STAT 9610 Lecture Notes

Eugene Katsevich

Table of contents

Chapter 1

Introduction

1.1 Welcome

This is a set of lecture notes developed for the PhD statistics course "STAT 9610: Statistical Methodology" at the University of Pennsylvania. Much of the content is adapted from Alan Agresti's book *Foundations of Linear and Generalized Linear Models* (2015). These notes may contain typos and errors; if you find any such issues or have other suggestions for improvement, please notify the instructor via Ed Discussion.

1.2 Preview: Linear and generalized linear models

See also Agresti 1.1, Dunn and Smyth 1.1-1.2, 1.5-1.6, 1.8-1.12

The overarching statistical goal addressed in this class is to learn about relationships between a response y and predictors $x_0, x_1, \ldots, x_{p-1}$. This abstract formulation encompasses an extremely wide variety of applications. The most widely used set of statistical models to address such problems are *generalized linear models*, which are the focus of this class.

Let's start by recalling the *linear model*, the most fundamental of the generalized linear models. In this case, the response is continuous ($y \in \mathbb{R}$) and modeled as:

$$y = \beta_0 x_0 + \cdots + \beta_{p-1} x_{p-1} + \epsilon, \tag{1.1}$$

where

$$\epsilon \sim (0, \sigma^2), \quad \text{i.e. } \mathbb{E}[\epsilon] = 0 \text{ and } \mathrm{Var}[\epsilon] = \sigma^2. \tag{1.2}$$

We view the predictors $x_0, \ldots, x_{p-1}$ as fixed, so the only source of randomness in y is ϵ. Another way of writing the linear model is:

$$\mu \equiv \mathbb{E}[y] = \beta_0 x_0 + \cdots + \beta_{p-1} x_{p-1} \equiv \eta.$$

Not all responses are continuous, however. In some cases, we have binary responses ($y \in \{0, 1\}$) or count responses ($y \in \mathbb{Z}$). In these cases, there is a mismatch between the:

$$\textit{linear predictor } \eta \equiv \beta_0 x_0 + \cdots + \beta_{p-1} x_{p-1}$$

and the

$$\textit{mean response } \mu \equiv \mathbb{E}[y].$$

The linear predictor can take arbitrary real values ($\eta \in \mathbb{R}$), but the mean response can lie in a restricted range, depending on the response type. For example, $\mu \in [0, 1]$ for binary y and $\mu \in [0, \infty)$ for count y.

For these kinds of responses, it makes sense to model a *transformation* of the mean as linear, rather than the mean itself:

$$g(\mu) = g(\mathbb{E}[y]) = \beta_0 x_0 + \cdots + \beta_{p-1} x_{p-1} = \eta.$$

This transformation g is called the link function. For binary y, a common choice of link function is the *logit link*, which transforms a probability into a log-odds:

$$\text{logit}(\pi) \equiv \log \frac{\pi}{1-\pi}.$$

So the predictors contribute linearly on the log-odds scale rather than on the probability scale. For count y, a common choice of link function is the *log link*.

Models of the form

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

are called *generalized linear models* (GLMs). They specialize to linear models for the identity link function, i.e., $g(\mu) = \mu$. The focus of this course is methodologies to learn about the coefficients $\boldsymbol{\beta} \equiv (\beta_0, \ldots, \beta_{p-1})^T$ of a GLM based on a sample $(\boldsymbol{X}, \boldsymbol{y}) \equiv \{(x_{i,0}, \ldots, x_{i,p-1}, y_i)\}_{i=1}^n$ drawn from this distribution. Learning about the coefficient vector helps us learn about the relationship between the response and the predictors.

1.3 Course outline

This course is broken up into six units:

- **Unit 1: Linear models: Estimation.** The *least squares* point estimate $\widehat{\boldsymbol{\beta}}$ of $\boldsymbol{\beta}$ based on a dataset $(\boldsymbol{X}, \boldsymbol{y})$ under the linear model assumptions.
- **Unit 2: Linear models: Inference.** Under the additional assumption that $\epsilon \sim N(0, \sigma^2)$, how to carry out statistical inference (hypothesis testing and confidence intervals) for the coefficients.
- **Unit 3: Linear models: Misspecification.** What to do when the linear model assumptions are not correct: What issues can arise, how to diagnose them, and how to fix them.
- **Unit 4: GLMs: General theory.** Estimation and inference for GLMs (generalizing Chapters 1 and 2). GLMs fit neatly into a unified theory based on *exponential families*.

- **Unit 5: GLMs: Special cases.** Looking more closely at the most important special cases of GLMs, including logistic regression and Poisson regression.
- **Unit 6: Multiple testing.** How to adjust for multiple hypothesis testing, both in the context of GLMs and more generally.

1.4 Notation

We will use the following notations in this course. Vector and matrix quantities will be bolded, whereas scalar quantities will not be. Capital letters will be used for matrices, and lowercase for vectors and scalars. No notational distinction will be made between random quantities and their realizations. The letters $i = 1, \ldots, n$ and $j = 0, \ldots, p-1$ will index samples and predictors, respectively. The predictors $\{x_{ij}\}_{i,j}$ will be gathered into an $n \times p$ matrix $\boldsymbol{X}$. The rows of $\boldsymbol{X}$ correspond to samples, with the ith row denoted $\boldsymbol{x}_{i*}$. The columns of $\boldsymbol{X}$ correspond to predictors, with the jth column denoted $\boldsymbol{x}_{*j}$. The responses $\{y_i\}_i$ will be gathered into an $n \times 1$ response vector $\boldsymbol{y}$. The notation $\equiv$ will be used for definitions.

Part I

Linear models: Estimation

In this unit, we will focus on estimation of coefficients in the linear regression model (eqs. 1.1 and 1.2). We start by discussing the interpretation of linear models (Chapter 2). Then, we discuss least squares estimates from the algebraic, geometric, and probabilistic perspectives (Chapter 3). We then discuss important properties of least squares estimates, including orthogonality relationships least squares estimation implies (Chapter 4) and the effects of collinearity (Chapter 5). We conclude with an R demo (Chapter 6).

Chapter 2

Interpreting linear models

2.1 Predictors and coefficients

See also Agresti 1.2, Dunn and Smyth 1.4, 1.7, 2.7

The types of predictors x_j (e.g. binary or continuous) has less of an effect on the regression than the type of response, but it is still important to pay attention to the former.

Intercepts. It is common to include an *intercept* in a linear regression model, a predictor x_0 such that $x_{i0} = 1$ for all i. When an intercept is present, we index it as the 0th predictor. The simplest kind of linear model is the *intercept-only model* or the *one-sample model*:

$$y = \beta_0 + \epsilon. \tag{2.1}$$

The parameter β_0 is the mean of the response.

Binary predictors. In addition to an intercept, suppose we have a binary predictor $x_1 \in \{0, 1\}$ (e.g. $x_1 = 1$ for patients who took blood pressure medication and $x_1 = 0$ for those who didn't). This leads to the following linear model:

$$y = \beta_0 + \beta_1 x_1 + \epsilon. \tag{2.2}$$

Here, β_0 is the mean response (say blood pressure) for observations with $x_1 = 0$ and $\beta_0 + \beta_1$ is the mean response for observations with $x_1 = 1$. Therefore, the parameter β_1 is the difference in mean response between observations with $x_1 = 1$ and $x_1 = 0$. This parameter is sometimes called the *effect* or *effect size* of x_1, though a causal relationship might or might not be present. The model (2.2) is sometimes called the *two-sample model*, because the response data can be split into two "samples": those corresponding to $x_1 = 0$ and those corresponding to $x_1 = 1$.

Categorical predictors. A binary predictor is a special case of a categorical predictor: A predictor taking two or more discrete values. Suppose we have a predictor $w \in \{w_0, w_1, \ldots, w_{C-1}\}$, where $C \geq 2$ is the number of categories and $w_0, \ldots, w_{C-1}$ are the *levels* of w. E.g. suppose $\{w_0, \ldots, w_{C-1}\}$ is the collection of U.S. states, so that $C = 50$. If we want to regress a response on the categorical predictor w, we cannot simply set $x_1 = w$ in the context of the linear regression (2.2). Indeed, w does not necessarily take numerical values. Instead, we need to add a predictor x_j for each of the levels of w. In particular, define $x_j \equiv 1(w = w_j)$ for $j = 1, \ldots, C - 1$ and consider the regression

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_{C-1} x_{C-1} + \epsilon. \tag{2.3}$$

Here, category 0 is the *base category*, and β_0 represents the mean response in the base category. The coefficient β_j represents the difference in mean response between the jth category and the base category.

Quantitative predictors. A quantitative predictor is one that can take on any real value. For example, suppose that $x_1 \in \mathbb{R}$, and consider the linear model

$$y = \beta_0 + \beta_1 x_1 + \epsilon. \tag{2.4}$$

Now, the interpretation of β_1 is that an increase in x_1 by 1 is associated with an increase in y by β_1. We must be careful to avoid saying "an increase in x_1 by 1 *causes* y to increase by β_1" unless we make additional causal assumptions. Note that the units of x_1 matter. If x_1 is the height of a person, then the value and the interpretation of β_1 changes depending on whether that height is measured in feet or in meters.

Ordinal predictors. There is an awkward category of predictor in between categorical and continuous called *ordinal*. An ordinal predictor is one that takes a discrete number of values, but these values have an intrinsic ordering, e.g. $x_1 \in \{\texttt{small}, \texttt{medium}, \texttt{large}\}$. It can be treated as categorical at the cost of losing the ordering information, or as continuous if one is willing to assign quantitative values to each category.

Multiple predictors. A linear regression need not contain just one predictor (aside from an intercept). For example, let's say x_1 and x_2 are two predictors. Then, a linear model with both predictors is

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \epsilon. \tag{2.5}$$

When there are multiple predictors, the interpretation of coefficients must be revised somewhat. For example, β_1 in the above regression is the effect of an increase in x_1 by 1 *while holding x_2 constant* or *while adjusting for x_2* or *while controlling for x_2*. If y is blood pressure, x_1 is a binary predictor indicating blood pressure medication taken and x_2 is sex, then β_1 is the effect of the medication on blood pressure while controlling for sex. In general, the coefficient of a predictor depends on what other predictors are in the model. As an extreme case, suppose the medication has no actual effect, but that men generally have higher blood pressure and higher rates of taking the medication. Then, the coefficient β_1 in the single regression model (2.2) would be nonzero but the coefficient in the multiple regression model (2.5) would be equal to zero. In this case, sex acts as a *confounder*.

Interactions. Note that the multiple regression model (2.5) has the built-in assumption that the effect of x_1 on y is the same for any fixed value of x_2 (and vice versa). In some cases, the effect of one variable on the response may depend on the value of another variable. In this case, it's appropriate to add another predictor called an *interaction*. Suppose x_2 is quantitative (e.g. years of job experience) and x_2 is binary (e.g. sex, with $x_2 = 1$ meaning male). Then, we can define a third predictor x_3 as the product of the first two, i.e. $x_3 = x_1 x_2$. This gives the regression model

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 + \epsilon. \tag{2.6}$$

Now, the effect of adding another year of job experience is β_1 for females and $\beta_1 + \beta_3$ for males. The coefficient β_3 is the difference in the effect of job experience between males and females.

2.2 Linear model spaces

See also Agresti 1.3-1.4, Dunn and Smyth 2.1, 2.2, 2.5.1

The matrix $\boldsymbol{X}$ is called the *model matrix* or the *design matrix*. Concatenating the linear model equations across observations gives us an equivalent formulation:

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}; \quad \mathbb{E}[\boldsymbol{\epsilon}] = \boldsymbol{0}, \ \mathrm{Var}[\boldsymbol{\epsilon}] = \sigma^2 \boldsymbol{I_n}$$

or

$$\mathbb{E}[\boldsymbol{y}] = \boldsymbol{X}\boldsymbol{\beta} = \boldsymbol{\mu}.$$

As $\boldsymbol{\beta}$ varies in $\mathbb{R}^p$, the set of possible vectors $\boldsymbol{\mu} \in \mathbb{R}^n$ is defined

$$C(\boldsymbol{X}) \equiv \{\boldsymbol{\mu} = \boldsymbol{X}\boldsymbol{\beta} : \boldsymbol{\beta} \in \mathbb{R}^p\}.$$

$C(\boldsymbol{X})$, called the *model vector space*, is a subspace of $\mathbb{R}^n$: $C(\boldsymbol{X}) \subseteq \mathbb{R}^n$. Since

$$\boldsymbol{X}\boldsymbol{\beta} = \beta_0 \boldsymbol{x_{*0}} + \cdots + \beta_{p-1} \boldsymbol{x_{*p-1}},$$

the model vector space is the column space of the matrix $\boldsymbol{X}$ (Figure 2.1).

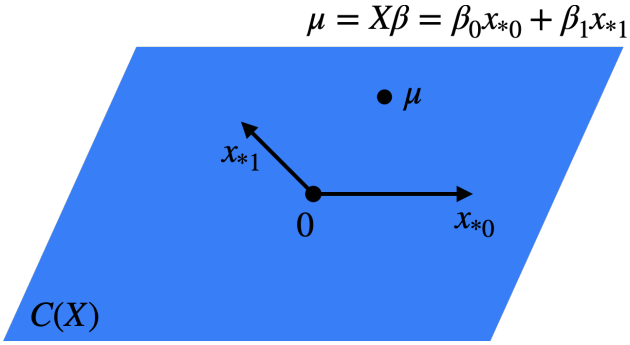


Figure 2.1: The model vector space.

The *dimension* of $C(\boldsymbol{X})$ is the rank of $\boldsymbol{X}$, i.e. the number of linearly independent columns of $\boldsymbol{X}$. If $\mathrm{rank}(\boldsymbol{X}) < p$, this means that there are two different vectors $\boldsymbol{\beta}$ and $\boldsymbol{\beta'}$ such that $\boldsymbol{X}\boldsymbol{\beta} = \boldsymbol{X}\boldsymbol{\beta'}$. Therefore, we have two values of the parameter vector that give the same model for $\boldsymbol{y}$. This makes $\boldsymbol{\beta}$ *not identifiable*, and makes it impossible to reliably determine $\boldsymbol{\beta}$ based on the data. For this reason, we will generally assume that $\boldsymbol{\beta}$ is *identifiable*, i.e. $\boldsymbol{X}\boldsymbol{\beta} \neq \boldsymbol{X}\boldsymbol{\beta'}$ if $\boldsymbol{\beta} \neq \boldsymbol{\beta'}$. This is equivalent to the assumption that $\mathrm{rank}(\boldsymbol{X}) = p$. Note that this cannot hold when $p > n$, so for the majority of the course we will assume that $p \leq n$. In this case, $\mathrm{rank}(\boldsymbol{X}) = p$ if and only if $\boldsymbol{X}$ has *full-rank*.

As an example when $p \leq n$ but when $\boldsymbol{\beta}$ is still not identifiable, consider the case of a categorical predictor. Suppose the categories of w were $\{w_1, \ldots, w_{C-1}\}$, i.e. the baseline category w_0 did not exist. In this case, the model (2.3) would not be identifiable because $x_0 = 1 = x_1 + \cdots + x_{C-1}$ and thus $x_{*0} = 1 = x_{*1} + \cdots + x_{*,C-1}$. Indeed, this means that one of the predictors can be expressed as a linear combination of the others, so $\boldsymbol{X}$ cannot have full rank. A simpler way of phrasing the problem is that we are describing $C-1$ intrinsic parameters (the means in each of the $C-1$ groups) with C model parameters. There must therefore be some redundancy. For this reason, if we include an intercept term in the model then we must designate one of our categories as the baseline and exclude its indicator from the model.

Chapter 3

Least squares estimation

3.1 Algebraic perspective

See also Agresti 2.1.1, Dunn and Smyth 2.4.1, 2.5.2

Now, suppose that we are given a dataset $(\boldsymbol{X}, \boldsymbol{y})$. How do we go about estimating $\boldsymbol{\beta}$ based on this data? The canonical approach is the *method of least squares*:

$$\widehat{\boldsymbol{\beta}} \equiv \arg\min_{\boldsymbol{\beta}} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|^2.$$

The quantity

$$\|\boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}\|^2 = \|\boldsymbol{y} - \widehat{\boldsymbol{\mu}}\|^2 = \sum_{i=1}^{n}(y_i - \widehat{\mu}_i)^2$$

is called the *residual sum of squares (RSS)*, and it measures the lack of fit of the linear regression model. We therefore want to choose $\widehat{\boldsymbol{\beta}}$ to minimize this lack of fit. Letting $L(\boldsymbol{\beta}) = \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|^2$, we can do some calculus to derive that

$$\frac{\partial}{\partial \boldsymbol{\beta}} L(\boldsymbol{\beta}) = -\boldsymbol{X}^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}).$$

Setting this vector of partial derivatives equal to zero, we arrive at the *normal equations*:

$$-\boldsymbol{X}^T(\boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}) = 0 \quad \Longleftrightarrow \quad \boldsymbol{X}^T\boldsymbol{X}\widehat{\boldsymbol{\beta}} = \boldsymbol{X}^T\boldsymbol{y}. \tag{3.1}$$

If $\boldsymbol{X}$ is full rank, the matrix $\boldsymbol{X}^T\boldsymbol{X}$ is invertible and we can therefore conclude that

$$\widehat{\boldsymbol{\beta}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}. \tag{3.2}$$

3.2 Probabilistic perspective

See also Agresti 2.7.1

3.2.1 Least squares as maximum likelihood estimation

Note that if $\boldsymbol{\epsilon}$ is assumed to be $N(0, \sigma^2 \boldsymbol{I_n})$, then the least squares solution would also be the maximum likelihood solution. Indeed, for $y_i \sim N(\mu_i, \sigma^2)$, the log-likelihood is:

$$\log \left[\prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i - \mu_i)^2}{2\sigma^2} \right) \right] = \text{constant} - \frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - \mu_i)^2.$$

3.2.2 Gauss-Markov theorem

Now that we have derived the least squares estimator, we can compute its bias and variance. To obtain the bias, we first calculate that:

$$\mathbb{E}[\widehat{\boldsymbol{\beta}}] = \mathbb{E}[(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}] = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\mathbb{E}[\boldsymbol{y}] = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\beta} = \boldsymbol{\beta}.$$

Therefore, the least squares estimator is unbiased. To obtain the variance, we compute:

$$\begin{aligned} \text{Var}[\widehat{\boldsymbol{\beta}}] &= \text{Var}[(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}] \\ &= (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\text{Var}[\boldsymbol{y}]\boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1} \\ &= (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T(\sigma^2\boldsymbol{I_n})\boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1} \\ &= \sigma^2(\boldsymbol{X}^T\boldsymbol{X})^{-1}. \end{aligned} \tag{3.3}$$

Theorem 3.1 (Gauss-Markov theorem)**.** *For homoskedastic linear models (eqs. (1.1) and (1.2)), the least squares coefficient estimates have the smallest covariance matrix (in the sense of positive semidefinite matrices) among all linear unbiased estimates of* $\boldsymbol{\beta}$*.*

3.3 Geometric perspective

See also Agresti 2.2.1-2.2.3

The following is the key geometric property of least squares (Figure 3.1).

Proposition 3.1. *The mapping* $\boldsymbol{y} \mapsto \widehat{\boldsymbol{\mu}} = \boldsymbol{X}\widehat{\boldsymbol{\beta}} \in C(\boldsymbol{X})$ *is an* orthogonal projection *onto* $C(\boldsymbol{X})$*, with projection matrix*

$$\boldsymbol{H} \equiv \boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T \quad (\textit{the hat matrix}). \tag{3.4}$$

Geometrically, this makes sense since we define $\widehat{\boldsymbol{\beta}}$ so that $\widehat{\boldsymbol{\mu}} \in C(\boldsymbol{X})$ is as close to $\boldsymbol{y}$ as possible. The shortest path between a point and a plane is the perpendicular. A simple example of $\boldsymbol{H}$ can be obtained by considering the intercept-only regression.

Proof. To prove that $\boldsymbol{y} \mapsto \widehat{\boldsymbol{\mu}}$ is an orthogonal projection onto $C(\boldsymbol{X})$, it suffices to show that:

$$\boldsymbol{v}^T(\boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}) = 0 \text{ for each } \boldsymbol{v} \in C(\boldsymbol{X}).$$

Since the columns $\{\boldsymbol{x_{*0}}, \ldots, \boldsymbol{x_{*p-1}}\}$ of $\boldsymbol{X}$ form a basis for $C(\boldsymbol{X})$, it suffices to show that $\boldsymbol{x}_{*j}^T(\boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}) = 0$ for each $j = 0, \ldots, p-1$. This is a consequence of the normal equations $\boldsymbol{X}^T(\boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}) = 0$ derived in (3.1).

To show that the projection matrix is $\boldsymbol{H}$ (3.4), it suffices to check that:

$$\widehat{\boldsymbol{\mu}} = \boldsymbol{X}\widehat{\boldsymbol{\beta}} = \boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y} \equiv \boldsymbol{H}\boldsymbol{y}.$$

□

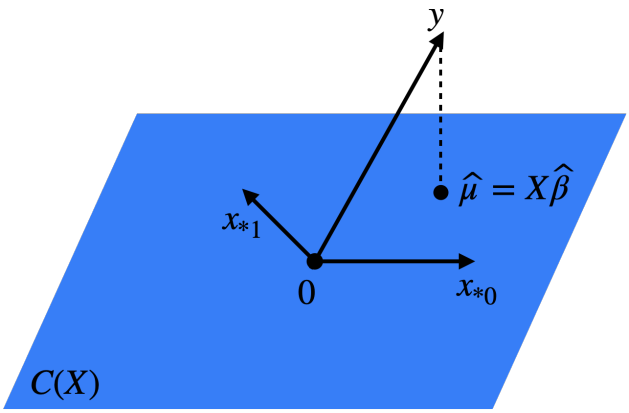


Figure 3.1: Least squares as orthogonal projection.

Proposition 3.2. *If $\boldsymbol{P}$ is an orthogonal projection onto a subspace $\boldsymbol{W}$, then:*

1. *$\boldsymbol{P}$ is idempotent, i.e., $\boldsymbol{P}^2 = \boldsymbol{P}$.*
2. *For all $\boldsymbol{v} \in \boldsymbol{W}$, we have $\boldsymbol{P}\boldsymbol{v} = \boldsymbol{v}$, and for all $\boldsymbol{v} \in \boldsymbol{W}^\perp$, we have $\boldsymbol{P}\boldsymbol{v} = 0$.*
3. *$trace(\boldsymbol{P}) = dim(\boldsymbol{W})$.*

One consequence of the geometric interpretation of least squares is that the fitted values $\widehat{\boldsymbol{\mu}}$ depend on $\boldsymbol{X}$ only through $C(\boldsymbol{X})$. As we will see in Homework 1, there are many different model matrices $\boldsymbol{X}$ leading to the same model space. Essentially, this reflects the fact that there are many different bases for the same vector space. Consider, for example, changing the units on the columns of $\boldsymbol{X}$. It can be verified that not just the fitted values $\widehat{\boldsymbol{\mu}}$ but also the predictions on a new set of features remain invariant to reparametrization (this follows from parts (a) and (b) of Homework 1 Problem 1). Therefore, while reparametrization can have a huge impact on the fitted coefficients, it has no impact on the predictions of linear regression.

Chapter 4

Analysis of variance

See also Agresti 2.4.2, 2.4.3, 2.4.6, Dunn and Smyth 2.9

4.1 Analysis of variance

The orthogonality property of least squares, together with the Pythagorean theorem, leads to a fundamental relationship called *the analysis of variance*.

Let's say that $S \subset \{0, 1, \ldots, p-1\}$ is a subset of the predictors we wish to exclude from the model. First regress $\boldsymbol{y}$ on $\boldsymbol{X}$ to get $\widehat{\boldsymbol{\beta}}$ as usual. Then, we consider the *partial model matrix* $\boldsymbol{X}_{*,\text{-}\boldsymbol{S}}$ obtained by selecting all predictors except those in S. Regressing $\boldsymbol{y}$ on $\boldsymbol{X}_{*,\text{-}\boldsymbol{S}}$ results in $\widehat{\boldsymbol{\beta}}_{\text{-}\boldsymbol{S}}$ (note: $\widehat{\boldsymbol{\beta}}_{\text{-}\boldsymbol{S}}$ is not necessarily obtained from $\widehat{\boldsymbol{\beta}}$ by extracting the coefficients corresponding to -S).

Theorem 4.1.

$$\|\boldsymbol{y} - \boldsymbol{X}_{*,\text{-}\boldsymbol{S}}\widehat{\boldsymbol{\beta}}_{\text{-}\boldsymbol{S}}\|^2 = \|\boldsymbol{X}\widehat{\boldsymbol{\beta}} - \boldsymbol{X}_{*,\text{-}\boldsymbol{S}}\widehat{\boldsymbol{\beta}}_{\text{-}\boldsymbol{S}}\|^2 + \|\boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}\|^2. \tag{4.1}$$

Proof. Consider the three points $\boldsymbol{y}$, $\boldsymbol{X}\widehat{\boldsymbol{\beta}}$, $\boldsymbol{X}_{*,\text{-}\boldsymbol{S}}\widehat{\boldsymbol{\beta}}_{\text{-}\boldsymbol{S}} \in \mathbb{R}^n$. Since $\boldsymbol{X}\widehat{\boldsymbol{\beta}}$ and $\boldsymbol{X}_{*,\text{-}\boldsymbol{S}}\widehat{\boldsymbol{\beta}}_{\text{-}\boldsymbol{S}}$ are both in $C(\boldsymbol{X})$, it follows by the orthogonal projection property that $\boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}$ is orthogonal to $\boldsymbol{X}\widehat{\boldsymbol{\beta}} - \boldsymbol{X}_{*,\text{-}\boldsymbol{S}}\widehat{\boldsymbol{\beta}}_{\text{-}\boldsymbol{S}}$. In other words, these three points form a right triangle (Figure 4.1). The relationship (4.1) is then a consequence of the Pythagorean theorem.

□

We will rely on this fundamental relationship throughout this course. One important special case is when $S = \{1, \ldots, p-1\}$, i.e., the model without S is the intercept-only model. In this case, $\boldsymbol{X}_{*,\text{-}\boldsymbol{S}} = \mathbf{1}_{\boldsymbol{n}}$ and $\widehat{\boldsymbol{\beta}}_{\text{-}\boldsymbol{S}} = \bar{y}$. Therefore, equation (4.1) implies the following.

Proposition 4.1.

$$\|\boldsymbol{y} - \bar{y}\mathbf{1}_{\boldsymbol{n}}\|^2 = \|\boldsymbol{X}\widehat{\boldsymbol{\beta}} - \bar{y}\mathbf{1}_{\boldsymbol{n}}\|^2 + \|\boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}\|^2.$$

Equivalently, we can rewrite this equation as follows:

$$\text{SST} \equiv \sum_{i=1}^{n}(y_i - \bar{y})^2 = \sum_{i=1}^{n}(\widehat{\mu}_i - \bar{y})^2 + \sum_{i=1}^{n}(y_i - \widehat{\mu}_i)^2 \equiv \text{SSR} + \text{SSE}. \tag{4.2}$$

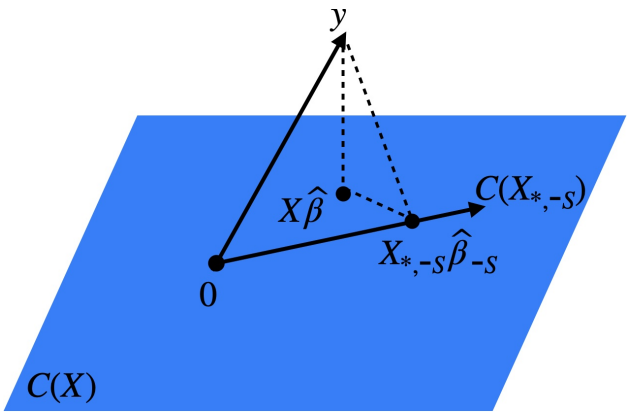


Figure 4.1: Pythagorean theorem for regression on a subset of predictors.

4.2 R^2 and multiple correlation

The ANOVA decomposition (4.2) of the variation in $\boldsymbol{y}$ into that explained by the linear regression model (SSR) and that left over (SSE) leads naturally to the definition of R^2 as the fraction of variation in $\boldsymbol{y}$ explained by the linear regression model:

$$R^2 \equiv \frac{\text{SSR}}{\text{SST}} = \frac{\sum_{i=1}^n (\widehat{\mu}_i - \bar{y})^2}{\sum_{i=1}^n (y_i - \bar{y})^2} = \frac{\|\boldsymbol{X}\widehat{\boldsymbol{\beta}} - \bar{y}\mathbf{1}_n\|^2}{\|\boldsymbol{y} - \bar{y}\mathbf{1}_n\|^2}.$$

By the decomposition (4.2), we have $R^2 \in [0, 1]$. The closer R^2 is to 1, the more closely the data follow the fitted linear regression model. This intuition is formalized in the following result.

Proposition 4.2. *R^2 is the squared sample correlation between $\boldsymbol{X}\widehat{\boldsymbol{\beta}}$ and $\boldsymbol{y}$.*

For this reason, the positive square root of R^2 is called the *multiple correlation coefficient*.

