - \beta) \approx \frac{1}{n} \sum_{i=1}^n u(X_i, Y_i, \beta) \stackrel{\text{as}}{\sim} \mathcal{N}(0, C_1/n)$$

where

$$C_1 := \mathbf{V}(u(X, Y, \beta)) = \mathbf{E}(u(X, Y, \beta)u(X, Y, \beta)^T).$$

With

$$\hat{C} := \sum_{i=1}^n u(X_i, Y_i, \hat{\beta})u(X_i, Y_i, \hat{\beta})^T$$

then $\hat{C}/n$ is an estimator of C_1 , and with $Z \sim \mathcal{N}(0, \mathbf{I})$ we arrive at the distributional approximation

$$\hat{\beta} - \beta \stackrel{\mathcal{D}}{\approx} n\hat{J}^{-1}(\hat{C}/n^2)^{1/2}Z = \hat{J}^{-1}\hat{C}^{1/2}Z \sim \mathcal{N}(0, \hat{J}^{-1}\hat{C}\hat{J}^{-T}).$$

The estimator

$$\hat{\Sigma} := \hat{J}^{-1}\hat{C}\hat{J}^{-T} \quad (13.11)$$

of the asymptotic covariance matrix is known as the *sandwich estimator*.

If the estimating equation is a score equation, Bartlett's second identity states that *when the model is correctly specified* $C_1 = \mathbf{E}(D_{\beta}u(X, Y, \beta))$, in which case $\hat{C}\hat{J}^{-T} \approx \mathbf{I}$. So for a correctly specified model we can simply use the inverse of the observed Fisher information $\hat{J}^{-1}$ as an estimator of the asymptotic covariance matrix. The sandwich estimator is an alternative that does not assume Bartlett's second identity, and it is (somewhat) robust against model misspecifications.

Example 13.5. For the estimating equation based on generalized linear models,

$$\hat{J} = \mathcal{J}^{\text{obs}} = \mathbf{X}^T \hat{\mathbf{W}}^{\text{obs}} \mathbf{X},$$

The arguments made here are not rigorous but can be made so by sufficiently many regularity assumptions. The important point is to understand the motivation for the *sandwich estimator* defined below as a robust variance estimator.

where $\hat{\mathbf{W}}^{\text{obs}}$ is the diagonal matrix given by Equation 5.1 evaluated in $\hat{\beta}$. We also find that

$$\hat{C} = \sum_{i=1}^n \frac{(\hat{\mu}'_i)^2}{\mathcal{V}(\hat{\mu}_i)^2} (Y_i - \hat{\mu}_i)^2 X_i X_i^T = \mathbf{X}^T \widetilde{\mathbf{W}} \mathbf{X},$$

where $\widetilde{\mathbf{W}}$ is diagonal with entries

$$\tilde{w}_{ii} = \frac{(\hat{\mu}'_i)^2}{\mathcal{V}(\hat{\mu}_i)^2} (Y_i - \hat{\mu}_i)^2.$$

For generalized linear models we typically use the estimated Fisher information rather than the observed Fisher information, that is, we replace $\hat{\mathbf{W}}^{\text{obs}}$ by $\hat{\mathbf{W}}$, which is the diagonal weight matrix with weights

$$\hat{w}_{ii} = \frac{(\hat{\mu}'_i)^2}{\mathcal{V}(\hat{\mu}_i)}.$$

We then arrive at the sandwich estimator for the (asymptotic) covariance matrix of the glm estimator:

$$\tilde{\Sigma} = (\mathbf{X}^T \hat{\mathbf{W}} \mathbf{X})^{-1} (\mathbf{X}^T \widetilde{\mathbf{W}} \mathbf{X}) (\mathbf{X}^T \hat{\mathbf{W}} \mathbf{X})^{-1}. \quad (13.12)$$

The weights $\tilde{w}_{ii}$ can be represented in two ways in terms of $\hat{w}_{ii}$ and either the Pearson residuals or the working residuals. First we see that

$$\tilde{w}_{ii} = \hat{w}_{ii} \frac{(Y_i - \hat{\mu}_i)^2}{\mathcal{V}(\hat{\mu}_i)}, \quad (13.13)$$

where the weights $\hat{w}_{ii}$ are multiplied by the squared Pearson residual, and then that

$$\tilde{w}_{ii} = \left(\hat{w}_{ii} \frac{Y_i - \hat{\mu}_i}{\hat{\mu}'_i} \right)^2, \quad (13.14)$$

where the weights $\hat{w}_{ii}$ are multiplied by the working residuals and everything is then squared.

Note that if the mean is correctly specified (GA1 holds) then $\mathbf{E}((Y_i - \mu_i)^2 | X_i) = \mathbf{V}(Y_i | X_i)$, and⁷⁴

$$\mathbf{E}(\tilde{w}_{ii} | X_i) \approx \hat{w}_{ii} \frac{\mathbf{V}(Y_i | X_i)}{\mathcal{V}(\hat{\mu}_i)}.$$

If the variance function is also correctly specified (GA2 holds), then $\mathbf{V}(Y_i | X_i) = \varphi \mathcal{V}(\mu_i)$ and $\mathbf{E}(\tilde{w}_{ii} | X_i) \approx \varphi \hat{w}_{ii}$,

⁷⁴ We don't get exact equality because the weights are computed using the estimated parameters.

and the sandwich formula (13.12) will be relatively close to $\hat{\varphi}(\mathbf{X}^T \hat{\mathbf{W}} \mathbf{X})^{-1}$. But if either or both of GA1 or GA2 are wrong, the weights $\tilde{w}_{ii}$ can be interpreted as adjustments for the misspecification of the model.

Standard error estimates using the sandwich estimator can be computed using the sandwich R package. We can use these standard error estimates for construction of standard confidence intervals. This will be illustrated in Section 13.4.2.

It almost sounds too good to be true that we can compute valid confidence intervals without making *any* model assumptions – and in a certain sense it is, as argued forcefully by Freedman (2006). If the mean value is wrongly specified, it is often not possible to interpret the target parameter, which makes it very unclear what the value of the confidence interval is – even if it technically has the correct coverage. Thus the robust intervals for generalized linear models are primarily useful when GA2 might be violated but GA1 holds.

Freedman, David A. 2006. “On The So-Called ‘Huber Sandwich Estimator’ and ‘Robust Standard Errors.’” *The American Statistician* 60 (4): 299–302.