Proof. The first step is to observe that the mean of $\boldsymbol{X}\widehat{\boldsymbol{\beta}}$ is $\bar{y}$ (this follows from the normal equations). Therefore, the sample correlation between $\boldsymbol{X}\widehat{\boldsymbol{\beta}}$ and $\boldsymbol{y}$ is the inner product of the unit-normalized vectors $\boldsymbol{X}\widehat{\boldsymbol{\beta}} - \bar{y}\mathbf{1}$ and $\boldsymbol{y} - \bar{y}\mathbf{1}$, which is the cosine of the angle between them. From the geometry of Figure 4.1, we find that the cosine of the angle between $\boldsymbol{X}\widehat{\boldsymbol{\beta}} - \bar{y}\mathbf{1}$ and $\boldsymbol{y} - \bar{y}\mathbf{1}$ is $\|\boldsymbol{X}\widehat{\boldsymbol{\beta}} - \bar{y}\mathbf{1}\|/\|\boldsymbol{y} - \bar{y}\mathbf{1}\|$. Squaring this relation gives the desired conclusion.

□

4.3 R^2 increases as predictors are added

The R^2 is an *in-sample* measure, i.e., it uses the same data to fit the model and to assess the quality of the fit. Therefore, it is generally an optimistic measure of the (out-of-sample) prediction error. One manifestation of this is that the R^2 increases if any predictors are added to the model (even if these predictors are "junk"). To see this, it suffices to show that SSE decreases as we add predictors. Without loss of generality, suppose that we start with a model with all predictors except those in $S \subset \{0, 1, \ldots, p-1\}$ and compare it to the model including all the predictors $\{0, 1, \ldots, p-1\}$. We can read off from the Pythagorean theorem (4.1) that:

$$\text{SSE}(\boldsymbol{X}_{*,\text{-}S}, \boldsymbol{y}) \equiv \|\boldsymbol{y} - \boldsymbol{X}_{*,\text{-}S}\widehat{\boldsymbol{\beta}}_{\text{-}S}\|^2 \geq \|\boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}\|^2 \equiv \text{SSE}(\boldsymbol{X}, \boldsymbol{y}).$$

Adding many junk predictors will have the effect of degrading predictive performance but will nevertheless increase R^2.

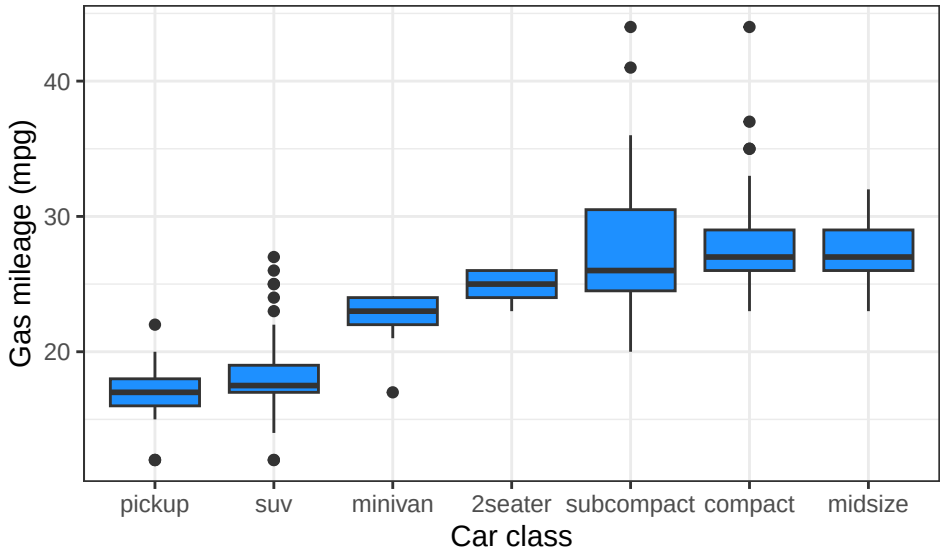
4.4 Special cases

4.4.1 The C-groups model

See also Agresti 2.3.2-2.3.3

4.4.1.1 ANOVA decomposition for C groups model

Let's consider the special case of the ANOVA decomposition (4.2) when the model matrix $\boldsymbol{X}$ represents a single categorical predictor w. In this case, each observation i is associated with one of the C classes of w, which we denote $c(i) \in \{1, \ldots, C\}$. Let's consider the C groups of observations $\{i : c(i) = c\}$ for $c \in \{1, \ldots, C\}$. For example, w may be the type of a car (compact, midsize, minivan, etc.) and y might be its fuel efficiency in miles per gallon.



It is easy to check that the least squares fitted values $\widehat{\mu}_i$ are simply the means of the corresponding groups:

$$\widehat{\mu}_i = \bar{y}_{c(i)}, \quad \text{where } \bar{y}_{c(i)} \equiv \frac{\sum_{i:c(i)=c} y_i}{|\{i : c(i) = c\}|}.$$

Therefore, we have:

$$\text{SSR} = \sum_{i=1}^{n}(\widehat{\mu}_i - \bar{y})^2 = \sum_{i=1}^{n}(\bar{y}_{c(i)} - \bar{y})^2 \equiv \text{between-groups sum of squares (SSB)}.$$

and

$$\text{SSE} = \sum_{i=1}^{n}(y_i - \widehat{\mu}_i)^2 = \sum_{i=1}^{n}(y_i - \bar{y}_{c(i)})^2 \equiv \text{within-groups sum of squares (SSW)}.$$

We therefore obtain the following corollary of the ANOVA decomposition (4.2):

$$\text{SST} = \text{SSB} + \text{SSW}. \tag{4.3}$$

4.4.2 Simple linear regression

See also Agresti 2.1.3

Consider a linear regression model with an intercept and one quantitative predictor, x:

$$y = \beta_0 + \beta_1 x + \epsilon. \tag{4.4}$$

This is the simple linear regression model. In this case, we can compute that

$$\widehat{\beta}_1 = \frac{\sigma_y}{\sigma_x}\rho_{xy}, \tag{4.5}$$

where ρ_{xy} is the sample correlation between x and y, σ_x^2 is the sample variance of x, and σ_y^2 is the sample variance of y. Furthermore, we have

$$\widehat{\beta}_0 = \bar{y} - \widehat{\beta}_1 \bar{x}. \tag{4.6}$$

4.4.2.1 ANOVA decomposition for simple linear regression

Figure 4.2 gives an interpretation of the ANOVA decomposition (4.2) in the case of the simple linear regression model (4.4).

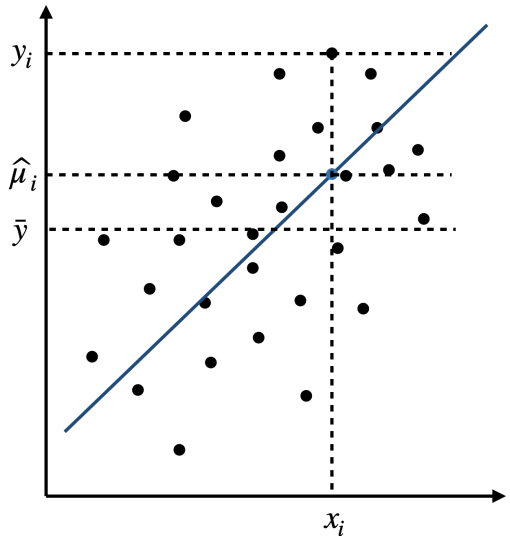


Figure 4.2: ANOVA decomposition for simple linear regression.

4.4.2.2 Connection between R^2 and correlation

There is a connection between R^2 and correlation in simple linear regression.

Proposition 4.3. *Let ρ_{xy} denote the sample correlation between x and y, and let R^2_{xy} be the R^2 from the simple linear regression (4.4). Then, we have:*

$$R^2 = \rho^2_{xy}.$$

Proof. This fact is a consequence of Proposition 4.2.

□

4.4.2.3 Regression to the mean

Simple linear regression can be used to study the relationship between the same quantity across time (or generations). For example, let x and y be the height of a parent and child. This example motivated Sir Francis Galton to study linear regression in the first place. Alternatively, x and y can be a student's score on a standardized test in two consecutive years, or the number of games won by a given sports team in two consecutive seasons. In this situation, it is reasonable to assume that the sample standard deviations of x and y are the same (or to normalize these variables to achieve this). In this case, equations (4.5) and (4.6) simplify to:

$$\widehat{\beta}_0 = \bar{y} - \rho_{xy}\bar{x} \quad \text{and} \quad \widehat{\beta}_1 = \rho_{xy}. \tag{4.7}$$

It follows that:

$$|\widehat{\mu}_i - \bar{y}| = |\widehat{\beta}_0 + \widehat{\beta}_1 x_i - \bar{y}| = |\rho_{xy}(x_i - \bar{x})| = |\rho_{xy}| \cdot |x_i - \bar{x}|.$$

Since $|\rho_{xy}| < 1$ unless $\boldsymbol{x}$ and $\boldsymbol{y}$ are perfectly correlated (by the Cauchy-Schwarz inequality), this means that:

$$|\widehat{\mu}_i - \bar{y}| < |x_i - \bar{x}| \quad \text{for each } i. \tag{4.8}$$

Therefore, we expect y_i to be closer to its mean than x_i is to its mean. This phenomenon is called *regression to the mean* (and is in fact the origin of the term "regression"). Many mistakenly attribute a causal mechanism to this phenomenon, when in reality it is simply a statistical artifact. For example, suppose x_i is the number of games a sports team won last season and y_i is the number of games it won this season. It is widely observed that teams with exceptional performance in a given season suffer a "winner's curse," performing worse in the next season. The reason for the winner's curse is simple: teams perform exceptionally well due to a combination of skill and luck. While skill stays roughly constant from year to year, the team which performed exceptionally well in a given season is unlikely to get as lucky as it did the next season.

Chapter 5

Collinearity and adjustment

See also Agresti 2.2.4, 2.5.6, 2.5.7, 4.6.5

An important part of linear regression analysis is the dependence of the least squares coefficient for a predictor (x_j) on what other predictors are in the model ($\boldsymbol{x}_{\text{-}j}$). This relationship is dictated by the extent to which $\boldsymbol{x}_{*j}$ is correlated with $\boldsymbol{X}_{*,\text{-}j}$. To explore this phenomenon, it will be useful to compare two different regressions:

- Regress $\boldsymbol{y}$ on *just* $\boldsymbol{x}_{*j}$. Let the resulting coefficient for x_j be $\widehat{\beta}_j$.
- Regress $\boldsymbol{y}$ on *all of* $\boldsymbol{X}$ (i.e., on both $\boldsymbol{x}_{*j}$ and $\boldsymbol{X}_{*,\text{-}j}$). Let the resulting coefficients for x_j and $\boldsymbol{x}_{\text{-}j}$ be $\widehat{\beta}_{j|\text{-}j}$ and $\widehat{\beta}_{\text{-}j|j}$, respectively.

5.1 Least squares estimates in the orthogonal case

The simplest case to analyze is when $\boldsymbol{x}_{*j}$ is orthogonal to $\boldsymbol{X}_{*,\text{-}j}$ in the sense that

$$\boldsymbol{x}_{*j}^T \boldsymbol{X}_{*,\text{-}j} = \boldsymbol{0}. \tag{5.1}$$

In this case, we can derive the least squares coefficient vector $\widehat{\boldsymbol{\beta}} = (\widehat{\beta}_{j|\text{-}j}, \widehat{\boldsymbol{\beta}}_{\text{-}j|j})$ in the regression of $\boldsymbol{y}$ on $\boldsymbol{X}$:

$$\begin{aligned}
\begin{pmatrix} \widehat{\beta}_{j|\text{-}j} \\ \widehat{\boldsymbol{\beta}}_{\text{-}j|j} \end{pmatrix} &= (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y} \\
&= \begin{pmatrix} \boldsymbol{x}_{*j}^T\boldsymbol{x}_{*j} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{X}_{*,\text{-}j}^T\boldsymbol{X}_{*,\text{-}j} \end{pmatrix}^{-1} \begin{pmatrix} \boldsymbol{x}_{*j}^T \\ \boldsymbol{X}_{*,\text{-}j}^T \end{pmatrix} \boldsymbol{y} \\
&= \begin{pmatrix} (\boldsymbol{x}_{*j}^T\boldsymbol{x}_{*j})^{-1}\boldsymbol{x}_{*j}^T\boldsymbol{y} \\ (\boldsymbol{X}_{*,\text{-}j}^T\boldsymbol{X}_{*,\text{-}j})^{-1}\boldsymbol{X}_{*,\text{-}j}^T\boldsymbol{y} \end{pmatrix} \\
&= \begin{pmatrix} \widehat{\beta}_j \\ \widehat{\boldsymbol{\beta}}_{\text{-}j} \end{pmatrix}.
\end{aligned} \tag{5.2}$$

Therefore, the least squares coefficient of x_j is the same regardless of whether the other predictors are included in the regression, i.e.

$$\widehat{\beta}_{j|\text{-}j} = \widehat{\beta}_j. \tag{5.3}$$

5.2 Least squares estimates via orthogonalization

The orthogonality assumption (5.1) is almost never satisfied in practice. Usually, $\boldsymbol{x}_{*j}$ has a nonzero projection $\boldsymbol{X}_{*,\text{-}j}\widehat{\boldsymbol{\gamma}}$ onto $C(\boldsymbol{X}_{*,\text{-}j})$:

$$\boldsymbol{x}_{*j} = \boldsymbol{X}_{*,\text{-}j}\widehat{\boldsymbol{\gamma}} + \boldsymbol{x}_{*j}^{\perp},$$

where $\boldsymbol{x}_{*j}^{\perp}$ is the residual from regressing $\boldsymbol{x}_{*j}$ onto $\boldsymbol{X}_{*,\text{-}j}$ and is therefore orthogonal to $C(\boldsymbol{X}_{*,\text{-}j})$. In other words, $\boldsymbol{x}_{*j}^{\perp}$ is the projection of $\boldsymbol{x}_{*j}$ onto the orthogonal complement of $C(\boldsymbol{X}_{*,\text{-}j})$. Another way of framing this relationship is that $\boldsymbol{x}_{*j}^{\perp}$ is the result of *adjusting* $\boldsymbol{x}_{*j}$ for $\boldsymbol{X}_{*,\text{-}j}$.

With this decomposition, let us change basis from $(\boldsymbol{x}_{*j}, \boldsymbol{X}_{*,\text{-}j})$ to $(\boldsymbol{x}_{*j}^{\perp}, \boldsymbol{X}_{*,\text{-}j})$ by the process explored in Homework 1 Question 1. Let us write:

$$\begin{aligned}
&\boldsymbol{y} = \boldsymbol{x}_{*j}\beta_{j|\text{-}j} + \boldsymbol{X}_{*,\text{-}j}\boldsymbol{\beta}_{\text{-}j|j} + \boldsymbol{\epsilon} \\
&\iff \boldsymbol{y} = (\boldsymbol{X}_{*,\text{-}j}\widehat{\boldsymbol{\gamma}} + \boldsymbol{x}_{*j}^{\perp})\beta_{j|\text{-}j} + \boldsymbol{X}_{*,\text{-}j}\boldsymbol{\beta}_{\text{-}j|j} + \boldsymbol{\epsilon} \\
&\iff \boldsymbol{y} = \boldsymbol{x}_{*j}^{\perp}\beta_{j|\text{-}j} + \boldsymbol{X}_{*,\text{-}j}\boldsymbol{\beta}'_{\text{-}j|j} + \boldsymbol{\epsilon}.
\end{aligned}$$

What this means is that $\widehat{\beta}_{j|\text{-}j}$, the least squares coefficient of $\boldsymbol{x}_{*j}$ in the regression of $\boldsymbol{y}$ on $(\boldsymbol{x}_{*j}, \boldsymbol{X}_{*,\text{-}j})$, is also the least squares coefficient of $\boldsymbol{x}_{*j}^{\perp}$ in the regression of $\boldsymbol{y}$ on $(\boldsymbol{x}_{*j}^{\perp}, \boldsymbol{X}_{*,\text{-}j})$. However, since $\boldsymbol{x}_{*j}^{\perp}$ is orthogonal to $\boldsymbol{X}_{*,\text{-}j}$ by construction, we can use the result (5.2) to conclude that:

$\widehat{\beta}_{j|\text{-}j}$ is the least squares coefficient of $\boldsymbol{x}_{*j}^{\perp}$ in the *univariate* regression of $\boldsymbol{y}$ on $\boldsymbol{x}_{*j}^{\perp}$.

We can solve this univariate regression explicitly to obtain:

$$\widehat{\beta}_{j|\text{-}j} = \frac{(\boldsymbol{x}_{*j}^{\perp})^T\boldsymbol{y}}{\|\boldsymbol{x}_{*j}^{\perp}\|^2}. \tag{5.4}$$

5.3 Adjustment and partial correlation

Equivalently, letting $\widehat{\boldsymbol{\beta}}_{\text{-}j}$ be the least squares estimate in the regression of $\boldsymbol{y}$ on $\boldsymbol{X}_{*,\text{-}j}$ (note that this is *not* the same as $\widehat{\boldsymbol{\beta}}_{\text{-}j|j}$), we can write:

$$\widehat{\beta}_{j|\text{-}j} = \frac{(\boldsymbol{x}_{*j}^{\perp})^T(\boldsymbol{y} - \boldsymbol{X}_{*,\text{-}j}\widehat{\boldsymbol{\beta}}_{\text{-}j})}{\|\boldsymbol{x}_{*j}^{\perp}\|^2} \equiv \frac{(\boldsymbol{x}_{*j}^{\perp})^T\boldsymbol{y}^{\perp}}{\|\boldsymbol{x}_{*j}^{\perp}\|^2}.$$

We can interpret this result as follows:

Theorem 5.1. *The linear regression coefficient $\widehat{\beta}_{j|\text{-}j}$ results from first adjusting $\boldsymbol{y}$ and $\boldsymbol{x}_{*j}$ for the effects of all other variables, and then regressing the residuals from $\boldsymbol{y}$ onto the residuals from $\boldsymbol{x}_{*j}$.*

In this sense, *the least squares coefficient for a predictor in a multiple linear regression reflects the effect of the predictor on the response after controlling for the effects of all other predictors.*

Econometricians calls this the Frisch-Waugh-Lovell (FWL) theorem, to acknowledge economists Ragnar Frisch and Frederick V. Waugh, who first derived the result in 1933, and Michael C. Lovell, who later rediscovered and extended it in 1963. In the statistical literature, this fact was known at least as early as 1907, when Yule documented it in his paper "On the Theory of Correlation for any Number of Variables, treated by a New System of Notation."

A related quantity is the *partial correlation* between $\boldsymbol{x}_{*j}$ and $\boldsymbol{y}$ after controlling for $\boldsymbol{X}_{*,-j}$, defined as the empirical correlation between $\boldsymbol{x}_{*j}^{\perp}$ and $\boldsymbol{y}^{\perp}$:

$$\rho(\boldsymbol{x}_{*j}, \boldsymbol{y} | \boldsymbol{X}_{*,-j}) \equiv \frac{(\boldsymbol{x}_{*j}^{\perp})^T (\boldsymbol{y}^{\perp})}{\|\boldsymbol{x}_{*j}^{\perp}\| \|\boldsymbol{y}^{\perp}\|}.$$

We can then connect the least squares coefficient $\widehat{\beta}_{j|\text{-}j}$ to this partial correlation:

$$\widehat{\beta}_{j|\text{-}j} = \frac{(\boldsymbol{x}_{*j}^{\perp})^T \boldsymbol{y}^{\perp}}{\|\boldsymbol{x}_{*j}^{\perp}\|^2} = \frac{\|\boldsymbol{y}^{\perp}\|}{\|\boldsymbol{x}_{*j}^{\perp}\|} \rho(\boldsymbol{x}_{*j}, \boldsymbol{y} | \boldsymbol{X}_{*,-j}),$$

in a similar spirit to equation (4.5).

5.4 Effects of collinearity

Collinearity between a predictor x_j and the other predictors tends to make the estimate $\widehat{\beta}_{j|\text{-}j}$ unstable. Intuitively, this makes sense because it becomes harder to distinguish between the effects of predictor x_j and those of the other predictors on the response. To find the variance of $\widehat{\beta}_{j|\text{-}j}$ for a model matrix $\boldsymbol{X}$, we could in principle use the formula (3.3). However, this formula involves the inverse of the matrix $\boldsymbol{X}^T \boldsymbol{X}$, which is hard to reason about. Instead, we can employ the formula (5.4) to calculate directly that:

$$\text{Var}[\widehat{\beta}_{j|\text{-}j}] = \frac{\sigma^2}{\|\boldsymbol{x}_{*j}^{\perp}\|^2}. \tag{5.5}$$

We see that the variance of $\widehat{\beta}_{j|\text{-}j}$ is inversely proportional to $\|\boldsymbol{x}_{*j}^{\perp}\|^2$. This means that the greater the collinearity, the less of $\boldsymbol{x}_{*j}$ is left over after adjusting for $\boldsymbol{X}_{*,-j}$, and the greater the variance of $\widehat{\beta}_{j|\text{-}j}$. To quantify the effect of this adjustment, suppose there were no other predictors other than the intercept term. Then, we would have:

$$\text{Var}[\widehat{\beta}_{j|1}] = \frac{\sigma^2}{\|\boldsymbol{x}_{*j} - \bar{x}_j \mathbf{1}_n\|^2}.$$

Therefore, we can rewrite the variance (5.5) as:

$$\begin{aligned} \text{Var}[\widehat{\beta}_{j|\text{-}j}] &= \frac{\|\boldsymbol{x}_{*j} - \bar{x}_j \mathbf{1}_n\|^2}{\|\boldsymbol{x}_{*j} - \boldsymbol{X}_{*,-j} \widehat{\boldsymbol{\gamma}}\|^2} \cdot \text{Var}[\widehat{\beta}_{j|1}] \\ &= \frac{1}{1 - R_j^2} \cdot \text{Var}[\widehat{\beta}_{j|1}] \equiv \text{VIF}_j \cdot \text{Var}[\widehat{\beta}_{j|1}], \end{aligned} \tag{5.6}$$

where R_j^2 is the R^2 value when regressing $\boldsymbol{x}_{*j}$ on $\boldsymbol{X}_{*,\text{-}j}$ and VIF stands for *variance inflation factor*. The higher R_j^2, the more of the variance in $\boldsymbol{x}_{*j}$ is explained by other predictors, the higher the variance in $\widehat{\beta}_{j|\text{-}j}$.

5.5 Application: Average treatment effect estimation in causal inference

Suppose we'd like to study the effect of an exposure or treatment (e.g. taking a blood pressure medication) on a response y (e.g. blood pressure). In the Neyman-Rubin causal model, for a given individual i we denote by $y_i(1)$ and $y_i(0)$ the outcomes that would have occurred had the individual received the treatment and the control, respectively. These are called *potential outcomes*. Let $t_i \in \{0, 1\}$ indicate whether the ith individual actually received treatment or control. Therefore, the observed outcome is[1]

$$y_i^{\text{obs}} = y_i(t_i). \tag{5.7}$$

Based on the data $\{(t_i, y_i^{\text{obs}})\}_{i=1,\dots,n}$, the most basic goal is to estimate the

$$\textit{average treatment effect } \tau \equiv \mathbb{E}[y(1) - y(0)],$$

where averaging is done over the population of individuals (often called *units* in causal inference). Of course, we do not observe both $y_i(1)$ and $y_i(0)$ for any unit i. Additionally, usually in observational studies we have *confounding variables* $w_2, \dots, w_{p-1}$: variables that influence both the treatment assignment and the response (e.g. degree of health-seeking activity). It is important to control for these confounders in order to get an unbiased estimate of the treatment effect. Suppose the following linear model holds:

$$y(t) = \beta_0 + \beta_1 t + \beta_2 w_2 + \cdots + \beta_{p-1} w_{p-1} + \epsilon \quad \text{for } t \in \{0, 1\}, \quad \text{where } \epsilon \perp\!\!\!\perp t.$$

This assumption can be broken down into the following statements:

- the treatment effect is constant across units;
- the response is a linear function of the treatment and observed confounders;
- there is no unmeasured confounding.

Under these assumptions, we find that

$$\tau \equiv \mathbb{E}[y(1) - y(0)] = \beta_1.$$

Using the relationship (5.7), we find that

$$y^{\text{obs}} = \beta_0 + \beta_1 t + \beta_2 w_2 + \cdots + \beta_{p-1} w_{p-1} + \epsilon \quad \text{for } t \in \{0, 1\}.$$

In this case, the average treatment effect τ is *identified* as the coefficient β_1 in the above regression, i.e. $\tau = \beta_1$. In other words, the causal parameter coincides with a parameter of the statistical

[1] The Fisher information is the expectation of the Hessian, but for canonical links, the Hessian is non-random, so the two coincide.

model for the observed data. Therefore, the least squares estimate $\widehat{\beta}_1$ is an unbiased estimate of the average treatment effect.

In this context, we can interpret Theorem 5.1 as follows: To get an estimate of the causal effect of an exposure on an outcome in the presence of confounders, first *adjust* both exposure and outcome for the confounders, and then estimate the effect of the adjusted exposure on the adjusted outcome via a univariate linear model. This is the essence of *covariate adjustment* in causal inference.

> **Note**
>
> Causal inference is a vast field, which lies mostly beyond the scope of STAT 9610; see STAT 9210 instead.

Chapter 6

R demo

See also Agresti 2.6, Dunn and Smyth 2.6

The R demo will be based on the `ScotsRaces` data from the Agresti textbook. Data description (quoted from the textbook):

> "Each year the Scottish Hill Runners Association publishes a list of hill races in Scotland for the year. The table below shows data on the record time for some of the races (in minutes). Explanatory variables listed are the distance of the race (in miles) and the cumulative climb (in thousands of feet)."

We will also familiarize ourselves with several important functions from the `tidyverse` packages, including the `ggplot2` package for data visualization and `dplyr` package for data manipulation.

```
library(tidyverse) # for data import, manipulation, and plotting
library(GGally)    # for ggpairs() function
library(ggrepel)   # for geom_text_repel() function
library(car)       # for vif() function
library(conflicted)
conflicts_prefer(dplyr::filter)
```

```
# read the data into R
scots_races <- read_tsv("data/ScotsRaces.dat") # read_tsv from readr for data import
scots_races
```

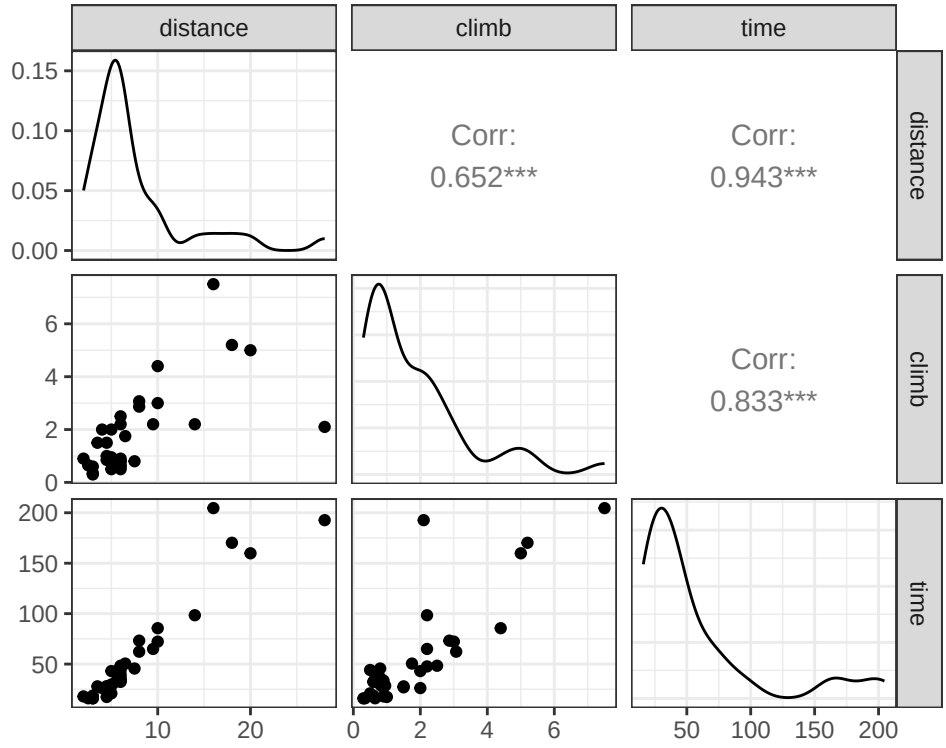
```
# A tibble: 35 x 4
   race                   distance climb  time
   <chr>                     <dbl> <dbl> <dbl>
 1 GreenmantleNewYearDash      2.5  0.65  16.1
 2 Carnethy5HillRace           6    2.5   48.4
 3 CraigDunainHillRace         6    0.9   33.6
 4 BenRhaHillRace              7.5  0.8   45.6
 5 BenLomondHillRace           8    3.07  62.3
 6 GoatfellHillRace            8    2.87  73.2
 7 BensofJuraFellRace         16    7.5  205.
```

```
 8 CairnpappleHillRace          6    0.8   36.4
 9 ScoltyHillRace               5    0.8   29.8
10 TraprainLawRace              6    0.65  39.8
# i 25 more rows
```

6.1 Exploration

Before modeling our data, let's first explore it.

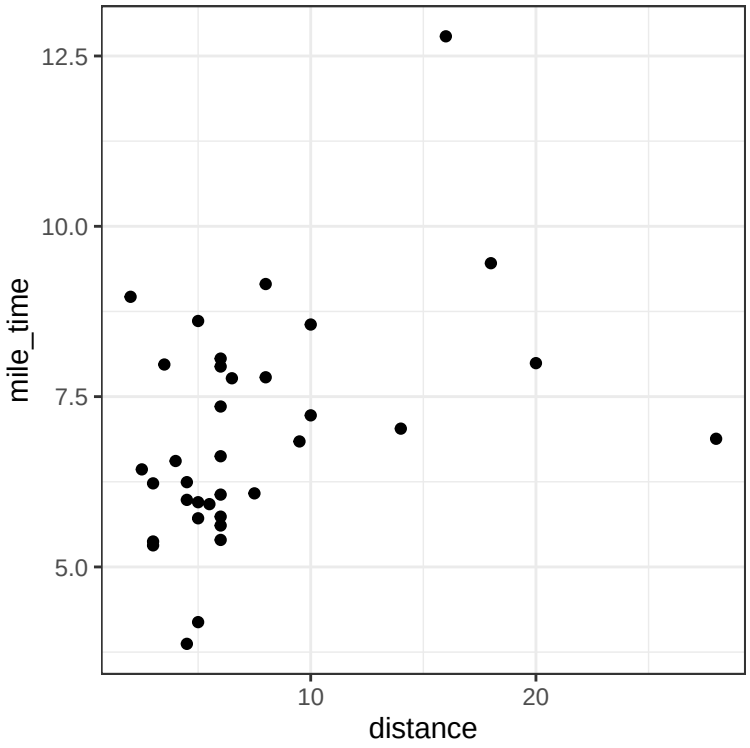
```
# pairs plot

# Q: What are the typical ranges of the variables?
# Q: What are the relationships among the variables?

scots_races |>
  select(-race) |> # select() from dplyr for selecting columns
  ggpairs() # ggpairs() from GGally to create pairs plot
```



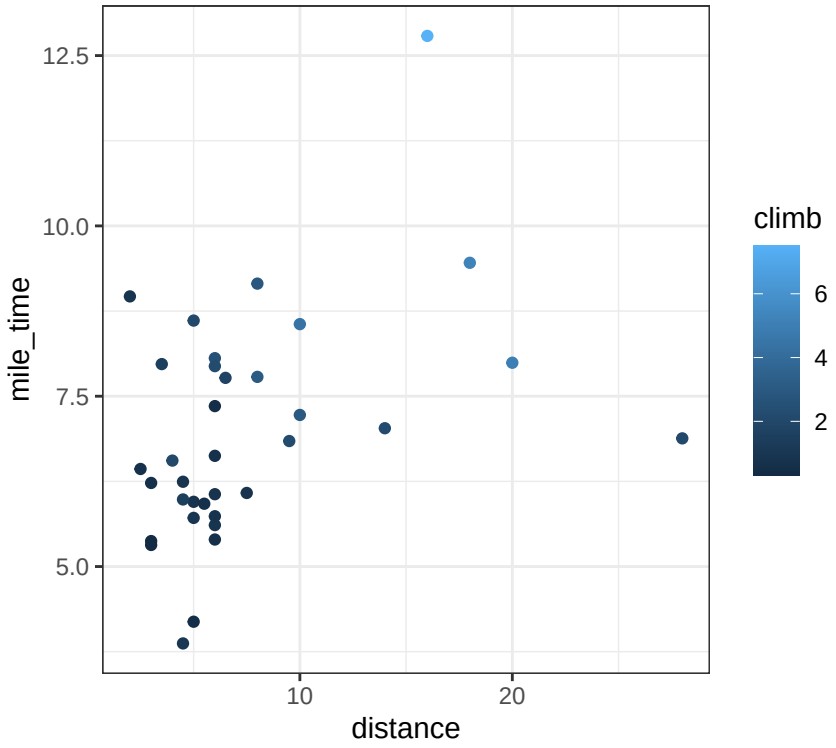

```
# mile time versus distance

# Q: How does mile time vary with distance?
# Q: What races deviate from this trend?
# Q: How does climb play into it?
```

```r
# add mile time variable to scots_races
scots_races <- scots_races |>
  mutate(mile_time = time / distance) # mutate() from dplyr to add column
```

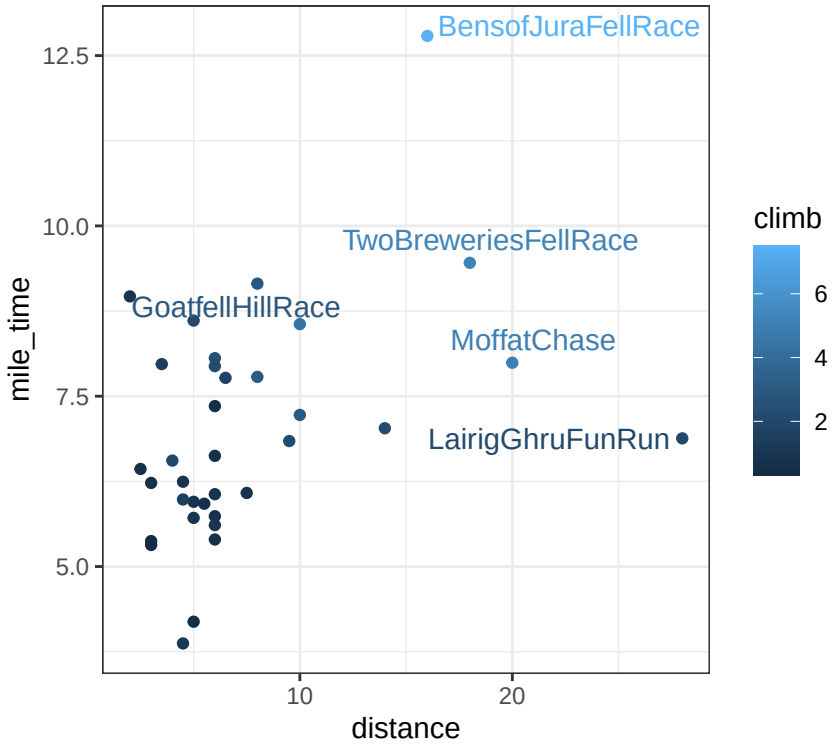
```r
# plot mile time versus distance
scots_races |>
  ggplot(aes(x = distance, y = mile_time)) +
  geom_point()
```



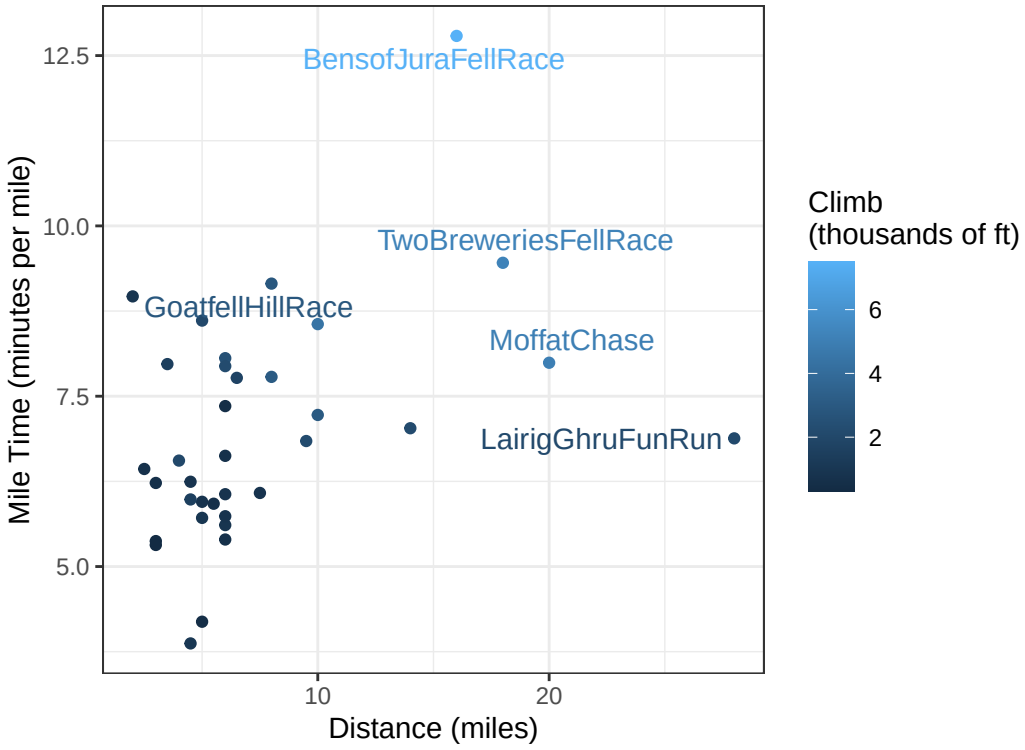

```r
# add climb information as point color
scots_races |>
  ggplot(aes(x = distance, y = mile_time, colour = climb)) +
  geom_point()
```




```
# highlight extreme points
scots_races_extreme <- scots_races |>
  filter(distance > 15 | mile_time > 9) # filter() from dplyr to subset rows

# plot mile time versus distance
scots_races |>
  ggplot(aes(x = distance, y = mile_time, label = race, colour = climb)) +
  geom_point() +
  geom_text_repel(aes(label = race), data = scots_races_extreme)
```




```
# clean up plot
scots_races |>
  ggplot(aes(x = distance, y = mile_time, label = race, color = climb)) +
  geom_point() +
  geom_text_repel(aes(label = race), data = scots_races_extreme) +
  labs(
    x = "Distance (miles)",
    y = "Mile Time (minutes per mile)",
    color = "Climb\n(thousands of ft)"
  )
```



6.2 Linear model coefficient interpretation

Let's fit some linear models and interpret the coefficients.

```
# Q: What is the effect of an extra mile of distance on time?

lm_fit <- lm(time ~ distance + climb, data = scots_races)
coef(lm_fit)
```

```
(Intercept)    distance       climb
 -13.108551    6.350955   11.780133
```

```
# Linear model with interaction

# Q: What is the effect of an extra mile of distance on time
#  for a run with low climb?

# Q: What is the effect of an extra mile of distance on time
#  for a run with high climb?

lm_fit_int <- lm(time ~ distance * climb, data = scots_races)
coef(lm_fit_int)
```

```
  (Intercept)       distance          climb distance:climb
   -0.7671925      4.9622542      3.7132519      0.6598256
```

```
  scots_races |>
    summarise(min_climb = min(climb), max_climb = max(climb))
```

```
# A tibble: 1 x 2
  min_climb max_climb
      <dbl>     <dbl>
1       0.3       7.5
```

Let's take a look at the regression summary for `lm_fit`:

```
  lm_fit <- lm(time ~ distance + climb, data = scots_races)
  summary(lm_fit)
```

```
Call:
lm(formula = time ~ distance + climb, data = scots_races)

Residuals:
    Min      1Q  Median      3Q     Max
-16.654  -4.842   1.110   4.667  27.762

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -13.1086     2.5608  -5.119 1.41e-05 ***
distance      6.3510     0.3578  17.751  < 2e-16 ***
climb        11.7801     1.2206   9.651 5.37e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 8.734 on 32 degrees of freedom
Multiple R-squared:  0.9717,    Adjusted R-squared:   0.97
F-statistic: 549.9 on 2 and 32 DF,  p-value: < 2.2e-16
```

We get a coefficient of 6.35 with standard error 0.36 for `distance`, where the standard error is an estimate of the quantity (5.5).

6.3 R^2 and sum-of-squared decompositions.

We can extract the R^2 from this fit by reading it off from the bottom of the summary, or by typing

```
  summary(lm_fit)$r.squared
```

```
[1] 0.971725
```

We can construct sum-of-squares decompositions (4.1) using the `anova` function. This function takes as arguments the partial model and the full model. For example, consider the partial model `time ~ distance`.

```r
lm_fit_partial <- lm(time ~ distance, data = scots_races)
anova(lm_fit_partial, lm_fit)
```

```
Analysis of Variance Table

Model 1: time ~ distance
Model 2: time ~ distance + climb
  Res.Df    RSS Df Sum of Sq     F    Pr(>F)
1     33 9546.9
2     32 2441.3  1    7105.6 93.14 5.369e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We find that adding the predictor `climb` reduces the RSS by 7106, from 9547 to 2441. As another example, we can compute the R^2 by comparing the full model with the null model:

```r
lm_fit_null <- lm(time ~ 1, data = scots_races)
anova(lm_fit_null, lm_fit)
```

```
Analysis of Variance Table

Model 1: time ~ 1
Model 2: time ~ distance + climb
  Res.Df   RSS Df Sum of Sq      F    Pr(>F)
1     34 86340
2     32  2441  2     83899 549.87 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Therefore, the R^2 is $83899/86340 = 0.972$, consistent with the above regression summary.

6.4 Adjustment and collinearity.

We can also test the adjustment formula (5.4) numerically. Let's consider the coefficient of `distance` in the regression `time ~ distance + climb`. We can obtain this coefficient by first regressing `climb` out of `distance` and `time`:

```r
lm_dist_on_climb <- lm(distance ~ climb, data = scots_races)
lm_time_on_climb <- lm(time ~ climb, data = scots_races)

scots_races_resid <- tibble(
  dist_residuals = residuals(lm_dist_on_climb),
  time_residuals = residuals(lm_time_on_climb)
)

lm_adjusted <- lm(time_residuals ~ dist_residuals - 1,
  data = scots_races_resid
```

```
  )
  summary(lm_adjusted)
```

```
Call:
lm(formula = time_residuals ~ dist_residuals - 1, data = scots_races_resid)

Residuals:
    Min      1Q  Median      3Q     Max
-16.654  -4.842   1.110   4.667  27.762

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
dist_residuals   6.3510     0.3471    18.3   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 8.474 on 34 degrees of freedom
Multiple R-squared:  0.9078,    Adjusted R-squared:  0.9051
F-statistic: 334.8 on 1 and 34 DF,  p-value: < 2.2e-16
```

We find a coefficient of 6.35 with standard error 0.35, which matches that obtained in the original regression.

We can get the partial correlation between `distance` and `time` by taking the empirical correlation between the residuals. We can compare this quantity to the usual correlation.

```
  scots_races_resid |>
    summarise(cor(dist_residuals, time_residuals)) |>
    pull()
```

```
[1] 0.9527881
```

```
  scots_races |>
    summarise(cor(distance, time)) |>
    pull()
```

```
[1] 0.9430944
```

In this case, the two correlation quantities are similar.

To obtain the variance inflation factors defined in equation (5.6), we can use the `vif` function from the `car` package:

```
  vif(lm_fit)
```

```
distance    climb
1.740812 1.740812
```

Why are these two VIF values the same?

Part II

Linear models: Inference

We now understand the least squares estimator $\widehat{\boldsymbol{\beta}}$ from geometric and algebraic points of view. In Unit 2, we switch to a probabilistic perspective to derive inferential statements for linear models, in the form of hypothesis tests and confidence intervals. In order to facilitate this, we will assume that the error terms are normally distributed:

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \quad \text{where } \boldsymbol{\epsilon} \sim N(\boldsymbol{0}, \sigma^2 \boldsymbol{I}_n).$$

We first establish some building blocks necessary for linear models inference, primarily related to manipulating the normal distribution (Chapter 7). Then, we discuss univariate and multivariate hypothesis testing in linear models (Chapter 8), as well as the power of these hypothesis tests (Chapter 9). We then move on to the construction of confidence intervals and confidence regions (Chapter 10). We conclude with a discussion of practical considerations (Chapter 11) and an R demo (Chapter 12).

Chapter 7

Building blocks

See also Agresti 3.1.1, 3.1.2, 3.1.4

First we put in place some building blocks: The multivariate normal distribution (Section 7.1), the distributions of linear regression estimates and residuals (Section 7.2), and estimation of the noise variance σ^2 (Section 7.3).

7.1 The multivariate normal distribution

Recall that a random vector $\boldsymbol{w} \in \mathbb{R}^d$ has a multivariate normal distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ if it has probability density

$$p(\boldsymbol{w}) = \frac{1}{\sqrt{(2\pi)^d \det(\boldsymbol{\Sigma})}} \exp\left(-\frac{1}{2}(\boldsymbol{w}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{w}-\boldsymbol{\mu})\right).$$

It is also possible to define normal distributions in the case $\det(\boldsymbol{\Sigma}) = 0$. These distributions are supported on subspaces of $\mathbb{R}^d$, and do not have densities with respect to the Lebesgue measure.

These random vectors have lots of special properties, including:

- **Linear transformation**: If $\boldsymbol{w} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, then $\boldsymbol{A}\boldsymbol{w} + \boldsymbol{b} \sim N(\boldsymbol{A}\boldsymbol{\mu} + \boldsymbol{b}, \boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^\top)$.
- **Independence**: If
$$\begin{pmatrix} \boldsymbol{w}_1 \\ \boldsymbol{w}_2 \end{pmatrix} \sim N\left(\begin{pmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{pmatrix}, \begin{pmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{12}^\top & \boldsymbol{\Sigma}_{22} \end{pmatrix}\right),$$
then $\boldsymbol{w}_1 \perp\!\!\!\perp \boldsymbol{w}_2$ if and only if $\boldsymbol{\Sigma}_{12} = \boldsymbol{0}$.

An important distribution related to the multivariate normal is the χ^2_d (chi-squared with d degrees of freedom) distribution, defined as

$$\chi^2_d \equiv \sum_{j=1}^{d} w_j^2 \quad \text{for} \quad w_1, \ldots, w_d \overset{\text{i.i.d.}}{\sim} N(0,1).$$

7.2 The distributions of linear regression estimates and residuals

See also Dunn and Smyth 2.8.2

The most important distributional result in linear regression is that

$$\widehat{\boldsymbol{\beta}} \sim N(\boldsymbol{\beta}, \sigma^2(\boldsymbol{X}^\top \boldsymbol{X})^{-1}). \tag{7.1}$$

Indeed, by the linear transformation property of the multivariate normal distribution,$\boldsymbol{y} \sim N(\boldsymbol{X}\boldsymbol{\beta}, \sigma^2 \boldsymbol{I}_n)$ implies that

$$\begin{aligned}
\widehat{\boldsymbol{\beta}} &= (\boldsymbol{X}^\top \boldsymbol{X})^{-1}\boldsymbol{X}^\top \boldsymbol{y} \\
&\sim N((\boldsymbol{X}^\top \boldsymbol{X})^{-1}\boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{\beta}, (\boldsymbol{X}^\top \boldsymbol{X})^{-1}\boldsymbol{X}^\top \sigma^2 \boldsymbol{I}_n \boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{X})^{-1}) \\
&= N(\boldsymbol{\beta}, \sigma^2(\boldsymbol{X}^\top \boldsymbol{X})^{-1}).
\end{aligned}$$

Next, let's consider the joint distribution of $\widehat{\boldsymbol{\mu}} = \boldsymbol{X}\widehat{\boldsymbol{\beta}}$ and $\widehat{\boldsymbol{\epsilon}} = \boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}$. We have

$$\begin{aligned}
\begin{pmatrix} \widehat{\boldsymbol{\mu}} \\ \widehat{\boldsymbol{\epsilon}} \end{pmatrix} &= \begin{pmatrix} \boldsymbol{H}\boldsymbol{y} \\ (\boldsymbol{I} - \boldsymbol{H})\boldsymbol{y} \end{pmatrix} \\
&= \begin{pmatrix} \boldsymbol{H} \\ \boldsymbol{I} - \boldsymbol{H} \end{pmatrix} \boldsymbol{y} \\
&\sim N\left(\begin{pmatrix} \boldsymbol{H} \\ \boldsymbol{I} - \boldsymbol{H} \end{pmatrix} \boldsymbol{X}\boldsymbol{\beta}, \begin{pmatrix} \boldsymbol{H} \\ \boldsymbol{I} - \boldsymbol{H} \end{pmatrix} \cdot \sigma^2 \boldsymbol{I} \begin{pmatrix} \boldsymbol{H} & \boldsymbol{I} - \boldsymbol{H} \end{pmatrix} \right) \\
&= N\left(\begin{pmatrix} \boldsymbol{X}\boldsymbol{\beta} \\ \boldsymbol{0} \end{pmatrix}, \begin{pmatrix} \sigma^2 \boldsymbol{H} & \boldsymbol{0} \\ \boldsymbol{0} & \sigma^2(\boldsymbol{I} - \boldsymbol{H}) \end{pmatrix} \right).
\end{aligned}$$

In other words,

$$\widehat{\boldsymbol{\mu}} \sim N(\boldsymbol{X}\boldsymbol{\beta}, \sigma^2 \boldsymbol{H}) \quad \text{and} \quad \widehat{\boldsymbol{\epsilon}} \sim N(\boldsymbol{0}, \sigma^2(\boldsymbol{I} - \boldsymbol{H})), \quad \text{with} \quad \widehat{\boldsymbol{\mu}} \perp\!\!\!\perp \widehat{\boldsymbol{\epsilon}}. \tag{7.2}$$

The statistical independence between $\widehat{\boldsymbol{\mu}}$ and $\widehat{\boldsymbol{\epsilon}}$ is a result of the fact that these two quantities are projections of $\boldsymbol{y}$ onto two orthogonal subspaces: $C(\boldsymbol{X})$ and $C(\boldsymbol{X})^\perp$ (Figure 7.1).

Since $\widehat{\boldsymbol{\beta}}$ is a deterministic function of $\widehat{\boldsymbol{\mu}}$ (in particular, $\widehat{\boldsymbol{\beta}} = (\boldsymbol{X}^\top \boldsymbol{X})^{-1}\boldsymbol{X}^\top \widehat{\boldsymbol{\mu}}$), it also follows that

$$\widehat{\boldsymbol{\beta}} \perp\!\!\!\perp \widehat{\boldsymbol{\epsilon}}. \tag{7.3}$$

7.3 Estimation of the noise variance σ^2

See also Dunn and Smyth 2.4.2, 2.5.3

We can't quite do inference for $\boldsymbol{\beta}$ based on the distributional result (7.1) because the noise variance σ^2 is unknown to us. Intuitively, since $\sigma^2 = \mathbb{E}[\epsilon_i^2]$, we can get an estimate of σ^2 by looking at the quantity $\|\widehat{\boldsymbol{\epsilon}}\|^2$. To get the distribution of this quantity, we need the following lemma:

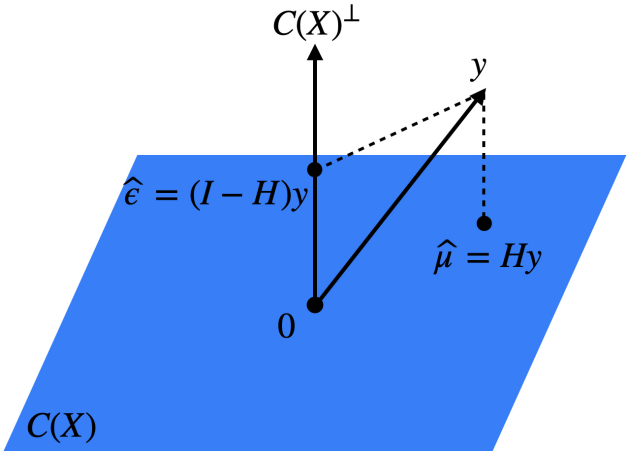


Figure 7.1: The fitted vector $\widehat{\boldsymbol{\mu}}$ and the residual vector $\widehat{\boldsymbol{\epsilon}}$ are projections of $\boldsymbol{y}$ onto orthogonal subspaces.

Lemma 7.1. *Let $\boldsymbol{w} \sim N(\boldsymbol{0}, \boldsymbol{P})$ for some projection matrix $\boldsymbol{P}$. Then, $\|\boldsymbol{w}\|^2 \sim \chi^2_d$, where $d = \text{trace}(\boldsymbol{P})$ is the dimension of the subspace onto which $\boldsymbol{P}$ projects.*

Proof. Let $\boldsymbol{P} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^\top$ be an eigenvalue decomposition of $\boldsymbol{P}$, where $\boldsymbol{U}$ is orthogonal and $\boldsymbol{D}$ is a diagonal matrix with $D_{ii} \in \{0, 1\}$. We have $\boldsymbol{w} \overset{d}{=} \boldsymbol{U}\boldsymbol{D}\boldsymbol{z}$ for $\boldsymbol{z} \sim N(0, \boldsymbol{I}_n)$. Therefore,

$$\|\boldsymbol{w}\|^2 = \|\boldsymbol{D}\boldsymbol{z}\|^2 = \sum_{i:D_{ii}=1} z_i^2 \sim \chi^2_d, \quad \text{where } d = |\{i : D_{ii} = 1\}| = \text{trace}(D) = \text{trace}(\boldsymbol{P}).$$

□

Recall that $\boldsymbol{I} - \boldsymbol{H}$ is a projection onto the $(n-p)$-dimensional space $C(\boldsymbol{X})^\perp$, so by Lemma 7.1 and equation (7.2), we have

$$\|\widehat{\boldsymbol{\epsilon}}\|^2 \sim \sigma^2 \chi^2_{n-p}. \tag{7.4}$$

From this result, it follows that $\mathbb{E}[\|\widehat{\boldsymbol{\epsilon}}\|^2] = \sigma^2(n-p)$, so

$$\widehat{\sigma}^2 \equiv \frac{1}{n-p}\|\widehat{\boldsymbol{\epsilon}}\|^2 \tag{7.5}$$

is an unbiased estimate for σ^2. Why does the denominator need to be $n-p$ rather than n for the estimator above to be unbiased? The reason for this is that the residuals $\widehat{\boldsymbol{\epsilon}}$ are the projection of the true noise vector $\boldsymbol{\epsilon} \in \mathbb{R}^n$ onto the $(n-p)$-dimensional subspace $C(\boldsymbol{X})^\perp$ (Figure 7.2). To see this, note that

$$\widehat{\boldsymbol{\epsilon}} = (\boldsymbol{I} - \boldsymbol{H})\boldsymbol{y} = (\boldsymbol{I} - \boldsymbol{H})(\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}) = (\boldsymbol{I} - \boldsymbol{H})\boldsymbol{\epsilon}.$$

Therefore, the norm of the residual vector will be smaller than that of the noise vector, especially to the extent that p is close to n.

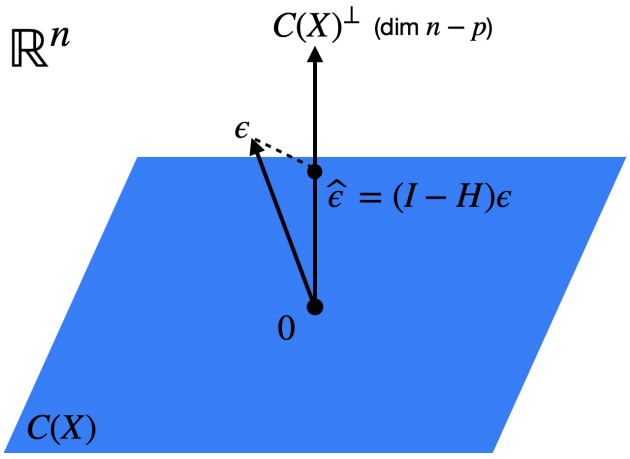


Figure 7.2: The residual vector $\widehat{\boldsymbol{\epsilon}}$ is the projection of the noise vector $\boldsymbol{\epsilon}$ onto $C(\boldsymbol{X})^\perp$.

Chapter 8

Hypothesis testing

See also Agresti 3.2.1, 3.2.2, 3.2.4, 3.2.8

Typically, two types of null hypotheses are tested in a regression setting: those involving one-dimensional parameters and those involving multi-dimensional parameters. For example, consider the null hypotheses $H_0 : \beta_j = 0$ and $H_0 : \boldsymbol{\beta}_S = \mathbf{0}$ for $S \subseteq \{0, 1, \ldots, p-1\}$, respectively. We discuss tests of these two kinds of hypotheses in Sections 8.1 and 8.2, and then discuss the power of these tests in Chapter 9.