13.3 Bootstrap

Bootstrapping is an alternative to asymptotic theory, or other theoretical arguments, for computing approximations of the sampling distribution of a statistic or a combinatorial statistic of interest. We will in this section focus on how bootstrapping can be used to construct confidence intervals for a parameter of interest. We will first describe the basic resampling framework of bootstrapping and outline various bootstrap resampling algorithms before we discuss in greater detail how the bootstrap is used for calibration of interval estimators.

13.3.1 Bootstrap basics

Bootstrapping is usually based on simulations rather than analytic methods. Specifically, on *resampling* of new datasets.

The two basic steps of the bootstrap are:

1. Sample B independent new datasets $\mathbf{Y}_1^*, \dots, \mathbf{Y}_B^*$.
2. Compute the statistic of interest from each resampled data set.

The statistic is typically either a parameter estimator, a test statistic or a combinant statistic. Its resampling distribution is also known as its *bootstrap distribution*.

From the B datasets it is possible to compute estimates of descriptive quantities of the bootstrap distribution, such as its standard deviation, the bias of an estimator, or quantiles of a combinant statistic. Section 13.3.3 discusses the principles behind using the bootstrap distribution to calibrate combinant-based intervals to have the desired coverage. These principles have nothing to do with the actual resampling but have everything to do with how well the distribution we resample from approximates the target sampling distribution. We postpone this discussion until we have developed the more operational resampling aspect of bootstrapping.

In practice, everything we are interested in about the bootstrap distribution is estimated from the empirical distribution based on the B resampled datasets. The finite number B of resampled datasets introduces a sampling error, and B should be chosen as large as possible to avoid this error to influence on the reported results. An appropriate choice is dictated by the trade-off between computational costs and the accuracy required by the application. We will give more advice on the choice of B below depending on the specific application of the bootstrap.

One of the most basic usages of the bootstrap is as a tool to compute an estimate of the standard error of an estimator.

Example 13.6 (Bootstrap standard error estimation). Let γ denote the parameter of interest and let $\hat{\gamma} = \hat{\gamma}(\mathbf{Y})$ denote the estimate based on the the data $\mathbf{Y}$. From the resampled data $\mathbf{Y}_1^*, \dots, \mathbf{Y}_B^*$ we compute reestimates

$$\hat{\gamma}_b^* = \hat{\gamma}(\mathbf{Y}_b^*)$$

of the parameter of interest. Then we estimate the *standard error* of $\hat{\gamma}$ as

$$\widehat{\text{se}} = \sqrt{\frac{1}{B-1} \sum_{b=1}^B (\hat{\gamma}_b^* - \bar{\gamma}^*)^2}$$

with $\bar{\gamma}^* = \frac{1}{B} \sum_{b=1}^B \hat{\gamma}_b^*$. We can use $\widehat{\text{se}}$ to construct the standard 95% confidence interval

$$\hat{\gamma} \pm 1.96 \cdot \widehat{\text{se}}.$$

For estimation of the standard error, $B = 200$ is usually enough for acceptable precision.

13.3.2 Bootstrap algorithms

The two major resampling algorithms are known as *nonparametric* and *parametric* bootstrapping. They have an abstract form and some specific variants for regression models. A third resampling algorithm, known as *residual resampling*, is specific to regression models, and the *wild bootstrap*, that we will not cover, should also be mentioned as a regression specific bootstrap algorithm.

Nonparametric bootstrapping

Nonparametric bootstrapping does not require a parametric model, but it does require some distributional assumptions that allow us to sample new data sets from an empirical distribution. In the abstract formulation the assumption is that $\mathbf{Y} = (Y_1, \dots, Y_n)$ with $Y_1, \dots, Y_n$ i.i.d. Each resampled data set is obtained by sampling n observations from $Y_1, \dots, Y_n$ *with replacement*. This is in practice carried out by sampling n independent indices uniformly from $\{1, \dots, n\}$. A resampled data set

$$\mathbf{Y}^* = (Y_1^*, \dots, Y_n^*)$$

thus consists of n i.i.d. observations from the empirical distribution

$$\sum_{i=1}^n \delta_{Y_i}.$$

In a regression context the response variables $Y_1, \dots, Y_n$ are **not** identically distributed conditionally on the predictors. Thus we cannot just resample the responses with replacement, say. But under assumption A6 the *pairs* (Y_i, X_i) are i.i.d.. This may be a reasonable assumption when the predictors and the response are observed jointly, while it makes little sense for an experiment where the predictors are fixed by a design.

Pair sampling for nonparametric bootstrapping in a regression setup proceeds under assumption A6 by resampling n pairs independently from the dataset $(X_1, Y_1), \dots, (X_n, Y_n)$ *with replacement*. That is, the resampled dataset consists of n

i.i.d. observations $(X_1^*, Y_1^*), \dots, (X_n^*, Y_n^*)$ from the empirical distribution

$$\sum_{i=1}^n \delta_{(X_i, Y_i)}.$$

The actual implementation is still done in practice by sampling n independent indices uniformly from $\{1, \dots, n\}$.

Parametric bootstrapping

Parametric bootstrapping amounts to resampling data from the parametrically specified model of the data. In the abstract formulation the distribution of $\mathbf{Y}$ is assumed to be in a parametrized family P_β of probability measures, and $\hat{\beta}$ is the estimate of β based on $\mathbf{Y}$. Then we resample $\mathbf{Y}^*$ as n i.i.d. observations from $P_{\hat{\beta}}$.

In the regression setup we can do parametric bootstrapping either conditionally or unconditionally on the predictors. Both require the strong distributional model assumption GA3.

Conditional parametric bootstrapping amounts to resampling the responses $\mathbf{Y}^* = (Y_1^*, \dots, Y_n^*)$ under assumption A5. That is, we resample the coordinates $Y_i^* \mid X_i$ independently from the fitted exponential dispersion model $\mathcal{E}(\theta(\hat{\eta}_i), \nu_{\hat{\varphi}})$. Contrary to nonparametric pair sampling, this form of bootstrapping does not assume A6 and makes good sense for a designed experiment.

Unconditional parametric bootstrapping requires additionally a parametric model of the predictor distribution. Resampling proceeds under assumption A6 and consists of two steps:

1. Sample $\mathbf{X}^* = (X_1^*, \dots, X_n^*)$ i.i.d. from the fitted predictor distribution. Set $\hat{\eta}_i^* = (X_i^*)^T \hat{\beta}$.
2. Sample $Y_1^*, \dots, Y_n^*$ given $\mathbf{X}^*$ independently with $Y_i^* \mid \mathbf{X}^* \sim \mathcal{E}(\theta(\hat{\eta}_i^*), \nu_{\hat{\varphi}})$.

We would rarely want to do unconditional parametric bootstrapping as this requires a full model of the predictors. But we could implement a hybrid algorithm where step 1. above is replaced by resampling the predictors from the empirical distribution. That is, the resampling of the predictors is nonparametric, but the resampling given the predictors is parametric.

Residual resampling

Residual resampling is a version of nonparametric bootstrapping for regression models where residuals and not responses are resampled. The algorithm only requires assumption A5, just as conditional parametric bootstrapping, and it thus makes sense for designed experiments, but it does not require a parametric model specification of the conditional response distribution.

Residual resampling is simplest to formulate for the additive noise model where

$$Y_i = \mu(X_i^T \beta) + \varepsilon_i$$

with $\varepsilon_1, \dots, \varepsilon_n$ i.i.d. given $\mathbf{X}$. In this case the raw residuals are

$$\hat{\varepsilon}_i = Y_i - \mu(X_i^T \hat{\beta}) = Y_i - \hat{\mu}_i.$$

We resample a new data set by resampling raw residuals *with replacement*. That is, a resampled response vector $\mathbf{Y}^* = (Y_1^*, \dots, Y_n^*)$ is constructed as follows:

1. Sample the residuals $\varepsilon_1^*, \dots, \varepsilon_n^*$ independently from $\hat{\varepsilon}_1, \dots, \hat{\varepsilon}_n$ with replacement.
2. Set $Y_i^* = \hat{\mu}_i + \varepsilon_i^*$ for $i = 1, \dots, n$.

For residual resampling the predictor variables are held fixed just as for conditional parametric bootstrapping. The resampling of residuals is in practice carried out as for the other nonparametric bootstrapping procedures by sampling indices uniformly. It is possible to combine the residual resampling above with a nonparametric bootstrap resampling of the predictors – if we are willing to make the stronger assumption A6.

It is not straightforward to extend residual resampling to all generalized linear models. We would need an one-to-one map $t_\mu : \overline{\mathcal{J}} \rightarrow \mathbb{R}$ with the property that the distribution of $\varepsilon = t_\mu(Y)$ and X are independent when $\mathbf{E}(Y | X) = \mu$ and $\mathbf{V}(Y | X) = \varphi\mathcal{V}(\mu)$. For the additive noise model this map is $t_\mu(y) = y - \mu$. If we can find such a map t_μ , the residuals are defined as $\hat{\varepsilon}_i = t_{\hat{\mu}_i}(Y_i)$ and we replace step 2. above by

2. Set $Y_i^* = t_{\hat{\mu}_i}^{-1}(\varepsilon_i^*)$ for $i = 1, \dots, n$.

The additive noise model can be generalized to the model where

$$Y_i = \mu(X_i^T \beta) + \sqrt{\mathcal{V}(\mu(X_i^T \beta))} \varepsilon_i \quad (13.15)$$

with $\varepsilon_1, \dots, \varepsilon_n$ i.i.d. given $\mathbf{X}$. Then we can take

$$t_\mu(y) = \frac{y - \mu}{\sqrt{\mathcal{V}(\mu)}}.$$

The resulting resampled residuals are the Pearson residuals, but for the corresponding bootstrap distribution to resemble to sampling distribution of the responses correctly, we need the specific model assumption (13.15).

13.3.3 Bootstrap calibration

The bootstrap can be used to compute an approximation to the distribution of any combinant statistic R . This approximation can then be used to calibrate an R -based interval construction to have the desired (nominal) coverage.

The bootstrap was introduced in a seminal paper by Bradley Efron (1979), who very clearly presented the basic ideas. The goal is to approximate the distribution of a combinant statistic $R(\mathbf{Y}, \gamma)$ when $\mathbf{Y} \sim P$ for an unknown P with $\gamma = \gamma(P)$ a functional of P . The main principle behind bootstrapping is the distributional approximation

$$R(\mathbf{Y}, \gamma) \stackrel{\mathcal{D}}{\approx} R(\mathbf{Y}^*, \hat{\gamma}) \quad (13.16)$$

with $\mathbf{Y}^* \sim \hat{P}$. Here $\hat{P}$ is an estimate of P based on $\mathbf{Y}$ and $\hat{\gamma}$ is an estimate of γ . That is, we approximate the *unknown sampling distribution* of $R(\mathbf{Y}, \gamma)$ by the *known bootstrap distribution* of $R(\mathbf{Y}^*, \hat{\gamma})$.

Note that it is possible for $\hat{\gamma} \neq \gamma(\hat{P})$. Indeed, $\hat{\gamma}$ is in a regression context typically a plug-in estimate $\gamma(\hat{\beta})$ based on the parametric estimate of β , while $\hat{P}$ is (derived from) an empirical distribution.

Example 13.7 (Bootstrapping a pivot). Suppose that we have a parametrized model P_β and $\gamma = \gamma(\beta)$. With $\hat{\beta}$ the estimate of β we can use the plug-in estimates $\hat{\gamma} = \gamma(\hat{\beta})$ and

Efron, B. 1979. “Bootstrap Methods: Another Look at the Jackknife.” *The Annals of Statistics* 7 (1): 1–26.

$\hat{P} = P_{\hat{\beta}}$ of γ and P , respectively. The statistic R is a pivot if the sampling distribution of

$$R(\mathbf{Y}, \gamma(\beta)), \quad \mathbf{Y} \sim P_{\beta}$$

does not depend upon β . If R is a pivot, the approximation (13.16) is a distributional identity. That is, there is no approximation error, and the sampling distribution can be computed exactly as the distribution of $R(\mathbf{Y}^*, \gamma(\hat{\beta}))$ for $\mathbf{Y}^* \sim P_{\hat{\beta}}$.

The standardized Z -score in Theorem 2.3, see also Example 13.2, is an example of a pivot, whose distribution can be found analytically. The analytic and the parametric bootstrap solutions coincide in this case.