8.1 Testing a one-dimensional parameter

See also Dunn and Smyth 2.8.3

8.1.1 t-test for a single coefficient

The most common question to ask in a linear regression context is: Is the jth predictor associated with the response when controlling for the other predictors? In the language of hypothesis testing, this corresponds to the null hypothesis:

$$H_0 : \beta_j = 0 \tag{8.1}$$

According to equation (7.1), we have $\widehat{\beta}_j \sim N(0, \sigma^2/s_j^2)$, where, as we learned in Chapter 1:

$$s_j^2 \equiv [(\boldsymbol{X}^T\boldsymbol{X})^{-1}_{jj}]^{-1} = \|\boldsymbol{x}_{*j}^{\perp}\|^2.$$

Therefore,

$$\frac{\widehat{\beta}_j}{\sigma/s_j} \sim N(0, 1), \tag{8.2}$$

and we are tempted to define a level α test of the null hypothesis (8.1) based on this normal distribution. While this is infeasible since we don't know σ^2, we can substitute in the unbiased estimate (7.5) derived in Section 7.3. Then,

$$\mathrm{SE}(\widehat{\beta}_j) \equiv \frac{\widehat{\sigma}}{s_j}$$

is the standard error of $\widehat{\beta}_j$, which is an approximation to the standard deviation of $\widehat{\beta}_j$. Dividing $\widehat{\beta}_j$ by its standard error gives us the t-statistic:

$$t_j \equiv \frac{\widehat{\beta}_j}{\mathrm{SE}(\widehat{\beta}_j)} = \frac{\widehat{\beta}_j}{\sqrt{\frac{1}{n-p}\|\widehat{\boldsymbol{\epsilon}}\|^2}/s_j}.$$

This statistic is *pivotal*, in the sense that it has the same distribution for any $\boldsymbol{\beta}$ such that $\beta_j = 0$. Indeed, we can rewrite it as:

$$t_j = \frac{\frac{\widehat{\beta}_j}{\sigma/s_j}}{\sqrt{\frac{\sigma^{-2}\|\widehat{\boldsymbol{\epsilon}}\|^2}{n-p}}}.$$

Recalling the independence of $\widehat{\boldsymbol{\beta}}$ and $\widehat{\boldsymbol{\epsilon}}$ (7.3), the scaled chi-square distribution of $\|\widehat{\boldsymbol{\epsilon}}\|^2$ (7.4), and the standard normal distribution of $\frac{\widehat{\beta}_j}{\sigma/s_j}$ (8.2), we find that, under $H_0 : \beta_j = 0$,

$$t_j \sim \frac{N(0,1)}{\sqrt{\frac{1}{n-p}\chi^2_{n-p}}}, \quad \text{with numerator and denominator independent.}$$

This distribution is called the t distribution with $n-p$ degrees of freedom and is denoted t_{n-p}. This paves the way for the two-sided t-test:

$$\phi_t(\boldsymbol{X}, \boldsymbol{y}) = 1(|t_j| > t_{n-p}(1-\alpha/2)),$$

where $t_{n-p}(1-\alpha/2)$ denotes the $1-\alpha/2$ quantile of t_{n-p}. Note that, by the law of large numbers,

$$\frac{1}{n-p}\chi^2_{n-p} \xrightarrow{P} 1 \quad \text{as} \quad n-p \to \infty,$$

so for large $n-p$ we have $t_j \sim t_{n-p} \approx N(0,1)$. Hence, the t-test is approximately equal to the following z-test:

$$\phi_t(\boldsymbol{X}, \boldsymbol{y}) \approx \phi_z(\boldsymbol{X}, \boldsymbol{y}) \equiv 1(|t_j| > z(1-\alpha/2)),$$

where $z(1-\alpha/2)$ is the $1-\alpha/2$ quantile of $N(0,1)$. The t-test can also be defined in a one-sided fashion if power against one-sided alternatives is desired.

8.1.2 Example: One-sample model

Consider the intercept-only linear regression model $y = \beta_0 + \epsilon$, and let usapply the t-test derived above to test the null hypothesis $H_0 : \beta_0 = 0$. We have $\widehat{\beta}_0 = \bar{y}$. Furthermore, we have

$$\mathrm{SE}^2(\widehat{\beta}_0) = \frac{\widehat{\sigma}^2}{n}, \quad \text{where} \quad \widehat{\sigma}^2 = \frac{1}{n-1}\|\boldsymbol{y} - \bar{y}\mathbf{1}_n\|^2.$$

Hence, we obtain the t statistic:

$$t = \frac{\widehat{\beta}_0}{\mathrm{SE}(\widehat{\beta}_0)} = \frac{\sqrt{n}\bar{y}}{\sqrt{\frac{1}{n-1}\|\boldsymbol{y} - \bar{y}\mathbf{1}_n\|^2}}.$$

According to the theory above, this test statistic has a null distribution of t_{n-1}.

8.1.3 Example: Two-sample model

Suppose we have $x_1 \in \{0, 1\}$, in which case the linear regression $y = \beta_0 + \beta_1 x_1 + \epsilon$ becomes a two-sample model. We can rewrite this model as:

$$y_i \sim \begin{cases} N(\beta_0, \sigma^2) & \text{for } x_i = 0; \\ N(\beta_0 + \beta_1, \sigma^2) & \text{for } x_i = 1. \end{cases}$$

It is often of interest to test the null hypothesis $H_0 : \beta_1 = 0$, i.e., that the two groups have equal means. Let usdefine:

$$\bar{y}_0 \equiv \frac{1}{n_0}\sum_{i:x_i=0} y_i, \quad \bar{y}_1 \equiv \frac{1}{n_1}\sum_{i:x_i=1} y_i, \quad \text{where} \quad n_0 = |\{i : x_i = 0\}| \text{ and } n_1 = |\{i : x_i = 1\}|.$$

Then, we have seen before that $\widehat{\beta}_0 = \bar{y}_0$ and $\widehat{\beta}_1 = \bar{y}_1 - \bar{y}_0$. We can compute that:

$$s_1^2 \equiv \|\boldsymbol{x}_{*1}^{\perp}\|^2 = \|\boldsymbol{x}_{*1} - \frac{n_1}{n}\mathbf{1}\|^2 = n_1\frac{n_0^2}{n^2} + n_0\frac{n_1^2}{n^2} = \frac{n_0 n_1}{n} = \frac{1}{\frac{1}{n_0} + \frac{1}{n_1}}$$

and

$$\widehat{\sigma}^2 = \frac{1}{n-2}\left(\sum_{i:x_i=0}(y_i - \bar{y}_0)^2 + \sum_{i:x_i=1}(y_i - \bar{y}_1)^2\right).$$

Therefore, we arrive at a t-statistic of:

$$t = \frac{\sqrt{\frac{1}{\frac{1}{n_0}+\frac{1}{n_1}}}(\bar{y}_1 - \bar{y}_0)}{\sqrt{\frac{1}{n-2}\left(\sum_{i:x_i=0}(y_i - \bar{y}_0)^2 + \sum_{i:x_i=1}(y_i - \bar{y}_1)^2\right)}}.$$

Under the null hypothesis, this statistic has a distribution of t_{n-2}.

8.1.4 t-test for a contrast among coefficients

Given a vector $\boldsymbol{c} \in \mathbb{R}^p$, the quantity $\boldsymbol{c}^T\boldsymbol{\beta}$ is sometimes called a *contrast*. For example, suppose $\boldsymbol{c} = (1, -1, 0, \ldots, 0)$. Then, $\boldsymbol{c}^T\boldsymbol{\beta} = \beta_1 - \beta_2$ is the difference in effects of the first and second predictors. We are sometimes interested in testing whether such a contrast is equal to zero, i.e., $H_0 : \boldsymbol{c}^T\boldsymbol{\beta} = 0$. While this hypothesis can involve two or more of the predictors, the parameter $\boldsymbol{c}^T\boldsymbol{\beta}$ is still one-dimensional, and therefore we can still apply a t-test. Going back to the distribution $\widehat{\boldsymbol{\beta}} \sim N(\boldsymbol{\beta}, \sigma^2(\boldsymbol{X}^T\boldsymbol{X})^{-1})$, we find that:

$$\boldsymbol{c}^T\widehat{\boldsymbol{\beta}} \sim N(\boldsymbol{c}^T\boldsymbol{\beta}, \sigma^2\boldsymbol{c}^T(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{c}). \tag{8.3}$$

Therefore, under the null hypothesis that $\boldsymbol{c}^T\boldsymbol{\beta} = 0$, we can derive that:

$$\frac{\boldsymbol{c}^T\widehat{\boldsymbol{\beta}}}{\widehat{\sigma}\sqrt{\boldsymbol{c}^T(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{c}}} \sim t_{n-p}, \tag{8.4}$$

giving us another t-test. Note that the t-tests described above can be recovered from this more general formulation by setting $\boldsymbol{c} = \boldsymbol{e}_j$, the indicator vector with the jth coordinate equal to 1 and all others equal to zero.

8.2 Testing a multi-dimensional parameter

See also Dunn and Smyth 2.10.1

8.2.1 F-test for a group of coefficients

Now we move on to the case of testing a multi-dimensional parameter: $H_0 : \boldsymbol{\beta}_S = \boldsymbol{0}$ for some $S \subseteq \{0, 1, \ldots, p-1\}$. In other words, we would like to test

$$H_0 : \boldsymbol{y} = \boldsymbol{X}_{*,\text{-}S}\boldsymbol{\beta}_{\text{-}S} + \boldsymbol{\epsilon} \quad \text{versus} \quad H_1 : \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}.$$

To test this hypothesis, let us fit least squares coefficients $\widehat{\boldsymbol{\beta}}_{\text{-}S}$ and $\widehat{\boldsymbol{\beta}}$ for the partial model as well as the full model. If the partial model fits well, then the residuals $\boldsymbol{y} - \boldsymbol{X}_{*,\text{-}S}\widehat{\boldsymbol{\beta}}_{\text{-}S}$ from this model will not be much larger than the residuals $\boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}$ from the full model. To quantify this intuition, let us recall our analysis of variance decomposition from Chapter 1:

$$\|\boldsymbol{y} - \boldsymbol{X}_{*,\text{-}S}\widehat{\boldsymbol{\beta}}_{\text{-}S}\|^2 = \|\boldsymbol{X}\widehat{\boldsymbol{\beta}} - \boldsymbol{X}_{*,\text{-}S}\widehat{\boldsymbol{\beta}}_{\text{-}S}\|^2 + \|\boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}\|^2.$$

Let us consider the ratio

$$\frac{\|\boldsymbol{y} - \boldsymbol{X}_{*,\text{-}S}\widehat{\boldsymbol{\beta}}_{\text{-}S}\|^2 - \|\boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}\|^2}{\|\boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}\|^2} = \frac{\|\boldsymbol{X}\widehat{\boldsymbol{\beta}} - \boldsymbol{X}_{*,\text{-}S}\widehat{\boldsymbol{\beta}}_{\text{-}S}\|^2}{\|\boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}\|^2}, \tag{8.5}$$

which is the relative increase in the residual sum of squares when going from the full model to the partial model. To interpret this ratio geometrically, let us first examine the quantity $\boldsymbol{X}\widehat{\boldsymbol{\beta}} - \boldsymbol{X}_{*,\text{-}S}\widehat{\boldsymbol{\beta}}_{\text{-}S}$

from the numerator. Letting $\boldsymbol{H}$ and $\boldsymbol{H}_{\text{-}S}$ be the projection matrices for the full and partial models, we have

$$\boldsymbol{X}\widehat{\boldsymbol{\beta}} - \boldsymbol{X}_{*,\text{-}S}\widehat{\boldsymbol{\beta}}_{\text{-}S} = (\boldsymbol{H} - \boldsymbol{H}_{\text{-}S})\boldsymbol{y}.$$

It turns out the the matrix $\boldsymbol{H} - \boldsymbol{H}_{\text{-}S}$ is a projection matrix:

Proposition 8.1. *The matrix $\boldsymbol{H} - \boldsymbol{H}_{\text{-}S}$ is a projection matrix onto the space $C(\boldsymbol{X}_{*S}^{\perp})$ spanned by the columns of $\boldsymbol{X}_{*S}$ adjusted for $\boldsymbol{X}_{*,\text{-}S}$.*

Figure 8.1 illustrates this relationship.

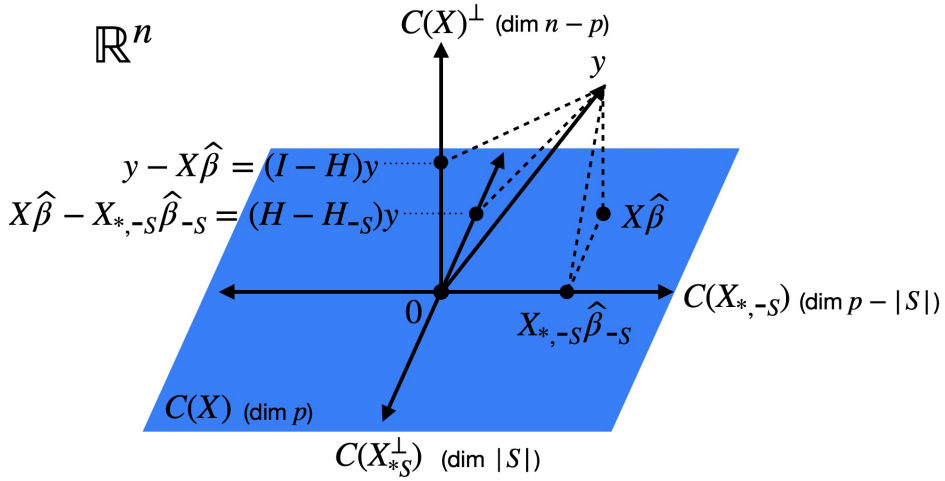


Figure 8.1: Geometry of the F-test. Orthogonality relationships stem from $C(\boldsymbol{X}_{*,\text{-}S}) \perp C(\boldsymbol{X}_{*S}^{\perp}) \perp C(\boldsymbol{X})^{\perp}$.

Proof. Let $\boldsymbol{v} \in C(\boldsymbol{X}_{*S}^{\perp})$. Because $\boldsymbol{v}$ is orthogonal to $C(\boldsymbol{X}_{*,\text{-}S})$ by construction, we have $(\boldsymbol{H} - \boldsymbol{H}_{\text{-}S})\boldsymbol{v} = \boldsymbol{H}\boldsymbol{v} - \boldsymbol{H}_{\text{-}S}\boldsymbol{v} = \boldsymbol{v} - \boldsymbol{0} = \boldsymbol{v}$. On the other hand, let $v \in C(\boldsymbol{X}_{*,\text{-}S})$. Then, we have $(\boldsymbol{H} - \boldsymbol{H}_{\text{-}S})\boldsymbol{v} = \boldsymbol{H}\boldsymbol{v} - \boldsymbol{H}_{\text{-}S}\boldsymbol{v} = \boldsymbol{v} - \boldsymbol{v} = \boldsymbol{0}$. Finally, let $v \in C(\boldsymbol{X})^{\perp}$. Then, we have $(\boldsymbol{H} - \boldsymbol{H}_{\text{-}S})\boldsymbol{v} = \boldsymbol{H}\boldsymbol{v} - \boldsymbol{H}_{\text{-}S}\boldsymbol{v} = \boldsymbol{0} - \boldsymbol{0} = \boldsymbol{0}$. From these three observations, it follows that $\boldsymbol{H} - \boldsymbol{H}_{\text{-}S}$ is a projection matrix onto $C(\boldsymbol{X}_{*S}^{\perp})$.

□

With this additional intuition, let us rewrite the ratio (8.5) as

$$\frac{\|\boldsymbol{X}\widehat{\boldsymbol{\beta}} - \boldsymbol{X}_{*,\text{-}S}\widehat{\boldsymbol{\beta}}_{\text{-}S}\|^2}{\|\boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}\|^2} = \frac{\|(\boldsymbol{H} - \boldsymbol{H}_{\text{-}S})\boldsymbol{y}\|^2}{\|(\boldsymbol{I} - \boldsymbol{H})\boldsymbol{y}\|^2},$$

revealing that the numerator and denominator are the squared norms of the projections of $\boldsymbol{y}$ onto $C(\boldsymbol{X}_{*S}^{\perp})$ and $C(\boldsymbol{X})^{\perp}$, respectively (Figure 8.1). The numerator is expected to be large if $\boldsymbol{\beta}_S \neq 0$, so $\boldsymbol{y}$ will have a large projection onto $C(\boldsymbol{X}_{*S}^{\perp})$. We can view the denominator as a normalization term.

Now, let us derive the distribution of this test statistic under the null hypothesis. If $\boldsymbol{\beta}_S = 0$, then we have $\boldsymbol{y} = \boldsymbol{X}_{*,\text{-}S}\boldsymbol{\beta}_{\text{-}S} + \boldsymbol{\epsilon}$, and

$$(\boldsymbol{H} - \boldsymbol{H}_{\text{-}S})\boldsymbol{X}_{*,\text{-}S}\boldsymbol{\beta}_{\text{-}S} = (\boldsymbol{I} - \boldsymbol{H})\boldsymbol{X}_{*,\text{-}S}\boldsymbol{\beta}_{\text{-}S} = 0$$

because $\boldsymbol{X}_{*,\text{-}S}\boldsymbol{\beta}_{\text{-}S} \in C(\boldsymbol{X}_{*,\text{-}S})$, and the latter space is orthogonal to both $C(\boldsymbol{X}^{\perp}_{*,S})$ and $C(\boldsymbol{X})^{\perp}$. It follows that

$$\frac{\|(\boldsymbol{H} - \boldsymbol{H}_{\text{-}S})\boldsymbol{y}\|^2}{\|(\boldsymbol{I} - \boldsymbol{H})\boldsymbol{y}\|^2} = \frac{\|(\boldsymbol{H} - \boldsymbol{H}_{\text{-}S})\boldsymbol{\epsilon}\|^2}{\|(\boldsymbol{I} - \boldsymbol{H})\boldsymbol{\epsilon}\|^2}.$$

Since the projection matrices in the numerator and denominator project onto orthogonal subspaces, we have $(\boldsymbol{H} - \boldsymbol{H}_{\text{-}S})\boldsymbol{\epsilon} \perp\!\!\!\perp (\boldsymbol{I} - \boldsymbol{H})\boldsymbol{\epsilon}$, with $\|(\boldsymbol{H} - \boldsymbol{H}_{\text{-}S})\boldsymbol{\epsilon}\|^2 \sim \sigma^2\chi^2_{|S|}$ and $\|(\boldsymbol{I} - \boldsymbol{H})\boldsymbol{\epsilon}\|^2 \sim \sigma^2\chi^2_{n-p}$. Renormalizing numerator and denominator to have expectation 1 under the null, we arrive at the F-statistic

$$F \equiv \frac{(\|\boldsymbol{y} - \boldsymbol{X}_{*,\text{-}S}\widehat{\boldsymbol{\beta}}_{\text{-}S}\|^2 - \|\boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}\|^2)/|S|}{\|\boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}\|^2/(n-p)}.$$

We have derived that under the null hypothesis,

$$F \sim \frac{\chi^2_{|S|}/|S|}{\chi^2_{n-p}/(n-p)}, \quad \text{with numerator and denominator independent.}$$

This distribution is called the F-distribution with $|S|$ and $n-p$ degrees of freedom, and is denoted $F_{|S|,n-p}$. Denoting by $F_{|S|,n-p}(1-\alpha)$ the $1-\alpha$ quantile of this distribution, we arrive at the F-test

$$\phi_F(\boldsymbol{X}, \boldsymbol{y}) \equiv 1(F > F_{|S|,n-p}(1-\alpha)).$$

Note that the F-test searches for deviations of $\boldsymbol{\beta}_S$ from zero in all directions, and does not have one-sided variants like the t-test.

8.2.2 Example: Testing for any significant coefficients except the intercept

Suppose $\boldsymbol{x}_{*,0} = \mathbf{1}_n$ is an intercept term. Then, consider the null hypothesis $H_0 : \beta_1 = \cdots = \beta_{p-1} = 0$. In other words, the null hypothesis is the intercept-only model, and the alternative hypothesis is the regression model with an intercept and $p-1$ additional predictors. In this case, $S = \{1, \ldots, p-1\}$ and $\text{-}S = \{0\}$. The corresponding F statistic is

$$F \equiv \frac{(\|\boldsymbol{y} - \bar{y}\mathbf{1}\|^2 - \|\boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}\|^2)/(p-1)}{\|\boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}\|^2/(n-p)} = \frac{\|\boldsymbol{X}\widehat{\boldsymbol{\beta}} - \bar{y}\mathbf{1}\|^2/(p-1)}{\|\boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}\|^2/(n-p)}$$

with null distribution $F_{p-1,n-p}$.

8.2.3 Example: Testing for equality of group means in C-groups model

As a further special case, consider the C-groups model from Chapter 1. Recall the ANOVA decomposition

$$\sum_{i=1}^{n}(y_i - \bar{y})^2 = \sum_{i=1}^{n}(\bar{y}_{c(i)} - \bar{y})^2 + \sum_{i=1}^{n}(y_i - \bar{y}_{c(i)})^2 = \text{SSB} + \text{SSW}.$$

The F-statistic in this case becomes

$$F = \frac{\sum_{i=1}^{n}(\bar{y}_{c(i)} - \bar{y})^2/(C-1)}{\sum_{i=1}^{n}(y_i - \bar{y}_{c(i)})^2/(n-C)} = \frac{\text{SSB}/(C-1)}{\text{SSW}/(n-C)},$$

with null distribution $F_{C-1,n-C}$.

Chapter 9

Power

See also Agresti 3.2.5

So far we've been focused on finding the null distributions of various test statistics in order to construct tests with Type-I error control. Now let's shift our attention to examining the power of these tests.

9.1 The power of a t-test

9.1.1 Power formula

Consider the t-test of the null hypothesis $H_0 : \beta_j = 0$. Suppose that, in reality, $\beta_j \neq 0$. What is the probability the t-test will reject the null hypothesis? To answer this question, recall that $\widehat{\beta}_j \sim N(\beta_j, \sigma^2/s_j^2)$. Therefore,

$$t = \frac{\widehat{\beta}_j}{\mathrm{SE}(\widehat{\beta}_j)} = \frac{\beta_j}{\mathrm{SE}(\widehat{\beta}_j)} + \frac{\widehat{\beta}_j - \beta_j}{\mathrm{SE}(\widehat{\beta}_j)} \dot{\sim} N\left(\frac{\beta_j s_j}{\sigma}, 1\right) \tag{9.1}$$

Here we have made the approximation $\mathrm{SE}(\widehat{\beta}_j) \approx \frac{\sigma}{s_j}$, which is pretty good when $n - p$ is large. Therefore, the power of the two-sided t-test is

$$\mathbb{E}[\phi_t] = \mathbb{P}[\phi_t = 1] \approx \mathbb{P}[|t| > z_{1-\alpha/2}] \approx \mathbb{P}\left[\left|N\left(\frac{\beta_j s_j}{\sigma}, 1\right)\right| > z_{1-\alpha/2}\right]$$

Therefore, the quantity $\frac{\beta_j s_j}{\sigma}$ determines the power of the t-test. To understand s_j a little better, let's assume that the rows $\boldsymbol{x}_{i*}$ of the model matrix are drawn i.i.d. from some distribution $(x_0, \ldots, x_{p-1})$. Then we have roughly

$$\boldsymbol{x}_{*j}^{\perp} \approx \boldsymbol{x}_{*j} - \mathbb{E}[\boldsymbol{x}_{*j} | \boldsymbol{X}_{*,-j}],$$

so $x_{ij}^{\perp} \approx x_{ij} - \mathbb{E}[x_{ij} | \boldsymbol{x}_{i,-j}]$. Hence,

$$s_j^2 \equiv \|\boldsymbol{x}_{*j}^{\perp}\|^2 \approx n\mathbb{E}[(x_j - \mathbb{E}[x_j | \boldsymbol{x}_{\text{-}j}])^2] = n\mathbb{E}[\mathrm{Var}[x_j | \boldsymbol{x}_{\text{-}j}]].$$

Hence, we can rewrite the alternative distribution (9.1) as

$$t \dot{\sim} N\left(\frac{\beta_j \cdot \sqrt{n} \cdot \sqrt{\mathbb{E}[\text{Var}[x_j|\boldsymbol{x}_{\text{-}j}]]}}{\sigma}, 1\right) \tag{9.2}$$

We can see clearly now how the power of the t-test varies with the effect size β_j, the sample size n, the degree of collinearity $\mathbb{E}[\text{Var}[x_j|\boldsymbol{x}_{\text{-}j}]]$, and the noise standard deviation σ.

9.1.2 Power of the t-test when predictors are added to the model

As we know, the outcome of a regression is a function of the predictors that are used. What happens to the t-test p-value for $H_0 : \beta_j = 0$ when a predictor is added to the model? To keep things simple, let's consider the

$$\text{true underlying model: } y = \beta_0 x_0 + \beta_1 x_1 + \epsilon.$$

Let's consider the power of testing $H_0 : \beta_0 = 0$ in the regression models

$$\text{model 0: } y = \beta_0 x_0 + \epsilon \quad \text{versus} \quad \text{model 1: } y = \beta_0 x_0 + \beta_1 x_1 + \epsilon.$$

There are four cases based on $\text{cor}[\boldsymbol{x}_{*0}, \boldsymbol{x}_{*1}]$ and the value of β_1 in the true model:

1. $\text{cor}[\boldsymbol{x}_{*0}, \boldsymbol{x}_{*1}] \neq 0$ and $\beta_1 \neq 0$. In this case, in model 0 we have omitted an important variable that is correlated with $\boldsymbol{x}_{*0}$. Therefore, the meaning of β_0 differs between model 0 and model 1, so it may not be meaningful to compare the p-values arising from these two models.
2. $\text{cor}[\boldsymbol{x}_{*0}, \boldsymbol{x}_{*1}] \neq 0$ and $\beta_1 = 0$. In this case, we are adding a null predictor that is correlated with x_{*0}. Recall that the power of the t-test hinges on the quantity $\frac{\beta_j \cdot \sqrt{n} \cdot \sqrt{\mathbb{E}[\text{Var}[x_j|\boldsymbol{x}_{\text{-}j}]]}}{\sigma}$. Adding the predictor x_1 has the effect of reducing the conditional predictor variance $\mathbb{E}[\text{Var}[x_j|\boldsymbol{x}_{\text{-}j}]]$, therefore reducing the power. This is a case of *predictor competition*.
3. $\text{cor}[\boldsymbol{x}_{*0}, \boldsymbol{x}_{*1}] = 0$ and $\beta_1 \neq 0$. In this case, we are adding a non-null predictor that is orthogonal to $\boldsymbol{x}_{*0}$. While the conditional predictor variance $\mathbb{E}[\text{Var}[x_j|\boldsymbol{x}_{\text{-}j}]]$ remains the same due to orthogonality, the residual variance σ^2 is reduced when going from model 0 to model 1.[1] Therefore, in this case adding x_1 to the model increases the power for testing $H_0 : \beta_0 = 0$. This is a case of *predictor collaboration*.
4. $\text{cor}[\boldsymbol{x}_{*0}, \boldsymbol{x}_{*1}] = 0$ and $\beta_1 = 0$. In this case, we are adding an orthogonal null variable, which does not change the conditional predictor variance or the residual variance, and therefore keeps the power of the test the same.

In conclusion, adding a predictor can either increase or decrease the power of a t-test.

9.1.3 Application: Adjusting for covariates in randomized experiments.

Case 3 above, i.e., $\text{cor}[\boldsymbol{x}_{*0}, \boldsymbol{x}_{*1}] = 0$ and $\beta_1 \neq 0$, arises in the context of randomized experiments in causal inference. In this case, y represents the outcome, x_0 represents the treatment, and x_1 represents a covariate. Because the treatment is randomized, there is no correlation between x_0

[1] If β_1 is small enough, then the unbiased estimate of the residual variance may actually increase due to a reduction in the residual degrees of freedom in the denominator.

and x_1. Therefore, it is not necessary to adjust for x_1 in order to get an unbiased estimate of the average treatment effect. However, it is known that adjusting for covariates can lead to more *precise* estimates of the treatment effect due to the phenomenon discussed in case 3 above. This point is also related to the discussion in Chapter 1 about the fact that if x_0 and x_1 are orthogonal, then the least squares coefficient $\widehat{\beta}_0$ is the same regardless of whether x_1 is included in the model. As we see here, either including x_1 in the model or adjusting y for x_1 is necessary to get better power.

9.2 The power of an F-test

Now let's turn our attention to computing the power of the F-test. We have

$$\begin{aligned} F &= \frac{\|\boldsymbol{X}\widehat{\boldsymbol{\beta}} - \boldsymbol{X}_{*,\text{-}S}\widehat{\boldsymbol{\beta}}_{-S}\|^2/|S|}{\|\boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}\|^2/|n-p|} \\ &= \frac{\|(\boldsymbol{H} - \boldsymbol{H}_{\text{-}S})\boldsymbol{y}\|^2/|S|}{\|(\boldsymbol{I} - \boldsymbol{H})\boldsymbol{y}\|^2/|n-p|} \\ &\approx \frac{\|(\boldsymbol{H} - \boldsymbol{H}_{\text{-}S})\boldsymbol{y}\|^2/|S|}{\sigma^2}. \end{aligned}$$

To calculate the distribution of the numerator, we need to introduce the notion of a *non-central chi-squared random variable.*

Definition 9.1. For some vector $\boldsymbol{\mu} \in \mathbb{R}^d$, suppose $\boldsymbol{z} \sim N(\boldsymbol{\mu}, \boldsymbol{I}_d)$. Then, we define the distribution of $\|\boldsymbol{z}\|^2$ as the noncentral chi-square random variable with d degrees of freedom and noncentrality parameter $\|\boldsymbol{\mu}\|^2$ and denote this distribution by $\chi^2_d(\|\boldsymbol{\mu}\|^2)$.

The following proposition states two useful facts about noncentral chi-square distributions.