The bootstrap was inspired by jackknife methods, which, among other things, had been introduced earlier for nonparametric bias and variance estimation. Like jackknife methods, bootstrapping was developed in a nonparametric framework, and the original resampling algorithm was nonparametric bootstrapping. For nonparametric bootstrapping we cannot expect R to be a pivot, though, and a systematic study of the “degree of pivotality” of different statistics is beyond the scope of this book. The take-home message is that for the bootstrap to be accurate and useful, the distribution of the statistic R should be insensitive to the error of the joint approximation $(\hat{P}, \hat{\gamma}) \approx (P, \gamma)$.

We present a couple of examples to illustrate how the bootstrap is used in practice for various combinant statistics for the calibration and computation of interval estimates.

Example 13.8. The simplest combinant statistic is arguably

$$R(\mathbf{Y}, \gamma) = \hat{\gamma} - \gamma.$$

With w_{α} and z_{α} the $\alpha/2$ and $1 - \alpha/2$ quantiles for R , the *nominal* level $(1 - \alpha)$ -confidence interval is

$$\{\gamma \in \mathbb{R} \mid \hat{\gamma} - \gamma \in (w_{\alpha}, z_{\alpha})\} = (\hat{\gamma} - z_{\alpha}, \hat{\gamma} - w_{\alpha}).$$

Estimates, $\hat{w}_{\alpha}$ and $\hat{z}_{\alpha}$, of the quantiles are obtained as the empirical quantiles for the bootstrapped sample

$$\hat{\gamma}_1^* - \hat{\gamma}, \dots, \hat{\gamma}_B^* - \hat{\gamma}.$$

Here $\hat{\gamma}_b^* = \hat{\gamma}(\mathbf{Y}_b^*)$ is the estimate based on the b -th resampled data set $\mathbf{Y}_b^*$. This is a simple procedure, but approximate pivotality of R is questionable.

Example 13.9. With $\hat{q}_\alpha$ and $\hat{r}_\alpha$ denoting the $\alpha/2$ and $(1 - \alpha/2)$ -quantiles for the empirical distribution of the reestimates

$$\hat{\gamma}_1^*, \dots, \hat{\gamma}_B^*,$$

the *percentile* confidence interval is defined simply as

$$(\hat{q}_\alpha, \hat{r}_\alpha).$$

There are theoretical arguments justifying this interval if there exists a monotone transformation of $\hat{\gamma}$, which is a pivot and whose distribution is symmetric. We may observe that $\hat{w}_\alpha = \hat{q}_\alpha - \hat{\gamma}$ and $\hat{z}_\alpha = \hat{r}_\alpha - \hat{\gamma}$, and the confidence interval based on the simple combinant statistic equals

$$(2\hat{\gamma} - \hat{r}_\alpha, 2\hat{\gamma} - \hat{q}_\alpha).$$

This interval is generally different from the percentile interval, but if the distribution of $\hat{\gamma}$ is close to being symmetric around⁷⁵ γ , the two intervals are almost identical. The percentile interval has the nice property of being invariant to monotone parameter transformations, but its actual coverage is sometimes poor.

⁷⁵ The distribution of $\hat{\gamma}$ is often symmetric, in which case percentile intervals and intervals based on the simple combinant statistic are almost the same.

Example 13.10. The combinant statistic can be the (profile) negative log-likelihood

$$R(\mathbf{Y}, \gamma) = 2(\ell_{\max} - \ell(\gamma)).$$

From the bootstrap sample

$$2(\ell_{\max,1}^* - \ell(\hat{\gamma})), \dots, 2(\ell_{\max,B}^* - \ell(\hat{\gamma}))$$

we compute the estimate $\hat{c}_\alpha$ of the $(1 - \alpha)$ -quantile, and the corresponding likelihood interval for γ with nominal coverage $1 - \alpha$ becomes

$$\{\gamma \mid 2(\ell_{\max} - \ell(\gamma)) < \hat{c}_\alpha\}.$$

Example 13.11. If we have an analytic estimate of the standard error, we can also bootstrap the Z -statistic or Z^2 .

With

$$R(\mathbf{Y}, \gamma) = \frac{(\hat{\gamma} - \gamma)^2}{\widehat{\text{se}}^2}$$

the bootstrap sample becomes

$$\frac{(\hat{\gamma}_1^* - \hat{\gamma})^2}{(\widehat{\text{se}}_1^*)^2}, \dots, \frac{(\hat{\gamma}_B^* - \hat{\gamma})^2}{(\widehat{\text{se}}_B^*)^2},$$

which we use to estimate $\hat{c}_\alpha$ as the empirical $(1 - \alpha)$ -quantile. The corresponding confidence interval for γ then becomes

$$\{\gamma \mid (\hat{\gamma} - \gamma)^2 < \hat{c}_\alpha \widehat{\text{se}}^2\} = \hat{\gamma} \pm \sqrt{\hat{c}_\alpha} \cdot \widehat{\text{se}}.$$

In all the examples above, the actual coverage will be close to $1 - \alpha$ whenever the combinant statistic is sufficiently close to being a pivot and B is large enough. The bootstrap does not work without assumptions, but it does not rely on strong distributional approximations, such as a $\mathcal{N}(0, 1)$ - or a χ_1^2 -distribution of the combinant statistic.

We should note that if we only use the bootstrap to estimate the standard error, the resulting standard interval, as in Example 13.6, still supposes that the Z -statistic

$$\frac{\hat{\gamma} - \gamma}{\widehat{\text{se}}}$$

is approximately $\mathcal{N}(0, 1)$ -distributed. It is possible to use the bootstrap for both estimating the standard error and for calibrating the interval based on the Z -statistic, but it requires nested bootstrapping with heavy computational costs.

As mentioned in Example 13.6, if we only want an estimate of the standard error, $B = 200$ is usually large enough. If we want to estimate quantiles to calibrate intervals to have 95% coverage, the rule of thumb is to take $B = 999$. If we aim for larger coverage⁷⁶ we need to take B even larger to get an acceptable relative accuracy of the empirical quantiles.

⁷⁶ From a testing perspective this is equivalent to computing a more accurate p -value for testing hypotheses about γ .

13.4 Advertising revisited

We revisit the advertising case from Chapter 10. The main focus in this section is to compute a confidence interval for the

effect of advertising as expressed by the regression parameter for the categorical variable `ad`.

We first load the and clean the data as in Chapter 10.

```
data(vegetables, package = "RwR")
vegetables <- filter(vegetables, !is.na(discount), week != 2)
```

13.4.1 Model based intervals

We fit a Poisson model with the same formula as used with the gamma model in Section 10.4 (with a natural cubic spline expansion of `discount`). We then compute a standard confidence interval for the regression parameter associated with the variable `ad` via `confint.lm()`. The default nominal coverage of all confidence intervals is 95%.

```
form <- sale ~ offset(log(normalSale)) + store + ad +
  ns(discount, knots = c(20, 30, 40), Boundary.knots = c(0, 50)) - 1
vegetables_glm <- glm(form, family = poisson, data = vegetables)
int_to_tib <- function(x, name)
  tibble(interval = name, lower = x[1], upper = x[2])
ad_int <- confint.lm(vegetables_glm, "ad1") |>
  int_to_tib("Pois standard")
ad_int
```

```
# A tibble: 1 x 3
  interval      lower upper
  <chr>         <dbl> <dbl>
1 Pois standard 0.446 0.563
```

The `vegetables_glm` object is of class `glm`. It means that when the generic function `confint()` is called with this object as argument, it calls the method `confint.glm()` for this class. This method computes likelihood intervals by profiling. We compare the resulting confidence interval to the standard confidence interval.

```
confint(vegetables_glm, "ad1") |>
  int_to_tib("Pois likelihood") |>
  add_row(ad_int) -> ad_int
ad_int
```

```
# A tibble: 2 x 3
  interval      lower      upper
  <chr>         <dbl>    <dbl>
1 Pois likelihood 0.44580 0.56305
2 Pois standard   0.44573 0.56319
```

We find that the two intervals are identical up to the fourth decimal. Thus for this example there is for practical purposes no difference between the two intervals.

The interval produced by either method is fairly narrow and identifies a clearly significant positive association between advertising and sales. However, as discussed in Chapter 10, we know that the Poisson model is misspecified so we have no reason to expect that this construction with *nominal level* 95% has 95% actual coverage.

Due to the overdispersion and/or the misspecification of the variance function by the Poisson model, we fit alternative models with the same mean value specification (same formula and same link function) but using the quasi Poisson family as well as the gamma model. For each model fit we then compute both standard and likelihood intervals. The results are shown in Table 13.1.

```
vegetables_glm_quasi <- glm(form, family = quasipoisson, data = vegetables)
vegetables_glm_gamma <- glm(form, family = Gamma("log"), data = vegetables)
confint.lm(vegetables_glm_quasi, "ad1") |>
  int_to_tib("Q-Pois standard") |>
  add_row(ad_int) -> ad_int
confint(vegetables_glm_quasi, "ad1") |>
  int_to_tib("Q-Pois likelihood") |>
  add_row(ad_int) -> ad_int
confint.lm(vegetables_glm_gamma, "ad1") |>
  int_to_tib("Gamma standard") |>
  add_row(ad_int) -> ad_int
confint(vegetables_glm_gamma, "ad1") |>
  int_to_tib("Gamma likelihood") |>
  add_row(ad_int) -> ad_int
```

interval	lower	upper
Gamma likelihood	0.10245	0.57064
Gamma standard	0.10891	0.56131
Q-Pois likelihood	0.30361	0.70461
Q-Pois standard	0.30367	0.70525
Pois likelihood	0.44580	0.56305

interval	lower	upper
Pois standard	0.44573	0.56319

Table 13.1: Six different computations of confidence intervals with nominal coverage 95%.

The intervals based on the quasi Poisson family are unsurprisingly somewhat wider because the estimated dispersion parameter is somewhat larger than 1. The likelihood interval is again almost identical to the standard interval for the quasi Poisson family, because it is also just a rescaling by the estimated dispersion parameter.

The intervals for the gamma model are even wider and are furthermore skewed toward zero. There is a slightly larger difference between the standard interval and the likelihood interval for the gamma model, but it is still a relatively small difference.

13.4.2 Robust standard errors

We turn to standard confidence intervals based on the sandwich estimator of the asymptotic covariance matrix. We first show how to compute both the ordinary estimate of the asymptotic covariance matrix and the robust estimate by using the formulas explicitly, and then we show below how to use the sandwich package.

We first compute the estimates of the asymptotic covariance matrices by extracting all the relevant components from the glm object. We illustrate these computations using the Poisson model. The weights $\hat{w}_{ii}$ are extracted using the `weights()` function with argument⁷⁷ `type = "working"`, and we compute the estimated Fisher information matrix in terms of the weights and the model matrix.

```
w_hat <- weights(vegetables_glm, type = "working")
X <- model.matrix(vegetables_glm)
J <- crossprod(X, w_hat * X)
```

⁷⁷ These are actually the weights from the last (m -th) iteration of the IWLS algorithm. That is, they are computed from $\hat{\beta}_{m-1}$, and they are

The inverse of J is the estimate of the asymptotic covariance matrix of $\hat{\beta}$. It can be computed for the glm object by the

```
Sigma_hat <- vcov(vegetables_glm)
(solve(J) - Sigma_hat) |>
  range()
```

```
[1] -6.418199e-13  1.229461e-12
```

Table 13.2: Numerical check that J equals the inverse of the covariance matrix reported by the `glm` object.

`vcov()` function. Table 13.2 shows that, up to numerical errors, our computations and those by `vcov()` are in agreement. We then compute the sandwich estimator using (13.12) and (13.14).

```
w_tilde <- (w_hat * residuals(vegetables_glm, type = "working"))^2
Sigma_tilde <- solve(J, crossprod(X, w_tilde * X)) %*% solve(J)
```

The R package `sandwich` implements methods for automatic computation of the sandwich estimator for a range of model objects in R. We compare our explicit computation above to that of the `sandwich()` function from the package.

```
(sandwich::sandwich(vegetables_glm) - Sigma_tilde) |> range()
```

```
[1] -1.429648e-10  1.267022e-10
```

Up to numerical errors our computation of the sandwich estimator and the computation using `sandwich()` agree⁷⁸.