Proposition 9.1. *The following two relations hold:*

1. *The mean of a $\chi^2_d(\|\boldsymbol{\mu}\|^2)$ random variable is $d + \|\boldsymbol{\mu}\|^2$.*
2. *If $\boldsymbol{P}$ is a projection matrix and $\boldsymbol{y} = \boldsymbol{\mu} + \boldsymbol{\epsilon}$, then $\frac{1}{\sigma^2}\|\boldsymbol{P}\boldsymbol{y}\|^2 \sim \chi^2_{tr(\boldsymbol{P})}\left(\frac{1}{\sigma^2}\|\boldsymbol{P}\boldsymbol{\mu}\|^2\right)$.*

It therefore follows that

$$\begin{aligned} F &\approx \frac{\|(\boldsymbol{H} - \boldsymbol{H}_{\text{-}S})\boldsymbol{y}\|^2/|S|}{\sigma^2} \\ &\sim \frac{1}{|S|}\chi^2_{|S|}\left(\|(\boldsymbol{H} - \boldsymbol{H}_{\text{-}S})\boldsymbol{X}\boldsymbol{\beta}\|^2\right) \\ &= \frac{1}{|S|}\chi^2_{|S|}\left(\frac{1}{\sigma^2}\|\boldsymbol{X}^{\perp}_{*,S}\boldsymbol{\beta}_S\|^2\right). \end{aligned}$$

Assuming as before that the rows of $\boldsymbol{X}$ are samples from a joint distribution, we can write

$$\|\boldsymbol{X}^{\perp}_{*,S}\boldsymbol{\beta}_S\|^2 \approx n\boldsymbol{\beta}_S^T\mathbb{E}[\mathrm{Var}[\boldsymbol{x}_S|\boldsymbol{x}_{\text{-}S}]]\boldsymbol{\beta}_S.$$

Therefore,

$$F \dot{\sim} \frac{1}{|S|}\chi^2_{|S|}\left(\frac{n\beta_S^T \mathbb{E}[\mathrm{Var}[\boldsymbol{x}_S|\boldsymbol{x}_{\text{-}S}]]\boldsymbol{\beta}_S}{\sigma^2}\right)$$

which is similar in spirit to equation (9.2). To get a better sense of what this relationship implies for the power of the F-test, we find from the first part of Proposition 9.1 that, under the alternative,

$$\begin{aligned}\mathbb{E}[F] &\approx \mathbb{E}\left[\frac{1}{|S|}\chi^2_{|S|}\left(\frac{n\beta_S^T \mathbb{E}[\mathrm{Var}[\boldsymbol{x}_S|\boldsymbol{x}_{\text{-}S}]]\boldsymbol{\beta}_S}{\sigma^2}\right)\right] \\ &= 1 + \frac{n\beta_S^T \mathbb{E}[\mathrm{Var}[\boldsymbol{x}_S|\boldsymbol{x}_{\text{-}S}]]\boldsymbol{\beta}_S}{|S| \cdot \sigma^2}.\end{aligned}$$

By contrast, under the null, the mean of the F-statistic is 1. The $|S|$ term in the denominator above suggests that testing larger sets of variables explaining the same amount of variation in $\boldsymbol{y}$ will hurt power. The test must accommodate for the fact that larger sets of variables will explain more of the variability in y even under the null hypothesis.

Chapter 10

Confidence intervals

See also Agresti 3.3, Dunn and Smyth 2.8.4-2.8.5

In addition to hypothesis testing, we often want to construct confidence intervals for various quantities. As with hypotheses testing, we will split the target quantities into two categories: univariate and multivariate.

10.1 Confidence intervals for univariate quantities

10.1.1 Confidence interval for a coefficient

Under $H_0 : \beta_j = 0$, we showed that $\frac{\widehat{\beta}_j}{\widehat{\sigma}/s_j} \sim t_{n-p}$. The same argument shows that for arbitrary β_j, we have

$$\frac{\widehat{\beta}_j - \beta_j}{\widehat{\sigma}/s_j} \sim t_{n-p}.$$

We can use this relationship to construct a confidence interval for β_j as follows:

$$\begin{aligned}
&1-\alpha \\
&\quad = \mathbb{P}[|t_{n-p}| \leq t_{n-p}(1-\alpha/2)] \\
&\quad = \mathbb{P}\left[\left|\frac{\widehat{\beta}_j - \beta_j}{\widehat{\sigma}/s_j}\right| \leq t_{n-p}(1-\alpha/2)\right] \\
&\quad = \mathbb{P}\left[\beta_j \in \left[\widehat{\beta}_j - \frac{\widehat{\sigma}}{s_j} t_{n-p}(1-\alpha/2), \widehat{\beta}_j + \frac{\widehat{\sigma}}{s_j} t_{n-p}(1-\alpha/2)\right]\right] \\
&\quad \equiv \mathbb{P}\left[\beta_j \in \left[\widehat{\beta}_j - \mathrm{SE}(\widehat{\beta}_j) t_{n-p}(1-\alpha/2), \widehat{\beta}_j + \mathrm{SE}(\widehat{\beta}_j) t_{n-p}(1-\alpha/2)\right]\right] \\
&\quad \equiv \mathbb{P}[\beta_j \in \mathrm{CI}(\beta_j)].
\end{aligned} \tag{10.1}$$

The confidence interval $\mathrm{CI}(\beta_j)$ defined above therefore has $1-\alpha$ coverage. Because of the duality between confidence intervals and hypothesis tests, the factors contributing to powerful tests (Chapter 9) also lead to shorter confidence intervals.

10.1.2 Confidence interval for $\mathbb{E}[y|\tilde{\boldsymbol{x}}]$

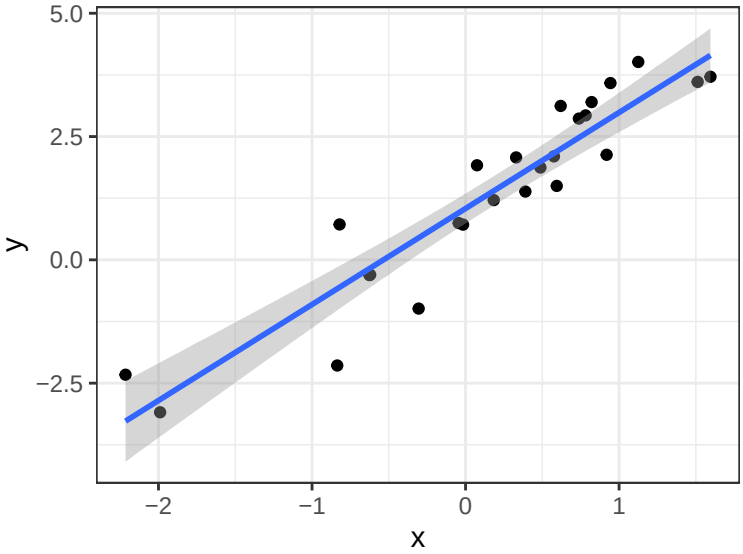
Suppose now that we have a new predictor vector $\tilde{\boldsymbol{x}} \in \mathbb{R}^p$. The mean of the response for this predictor vector is $\mathbb{E}[y|\tilde{\boldsymbol{x}}] = \tilde{\boldsymbol{x}}^T\boldsymbol{\beta}$. Plugging in $\tilde{\boldsymbol{x}}$ for $\boldsymbol{c}$ in the relation (8.3), we obtain

$$\frac{\tilde{\boldsymbol{x}}^T\widehat{\boldsymbol{\beta}} - \tilde{\boldsymbol{x}}^T\boldsymbol{\beta}}{\widehat{\sigma}\sqrt{\tilde{\boldsymbol{x}}^T(\boldsymbol{X}^T\boldsymbol{X})^{-1}\tilde{\boldsymbol{x}}}} \sim t_{n-p}.$$

From this, we can derive that

$$\begin{aligned} \mathrm{CI}(\tilde{\boldsymbol{x}}^T\boldsymbol{\beta}) &\equiv \tilde{\boldsymbol{x}}^T\widehat{\boldsymbol{\beta}} \pm \mathrm{SE}(\tilde{\boldsymbol{x}}^T\widehat{\boldsymbol{\beta}}) \cdot t_{n-p}(1-\alpha/2) \\ &\equiv \tilde{\boldsymbol{x}}^T\widehat{\boldsymbol{\beta}} \pm \widehat{\sigma}\sqrt{\tilde{\boldsymbol{x}}^T(\boldsymbol{X}^T\boldsymbol{X})^{-1}\tilde{\boldsymbol{x}}} \cdot t_{n-p}(1-\alpha/2) \end{aligned} \tag{10.2}$$

is a $1-\alpha$ confidence interval for $\tilde{\boldsymbol{x}}^T\boldsymbol{\beta}$. Consider the special case of the simple linear regression $y = \beta_0 + \beta_1 x + \epsilon$. Then, confidence intervals for $\beta_0 + \beta_1\tilde{x}$ for each $\tilde{x} \in \mathbb{R}$ sweep out *confidence bands* for the regression line.



We see that the width of the confidence band appears to be the smallest around the center of the data. To verify this, let $\bar{x}$ be the mean of the observed x values. Centering x leads to the following reparameterized regression:

$$y = \beta_0' + \beta_1(x - \bar{x}) + \epsilon.$$

The width of the confidence interval (10.2) is proportional to the square root of $\tilde{\boldsymbol{x}}^T(\boldsymbol{X}^T\boldsymbol{X})^{-1}\tilde{\boldsymbol{x}}$. Applying this to the centered vector $\tilde{\boldsymbol{x}} = (1, \tilde{x} - \bar{x})^T$ and the centered matrix $\boldsymbol{X} = (\boldsymbol{1}, \boldsymbol{x} - \bar{x}\boldsymbol{1})$, we get

$$\begin{aligned} \tilde{\boldsymbol{x}}^T(\boldsymbol{X}^T\boldsymbol{X})^{-1}\tilde{\boldsymbol{x}} &= (1, \tilde{x} - \bar{x}) \begin{pmatrix} n & 0 \\ 0 & \sum_i (x_i - \bar{x})^2 \end{pmatrix}^{-1} \begin{pmatrix} 1 \\ \tilde{x} - \bar{x} \end{pmatrix} \\ &= \frac{1}{n} + \frac{(\tilde{x} - \bar{x})^2}{\sum_i (x_i - \bar{x})^2}. \end{aligned}$$

We see that this quantity is minimized at $\tilde{x} = \bar{x}$, as expected.

10.1.3 Prediction interval for $y|\tilde{\boldsymbol{x}}$

Instead of creating a confidence interval for a point on the regression line, we may want to create a confidence interval for a new draw $\tilde{y}$ of y for $\boldsymbol{x} = \tilde{\boldsymbol{x}}$, i.e., a *prediction interval*. Note that

$$\begin{aligned}\tilde{y} - \tilde{\boldsymbol{x}}^T\widehat{\beta} &= \tilde{\boldsymbol{x}}^T\beta + \tilde{\epsilon} - \tilde{\boldsymbol{x}}^T\widehat{\beta} \\ &= \tilde{\epsilon} + \tilde{\boldsymbol{x}}^T(\beta - \widehat{\beta}) \\ &\sim N(0, \sigma^2 + \sigma^2\tilde{\boldsymbol{x}}^T(\boldsymbol{X}^T\boldsymbol{X})^{-1}\tilde{\boldsymbol{x}}).\end{aligned}$$

Therefore, we have

$$\frac{\tilde{y} - \tilde{\boldsymbol{x}}^T\widehat{\beta}}{\widehat{\sigma}\sqrt{1 + \tilde{\boldsymbol{x}}^T(\boldsymbol{X}^T\boldsymbol{X})^{-1}\tilde{\boldsymbol{x}}}} \sim t_{n-p},$$

which leads to the $1 - \alpha$ prediction interval

$$\begin{aligned}&\tilde{\boldsymbol{x}}^T\widehat{\boldsymbol{\beta}} \pm \widehat{\sigma}\sqrt{1 + \tilde{\boldsymbol{x}}^T(\boldsymbol{X}^T\boldsymbol{X})^{-1}\tilde{\boldsymbol{x}}} \cdot t_{n-p}(1 - \alpha/2) \\ &\equiv \tilde{\boldsymbol{x}}^T\widehat{\boldsymbol{\beta}} \pm \mathrm{SE}(\tilde{\boldsymbol{x}}^T\widehat{\boldsymbol{\beta}}) \cdot t_{n-p}(1 - \alpha/2).\end{aligned} \tag{10.3}$$

Remark: Prediction with confidence in machine learning.

The entire field of supervised machine learning is focused on accurately predicting $\tilde{y}$ from $\tilde{\boldsymbol{x}}$, usually using nonlinear functions $\widehat{f}(\tilde{\boldsymbol{x}})$. In addition to providing a guess for $\tilde{y}$, it is often useful to quantify the uncertainty in this guess. In other words, it is useful to come up with a prediction interval (or prediction region) $\mathrm{PI}(\tilde{y})$ such that

$$\mathbb{P}[\tilde{y} \in \mathrm{PI}(\tilde{y}) \mid \tilde{\boldsymbol{x}}] \geq 1 - \alpha. \tag{10.4}$$

For example, in safety-critical applications of machine learning like self-driving cars, it is essential to have confidence in predictions. Unfortunately, beyond the realm of linear regression, it is hard to come up with intervals satisfying (10.4) for each point $\tilde{\boldsymbol{x}}$. However, the emerging field of *conformal inference* provides guarantees on average over possible values of $\boldsymbol{x}$:

$$\mathbb{P}[y \in \mathrm{PI}(y)] = \mathbb{E}[\mathbb{P}[y \in \mathrm{PI}(y) \mid \boldsymbol{x}]] \geq 1 - \alpha. \tag{10.5}$$

Remarkably, these guarantees place no assumption on the machine learning method used and require only that the data points on which $\widehat{f}$ is trained are exchangeable (an even weaker condition than i.i.d.). While the unconditional guarantee (10.5) is weaker than the conditional one (10.4), it can be obtained for modern machine learning and deep learning models.

10.2 Confidence regions and simultaneous intervals

10.2.1 Confidence regions

A multivariate generalization of a confidence interval is a *confidence region*. We will discuss the construction of a confidence region for $\boldsymbol{\beta}$ in the linear regression model. A $1-\alpha$ confidence region for $\boldsymbol{\beta}$ is a set $\mathrm{CR}(\boldsymbol{\beta}) \subseteq \mathbb{R}^p$ such that

$$\mathbb{P}[\boldsymbol{\beta} \in \mathrm{CR}(\boldsymbol{\beta})] \geq 1-\alpha.$$

To construct such a region, note first that

$$\frac{\frac{1}{p}\|\boldsymbol{X}\widehat{\boldsymbol{\beta}} - \boldsymbol{X}\boldsymbol{\beta}\|^2}{\widehat{\sigma}^2} \sim F_{p,n-p}.$$

Hence, we have

$$\mathbb{P}[\|\boldsymbol{X}\widehat{\boldsymbol{\beta}} - \boldsymbol{X}\boldsymbol{\beta}\|^2 \leq p\widehat{\sigma}^2 F_{p,n-p}(1-\alpha)] \geq 1-\alpha.$$

Hence, the region

$$\mathrm{CR}(\boldsymbol{\beta}) \equiv \{\boldsymbol{\beta} : (\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta})^T \boldsymbol{X}^T \boldsymbol{X} (\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}) \leq p\widehat{\sigma}^2 F_{p,n-p}(1-\alpha)\} \subseteq \mathbb{R}^p$$

is a $1-\alpha$ confidence region for the vector $\boldsymbol{\beta}$. It's easy to see that $\mathrm{CR}(\boldsymbol{\beta})$ is an ellipse centered at $\widehat{\boldsymbol{\beta}}$.

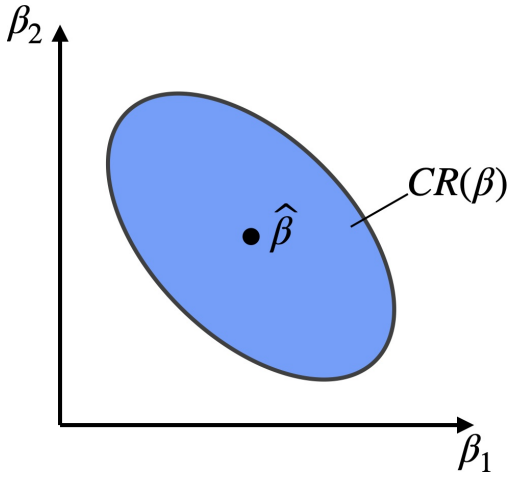


Figure 10.1: Confidence region for $\boldsymbol{\beta}$.

10.2.2 Simultaneous intervals

As a byproduct of confidence regions for the multivariate $\boldsymbol{\beta}$, we can construct *simultaneous intervals* for univariate quantities. To motivate the definition of simultaneous intervals, note that the intervals in Section 10.1 have *pointwise coverage*. For example, we have

$$\mathbb{P}[\beta_j \in \mathrm{CI}(\beta_j)] \geq 1-\alpha \quad \text{for each } j.$$

or

$$\mathbb{P}[\tilde{\boldsymbol{x}}^T\boldsymbol{\beta} \in \mathrm{CI}(\tilde{\boldsymbol{x}}^T\boldsymbol{\beta})] \geq 1-\alpha \quad \text{for each } \tilde{\boldsymbol{x}}.$$

Sometimes a stronger *simultaneous coverage* guarantee is desired, e.g.,

$$\mathbb{P}[\beta_j \in \mathrm{CI}^{\mathrm{sim}}(\beta_j) \text{ for each } j] \geq 1-\alpha \tag{10.6}$$

or

$$\mathbb{P}[\tilde{\boldsymbol{x}}^T\boldsymbol{\beta} \in \mathrm{CI}^{\mathrm{sim}}(\tilde{\boldsymbol{x}}^T\boldsymbol{\beta}) \text{ for each } \tilde{\boldsymbol{x}}] \geq 1-\alpha. \tag{10.7}$$

To obtain such simultaneous confidence intervals, we can leverage the fact that the confidence region $\mathrm{CR}(\boldsymbol{\beta})$ is for the entire vector $\boldsymbol{\beta}$. We can therefore define

$$\mathrm{CI}^{\mathrm{sim}}(\beta_j) \equiv \{\beta_j : \boldsymbol{\beta} \in \mathrm{CR}(\boldsymbol{\beta})\}.$$

Then, these confidence intervals will satisfy the simultaneous coverage property (10.6). We will obtain a more explicit expression for $\mathrm{CI}^{\mathrm{sim}}(\beta_j)$ shortly.

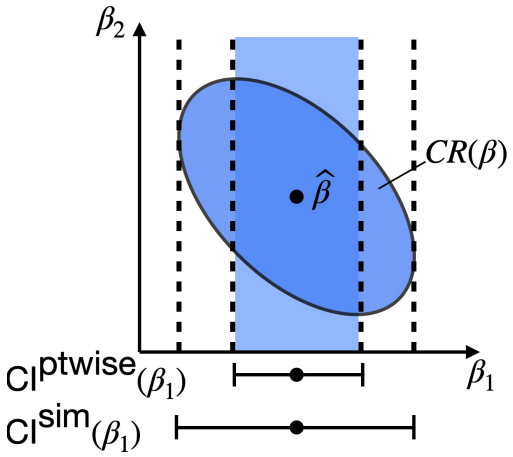


Figure 10.2: Confidence region and simultaneous and pointwise confidence intervals.

Similarly, we may define the simultaneous confidence regions

$$\mathrm{CI}^{\mathrm{sim}}(\tilde{\boldsymbol{x}}^T\boldsymbol{\beta}) \equiv \{\tilde{\boldsymbol{x}}^T\boldsymbol{\beta} : \boldsymbol{\beta} \in \mathrm{CR}(\boldsymbol{\beta})\}.$$

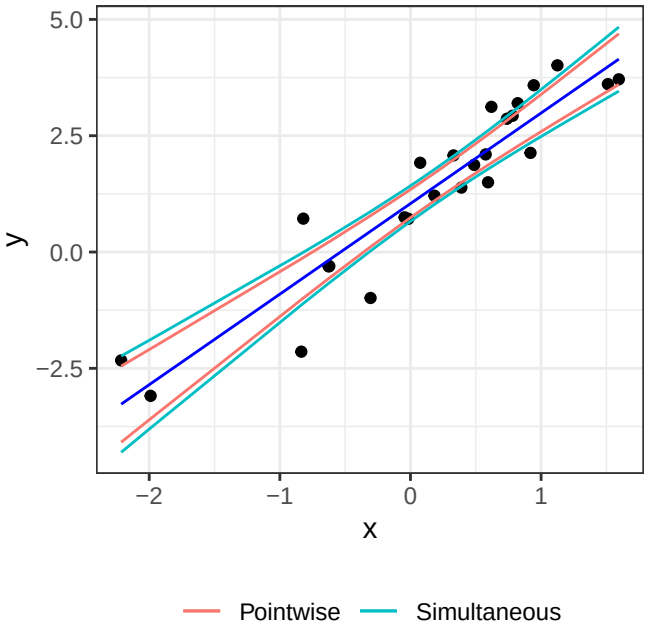
Let us find a more explicit expression for the latter interval. For notational ease, let us define $\boldsymbol{\Sigma} \equiv \boldsymbol{X}^T\boldsymbol{X}$. Then, note that if $\boldsymbol{\beta} \in \mathrm{CR}(\boldsymbol{\beta})$, then by the Cauchy-Schwarz inequality we have

$$\begin{aligned}(\tilde{\boldsymbol{x}}^T\widehat{\boldsymbol{\beta}} - \tilde{\boldsymbol{x}}^T\boldsymbol{\beta})^2 &= \|\tilde{\boldsymbol{x}}^T(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta})\|^2 \\ &= \|(\boldsymbol{\Sigma}^{-1/2}\tilde{\boldsymbol{x}})^T\boldsymbol{\Sigma}^{1/2}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta})\|^2 \\ &\leq \|(\boldsymbol{\Sigma}^{-1/2}\tilde{\boldsymbol{x}})\|^2\|\boldsymbol{\Sigma}^{1/2}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta})\|^2 \\ &\leq \tilde{\boldsymbol{x}}^T\boldsymbol{\Sigma}^{-1}\tilde{\boldsymbol{x}}p\widehat{\sigma}^2F_{p,n-p}(1-\alpha),\end{aligned}$$

i.e.,

$$\begin{aligned}\tilde{\boldsymbol{x}}^T\boldsymbol{\beta} &\in \tilde{\boldsymbol{x}}^T\widehat{\boldsymbol{\beta}} \pm \widehat{\sigma}\sqrt{\tilde{\boldsymbol{x}}^T(\boldsymbol{X}^T\boldsymbol{X})^{-1}\tilde{\boldsymbol{x}}}\sqrt{pF_{p,n-p}(1-\alpha)} \\ &\equiv \tilde{\boldsymbol{x}}^T\widehat{\boldsymbol{\beta}} \pm \mathrm{SE}(\tilde{\boldsymbol{x}}^T\widehat{\boldsymbol{\beta}}) \cdot \sqrt{pF_{p,n-p}(1-\alpha)}.\end{aligned} \tag{10.8}$$

Defining the above interval as $\mathrm{CI}^{\mathrm{sim}}(\tilde{\boldsymbol{x}}^T\boldsymbol{\beta})$ gives us the simultaneous coverage property (10.7). These simultaneous intervals are called *Working-Hotelling intervals*. Comparing to equation (10.3), we see that the simultaneous interval is the pointwise interval expanded by a factor of $\sqrt{pF_{p,n-p}(1-\alpha)}/t_{n-p}(1-\alpha/2)$. In the case of simple linear regression, we can obtain simultaneous confidence bands (Working-Hotelling bands) for the regression line.



Specializing to the case $\tilde{\boldsymbol{x}} \equiv \boldsymbol{e_j}$, we get an expression for the simultaneous intervals for each coordinate:

$$\begin{aligned}\mathrm{CI}^{\mathrm{sim}}(\beta_j) &\equiv \widehat{\beta}_j \pm \widehat{\sigma}\sqrt{(\boldsymbol{X}^T\boldsymbol{X})^{-1}_{jj}}\sqrt{pF_{p,n-p}(1-\alpha)} \\ &\equiv \mathrm{SE}(\widehat{\beta}_j)\sqrt{pF_{p,n-p}(1-\alpha)},\end{aligned} \tag{10.9}$$

which again is the pointwise interval (10.1) expanded by a factor of $\sqrt{pF_{p,n-p}(1-\alpha)}/t_{n-p}(1-\alpha/2)$.

Chapter 11

Practical considerations

11.1 Practical versus statistical significance

You can have a statistically significant effect that is not practically significant. The hypothesis testing framework is most useful in the case when the signal-to-noise ratio is relatively small. Otherwise, constructing a confidence interval for the effect size is a more meaningful approach.

11.2 Correlation versus causation, and Simpson's paradox

Causation can be elusive for several reasons. One is reverse causation, where it is not clear whether X causes Y or Y causes X. Another is confounding, where there is a third variable Z that causes both X and Y. For the latter reason, linear regression coefficients can be sensitive to the choice of other predictors to include and can be misleading if you omit important variables from the regression. A special and sometimes overlooked case of this is *Simpson's paradox*, where an important discrete variable is omitted. Consider the example in Figure 11.1. Sometimes this discrete variable may seem benign, such as the year in which the data was collected. Such variables might or might not be measured.

11.3 Dealing with correlated predictors

It depends on the goal. If we're trying to tease apart effects of correlated predictors, then we have no choice but to proceed as usual despite lower power. Otherwise, we can test predictors in groups via the F-test to get higher power at the cost of lower "resolution." Sometimes, it is recommended to simply remove predictors that are correlated with other predictors. This practice, however, is somewhat arbitrary and not recommended.

11.4 Model selection

We need to ask ourselves: Why do we want to do model selection? It can either be for prediction purposes or for inferential purposes. If it is for prediction purposes, then we can apply cross-validation to select a model and we don't need to think very hard about statistical significance. If it is for inference, then we need to be more careful. There are various classical model selection criteria (e.g., AIC, BIC), but it is not entirely clear what statistical guarantee we are getting for the

Kidney stone treatment [edit]

Another example comes from a real-life medical study[15] comparing the success rates of two treatments for kidney stones.[16] The table below shows the success rates and numbers of treatments for treatments involving both small and large kidney stones, where Treatment A includes open surgical procedures and Treatment B includes closed surgical procedures. The numbers in parentheses indicate the number of success cases over the total size of the group.

Stone size \ Treatment	Treatment A	Treatment B
Small stones	*Group 1* **93% (81/87)**	*Group 2* 87% (234/270)
Large stones	*Group 3* **73% (192/263)**	*Group 4* 69% (55/80)
Both	78% (273/350)	**83% (289/350)**

The paradoxical conclusion is that treatment A is more effective when used on small stones, and also when used on large stones, yet treatment B appears to be more effective when considering both sizes at the same time. In this example, the "lurking" variable (or confounding variable) causing the paradox is the size of the stones, which was not previously known to researchers to be important until its effects were included.

Which treatment is considered better is determined by which success ratio (successes/total) is larger. The reversal of the inequality between the two ratios when considering the combined data, which creates Simpson's paradox, happens because two effects occur together:

1. The sizes of the groups, which are combined when the lurking variable is ignored, are very different. Doctors tend to give cases with large stones the better treatment A, and the cases with small stones the inferior treatment B. Therefore, the totals are dominated by groups 3 and 2, and not by the two much smaller groups 1 and 4.
2. The lurking variable, stone size, has a large effect on the ratios; i.e., the success rate is more strongly influenced by the severity of the case than by the choice of treatment. Therefore, the group of patients with large stones using treatment A (group 3) does worse than the group with small stones, even if the latter used the inferior treatment B (group 2).

Based on these effects, the paradoxical result is seen to arise because the effect of the size of the stones overwhelms the benefits of the better treatment (A). In short, the less effective treatment B appeared to be more effective because it was applied more frequently to the small stones cases, which were easier to treat.[16]

Figure 11.1: An example of Simpson's paradox (source: Wikipedia).

resulting models. A simpler approach is to apply a t-test for each variable in the model, apply a multiple testing correction to the resulting p-values, and report the set of significant variables and the associated guarantee. Re-fitting the linear regression after model selection leads us into some dicey inferential territory due to selection bias. This is the subject of ongoing research, and the jury is still out on the best way of doing this.

Chapter 12

R demo

See also Agresti 3.4.1, 3.4.3, Dunn and Smyth 2.6, 2.14

Let's put into practice what we've learned in this chapter by analyzing data about house prices.

```
library(tidyverse)
library(GGally)

houses_data <- read_tsv("data/Houses.dat")
houses_data
```

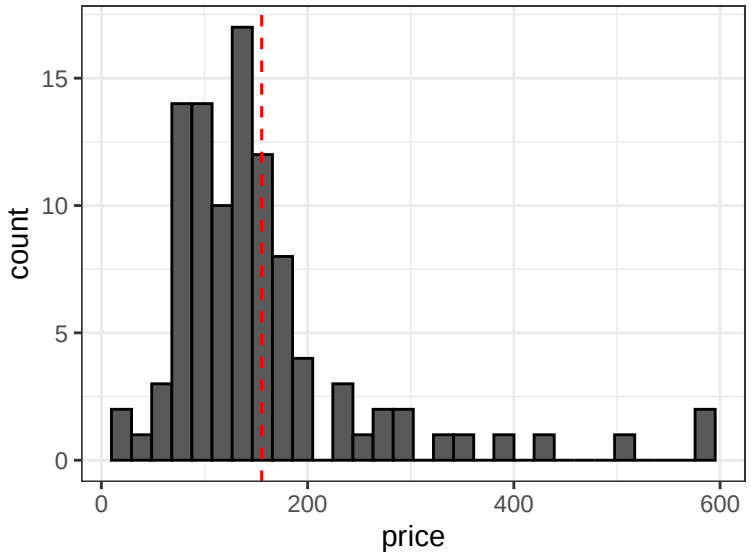
```
# A tibble: 100 x 7
    case taxes  beds baths   new price  size
   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
 1     1  3104     4     2     0  280.  2048
 2     2  1173     2     1     0  146.   912
 3     3  3076     4     2     0  238.  1654
 4     4  1608     3     2     0  200   2068
 5     5  1454     3     3     0  160.  1477
 6     6  2997     3     2     1  500.  3153
 7     7  4054     3     2     0  266.  1355
 8     8  3002     3     2     1  290.  2075
 9     9  6627     5     4     0  587   3990
10    10   320     3     2     0   70   1160
# i 90 more rows
```

12.1 Exploration

Let's first do a bit of exploration:

```
# visualize distribution of housing prices, superimposing the mean
houses_data |>
  ggplot(aes(x = price)) +
```

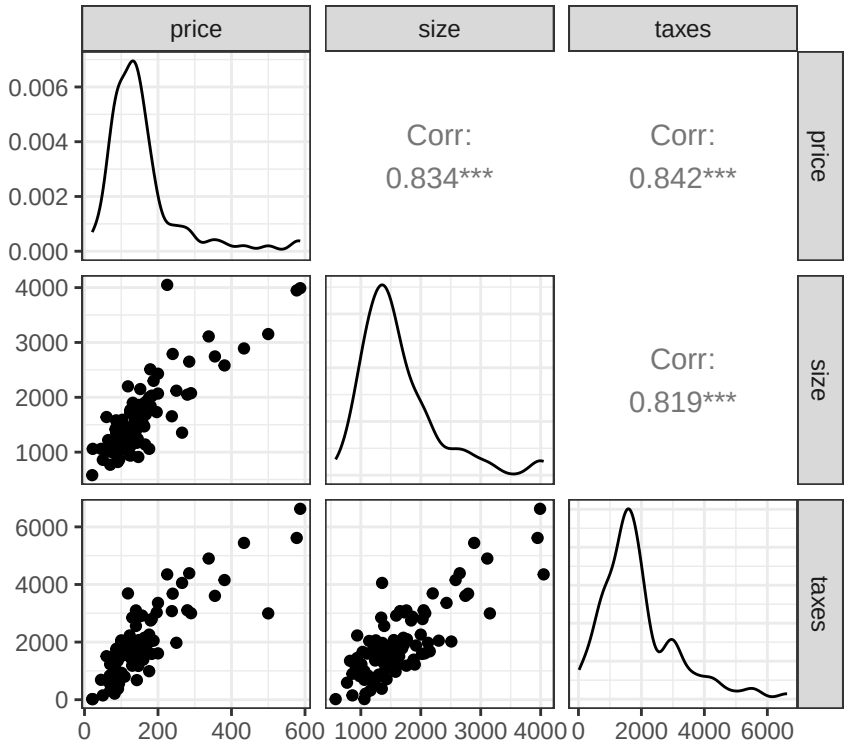
```
    geom_histogram(color = "black", bins = 30) +
    geom_vline(aes(xintercept = mean(price)),
      colour = "red",
      linetype = "dashed"
    )
```



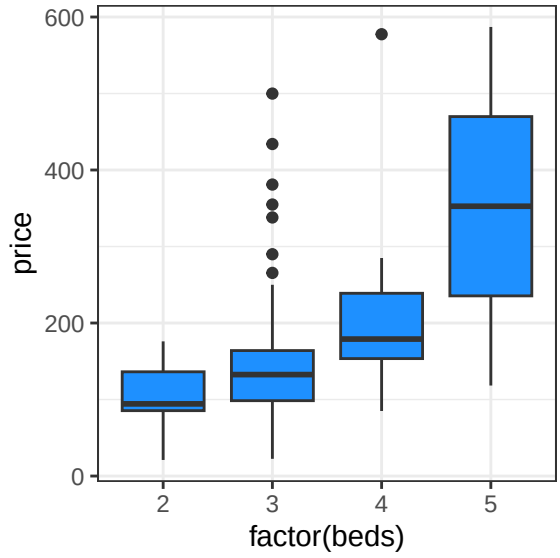

```
  # compare median and mean price
  houses_data |>
    summarise(
      mean_price = mean(price),
      median_price = median(price)
    )
```

```
# A tibble: 1 x 2
  mean_price median_price
       <dbl>        <dbl>
1       155.         133.
```

```
  # create a pairs plot of continuous variables
  houses_data |>
    select(price, size, taxes) |>
    ggpairs()
```

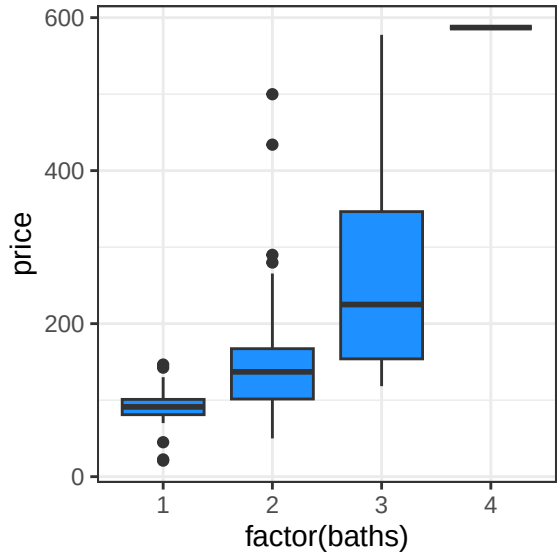



```
# see how price relates to beds
houses_data |>
  ggplot(aes(x = factor(beds), y = price)) +
  geom_boxplot(fill = "dodgerblue")
```

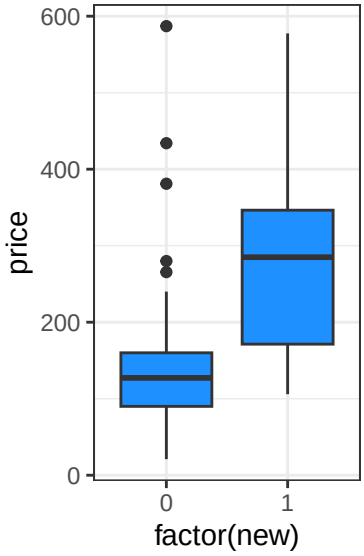



```
# see how price relates to baths
houses_data |>
```

```
  ggplot(aes(x = factor(baths), y = price)) +
  geom_boxplot(fill = "dodgerblue")
```




```
# see how price relates to new
houses_data |>
  ggplot(aes(x = factor(new), y = price)) +
  geom_boxplot(fill = "dodgerblue")
```



12.2 Hypothesis testing

Let's run a linear regression and interpret the summary. But first, we must decide whether to model beds/baths as categorical or continuous? We should probably model these as categorical, given the potentially nonlinear trend observed in the box plots.

```r
lm_fit <- lm(price ~ factor(beds) + factor(baths) + new + size,
  data = houses_data
)
summary(lm_fit)
```

```
Call:
lm(formula = price ~ factor(beds) + factor(baths) + new + size,
    data = houses_data)

Residuals:
     Min       1Q   Median       3Q      Max
-179.306  -32.037   -2.899   19.115  152.718

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)     -19.26307   18.01344  -1.069 0.287730
factor(beds)3   -16.46430   15.04669  -1.094 0.276749
factor(beds)4   -12.48561   21.12357  -0.591 0.555936
factor(beds)5  -101.14581   55.83607  -1.811 0.073366 .
factor(baths)2    2.39872   15.44014   0.155 0.876885
factor(baths)3   -0.70410   26.45512  -0.027 0.978825
factor(baths)4  273.20079   83.65764   3.266 0.001540 **
new              66.94940   18.50445   3.618 0.000487 ***
size              0.10882    0.01234   8.822 7.46e-14 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 51.17 on 91 degrees of freedom
Multiple R-squared:  0.7653,    Adjusted R-squared:  0.7446
F-statistic: 37.08 on 8 and 91 DF,  p-value: < 2.2e-16
```

We can read off the test statistics and p-values for each variable from the regression summary, as well as for the F-test against the constant model from the bottom of the summary.

Let's use an F-test to assess whether the categorical `baths` variable is important.

```r
lm_fit_partial <- lm(price ~ factor(beds) + new + size,
  data = houses_data
)
anova(lm_fit_partial, lm_fit)
```

```
Analysis of Variance Table

Model 1: price ~ factor(beds) + new + size
Model 2: price ~ factor(beds) + factor(baths) + new + size
  Res.Df    RSS Df Sum of Sq      F   Pr(>F)
1     94 273722
2     91 238289  3     35433 4.5104 0.005374 **
```

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

What if we had not coded `baths` as a factor?

```
lm_fit_not_factor <- lm(price ~ factor(beds) + baths + new + size,
  data = houses_data
)
anova(lm_fit_partial, lm_fit_not_factor)
```

```
Analysis of Variance Table

Model 1: price ~ factor(beds) + new + size
Model 2: price ~ factor(beds) + baths + new + size
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1     94 273722
2     93 273628  1     94.33 0.0321 0.8583
```

If we want to test for the equality of means across groups of a categorical predictor, without adjusting for other variables, we can use the ANOVA F-test. There are several equivalent ways of doing so:

```
# just use the summary function
lm_fit_baths <- lm(price ~ factor(baths), data = houses_data)
summary(lm_fit_baths)
```

```
Call:
lm(formula = price ~ factor(baths), data = houses_data)

Residuals:
    Min      1Q  Median      3Q     Max
-146.44  -45.88   -7.89   22.22  352.01

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)       90.21      19.51   4.624 1.17e-05 ***
factor(baths)2    57.68      21.72   2.656  0.00927 **
factor(baths)3   174.52      31.13   5.607 1.97e-07 ***
factor(baths)4   496.79      82.77   6.002 3.45e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 80.44 on 96 degrees of freedom
Multiple R-squared:  0.3881,    Adjusted R-squared:  0.369
F-statistic:  20.3 on 3 and 96 DF,  p-value: 2.865e-10
```

```
# use the anova function as before
lm_fit_const <- lm(price ~ 1, data = houses_data)
anova(lm_fit_const, lm_fit_baths)
```

```
Analysis of Variance Table

Model 1: price ~ 1
Model 2: price ~ factor(baths)
  Res.Df     RSS Df Sum of Sq      F    Pr(>F)
1     99 1015150
2     96  621130  3    394020 20.299 2.865e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# use the aov function
aov_fit <- aov(price ~ factor(baths), data = houses_data)
summary(aov_fit)
```

```
              Df Sum Sq Mean Sq F value   Pr(>F)
factor(baths)  3 394020  131340    20.3 2.86e-10 ***
Residuals     96 621130    6470
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can also use an F-test to test for the presence of an interaction with a multi-class categorical predictor.

```r
lm_fit_interaction <- lm(price ~ size * factor(beds), data = houses_data)
summary(lm_fit_interaction)
```

```
Call:
lm(formula = price ~ size * factor(beds), data = houses_data)

Residuals:
     Min       1Q   Median       3Q      Max
-232.643  -25.938   -0.942   19.172  155.517

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)          50.12619   48.22282   1.039 0.301310
size                  0.05037    0.04210   1.197 0.234565
factor(beds)3      -103.85734   52.20373  -1.989 0.049620 *
factor(beds)4      -143.90213   67.31359  -2.138 0.035185 *
factor(beds)5      -507.88205  144.10191  -3.524 0.000663 ***
size:factor(beds)3    0.07589    0.04368   1.738 0.085633 .
size:factor(beds)4    0.09234    0.04704   1.963 0.052638 .
size:factor(beds)5    0.21147    0.05957   3.550 0.000609 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 53.35 on 92 degrees of freedom
Multiple R-squared:  0.7421,    Adjusted R-squared:  0.7225
```

```
F-statistic: 37.81 on 7 and 92 DF,  p-value: < 2.2e-16
```

```
lm_fit_size <- lm(price ~ size + factor(beds), data = houses_data)
anova(lm_fit_size, lm_fit_interaction)
```

```
Analysis of Variance Table

Model 1: price ~ size + factor(beds)
Model 2: price ~ size * factor(beds)
  Res.Df    RSS Df Sum of Sq      F   Pr(>F)
1     95 300953
2     92 261832  3     39121 4.5819 0.004905 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Contrasts of regression coefficients can be tested using the `glht()` function from the `multcomp` package.

12.3 Confidence intervals

We can construct pointwise confidence intervals for each coefficient using `confint()`:

```
confint(lm_fit)
```

```
                     2.5 %      97.5 %
(Intercept)     -55.04455734  16.5184161
factor(beds)3   -46.35270691  13.4241025
factor(beds)4   -54.44498235  29.4737689
factor(beds)5  -212.05730801   9.7656895
factor(baths)2  -28.27123130  33.0686620
factor(baths)3  -53.25394742  51.8457394
factor(baths)4  107.02516067 439.3764122
new              30.19258305 103.7062177
size              0.08431972   0.1333284
```

To create simultaneous confidence intervals, we need a somewhat more manual approach. We start with the coefficients and standard errors:

```
coef(summary(lm_fit))
```

```
                   Estimate  Std. Error     t value     Pr(>|t|)
(Intercept)     -19.2630706 18.01344052 -1.06937209 2.877304e-01
factor(beds)3   -16.4643022 15.04669172 -1.09421410 2.767490e-01
factor(beds)4   -12.4856067 21.12356937 -0.59107467 5.559357e-01
factor(beds)5  -101.1458092 55.83607248 -1.81147786 7.336590e-02
factor(baths)2    2.3987153 15.44014266  0.15535578 8.768849e-01
factor(baths)3   -0.7041040 26.45511871 -0.02661504 9.788251e-01
factor(baths)4  273.2007864 83.65764044  3.26570036 1.540093e-03
```

```
new              66.9494004 18.50445029  3.61801617 4.872475e-04
size              0.1088241  0.01233621  8.82151661 7.460814e-14
```

Then we add lower and upper confidence interval endpoints based on the formula (10.9):

```
alpha <- 0.05
n <- nrow(houses_data)
p <- length(coef(lm_fit))
f_quantile <- qf(1 - alpha, df1 = p, df2 = n - p)
coef(summary(lm_fit)) |>
  as.data.frame() |>
  rownames_to_column(var = "Variable") |>
  select(Variable, Estimate, `Std. Error`) |>
  mutate(
    CI_lower = Estimate - `Std. Error` * sqrt(p * f_quantile),
    CI_upper = Estimate + `Std. Error` * sqrt(p * f_quantile)
  )
```

```
        Variable     Estimate  Std. Error      CI_lower    CI_upper
1    (Intercept)  -19.2630706 18.01344052  -95.38917389  56.8630327
2  factor(beds)3  -16.4643022 15.04669172  -80.05271036  47.1241059
3  factor(beds)4  -12.4856067 21.12356937 -101.75533960  76.7841262
4  factor(beds)5 -101.1458092 55.83607248 -337.11309238 134.8214739
5 factor(baths)2    2.3987153 15.44014266  -62.85244495  67.6498756
6 factor(baths)3   -0.7041040 26.45511871 -112.50535022 111.0971422
7 factor(baths)4  273.2007864 83.65764044  -80.34245635 626.7440292
8            new   66.9494004 18.50445029  -11.25174573 145.1505465
9           size    0.1088241  0.01233621    0.05669037   0.1609578
```

Note that the simultaneous intervals are substantially larger.

To construct pointwise confidence intervals for the fit, we can use the `predict()` function:

```
predict(lm_fit, newdata = houses_data, interval = "confidence") |> head()
```

```
       fit       lwr      upr
1 193.52176 165.22213 221.8214
2  79.98449  51.91430 108.0547
3 150.64507 122.28397 179.0062
4 191.71955 172.27396 211.1651
5 124.30169  81.34488 167.2585
6 376.74308 333.44559 420.0406
```

To get pointwise prediction intervals, we switch `"confidence"` to `"prediction"`:

```
predict(lm_fit, newdata = houses_data, interval = "prediction") |> head()
```

```
       fit       lwr      upr
1 193.52176  88.00908 299.0344
```

```
2  79.98449 -25.46688 185.4359
3 150.64507  45.11589 256.1743
4 191.71955  88.22951 295.2096
5 124.30169  13.95069 234.6527
6 376.74308 266.25901 487.2271
```

To construct simultaneous confidence intervals for the fit or predictions, we again need a slightly more manual approach. We call `predict()` again, but this time asking it for the standard errors rather than the confidence intervals:

```
predictions <- predict(lm_fit, newdata = houses_data, se.fit = TRUE)
head(predictions$fit)
```

```
       1         2         3         4         5         6
193.52176  79.98449 150.64507 191.71955 124.30169 376.74308
```

```
head(predictions$se.fit)
```

```
       1         2         3         4         5         6
14.246855 14.131352 14.277804  9.789472 21.625709 21.797212
```

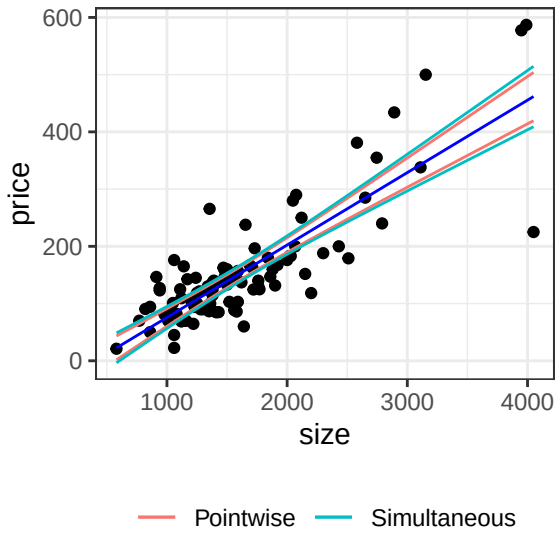
Now we can construct the simultaneous confidence intervals via the formula (10.8):

```
f_quantile <- qf(1 - alpha, df1 = p, df2 = n - p)
tibble(
  lower = predictions$fit - predictions$se.fit * sqrt(p * f_quantile),
  upper = predictions$fit + predictions$se.fit * sqrt(p * f_quantile)
)
```

```
# A tibble: 100 x 2
   lower upper
   <dbl> <dbl>
 1 133.   254.
 2  20.3  140.
 3  90.3  211.
 4 150.   233.
 5  32.9  216.
 6 285.   469.
 7  82.8  145.
 8 188.   331.
 9 371.   803.
10  57.3  128.
# i 90 more rows
```

In the case of simple linear regression, we can plot these pointwise and simultaneous confidence intervals as bands:

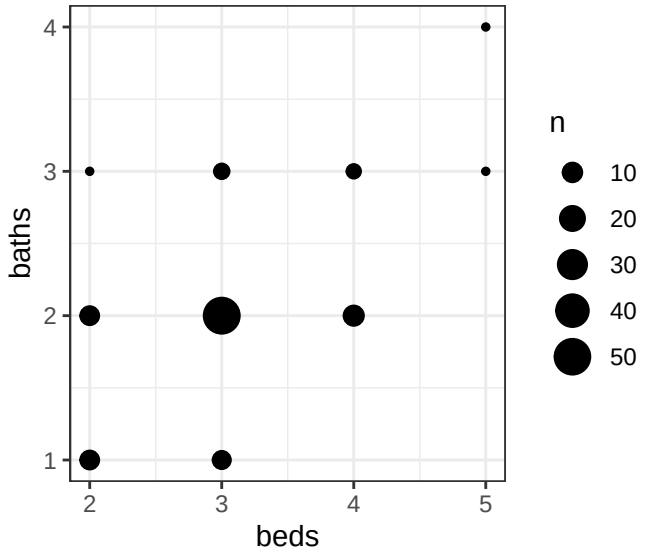
```r
# to produce confidence intervals for fits in general, use the predict() function
n <- nrow(houses_data)
p <- 2
alpha <- 0.05
lm_fit <- lm(price ~ size, data = houses_data)
predictions <- predict(lm_fit, se.fit = TRUE)
t_quantile <- qt(1 - alpha / 2, df = n - p)
f_quantile <- qf(1 - alpha, df1 = p, df2 = n - p)
houses_data |>
  mutate(
    fit = predictions$fit,
    se = predictions$se.fit,
    ptwise_width = t_quantile * se,
    simultaneous_width = sqrt(p * f_quantile) * se
  ) |>
  ggplot(aes(x = size)) +
  geom_point(aes(y = price)) +
  geom_line(aes(y = fit), color = "blue") +
  geom_line(aes(y = fit + ptwise_width, color = "Pointwise")) +
  geom_line(aes(y = fit - ptwise_width, color = "Pointwise")) +
  geom_line(aes(y = fit + simultaneous_width, color = "Simultaneous")) +
  geom_line(aes(y = fit - simultaneous_width, color = "Simultaneous")) +
  theme(legend.title = element_blank(), legend.position = "bottom")
```



12.4 Predictor competition and collaboration

Let's look at the power of detecting the association between `price` and `beds`. We can imagine that `beds` and `baths` are correlated:

```
houses_data |>
  ggplot(aes(x = beds, y = baths)) +
  geom_count()
```



So let's see how significant `beds` is, with and without `baths` in the model:

```
lm_fit_only_beds <- lm(price ~ factor(beds), data = houses_data)
summary(lm_fit_only_beds)
```

```
Call:
lm(formula = price ~ factor(beds), data = houses_data)

Residuals:
    Min      1Q  Median      3Q     Max
-234.35  -50.63  -15.69   24.56  365.86

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)     105.94      21.48   4.931 3.43e-06 ***
factor(beds)3    44.69      24.47   1.827 0.070849 .
factor(beds)4   105.70      32.35   3.268 0.001504 **
factor(beds)5   246.71      69.62   3.544 0.000611 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 93.65 on 96 degrees of freedom
Multiple R-squared:  0.1706,    Adjusted R-squared:  0.1447
F-statistic: 6.583 on 3 and 96 DF,  p-value: 0.0004294
```

```r
lm_fit_only_baths <- lm(price ~ factor(baths), data = houses_data)
lm_fit_beds_baths <- lm(price ~ factor(beds) + factor(baths), data = houses_data)
anova(lm_fit_only_baths, lm_fit_beds_baths)
```

```
Analysis of Variance Table

Model 1: price ~ factor(baths)
Model 2: price ~ factor(beds) + factor(baths)
  Res.Df    RSS Df Sum of Sq     F  Pr(>F)
1     96 621130
2     93 572436  3     48693 2.637 0.05424 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We see that the significance of `beds` dropped by two orders of magnitude. This is an example of predictor competition.

On the other hand, note that the variable `new` is not very correlated with `beds`:

```r
lm_fit <- lm(new ~ beds, data = houses_data)
summary(lm_fit)
```

```
Call:
lm(formula = new ~ beds, data = houses_data)

Residuals:
     Min       1Q   Median       3Q      Max
-0.15762 -0.11000 -0.11000 -0.08619  0.91381

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.03857    0.14950   0.258    0.797
beds         0.02381    0.04871   0.489    0.626

Residual standard error: 0.3157 on 98 degrees of freedom
Multiple R-squared:  0.002432,  Adjusted R-squared:  -0.007747
F-statistic: 0.2389 on 1 and 98 DF,  p-value: 0.6261
```

but we know it has a substantial impact on `price`. Let's look at the significance of the test that `beds` is not important when we add `new` to the model.

```r
lm_fit_only_new <- lm(price ~ new, data = houses_data)
lm_fit_beds_new <- lm(price ~ new + factor(beds), data = houses_data)
anova(lm_fit_only_new, lm_fit_beds_new)
```

```
Analysis of Variance Table

Model 1: price ~ new
```

```
Model 2: price ~ new + factor(beds)
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1     98 787781
2     95 619845  3    167936 8.5795 4.251e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Adding `new` to the model made the p-value more significant by a factor of 10. This is an example of predictor collaboration.

Part III

Linear models: Misspecification

In our discussion of linear model inference in Unit 2, we assumed the normal linear model throughout:

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \quad \text{where } \boldsymbol{\epsilon} \sim N(\boldsymbol{0}, \sigma^2 \boldsymbol{I}_n).$$

In this unit, we will discuss what happens when this model is misspecified:

- **Non-normality** (Section 13.1): $\boldsymbol{\epsilon} \sim (0, \sigma^2 \boldsymbol{I}_n)$ but not $N(0, \sigma^2 \boldsymbol{I}_n)$.
- **Heteroskedastic and/or correlated errors** (Section 13.2): $\boldsymbol{\epsilon} \sim (0, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma} \neq \sigma^2 \boldsymbol{I}$. This includes the case of heteroskedastic errors ($\boldsymbol{\Sigma}$ is diagonal but not a constant multiple of the identity) and correlated errors ($\boldsymbol{\Sigma}$ is not diagonal).
- **Model bias** (Section 13.3): It is not the case that $\mathbb{E}[\boldsymbol{y}] = \boldsymbol{X}\boldsymbol{\beta}$ for some $\boldsymbol{\beta} \in \mathbb{R}^p$.
- **Outliers** (Section 13.4): For one or more i, it is not the case that $y_i \sim N(\boldsymbol{x}_{i*}^T \boldsymbol{\beta}, \sigma^2)$.

For each type of misspecification, we will discuss its origins, consequences, detection, and fixes (Section 13.1-Section 13.4). We then discuss methodological approaches to address model misspecification, including asymptotic robust inference methods (Chapter 14), the bootstrap (Chapter 15), the permutation test (Chapter 16), and robust estimation (Chapter 17). We conclude with an R demo (Chapter 18).

Chapter 13

Overview

13.1 Non-normality

13.1.1 Origin

Non-normality occurs when the distribution of $y|\boldsymbol{x}$ is either skewed or has heavier tails than the normal distribution. This may happen, for example, if there is some discreteness in y.

13.1.2 Consequences

Non-normality is the most benign of linear model misspecifications. While we derived linear model inferences under the normality assumption, all the corresponding statements hold asymptotically without this assumption. Recall Homework 2 Question 1, or take for example the simpler problem of estimating the mean μ of a distribution based on n samples from it: We can test $H_0 : \mu = 0$ and build a confidence interval for μ even if the underlying distribution is not normal. So if n is relatively large and p is relatively small, you need not worry too much. If n is small and the errors are highly skewed or heavy-tailed, we may have issues with incorrect standard errors.

13.1.3 Detection

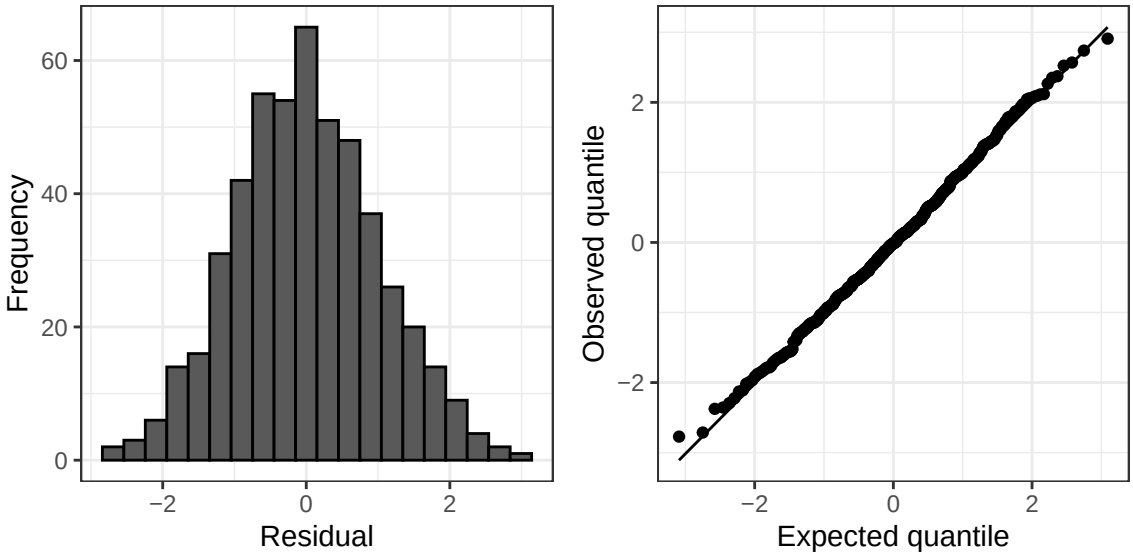
Non-normality is a property of the error terms ϵ_i. We do not observe these directly, but we can approximate them using the residuals:

$$\widehat{\epsilon}_i = y_i - \boldsymbol{x}_{i*}^T \widehat{\boldsymbol{\beta}}.$$

Recall from equation (7.2) that $\text{Var}[\widehat{\boldsymbol{\epsilon}}] = \sigma^2(\boldsymbol{I} - \boldsymbol{H})$. Letting h_i be the ith diagonal entry of $\boldsymbol{H}$, it follows that $\widehat{\epsilon}_i \sim (0, \sigma^2(1 - h_i))$. The *standardized residuals* are defined as:

$$r_i = \frac{\widehat{\epsilon}_i}{\widehat{\sigma}\sqrt{1 - h_i}}. \tag{13.1}$$

Under normality, we would expect $r_i \dot{\sim} N(0, 1)$. We can therefore assess normality by producing a histogram or normal QQ-plot of these residuals.



13.1.4 Fixes

As mentioned above, non-normality is not necessarily a problem that needs to be fixed, except in small samples. In small samples (but not too small!), we can apply the residual bootstrap for robust standard error computation and/or robust hypothesis testing.

13.2 Heteroskedastic and correlated errors

13.2.1 Origin

Heteroskedasticity can arise as follows. Suppose each observation y_i is actually the average of n_i underlying observations, each with variance σ^2. Then, the variance of y_i is σ^2/n_i, which will differ across i if n_i differ. It is also common to see the variance of a distribution increase as the mean increases (as in Figure 13.1), whereas for a linear model the variance of y stays constant as the mean of y varies.