In applications of the sandwich estimator we usually want to compute (standard) confidence intervals based on the standard errors it provides. We can do these computations ourselves but it can also be done conveniently by the general `coefci()` function from the `lmtest` package.

```
lmtest::coefci(vegetables_glm, parm = "ad1", vcov. = sandwich) |>
  int_to_tib("Pois sandwich") |>
  add_row(ad_int) -> ad_int
lmtest::coefci(vegetables_glm_gamma, parm = "ad1", vcov. = sandwich) |>
  int_to_tib("Gamma sandwich") |>
  add_row(ad_int) -> ad_int
```

⁷⁸ We have used (13.14) for the weights, which is also how they are computed by `sandwich()`. Had we used (13.13) we would have gotten larger differences. This is because the working weights `w_hat`, used as estimates of w_{ii} , are from the last iteration of IWLS, and the two different representations of $\tilde{w}_{..}$ therefore

The two robust intervals are included in Table 13.3. We note that the robust interval based on the Poisson model is notably wider than the model based intervals, and it is even wider than the intervals corrected for overdispersion via the quasi Poisson model. The robust interval for the gamma model is, in fact, narrower than the model based intervals but the differences between the robust interval and the model based intervals are much smaller for the gamma model than for the Poisson model.

interval	lower	upper
Gamma sandwich	0.16249	0.50772
Pois sandwich	0.26834	0.74057
Gamma likelihood	0.10245	0.57064
Gamma standard	0.10891	0.56131
Q-Pois likelihood	0.30361	0.70461
Q-Pois standard	0.30367	0.70525
Pois likelihood	0.44580	0.56305
Pois standard	0.44573	0.56319

Table 13.3: Eight different computations of confidence intervals with nominal coverage 95%.

13.4.3 Bootstrap intervals

In this section we first implement nonparametric bootstrapping using pair sampling. The implementation is based on `sample()` for sampling indices from 1 to n (the number of observations) with replacement. For each bootstrapped dataset we fit the same model using `glm` and here we only keep the parameter of interest for each fitted model. When doing bootstrapping in general it might be better to keep the entire model object returned by `glm()` and then subsequently extract what is needed – to avoid having to redo the bootstrap computations for other parameters of interest.

```
B <- 999
n <- nrow(vegetables)
beta <- numeric(B)
for(b in 1:B) {
  i <- sample(n, n, replace = TRUE)
  boot_glm <- glm(form, family = poisson, data = vegetables[i, ])
  beta[b] <- coefficients(boot_glm)["ad1"]
}
```

```
}
```

We also implement conditional parametric bootstrapping by sampling, conditionally on the predictors, from the fitted Poisson model. This is almost as easy to implement as non-parametric bootstrapping. It can be done by extracting the fitted means from the model and using the `rpois()` function for the simulation based on the fitted means. But the generic `simulate()` function does this for you, and it can be applied to a number of model objects in R, which will produce a vector of simulated response variables.

```
par_beta <- numeric(B)
vegetables_resamp <- vegetables
for(b in 1:B) {
  vegetables_resamp$sale <- simulate(vegetables_glm)[, 1]
  boot_glm <- glm(form, family = poisson, data = vegetables_resamp)
  par_beta[b] <- coefficients(boot_glm)["ad1"]
}
```

We will compute several bootstrap confidence intervals below, but first we take a closer look at the bootstrap distributions. Figure 13.1 shows the empirical distributions of the bootstrap estimates. We note that the distribution based on nonparametric bootstrapping is much wider than the distribution based on the Poisson model. We also compute the standard deviations for the bootstrap distributions, which we can use as estimates of the standard errors, which in turn can be used for computing standard confidence intervals.

```
beta_model <- coef(summary(vegetables_glm))["ad1", ]
beta_model_gamma <- coef(summary(vegetables_glm_gamma))["ad1", ]
boot_stat <- tibble(
  method = c("Pois model", "Pois par boot", "Pois nonpar boot", "Gamma model"),
  beta_hat = c(beta_model["Estimate"], mean(par_beta), mean(beta), beta_model_gamma["Estimate"]),
  se = c(beta_model["Std. Error"], sd(par_beta), sd(beta), beta_model_gamma["Std. Error"])
)
boot_stat
```

```
# A tibble: 4 x 3
  method      beta_hat      se
  <chr>      <dbl> <dbl>
1 Pois model    0.504 0.0299
2 Pois par boot  0.504 0.0293
3 Pois nonpar boot 0.488 0.177
4 Gamma model    0.335 0.115
```

Warning: Removed 1 row containing non-finite outside the scale ran
(`stat\_density()`).

We observe that the standard error based on parametric bootstrapping is almost identical to the model based analytic estimate of the standard error. This suggests that if the data were from a Poisson model, the model based estimate would, in fact, be quite accurate. The standard error based on nonparametric bootstrapping is, however, around six times larger. This is in accordance with our previous observations that the Poisson model does not fit the data.

A standard confidence interval can be computed based on the nonparametric bootstrap estimate of the standard error, see Table 13.4. Alternatively, we can compute a confidence interval based on the simple combinant statistic

$$R(\hat{\beta}_{ad1}, \beta_{ad1}) = \hat{\beta}_{ad1} - \beta_{ad1}.$$

Its distribution is approximated by the bootstrap distribution of

$$R(\hat{\beta}_{ad1}^*, \hat{\beta}_{ad1}) = \hat{\beta}_{ad1}^* - \hat{\beta}_{ad1},$$

where $\hat{\beta}_{ad1}^*$ denotes the parameter estimate for the nonparametrically bootstrapped sample. With $\hat{w}_\alpha$ and $\hat{z}_\alpha$ denoting the $\alpha/2$ and $1 - \alpha/2$ quantiles of the empirical bootstrap distribution of $\hat{\beta}_{ad1}^* - \hat{\beta}_{ad1}$, we compute the confidence interval as

$$\{\beta \in \mathbb{R} \mid \hat{\beta}_{ad1} - \beta \in (\hat{w}_\alpha, \hat{z}_\alpha)\} = (\hat{\beta}_{ad1} - \hat{z}_\alpha, \hat{\beta}_{ad1} - \hat{w}_\alpha).$$

Note how it is the smaller quantile that determines the upper limit of the interval and the larger quantile that determines the lower limit. This results in the following confidence interval with nominal coverage 95%.

```
beta_star <- beta - boot_stat$beta_hat[1]
beta_qt <- quantile(beta_star, probs = c(0.975, 0.025), type = 1)
boot_stat$beta_hat[1] - beta_qt
```

```
97.5%      2.5%
0.1927803 0.8718003
```

We can also bootstrap the distribution of the estimator based on the gamma model.

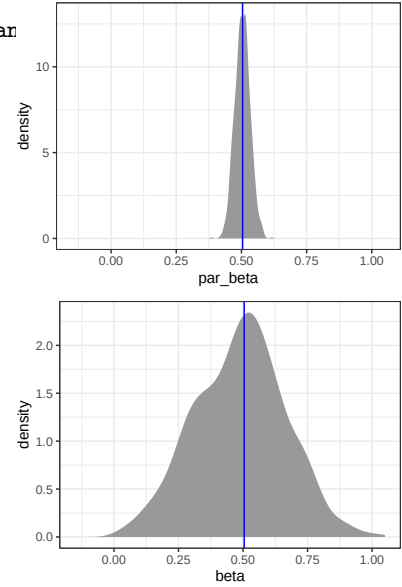


Figure 13.1: Parametric (top) and nonparametric (bottom) bootstrap distributions of $\hat{\beta}_{ad1}$ using the Poisson model.

```

beta <- numeric(B)
for(b in 1:B) {
  i <- sample(n, n, replace = TRUE)
  boot_glm <- glm(form, family = Gamma("log"), data = vegetab
  beta[b] <- coefficients(boot_glm)["ad1"]
}

```

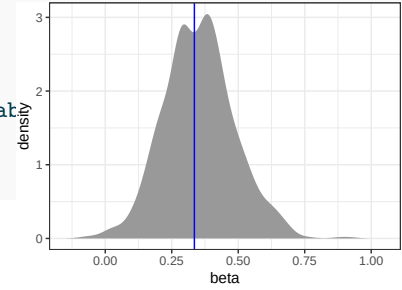


Figure 13.2: Nonparametric bootstrap distribution of $\hat{\beta}_{ad1}$ using the Gamma model.

We will only consider the nonparametric bootstrap distribution for the gamma model as shown in Figure 13.2. We note that this distribution is closer to zero and also a bit narrower than the nonparametric bootstrap distribution based on the Poisson model.

All four nonparametric bootstrap intervals are in Table 13.4. The gamma model results in slightly narrower intervals than the Poisson model, and in intervals that are also closer to zero. There are small differences between the standard intervals and the simple intervals. This is most pronounced for the Poisson model, where the simple interval is moved slightly to the right relative to the standard interval. This corresponds to a slight *left skewness* in the bootstrapped distribution, see Figure 13.1.

interval	lower	upper
Gamma boot se	0.07385	0.59636
Gamma boot simple	0.03442	0.57068
Pois boot simple	0.19278	0.87180
Pois boot se	0.15739	0.85152

Table 13.4: The four bootstrap confidence intervals with nominal coverage 95%.

Figure 13.3 compares all twelve intervals computed using the two different models for estimation and using interval estimators that are either model based, robust or bootstrap based. The intervals are quite different, and this example shows first of all that statistical inference based on a misspecified model (the Poisson model) can be quite misleading. We make the following additional observations.

- The nonparametric bootstrap intervals are generally the widest intervals. Using the gamma model standard errors or the robust sandwich estimator for the Poisson model is a huge improvement over the Poisson model

based standard errors, but the nonparametric bootstrap suggests even wider intervals.

- The parametric bootstrap does not adjust for model misspecification and it usually gives intervals close to standard intervals using model based standard errors. It can be useful if we don't have an analytic estimator of the standard error though.
- The gamma model and the Poisson model specify the same mean value model but result in different estimators (because they specify different variance models and thus different weights). Thus the estimates are different. If the gamma variance fits the data the best, the resulting estimator will also be more efficient. The same interval estimator gives narrower intervals for the gamma model estimator than for the Poisson model estimator, which suggest that the gamma model estimator is the most efficient.
- Irrespectively of which interval estimator we consider, the conclusion is still that there is a significant positive association between advertising and sale. But the evidence appears less strong and the magnitude much more undercertain than if we went with the initial conclusion based on the Poisson model only.

Using the nonparametric bootstrap to calibrate intervals to have the desired coverage usually performs quite well in practice for many estimators and for sufficiently large sample sizes. It is possible in special cases, or by relying on strong model assumptions, to analytically construct intervals with better coverage properties for small samples, but for moderate to large sample sizes the nonparametric bootstrap is hard to beat.

The main drawback of the bootstrap is a non-significant extra computational burden, since we need to refit the model B times. Though this computational cost scales linearly, it still means that if $B = 999$ as above, and if it takes just one minute to compute the estimator for one dataset, it will take more than half a day to compute the bootstrap distribution. Since the bootstrap computations are embarrassingly parallel, the cost can partly be alleviated with access to many computational cores. However, such access appears to trigger the use of computationally more costly estimators in practice, and the bootstrap B -factor on the computational cost remains a

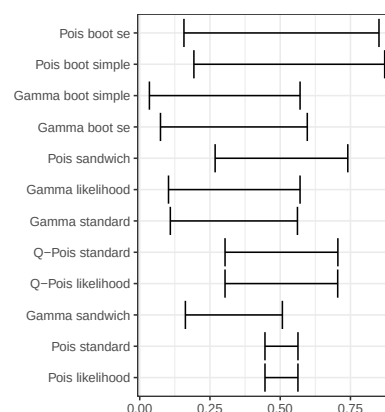


Figure 13.3: The twelve confidence intervals with nominal coverage 95% ordered according to width.

challenge. This means that computationally cheaper alternatives are always of interest, and that we might want to save the computational costs until we are fairly sure about what we really want to bootstrap.

We could, of course, also always lower B to save some computations. As a rule of thumb, a choice of $B = 200$ is actually sufficient if we just want an estimate of the standard error. The choice of $B = 999$ as above is recommended for the estimation of 2.5% and 97.5% quantiles of any combinant statistic. If we want to estimate even more extreme quantiles with any accuracy we need B to be even larger.

Exercises

Exercise 13.1. Assume that the sample space is $\mathbb{R}^k$ and the parameter of interest is the mean

$$\gamma(P) = \int y \, dP(y).$$

Assume that we have observations $Y_1, \dots, Y_n$ and let K denote the convex hull spanned by these observations (the smallest convex subset containing the observations). Let $\hat{P}$ be the NPMLE, for which

$$\gamma(\hat{P}) = \frac{1}{n} Y_i.$$

1. Argue that if the support of P' is contained in K then $\gamma(P') \in K$, and show that in this case there exists a

$$P = \sum_{i=1}^n P_i \varepsilon_{Y_i}.$$

such that $L(P) \geq L(P')$ and $\gamma(P) = \gamma(P')$.