Correlated errors can arise when observations have group, spatial, or temporal structure. Below are examples:

- **Group/clustered structure**: We have 10 samples $(\boldsymbol{x}_{i*}, y_i)$ each from 100 schools.
- **Spatial structure**: We have 100 soil samples from a 10×10 grid on a 1km $\times$ 1km field.
- **Temporal structure**: We have 366 COVID positivity rate measurements, one from each day of the year 2020.

The issue arises because there are common sources of variation among samples that are in the same group or spatially/temporally close to one another.

13.2.2 Consequences

All normal linear model inference from Unit 2 hinges on the assumption that $\boldsymbol{\epsilon} \sim N(\mathbf{0}, \sigma^2 \boldsymbol{I})$. If instead of $\sigma^2 \boldsymbol{I}$ we have $\text{Var}[\boldsymbol{\epsilon}] = \boldsymbol{\Sigma}$ for some matrix $\boldsymbol{\Sigma}$, then we may suffer two consequences: wrong inference (in terms of confidence interval coverage and hypothesis test levels) and inefficient inference (in terms of confidence interval width and hypothesis test power). One way of seeing

the consequence of heteroskedasticity for confidence interval coverage is the width of prediction intervals; see Figure 13.1 for intuition.

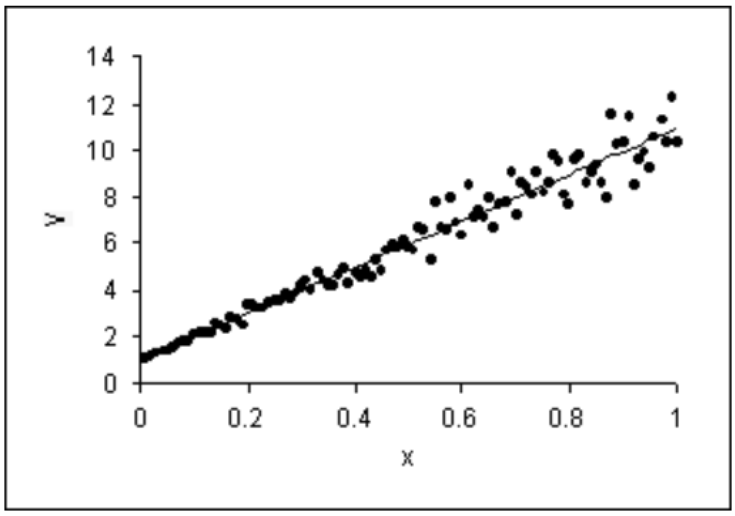


Figure 13.1: Heteroskedasticity in a simple bivariate linear model (image source: source).

Like with heteroskedastic errors, correlated errors can cause invalid standard errors. In particular, positively correlated errors typically cause standard errors to be smaller than they should be, leading to inflated Type-I error rates. For intuition, consider estimating the mean of a distribution based on n samples. Consider the cases when these samples are independent, compared to when they are perfectly correlated. The effective sample size in the former case is n and in the latter case is 1.

13.2.3 Detection

Heteroskedasticity is usually assessed via the *residual plot* (Figure 13.2). In this plot, the standardized residuals r_i (13.1) are plotted against the fitted values $\widehat{\mu}_i$. In the absence of heteroskedasticity, the spread of the points around the origin should be roughly constant as a function of $\widehat{\mu}$ (Figure 13.2(a)). A common sign of heteroskedasticity is the fan shape where variance increases as a function of $\widehat{\mu}$ (Figure 13.2(c)).

Residual plots once again come in handy to detect correlated errors. Instead of plotting the standardized residuals against the fitted values, we should plot the residuals against whatever variables we think might explain variation in the response that the regression does not account for. In the presence of group structures, we can plot residuals versus group (via a boxplot); in the presence of spatial or temporal structure, we can plot residuals as a function of space or time. If the residuals show a dependency on these variables, this suggests they are correlated. This dependency can be checked via formal means as well, e.g., via an ANOVA test in the case of groups or by estimating the autocorrelation function in the case of temporal structure.

13.3 Model bias

13.3.1 Origin

Model bias arises when predictors are left out of the regression model:

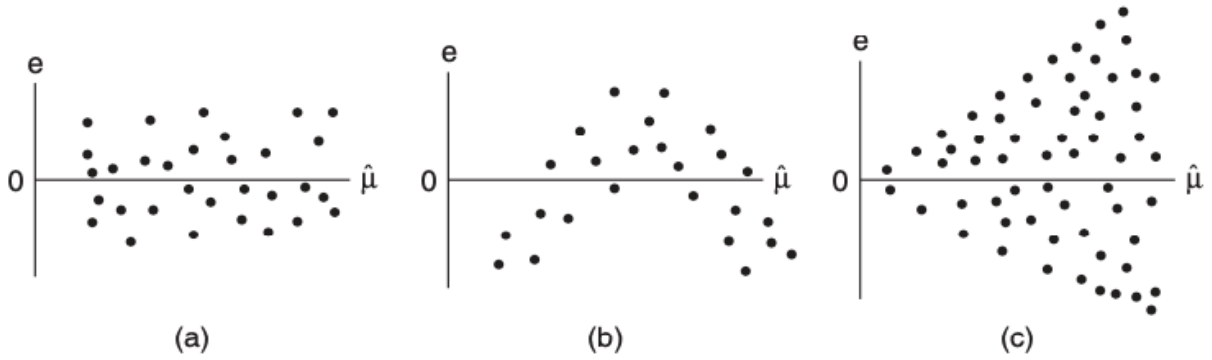


Figure 13.2: Residuals plotted against linear-model fitted values that reflect (a) model adequacy, (b) quadratic rather than linear relationship, and (c) nonconstant variance (image source: Agresti Figure 2.8).

$$\text{assumed model: } \boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}; \quad \text{actual model: } \boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{Z}\boldsymbol{\gamma} + \boldsymbol{\epsilon}. \tag{13.2}$$

We may not always know about or measure all the variables that impact a response $\boldsymbol{y}$.

Model bias can also arise when the predictors do not impact the response on the linear scale. For example:

$$\text{assumed model: } \mathbb{E}[\boldsymbol{y}] = \boldsymbol{X}\boldsymbol{\beta}; \quad \text{actual model: } g(\mathbb{E}[\boldsymbol{y}]) = \boldsymbol{X}\boldsymbol{\beta}. \tag{13.3}$$

13.3.2 Consequences

In cases of model bias, the parameters $\boldsymbol{\beta}$ in the assumed linear model lose their meanings. The least squares estimate $\widehat{\boldsymbol{\beta}}$ will be a biased estimate for the parameter we probably actually want to estimate. In the case (13.2) when predictors are left out of the regression model, these additional predictors $\boldsymbol{Z}$ will act as confounders and create bias in $\widehat{\boldsymbol{\beta}}$ as an estimate of the $\boldsymbol{\beta}$ parameters in the true model, unless $\boldsymbol{X}^T\boldsymbol{Z} = 0$. As discussed in Unit 2, this can lead to misleading conclusions.

13.3.3 Detection

Similarly to the detection of correlated errors, we can try to identify model bias by plotting the standardized residuals against predictors that may have been left out of the model. A good place to start is to plot standardized residuals against the predictors $\boldsymbol{X}$ (one at a time) that are in the model, since nonlinear transformations of these might have been left out. In this case, you would see something like Figure 13.2(b).

It is possible to formally test for model bias in cases when we have repeated observations of the response for each value of the predictor vector. In particular, suppose that $\boldsymbol{x}_{i*} = \boldsymbol{x}_c$ for $c = c(i)$ and predictor vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_C \in \mathbb{R}^p$. Then, consider testing the following hypothesis:

$$H_0 : y_i = \boldsymbol{x}_{i*}^T\boldsymbol{\beta} + \epsilon_i \quad \text{versus} \quad H_1 : y_i = \beta_{c(i)} + \epsilon_i.$$

The model under H_0 (the linear model) is nested in the model for H_1 (the saturated model), and we can test this hypothesis using an F-test called the *lack of fit F-test*.

13.3.4 Overview of fixes

To fix model bias in the case (13.2), ideally we would identify the missing predictors $\boldsymbol{Z}$ and add them to the regression model. This may not always be feasible or possible. To fix model bias in the case (13.3), it is sometimes advocated to find a transformation g (e.g., a square root or a logarithm) of $\boldsymbol{y}$ such that $\mathbb{E}[g(\boldsymbol{y})] = \boldsymbol{X}\boldsymbol{\beta}$. However, a better solution is to use a *generalized linear model*, which we will discuss starting in Unit 4.

13.4 Outliers

13.4.1 Origin

Outliers often arise due to measurement or data entry errors. An observation can be an outlier in $\boldsymbol{x}$, in y, or both.

13.4.2 Consequences

An outlier can have the effect of biasing the estimate $\widehat{\boldsymbol{\beta}}$. This occurs when an observation has outlying $\boldsymbol{x}$ as well as outlying y.

13.4.3 Detection

There are a few measures associated with an observation that can be used to detect outliers, though none are perfect. The first quantity is called the *leverage*, defined as:

$$\text{leverage of observation } i \equiv \text{corr}^2(y_i, \widehat{\mu}_i).$$

This quantity measures the extent to which the fitted value $\widehat{\mu}_i$ is sensitive to the (noise in the) observation y_i. It can be derived that:

$$\text{leverage of observation } i = h_i,$$

which is the ith diagonal element of the hat matrix $\boldsymbol{H}$. This is related to the fact that $\text{Var}[\widehat{\epsilon}_i] = \sigma^2(1 - h_i)$. The larger the leverage, the smaller the variance of the residual, so the closer the line passes to the ith observation. The leverage of an observation is larger to the extent that $\boldsymbol{x}_{i*}$ is far from $\bar{\boldsymbol{x}}$. For example, in the bivariate linear model $y_i = \beta_0 + \beta_1 x_i + \epsilon_i$,

$$h_i = \frac{1}{n} + \frac{(x_i - \bar{x})^2}{\sum_{i'=1}^{n}(x_{i'} - \bar{x})^2}.$$

Note that the average of the leverages is:

$$\frac{1}{n}\sum_{i=1}^{n} h_i = \frac{1}{n}\text{trace}(\boldsymbol{H}) = \frac{p}{n}.$$

An observation's leverage is considered large if it is significantly larger than this, e.g., three times larger.

Note that the leverage is not a function of y_i, so a high-leverage point might or might not be an outlier in y_i and therefore might or might not have a strong impact on the regression. To assess more directly whether an observation is *influential*, we can compare the least squares fits with and without that observation. To this end, we define the *Cook's distance*:

$$D_i = \frac{\sum_{i'=1}^{n}(\widehat{\mu}_{i'} - \widehat{\mu}_{i'}^{-i})^2}{p\widehat{\sigma}^2},$$

where $\widehat{\mu}_{i'}^{-i} = \boldsymbol{x}_{i*}^T \widehat{\boldsymbol{\beta}}^{-i}$ and $\widehat{\boldsymbol{\beta}}^{-i}$ is the least squares estimate based on $(\boldsymbol{X}_{-i,*}, \boldsymbol{y}_{-i})$. An observation is considered influential if it has Cook's distance greater than one.

There is a connection between Cook's distance and leverage:

$$D_i = \left(\frac{y_i - \widehat{\mu}_i}{\widehat{\sigma}\sqrt{1-h_{ii}}}\right)^2 \cdot \frac{h_{ii}}{p(1-h_{ii})}.$$

We recognize the first term as the standardized residual; therefore a point is influential if its residual and leverage are large.

Note that Cook's distance may not successfully identify outliers. For example, if there are groups of outliers, then they will *mask* each other in the calculation of Cook's distance.

13.4.4 Overview of fixes

If outliers can be detected, then the fix is to remove them from the regression. But, we need to be careful. Definitively determining whether observations are outliers can be tricky. Outlier detection can even be used as a way to commit fraud with data, as now-defunct blood testing start-up Theranos is alleged to have done. As an alternative to removing outliers, we can fit estimators $\widehat{\boldsymbol{\beta}}$ that are less sensitive to outliers; see Chapter 17.

Chapter 14

Asymptotic methods

In this section, we present a set of asymptotic methods for fixing heteroskedastic or correlated errors, in the setting that

$$\boldsymbol{y} \sim N(\boldsymbol{X}\boldsymbol{\beta}, \boldsymbol{\Sigma}). \tag{14.1}$$

These methods are based on estimating $\boldsymbol{\Sigma}$; they use this estimate to either (i) build a better estimate $\widehat{\boldsymbol{\beta}}$ (Section 14.1) or (ii) build better standard errors for the least squares estimate (Section 14.2). We discuss these two approaches in turn, followed by how to carry out inference based on the resulting estimates (Section 14.3).

14.1 Methods that build a better estimate of $\boldsymbol{\beta}$

14.1.1 Generalized least squares

Let us premultiply $\boldsymbol{y}$ by $\boldsymbol{\Sigma}^{-1/2}$ to obtain

$$\boldsymbol{\Sigma}^{-1/2}\boldsymbol{y} \sim N(\boldsymbol{\Sigma}^{-1/2}\boldsymbol{X}\boldsymbol{\beta}, \boldsymbol{I}).$$

Viewing $\boldsymbol{\Sigma}^{-1/2}\boldsymbol{y}$ as the new response and $\boldsymbol{\Sigma}^{-1/2}\boldsymbol{X}$ as the new model matrix, we can apply the usual least squares estimator to obtain

$$\tilde{\boldsymbol{\beta}}^{\text{GLS}} = (\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{y}. \tag{14.2}$$

This is the *generalized least squares* estimate for the model (14.1), and has the following distribution:

$$\tilde{\boldsymbol{\beta}}^{\text{GLS}} \sim N(\boldsymbol{\beta}, (\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X})^{-1}).$$

By the Gauss-Markov theorem, this is the best linear unbiased estimate of $\boldsymbol{\beta}$, recovering efficiency. We would like to carry out inference based on the latter distributional result analogously to how we did so in Chapter 2, as long as we can estimate $\boldsymbol{\Sigma}$ accurately enough.

14.1.2 Models for Σ

This class of methods typically postulates a parametric form for $\boldsymbol{\Sigma}$, denoted by $\boldsymbol{\Sigma}(\boldsymbol{\nu})$, where $\boldsymbol{\nu}$ is a vector of parameters, and then proceed by estimating $\boldsymbol{\nu}$. Below are a few examples of such parametric models:

- **Heteroskedastic errors.** In this case, we can assume that $\boldsymbol{\Sigma} = \operatorname{diag}(\sigma_1^2, \ldots, \sigma_n^2)$, where
$$\log \sigma_i^2 = \boldsymbol{x}_i^T \boldsymbol{\nu}.$$
- **Clustered errors.** Suppose that each observation i falls into a cluster $c(i)$. Then, we can postulate a *random effects model*
$$y_i = \boldsymbol{x}_i^T \boldsymbol{\beta} + \delta_{c(i)} + \tau_i, \quad \text{where} \quad \delta_c \overset{\text{i.i.d.}}{\sim} N(0, \sigma_\delta^2), \quad \tau_i \overset{\text{i.i.d.}}{\sim} N(0, \sigma_\tau^2).$$
This imposes a block-diagonal structure on $\boldsymbol{\Sigma}$, where each block corresponds to a cluster.
- **Temporal errors.** If the observations have a temporal structure, we might impose an AR(1) model on the residuals:
$$\epsilon_1 = \tau_1; \quad \epsilon_i = \rho \epsilon_{i-1} + \tau_i \quad \text{for } i > 1, \quad \text{where} \quad \tau_i \overset{\text{i.i.d.}}{\sim} N(0, \sigma^2).$$
This imposes an approximately banded structure on $\boldsymbol{\Sigma}$, with $\Sigma_{i_1 i_2} = \sigma^2 \rho^{|i_1 - i_2|}$.

> **Random versus fixed effects models**
>
> Random effects models deal address correlated errors but not with model bias. The difference is that, in the case of correlated errors, the errors may be correlated with themselves but not with the regressors. In the case of model bias, the errors may be correlated with the regressors. To address model bias in the presence of clustering structure, fixed effects are necessary. Fixed effects models decrease model bias at the cost of increased variance, because more parameters must be estimated. Random effects models are more susceptible to model bias but have lower variance.

14.1.3 Estimating $\boldsymbol{\Sigma}$

Given a parametric model for $\boldsymbol{\Sigma}$, we can estimate $\boldsymbol{\nu}$ by one of two approaches. The first approach, typical in statistics, is to maximize the likelihood as a function of $(\boldsymbol{\beta}, \boldsymbol{\nu})$. The second approach, typical in econometrics, is to estimate $\boldsymbol{\beta}$ using OLS, and then to fit $\boldsymbol{\nu}$ based on the residuals. This gives us the estimate $\widehat{\boldsymbol{\Sigma}} = \widehat{\boldsymbol{\Sigma}}(\widehat{\boldsymbol{\nu}})$.

14.1.4 Inferring about $\boldsymbol{\beta}$ based on the estimate $\widehat{\boldsymbol{\Sigma}}$

With an estimate $\widehat{\boldsymbol{\Sigma}}$ in hand, we can use it to build a (hopefully) better estimate of $\boldsymbol{\beta}$, using the following plug-in version of the GLS estimate (14.2):
$$\widehat{\boldsymbol{\beta}}^{\text{FGLS}} \equiv (\boldsymbol{X}^T \widehat{\boldsymbol{\Sigma}}^{-1} \boldsymbol{X})^{-1} \boldsymbol{X}^T \widehat{\boldsymbol{\Sigma}}^{-1} \boldsymbol{y}. \tag{14.3}$$

This is called the *feasible generalized least squares estimate* (FGLS) in econometrics, to contrast it with the infeasible estimate that assumes $\boldsymbol{\Sigma}$ is known exactly. Then, we can carry out inference based on the approximation distribution
$$\widehat{\boldsymbol{\beta}}^{\text{FGLS}} \dot{\sim} N(\boldsymbol{\beta}, (\boldsymbol{X}^T \widehat{\boldsymbol{\Sigma}}^{-1} \boldsymbol{X})^{-1}). \tag{14.4}$$

14.2 Methods that build better standard errors for OLS estimate

Sometimes we don't feel comfortable enough with our estimate of $\boldsymbol{\Sigma}$ to actually modify the least squares estimator. So we want to keep using our least squares estimator, but still get standard errors robust to heteroskedastic or correlated errors. There are several strategies to computing valid standard errors in such situations.

14.2.1 Sandwich standard errors

Let's say that $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim N(\boldsymbol{0}, \boldsymbol{\Sigma})$. Then, we can compute that the covariance matrix of the least squares estimate $\widehat{\boldsymbol{\beta}}$ is

$$\mathrm{Var}[\widehat{\boldsymbol{\beta}}] = (\boldsymbol{X}^T\boldsymbol{X})^{-1}(\boldsymbol{X}^T\boldsymbol{\Sigma}\boldsymbol{X})(\boldsymbol{X}^T\boldsymbol{X})^{-1}. \tag{14.5}$$

Note that this expression reduces to the usual $\sigma^2(\boldsymbol{X}^T\boldsymbol{X})^{-1}$ when $\boldsymbol{\Sigma} = \sigma^2\boldsymbol{I}$. It is called the sandwich variance because we have the $(\boldsymbol{X}^T\boldsymbol{\Sigma}\boldsymbol{X})$ term sandwiched between two $(\boldsymbol{X}^T\boldsymbol{X})^{-1}$ terms. If we have some estimate $\widehat{\boldsymbol{\Sigma}}$ of the covariance matrix, we can construct

$$\widehat{\mathrm{Var}}[\widehat{\boldsymbol{\beta}}] \equiv (\boldsymbol{X}^T\boldsymbol{X})^{-1}(\boldsymbol{X}^T\widehat{\boldsymbol{\Sigma}}\boldsymbol{X})(\boldsymbol{X}^T\boldsymbol{X})^{-1}. \tag{14.6}$$

Different estimates $\widehat{\boldsymbol{\Sigma}}$ are appropriate in different situations. Below we consider three of the most common choices: one for heteroskedasticity (due to Huber-White), one for group-correlated errors (due to Liang-Zeger), and one for temporally-correlated errors (due to Newey-West).

14.2.2 Specific instances of sandwich standard errors

Huber-White standard errors

Suppose $\boldsymbol{\Sigma} = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_n^2)$ for some variances $\sigma_1^2, \ldots, \sigma_n^2 > 0$. The Huber-White sandwich estimator is defined by (14.5), with

$$\widehat{\boldsymbol{\Sigma}} \equiv \mathrm{diag}(\widehat{\sigma}_1^2, \ldots, \widehat{\sigma}_n^2), \quad \text{where} \quad \widehat{\sigma}_i^2 = (y_i - \boldsymbol{x}_{i*}^T\widehat{\boldsymbol{\beta}})^2.$$

While each estimator $\widehat{\sigma}_i^2$ is very poor, Huber and White's insight was that the resulting estimate of the (averaged) quantity $\boldsymbol{X}^T\widehat{\boldsymbol{\Sigma}}\boldsymbol{X}$ is not bad. To see why, assume that $(\boldsymbol{x}_{i*}, y_i) \overset{\text{i.i.d.}}{\sim} F$ for some joint distribution F. Then, we have that

$$\begin{aligned}
\frac{1}{n}(\boldsymbol{X}^T\widehat{\boldsymbol{\Sigma}}\boldsymbol{X} - \boldsymbol{X}^T\boldsymbol{\Sigma}\boldsymbol{X}) &= \frac{1}{n}\sum_{i=1}^n(\widehat{\sigma}_i^2 - \sigma_i^2)\boldsymbol{x}_{i*}\boldsymbol{x}_{i*}^T \\
&= \frac{1}{n}\sum_{i=1}^n((\epsilon_i + \boldsymbol{x}_{i*}^T(\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}}))^2 - \sigma_i^2)\boldsymbol{x}_{i*}\boldsymbol{x}_{i*}^T \\
&= \frac{1}{n}\sum_{i=1}^n \epsilon_i^2\boldsymbol{x}_{i*}\boldsymbol{x}_{i*}^T + o_p(1) \\
&\xrightarrow{p} 0.
\end{aligned}$$

The last step holds by the law of large numbers, since $\mathbb{E}[\epsilon_i^2\boldsymbol{x}_{i*}\boldsymbol{x}_{i*}^T] = 0$ for each i.

Liang-Zeger standard errors

Next, let's consider the case of group-correlated errors. Suppose that the observations are *clustered*, with correlated errors among clusters but not between clusters. Suppose there are C clusters of observations, with the ith observation belonging to cluster $c(i) \in \{1, \ldots, C\}$. Suppose for the

sake of simplicity that the observations are ordered so that clusters are contiguous. Let $\widehat{\boldsymbol{\epsilon}}_c$ be the vector of residuals in cluster c, so that $\widehat{\boldsymbol{\epsilon}} = (\widehat{\boldsymbol{\epsilon}}_1, \ldots, \widehat{\boldsymbol{\epsilon}}_C)$. Then, the true covariance matrix is $\boldsymbol{\Sigma} = \text{block-diag}(\boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_C)$ for some positive definite $\boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_C$. The Liang-Zeger estimator is then defined by (14.5), with

$$\widehat{\boldsymbol{\Sigma}} \equiv \text{block-diag}(\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{1}}, \ldots, \widehat{\boldsymbol{\Sigma}}_{\boldsymbol{C}}), \quad \text{where} \quad \widehat{\boldsymbol{\Sigma}}_{\boldsymbol{c}} \equiv \widehat{\boldsymbol{\epsilon}}_c \widehat{\boldsymbol{\epsilon}}_c^T.$$

Note that the Liang-Zeger estimator is a generalization of the Huber-White estimator. Its justification is similar as well: while each $\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{c}}$ is a poor estimator, the resulting estimate of the (averaged) quantity $\boldsymbol{X}^T \widehat{\boldsymbol{\Sigma}} \boldsymbol{X}$ is not bad as long as the number of clusters is large. Liang-Zeger standard errors are referred to as "clustered standard errors" in the econometrics community. It is recommended to employ clustered standard errors even when using cluster-level fixed effects, in order to capture remaining within-cluster correlations.

Newey-West standard errors

Finally, consider the case when our observations i have a temporal structure, and we believe there to be nontrivial correlations between ϵ_{i1} and ϵ_{i2} for $|i1 - i2| \leq L$. Then, a natural extension of the Huber-White estimate of $\boldsymbol{\Sigma}$ is $\widehat{\Sigma}_{i1,i2} = \widehat{\epsilon}_{i1}\widehat{\epsilon}_{i2}$ for each pair $(i1, i2)$ such that $|i1 - i2| \leq L$. Unfortunately, this is not guaranteed to give a positive semidefinite matrix $\widehat{\boldsymbol{\Sigma}}$. Therefore, Newey and West proposed a slightly modified estimator:

$$\widehat{\boldsymbol{\Sigma}}_{i1,i2} = \max\left(0, 1 - \frac{|i1 - i2|}{L+1}\right) \widehat{\epsilon}_{i1}\widehat{\epsilon}_{i2}.$$

This estimator shrinks the off-diagonal estimates $\widehat{\epsilon}_{i1}\widehat{\epsilon}_{i2}$ based on their distance to the diagonal. It can be shown that this modification restores positive semidefiniteness of $\widehat{\boldsymbol{\Sigma}}$.

14.3 Inference based on an approximate covariance matrix

Whether based on the relations (14.4) or (14.6), we end up with an estimator $\widehat{\boldsymbol{\beta}}$ and an approximate covariance matrix $\widehat{\boldsymbol{\Omega}}$, so that

$$\widehat{\boldsymbol{\beta}} \dot{\sim} N(\boldsymbol{\beta}, \widehat{\boldsymbol{\Omega}}).$$

This allows us to construct confidence intervals and hypothesis tests for each β_j, by simply replacing $\text{SE}(\beta_j)$ with $\sqrt{\widehat{\Omega}_{jj}}$. For contrasts and prediction intervals, we can use the fact that $\boldsymbol{c}^T\boldsymbol{\beta} \dot{\sim} N(\boldsymbol{c}^T\boldsymbol{\beta}, \boldsymbol{c}^T\widehat{\boldsymbol{\Omega}}\boldsymbol{c})$, so that $\text{CE}(\boldsymbol{c}^T\boldsymbol{\beta}) = \sqrt{\boldsymbol{c}^T\widehat{\boldsymbol{\Omega}}\boldsymbol{c}}$. It is less obvious how to use the matrix $\widehat{\boldsymbol{\Omega}}$ to test the hypothesis $H_0 : \boldsymbol{\beta}_S = \boldsymbol{0}$. To this end, we can use a Wald test (we will discuss Wald tests in more detail in Unit 4). The Wald test statistic is

$$W = \widehat{\boldsymbol{\beta}}_S^T (\widehat{\boldsymbol{\Omega}}_{S,S})^{-1} \widehat{\boldsymbol{\beta}}_S,$$

which is asymptotically distributed as $\chi^2_{|S|}$ under the null hypothesis. This is based on the following result.

Lemma 14.1. *Let $\boldsymbol{Z} \sim N(0, \boldsymbol{\Sigma})$ be a d-dimensional random vector, with $\boldsymbol{\Sigma}$ invertible. Then,*

$$\boldsymbol{Z}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{Z} \sim \chi^2_d.$$

This gives us the test

$$\phi_{\text{Wald}}(\boldsymbol{X}, \boldsymbol{y}) = \mathbb{I}\left\{\widehat{\boldsymbol{\beta}}_S^T (\widehat{\boldsymbol{\Omega}}_{S,S})^{-1} \widehat{\boldsymbol{\beta}}_S > \chi^2_{|S|}(1-\alpha)\right\}. \tag{14.7}$$

As it turns out, under the usual linear model assumptions, this test is asymptotically equivalent to the usual F-test for the hypothesis $H_0 : \boldsymbol{\beta}_S = \mathbf{0}$.

Proposition 14.1. *The homoskedasticity-based F-statistic for the null hypothesis $H_0 : \boldsymbol{\beta}_S = \mathbf{0}$ can be expressed as*

$$F = \widehat{\boldsymbol{\beta}}_S^T (\widehat{\boldsymbol{\Omega}}_{S,S})^{-1} \widehat{\boldsymbol{\beta}}_S / |S|,$$

allowing us to rewrite the Wald test as

$$\phi_{Wald}(\boldsymbol{X}, \boldsymbol{y}) = \mathbb{I}\left\{F > \tfrac{1}{|S|}\chi^2_{|S|}(1-\alpha)\right\}.$$

Since $F_{|S|,n-p} \xrightarrow{d} \frac{1}{|S|}\chi^2_{|S|}$ as $n \to \infty$, it follows that the F-test and the Wald test are asymptotically equivalent.