2. Show that if the support of P' is not restricted then for all $\gamma \in \mathbb{R}^k$

$$\sup_{P': \gamma(P') = \gamma} L(P') = L(\hat{P}).$$

Exercise 13.2. The Gompertz distribution on $[0, \infty)$ has survival function

$$S(t) = \exp(-\eta(e^{\gamma t} - 1)) \quad t \geq 0,$$

for parameters $\eta, \gamma > 0$. We assume in the following that

$$(T_1, \delta_1), \dots, (T_n, \delta_n)$$

are n censored i.i.d. survival times following a Gompertz distribution. Introduce also the summary statistics

$$t = \sum_{i=1}^n T_i, \quad s = \sum_{i=1}^n \delta_i T_i \quad \text{and} \quad n_u = \sum_{i=1}^n \delta_i.$$

The values of these summary statistics for a dataset are given in the following table together with some additional statistics.

Statis- tic:	n	n_u	t	s	$\hat{\gamma}$	$\sum_{i=1}^n e^{\hat{\gamma} T_i}$	$\sum_{i=1}^n T_i^2 e^{\hat{\gamma} T_i}$
Value	50	32	16371384	0.09390	3396		9949983

1. Compute the hazard function for the Gompertz distribution. Show that the log-likelihood is

$$\ell(\eta, \gamma) = \log(\eta\gamma)n_u + \gamma s + \eta \left(n - \sum_{i=1}^n e^{\gamma T_i} \right).$$

2. Show that the MLE (provided it exists) satisfies the equations

$$\begin{aligned} \sum_{i=1}^n e^{\gamma T_i} &= n + \frac{n_u}{\eta} \\ \sum_{i=1}^n T_i e^{\gamma T_i} &= \frac{s}{\eta} + \frac{n_u}{\eta\gamma}. \end{aligned}$$

3. With $\mathcal{J}^{\text{obs}}$ denoting the observed Fisher information show that

$$((\mathcal{J}^{\text{obs}})^{-1})_{22} = \left(\frac{n_u}{\hat{\gamma}^2} + \hat{\eta} \sum_{i=1}^n T_i^2 e^{\hat{\gamma} T_i} - \frac{1}{n_u} \left(s + \frac{n_u}{\hat{\gamma}} \right)^2 \right)^{-1}.$$

4. Using the summary statistics from the table above show that

$$\hat{\eta} = 0.009564$$

and compute an approximate likelihood interval for γ .

From hereon it is assumed that the censoring distribution is exponential with rate parameter λ , and that the censoring variables are i.i.d.

5. Argue why $\hat{\lambda} = (n - n_u)/t$ is a sensible estimator of λ .
6. Simulate a new data set, $(\tilde{T}_1, \tilde{\delta}_1), \dots, (\tilde{T}_n, \tilde{\delta}_n)$, using the estimated parameters $\hat{\eta}$ and $\hat{\gamma}$ for the Gompertz survival distribution and $\hat{\lambda}$ for the exponential censoring distribution. Compute and plot the Kaplan-Meier estimator of the survival function for the simulated data and compare it to the theoretical survival function.

The implementation of the simulation is the first step in the construction of a parametric bootstrap algorithm, which, for instance, can be used to calibrate the likelihood interval.

7. Implement a parametric bootstrap algorithm⁷⁹ to obtain a bootstrap approximation of the distribution of the statistic

$$\frac{(\gamma - \hat{\gamma})^2}{((\mathcal{J}^{\text{obs}})^{-1})_{22}}.$$

Compare it with a χ^2_1 -distribution.

⁷⁹ Remember the parameter constraints!

Hint: The implementation will require the numerical solution of the score equation for the computation of the MLE. You will only need to numerically solve a univariate equation, and you can implement your own Newton algorithm. You can also use the R function `uniroot()`. Alternatively, you can use the R function `optim()` for direct minimization of the negative log-likelihood.

Exercise 13.3. This exercise is investigating the effect of model selection on the standard confidence interval constructions.

The exercise uses the `cholost` data set from the R package `RwR`, and you will consider models of the effect of *compliance* on cholesterol decrease. Data is from a trial where 164 men took cholestyramine for a number of years. Their decrease in blood-level cholesterol from the beginning to the end of the trial is the response variable (`y` in the data set). Not all of them took the intended dose. In the data set the variable `c` measures compliance. It is a monotone transformation of the proportion of the dose actually taken.

The conditional expectation of the decrease in cholesterol level Y as a function of the compliance C is denoted

$$\mathbf{E}(Y \mid C) = f(C),$$

and it is assumed throughout that $\mathbf{V}(Y \mid C) = \sigma^2$.

1. Fit the model

$$f(c) = \beta_0 + \beta_1 c$$

of the cholesterol decrease as a function of compliance.
Investigate the model fit⁸⁰.

⁸⁰ This is a small data set, so don't use hexagonal binning for plotting the data points!

In the following f is expanded using natural cubic splines. Using the R terminology, an expansion with df degrees of freedom is a model with $df - 1$ knots, df basis functions and $df + 1$ parameters (an intercept and one parameter for each of the basis functions).

2. Fit f as a natural cubic spline with 3 degrees of freedom. Give a graphical presentation of the resulting model. Compare the model to the linear regression model above.
3. In the remaining part of the exercise the focus is on the estimation of $f(c)$ for

$$c \in I := \{-2, -1, 0, 1, 2\}.$$

That is, the focus is on the computation of the estimators $\hat{f}(c)$ and their distribution for c taking these five particular values.

4. Use nonparametric bootstrapping to estimate standard errors of $\hat{f}(c)$ for $c \in I$ when fitting the natural cubic spline model with 3 degrees of freedom. Compare with the theoretical values of the standard errors.
5. Consider the 6 models where f is a natural cubic spline with 1–6 degrees of freedom. Compare the models and use the adjusted R^2 -statistic to select one of the 6 models.

The bootstrap above does not take into account any model selection such as the selection of degrees of freedom of the cubic spline expansion of f . The last question in this problem explores the effect of model selection on the confidence intervals.

6. Use nonparametric bootstrapping to estimate standard errors of $\hat{f}(c)$ for $c \in I$ when selection of the degrees of freedom in the range from 1 to 6 is part of the estimation procedure. Compare with the previously estimated standard errors.

Exercise 13.4. Let $\mathcal{L}(c)$ denote the likelihood regions from Definition 13.4.

1. Show that the MLE exists if and only if $\bigcap_{c>0} \mathcal{L}(c) \neq \emptyset$
2. Show that the MLE is unique if and only if $\bigcap_{c>0} \mathcal{L}(c)$ is a singleton set.