Proof. Recall from Chapter 8 that the F-test statistic can be expressed as

$$F = \frac{\|(\boldsymbol{H} - \boldsymbol{H}_{\text{-}S})\boldsymbol{y}\|^2/|S|}{\|(\boldsymbol{I} - \boldsymbol{H})\boldsymbol{y}\|^2/(n-p)} = \frac{\|(\boldsymbol{H} - \boldsymbol{H}_{\text{-}S})\boldsymbol{y}\|^2/|S|}{\widehat{\sigma}^2},$$

where $\boldsymbol{H} - \boldsymbol{H}_{\text{-}S}$ is the projection matrix onto $C(\boldsymbol{X}^{\perp}_{*,S})$. Now, let $\widehat{\boldsymbol{\beta}}$ be the least squares estimate in the regression of $\boldsymbol{y}$ on $\boldsymbol{X}$. Then, we have

$$\begin{aligned}(\boldsymbol{H} - \boldsymbol{H}_{\text{-}S})\boldsymbol{y} &= (\boldsymbol{H} - \boldsymbol{H}_{\text{-}S})(\boldsymbol{X}_{*,S}\widehat{\boldsymbol{\beta}}_S + \boldsymbol{X}_{*,\text{-}S}\widehat{\boldsymbol{\beta}}_{\text{-}S} + \widehat{\boldsymbol{\epsilon}}) \\ &= (\boldsymbol{H} - \boldsymbol{H}_{\text{-}S})\boldsymbol{X}_{*,S}\widehat{\boldsymbol{\beta}}_S \\ &= \boldsymbol{X}^{\perp}_{*,S}\widehat{\boldsymbol{\beta}}_S.\end{aligned}$$

Therefore, we have

$$\|(\boldsymbol{H} - \boldsymbol{H}_{\text{-}S})\boldsymbol{y}\|^2 = \widehat{\boldsymbol{\beta}}_S^T (\boldsymbol{X}^{\perp}_{*,S})^T \boldsymbol{X}^{\perp}_{*,S} \widehat{\boldsymbol{\beta}}_S.$$

Next, we claim that

$$(\boldsymbol{X}^{\perp}_{*,S})^T \boldsymbol{X}^{\perp}_{*,S} = \{[(\boldsymbol{X}^T\boldsymbol{X})^{-1}]_{S,S}]\}^{-1}. \tag{14.8}$$

To see this, note that

$$\widehat{\boldsymbol{\beta}}_S \sim N(\boldsymbol{\beta}_S, \sigma^2[(\boldsymbol{X}^T\boldsymbol{X})^{-1}]_{S,S}). \tag{14.9}$$

On the other hand, since $\widehat{\boldsymbol{\beta}}_S$ can be obtained by regressing $\boldsymbol{y}$ onto $\boldsymbol{X}^{\perp}_{*,S}$, we also have that

$$\widehat{\boldsymbol{\beta}}_S \sim N(\boldsymbol{\beta}_S, \sigma^2[(\boldsymbol{X}^{\perp}_{*,S})^T \boldsymbol{X}^{\perp}_{*,S}]^{-1}). \tag{14.10}$$

Combining 14.9 and 14.10, we verify the claimed relationship 14.8. Therefore, we have

$$F = \frac{\widehat{\boldsymbol{\beta}}_S^T[(\boldsymbol{X}^T\boldsymbol{X})^{-1}]_{S,S}\widehat{\boldsymbol{\beta}}_S/|S|}{\widehat{\sigma}^2}.$$

Recalling that $\widehat{\boldsymbol{\Omega}} = \widehat{\sigma}^2(\boldsymbol{X}^T\boldsymbol{X})^{-1}$, we find that

$$F = \widehat{\boldsymbol{\beta}}_S^T \left(\widehat{\boldsymbol{\Omega}}_{S,S}\right)^{-1} \widehat{\boldsymbol{\beta}}_S/|S|,$$

as desired.

□

Chapter 15

The bootstrap

15.1 Introduction to the bootstrap

The bootstrap can be used for either confidence interval construction or hypothesis testing, with confidence interval construction being a much more common application. For this reason, we will focus on confidence interval construction in remainder of this section as well as in Section 15.2 and Section 15.3. We will discuss bootstrap hypothesis testing in Section 15.4.

15.1.1 Usual inference paradigm

We typically carry out linear model inference for β_j by approximating the sampling distribution of $\widehat{\beta}_j$, or a derivative thereof, such as the t-statistic. Under the standard linear model assumptions, we have

$$g(\widehat{\boldsymbol{\beta}}, \boldsymbol{\beta}) \equiv \widehat{\beta}_j - \beta_j \sim N(0, \sigma^2[(\boldsymbol{X}^T\boldsymbol{X})^{-1}]_{jj}) \tag{15.1}$$

and

$$g(\widehat{\boldsymbol{\beta}}, \boldsymbol{\beta}) \equiv \frac{\widehat{\beta}_j - \beta_j}{\text{s.e.}(\widehat{\beta}_j)} \sim t_{n-p}. \tag{15.2}$$

Here, $g(\widehat{\boldsymbol{\beta}}, \boldsymbol{\beta})$ denotes a derivative quantity, whose distribution is the basis for inference. If the lower and upper quantiles $\mathbb{Q}_{\alpha/2}[g(\widehat{\boldsymbol{\beta}}, \boldsymbol{\beta})]$ and $\mathbb{Q}_{1-\alpha/2}[g(\widehat{\boldsymbol{\beta}}, \boldsymbol{\beta})]$ are known, then we can construct a $(1-\alpha)$ confidence interval for β_j as

$$\text{CI}(\beta_j) \equiv \{\beta_j : g(\widehat{\boldsymbol{\beta}}, \boldsymbol{\beta}) \in [\mathbb{Q}_{\alpha/2}[g(\widehat{\boldsymbol{\beta}}, \boldsymbol{\beta})], \mathbb{Q}_{1-\alpha/2}[g(\widehat{\boldsymbol{\beta}}, \boldsymbol{\beta})]]\}. \tag{15.3}$$

Under model misspecification, the distributions on the right-hand sides of equations (15.1) and (15.2) may no longer be valid.

15.1.2 Bootstrap inference paradigm

The *bootstrap* is an approach to more robust inference that obtains such sampling distributions by a technique known as *resampling*. The core idea of the bootstrap is to use the data to construct an approximation to the data-generating distribution and then to approximate the sampling distribution of any statistic by simulating from this approximate data-generating distribution. In more detail, the bootstrap paradigm is as follows:

Bootstrap paradigm to build confidence intervals for β_j

1. Use the data $(\boldsymbol{X}, \boldsymbol{y})$ to get an approximation $\widehat{F}$ for the data distribution F.
2. For each $b = 1, \ldots, B$,
   i) Sample a bootstrap dataset $(\boldsymbol{X}^{(b)}, \boldsymbol{y}^{(b)}) \sim \widehat{F}$;
   ii) Fit the least squares estimate $\widehat{\boldsymbol{\beta}}^{(b)}$ based on $(\boldsymbol{X}^{(b)}, \boldsymbol{y}^{(b)})$;
   iii) Construct a derivative quantity $g(\widehat{\boldsymbol{\beta}}^{(b)}, \widehat{\boldsymbol{\beta}})$, such as $\widehat{\beta}_j^{(b)} - \widehat{\beta}_j$.
3. Extract the empirical $\alpha/2$ and $1-\alpha/2$ quantiles of the derivative quantity $g(\widehat{\boldsymbol{\beta}}^{(b)}, \widehat{\boldsymbol{\beta}})$, denoted $\mathbb{Q}_{\alpha/2}[g(\widehat{\boldsymbol{\beta}}^{(b)}, \widehat{\boldsymbol{\beta}})]$ and $\mathbb{Q}_{1-\alpha/2}[g(\widehat{\boldsymbol{\beta}}^{(b)}, \widehat{\boldsymbol{\beta}})]$.
4. Construct a $(1-\alpha)$ confidence interval for β_j by analogy with (15.3):

$$\begin{aligned} &\mathrm{CI}^{\mathrm{boot}}(\beta_j) \\ &\quad \equiv \{\beta_j : g(\widehat{\boldsymbol{\beta}}, \boldsymbol{\beta}) \in [\mathbb{Q}_{\alpha/2}[g(\widehat{\boldsymbol{\beta}}^{(b)}, \widehat{\boldsymbol{\beta}})], \mathbb{Q}_{1-\alpha/2}[g(\widehat{\boldsymbol{\beta}}^{(b)}, \widehat{\boldsymbol{\beta}})]]\}. \end{aligned} \tag{15.4}$$

This approach, pioneered by Brad Efron in 1979, obviates the need for stringent assumptions and mathematical derivations to obtain limiting distributions, replacing these with added computation. The bootstrap is extremely flexible and can be adapted to apply in a variety of settings. Furthermore, bootstrap methods are typically more accurate than their asymptotic counterparts in finite samples. While the justification of the bootstrap is still asymptotic (requiring $\widehat{F}$ to approach F), the rate of convergence is often "second-order" $O(1/n)$ rather than the usual "first-order" $O(1/\sqrt{n})$ of standard asymptotic inference. This faster second-order convergence gives the bootstrap an advantage in finite samples.

15.1.3 Overview of bootstrap flavors

The bootstrap comes in a variety of flavors, dictated by the mechanism by which the data distribution F is learned (e.g. the *parametric*, *residual*, or *pairs* bootstraps), and the derivative quantity $g(\cdot, \cdot)$ on which inference is based (e.g. the *empirical bootstrap* and the *bootstrap-t*). These two sets of flavors can be mixed and matched.

15.2 Derivative quantities on which to base inference

15.2.1 The empirical bootstrap

The *empirical bootstrap*, the most common choice, is based on the quantity

$$g(\widehat{\boldsymbol{\beta}}, \boldsymbol{\beta}) = \widehat{\beta}_j - \beta_j,$$

We can derive that if

$$\mathbb{P}\left[\widehat{\beta}_j - \beta_j \in \left[\mathbb{Q}_{\alpha/2}[\widehat{\beta}_j^{(b)} - \widehat{\beta}_j], \mathbb{Q}_{1-\alpha/2}[\widehat{\beta}_j^{(b)} - \widehat{\beta}_j]\right]\right] \dot{\geq} 1-\alpha,$$

then

$$\mathbb{P}\left[\beta_j \in \left[\widehat{\beta}_j - \mathbb{Q}_{1-\alpha/2}[\widehat{\beta}_j^{(b)} - \widehat{\beta}_j], \widehat{\beta}_j - \mathbb{Q}_{\alpha/2}[\widehat{\beta}_j^{(b)} - \widehat{\beta}_j]\right]\right] \dot{\geq} 1-\alpha.$$

For this reason, we define

$$\mathrm{CI}^{\mathrm{boot}}(\beta_j) \equiv \left[\widehat{\beta}_j - \mathbb{Q}_{1-\alpha/2}[\widehat{\beta}_j^{(b)} - \widehat{\beta}_j], \widehat{\beta}_j - \mathbb{Q}_{\alpha/2}[\widehat{\beta}_j^{(b)} - \widehat{\beta}_j]\right].$$

! The percentile bootstrap

A commonly used alternative to the empirical bootstrap is the *percentile bootstrap*, defined by

$$\mathrm{CI}^{\mathrm{boot}}(\beta_j) \equiv \left[\mathbb{Q}_{\alpha/2}[\widehat{\beta}_j^{(b)}], \mathbb{Q}_{1-\alpha/2}[\widehat{\beta}_j^{(b)}]\right]. \tag{15.5}$$

Here, the resampling distribution of $\widehat{\beta}_j$ is used directly to construct the confidence interval. However, this approach does not fall within the bootstrap paradigm described above, and in particular, the formula (15.5) is not a special case of (15.4). The formula (15.5) can be viewed as seeking an interval within which $\widehat{\beta}_j$ (rather than β_j itself) falls with 95% probability. The percentile bootstrap is only justified when the distribution of $\widehat{\beta}_j^{(b)}$ is symmetric about $\widehat{\beta}_j$, in which case it coincides with the empirical bootstrap.

15.2.2 The bootstrap-t method

A weakness of the empirical bootstrap is that the quantity $g(\widehat{\boldsymbol{\beta}}, \boldsymbol{\beta}) = \widehat{\beta}_j - \beta_j$ has distribution $N(0, \sigma^2[(X^TX)^{-1}]_{jj})$ (recalling equation (15.1)), *which depends on the nuisance parameter* σ^2. When we approximate this distribution by bootstrapping, we implicitly are substituting in an estimate of σ^2, which is itself subject to sampling variability. The empirical bootstrap does not account for this variability, because the distribution $\widehat{F}$ on which the estimate of σ^2 is based is held fixed throughout. To see this more clearly, consider the following example.

Example 15.1 (Non-pivotality in the normal mean problem)**.** Suppose that $\boldsymbol{y} \sim N(\mathbf{1}\beta_0, \sigma^2\boldsymbol{I})$, and the goal is to construct a confidence interval for β_0. Defining $\widehat{\beta}_0 \equiv \bar{y}$ and $\widehat{\sigma}^2 \equiv \frac{1}{n-1}\sum_{i=1}^n (y_i - \bar{y})^2$, consider the empirical bootstrap based on resampling $\boldsymbol{y}^{(b)} \sim N(\mathbf{1}\widehat{\beta}_0, \widehat{\sigma}^2\boldsymbol{I})$. In this case, we will find that

$$\widehat{\beta}_0^{(b)} - \widehat{\beta}_0 = \bar{y}^{(b)} - \bar{y} \sim N(0, \widehat{\sigma}^2/n),$$

which will give rise to the bootstrap confidence interval

$$\mathrm{CI}^{\mathrm{boot}}(\beta_0) = \widehat{\beta}_0 \pm z_{1-\alpha/2}\tfrac{1}{\sqrt{n}}\widehat{\sigma}.$$

The uncertainty in the estimate $\widehat{\sigma}^2$ is not accounted for. We know from Unit 2 that, if the usual linear model assumptions are satisfied, we could account for this uncertainty by using a t-distribution instead of a normal distribution.

This issue can be addressed by bootstrapping a *pivotal* quantity $g(\widehat{\boldsymbol{\beta}}, \boldsymbol{\beta})$, i.e., a quantity whose distribution does not depend on unknown parameters (at least under standard assumptions). In the context of the linear model, the t-statistic (15.2) is pivotal. Bootstrapping the t-statistic, called the *bootstrap-t method*, is therefore a way to account for the uncertainty in the estimate of σ^2. To derive the bootstrap-t method, we approximate

$$\mathbb{P}\left[\frac{\widehat{\beta}_j - \beta_j}{\mathrm{s.e.}(\widehat{\beta}_j)} \in \left[\mathbb{Q}_{\alpha/2}\left(\frac{\widehat{\beta}_j^{(b)} - \widehat{\beta}_j}{\mathrm{s.e.}(\widehat{\beta}_j^{(b)})}\right), \mathbb{Q}_{1-\alpha/2}\left(\frac{\widehat{\beta}_j^{(b)} - \widehat{\beta}_j}{\mathrm{s.e.}(\widehat{\beta}_j^{(b)})}\right)\right]\right] \dot{\geq} 1 - \alpha,$$

which justifies the bootstrap-t confidence interval

$$\begin{aligned}&\mathrm{CI}^{\mathrm{boot}}(\beta_j)\\ &\equiv \left[\widehat{\beta}_j - \mathrm{s.e.}(\widehat{\beta}_j)\mathbb{Q}_{1-\alpha/2}\left(\frac{\widehat{\beta}_j^{(b)} - \widehat{\beta}_j}{\mathrm{s.e.}(\widehat{\beta}_j^{(b)})}\right), \widehat{\beta}_j - \mathrm{s.e.}(\widehat{\beta}_j)\mathbb{Q}_{\alpha/2}\left(\frac{\widehat{\beta}_j^{(b)} - \widehat{\beta}_j}{\mathrm{s.e.}(\widehat{\beta}_j^{(b)})}\right)\right].\end{aligned}$$

15.3 Techniques for learning the data distribution

The methods for learning the data distribution lie on a spectrum based on how flexibly they model this distribution. Methods with less flexibility are more stable (i.e. less variable) but less robust. On the other hand, methods with more flexibility are more robust but less stable. The following table shows the methods in increasing order of flexibility, including which types of model misspecification they are robust to.

Method	Non-normality	Hetero-skedasticity	Group correlation	Temporal correlation
Parametric	No	No	No	No
Residual	Yes	No	No	No
Pairs	Yes	Yes	No	No
Clustered	Yes	Yes	Yes	No
Moving Blocks	Yes	Yes	No	Yes

We will present these methods in the same order, starting from the parametric bootstrap.

15.3.1 The parametric bootstrap

The parametric bootstrap proceeds by specifying a parametric model for the data $(\boldsymbol{X}, \boldsymbol{y})$, such as the one from Unit 2:

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim N(0, \sigma^2 \boldsymbol{I}). \tag{15.6}$$

The model matrix $\boldsymbol{X}$ is kept fixed and only the distribution of $\boldsymbol{y}$ (conditionally on $\boldsymbol{X}$) is modeled. These parameters can be fit by maximum likelihood, as usual. Then, the bootstrapped datasets can be generated as follows:

$$\boldsymbol{X}^{(b)} = \boldsymbol{X}; \quad \boldsymbol{y}^{(b)} \sim N(\boldsymbol{X}\widehat{\boldsymbol{\beta}}, \widehat{\sigma}^2 \boldsymbol{I}).$$

In the context of the linear model, the parametric bootstrap is rarely used, particularly in the context of the standard parametric form (15.6) and the least squares estimator $\widehat{\boldsymbol{\beta}}$. Indeed, it offers no robustness to model misspecification and the distribution of the least squares estimator is known exactly, so there is no need to approximate it using resampling. In contexts beyond linear models or when estimators beyond the least squares estimator are of interest, the parametric bootstrap is useful because it can be used the approximate the distributions of analytically intractable estimators, substituting computation for math.

15.3.2 The residual bootstrap

The residual bootstrap is based on a *parametric* model for $\mathbb{E}[\boldsymbol{y} \mid \boldsymbol{X}]$ and a *nonparametric* model for the noise terms. In particular, suppose that

$$y_i = \boldsymbol{x}_{i*}^T \boldsymbol{\beta} + \epsilon_i, \quad \epsilon_i \overset{\text{i.i.d.}}{\sim} G \quad \text{for } i = 1, \ldots, n, \tag{15.7}$$

where G is an unknown distribution without a parametric form. Then, the data-generating distribution F is specified by the pair $(\boldsymbol{\beta}, G)$. As with the parametric bootstrap, the model matrix $\boldsymbol{X}$ is kept fixed and only the distribution of $\boldsymbol{y}$ (conditionally on $\boldsymbol{X}$) is modeled. The parameter

vector $\boldsymbol{\beta}$ can be fit by least squares, as usual. Then, the distribution of the noise terms ϵ_i can be estimated by the empirical distribution of the residuals $\widehat{\epsilon}_i$:

$$\widehat{G} = \frac{1}{n}\sum_{i=1}^{n} \delta_{\widehat{\epsilon}_i},$$

where $\delta_{\widehat{\epsilon}_i}$ is the Dirac delta function at $\widehat{\epsilon}_i$. The bootstrapped datasets can then be generated as follows:

$$x_{i*}^{(b)} = x_{i*}; \quad y_i^{(b)} = \boldsymbol{x}_{i*}^T\widehat{\boldsymbol{\beta}} + \epsilon_i^{(b)}, \quad \epsilon_i^{(b)} \overset{\text{i.i.d.}}{\sim} \widehat{G} \quad \text{for } i = 1, \ldots, n.$$

Note that the sampling of $\epsilon_i^{(b)} \overset{\text{i.i.d.}}{\sim} \widehat{G}$ is equivalent to sampling with replacement from the residuals $\widehat{\epsilon}_i$. By avoiding modeling ϵ_i as normal, the residual bootstrap is robust to non-normality. However, it is not robust to heteroskedasticity or correlated errors, because it models the ϵ_i as i.i.d. from some distribution.

15.3.3 Pairs bootstrap

Weakening the assumptions further, let us assume simply that

$$(\boldsymbol{x}_{i*}, y_i) \overset{\text{i.i.d.}}{\sim} F$$

for some joint distribution F. Unlike the parametric and residual bootstraps, the pairs bootstrap treats the predictors $\boldsymbol{X}$ as random rather than fixed. We can then fit $\widehat{F}$ as the empirical distribution of the data:

$$\widehat{F} = \frac{1}{n}\sum_{i=1}^{n} \delta_{(\boldsymbol{x}_{i*}, y_i)}.$$

The bootstrapped datasets can then be generated as follows:

$$(\boldsymbol{x}_{i*}^{(b)}, y_i^{(b)}) \overset{\text{i.i.d.}}{\sim} \widehat{F} \quad \text{for } i = 1, \ldots, n.$$

Note that sampling the pairs $(\boldsymbol{x}_{i*}^{(b)}, y_i^{(b)})$ from $\widehat{F}$ is equivalent to sampling with replacement from the rows of the original data. The benefit of the pairs bootstrap is that it does not assume homoskedasticity since the error variance is allowed to depend on $\boldsymbol{x}_{i*}$. Therefore, the pairs bootstrap addresses both non-normality and heteroskedasticity, though it does not address correlated errors (though variants of the pairs bootstrap do; see below). Note that the pairs bootstrap does not even assume that $\mathbb{E}[y_i] = \boldsymbol{x}_{i*}^T\boldsymbol{\beta}$ for some $\boldsymbol{\beta}$. However, in the presence of model bias, it is unclear for what parameters we are even doing inference. While the pairs bootstrap assumes less than the residual bootstrap, it may be somewhat less efficient in the case when the assumptions of the latter are met. However, the pairs bootstrap is the most commonly applied flavor of the bootstrap.

15.3.4 Clustered bootstrap

In the presence of clustered errors, the pairs bootstrap can be modified to the *clustered bootstrap*. The distributional assumption underlying the clustered bootstrap is the following:

$$\{(\boldsymbol{x}_{i*}, y_i) : c(i) = c\} \overset{\text{i.i.d.}}{\sim} F \quad \text{for } c = 1, \ldots, C, \tag{15.8}$$

where $c(i)$ is the cluster to which observation i belongs. Therefore, entire clusters are modeled as coming i.i.d. from some distribution across clusters. As with the pairs bootstrap, this distribution is estimated by the empirical distribution of the data, and resampling from this distribution amounts

to *sampling entire clusters* (rather than individual observations) from the original data, with replacement. This kind of resampling preserves the joint correlation structure within clusters. Note that the clustered bootstrap is a special case of the pairs bootstrap, where each pair forms its own cluster.

15.3.5 Moving blocks bootstrap

In the case of temporally (or spatially) correlated errors, the pairs bootstrap can be modified to the *moving blocks bootstrap*. The distributional assumption underlying the moving blocks bootstrap is the same as that of the clustered bootstrap (15.8), except the clusters are defined as contiguous blocks of observations. The distribution across blocks is fit as the empirical distribution of all blocks of a given size, and resampling from this distribution amounts to *sampling entire blocks* (rather than individual observations) from the original data, with replacement. This kind of resampling preserves the joint correlation structure within temporal or spatial blocks, though it ignores correlations across boundaries of these blocks. Like the clustered bootstrap, the moving blocks bootstrap is a special case of the pairs bootstrap.

15.4 Bootstrap hypothesis testing

The bootstrap inference paradigm described in Section 15.1.2 is primarily for constructing confidence intervals. For one-dimensional quantities like β_j, confidence intervals can be used to perform hypothesis tests via duality. However, it is more challenging to use the bootstrap to create confidence regions for multi-dimensional quantities like $\boldsymbol{\beta}_S$. Nevertheless, in some cases the bootstrap paradigm can be adapted to perform hypothesis tests directly.

15.4.1 Bootstrap testing paradigm

Bootstrap paradigm to test $H_0 : \boldsymbol{\beta}_S = 0$

1. Compute a test statistic $T(\boldsymbol{X}, \boldsymbol{y})$ measuring the evidence against H_0.
2. Use the data $(\boldsymbol{X}, \boldsymbol{y})$ to get an approximation $\widehat{F}$ for the data distribution F.
3. Find a null data distribution $\widehat{F}_0$ by "projecting" $\widehat{F}$ onto H_0.
4. For each $b = 1, \ldots, B$,
   i) Sample a null bootstrap dataset $(\boldsymbol{X}^{(b)}, \boldsymbol{y}^{(b)}) \sim \widehat{F}_0$;
   ii) Evaluate the test statistic on the resampled data to get $T(\boldsymbol{X}^{(b)}, \boldsymbol{y}^{(b)})$.
5. Evaluate the empirical quantile $\mathbb{Q}_{1-\alpha}(\{T(\boldsymbol{X}^{(b)}, \boldsymbol{y}^{(b)})\}_{b=1}^{B})$.
6. Reject if $T(\boldsymbol{X}, \boldsymbol{y}) > \mathbb{Q}_{1-\alpha}(\{T(\boldsymbol{X}^{(b)}, \boldsymbol{y}^{(b)})\}_{b=1}^{B})$.

Here, the key new step is the third, in which a null data distribution $\widehat{F}_0$ is derived from the approximate data distribution $\widehat{F}$. The challenge is that, depending on the form of $\widehat{F}$, this may or may not be possible. In fact, obtaining $\widehat{F}_0$ from $\widehat{F}$ is easily done whenever the model for the data involves a parameter vector $\boldsymbol{\beta}$, as is the case for the parametric and residual bootstraps (see the next section for more detail on the latter). In this case, $\widehat{F}_0$ can be obtained from $\widehat{F}$ by setting the coefficients $\boldsymbol{\beta}_S$ to zero. On the other hand, for the pairs bootstrap and its variants, it is not clear how to obtain $\widehat{F}_0$ from $\widehat{F}$. Finally, note that the bootstrap testing paradigm is in principle compatible with any test statistic T. A popular choice for T is the F-statistic for $H_0 : \boldsymbol{\beta}_S = 0$ from Unit 2.

15.4.2 Testing with the residual bootstrap

A commonly used bootstrap flavor for hypothesis testing is the *residual bootstrap*. Recalling the data-generating model (15.7), suppose $\widehat{F} = (\widehat{\boldsymbol{\beta}}, \widehat{G})$ is the fitted model. Then, we can define $\widehat{F}_0 = ((\mathbf{0}, \widehat{\boldsymbol{\beta}}_{\text{-}S}), \widehat{G})$. Therefore, the bootstrapped null data are drawn from the following distribution:

$$x_{i*}^{(b)} = x_{i*}; \quad y_i^{(b)} = \boldsymbol{x}_{i,\text{-}S}^T \widehat{\boldsymbol{\beta}}_{\text{-}S} + \epsilon_i^{(b)}, \quad \epsilon_i^{(b)} \overset{\text{i.i.d.}}{\sim} \widehat{G}.$$

As before, the bootstrapped residuals $\epsilon_i^{(b)}$ are sampled with replacement from the set of original residuals.

Chapter 16

The permutation test

Consider a linear regression model with intercept:

$$y_i = \beta_0 + \boldsymbol{x}_{i,\text{-}0}^T \boldsymbol{\beta}_{\text{-}0} + \epsilon_i, \quad \epsilon_i \overset{\text{i.i.d.}}{\sim} F, \quad i = 1, \ldots, n, \tag{16.1}$$

where F is an unknown distribution. Suppose we wish to test the null hypothesis that none of the variables in $\boldsymbol{x}_{\text{-}0}$ are associated with y:

$$H_0 : \boldsymbol{\beta}_{\text{-}0} = \mathbf{0}. \tag{16.2}$$

Furthermore, suppose the sample size n is small enough that neither asymptotic inference nor the bootstrap are reliable. In this case, we can use the *permutation test*, which controls Type-I error for any sample size n.

16.1 General formulation of the permutation test

Note that the null hypothesis can be formulated as follows:

$$H_0 : y_i \overset{\text{i.i.d.}}{\sim} G \quad \text{for some distribution } G.$$

If we view the model matrix $\boldsymbol{X}$ as random, then we can also formulate H_0 as an independence null hypothesis:

$$H_0 : \boldsymbol{x}_{\text{-}0} \perp\!\!\!\perp y.$$

Both of these reformulations suggest that, under the null hypothesis, the null distribution of the data $(\boldsymbol{X}, \boldsymbol{y})$ is invariant to permutations of $\boldsymbol{y}$, while keeping $\boldsymbol{X}$ fixed. In other words,

$$(\boldsymbol{X}, \pi\boldsymbol{y}) \overset{d}{=} (\boldsymbol{X}, \boldsymbol{y}), \quad \text{for all } \pi \in \mathcal{S}_n, \tag{16.3}$$

where $\mathcal{S}_n$ is the group of all permutations of $\{1, \ldots, n\}$ and $\pi\boldsymbol{y}$ is the permuted response vector. Therefore, we can use permuted instances of the data to approximate the null distribution of any test statistic under H_0. There are two instances of the permutation test: one based on the entire group $\mathcal{S}_n$ and the other based on a random sample of $\mathcal{S}_n$.

16.1.1 Permutation test based on the entire permutation group

Consider any test statistic $T : (\boldsymbol{X}, \boldsymbol{y}) \mapsto \mathbb{R}$. For example, this may be the usual F-statistic for testing the hypothesis (16.2) in the model (16.1). Then, the permutation test based on the entire permutation group is as follows:

Permutation test based on the entire permutation group

1. Compute the observed value of the test statistic $T(\boldsymbol{X}, \boldsymbol{y})$.
2. For each $\pi \in \mathcal{S}_n$, compute the test statistic on the permuted data, $T(\boldsymbol{X}, \pi\boldsymbol{y})$.
3. Compute the quantile $\mathbb{Q}_{1-\alpha}[\{T(\boldsymbol{X}, \pi\boldsymbol{y}) : \pi \in \mathcal{S}_n\}]$.
4. Reject if $T(\boldsymbol{X}, \boldsymbol{y}) > \mathbb{Q}_{1-\alpha}[\{T(\boldsymbol{X}, \pi\boldsymbol{y}) : \pi \in \mathcal{S}_n\}]$.

As claimed at the outset of this chapter, this test has non-asymptotic Type-I error control.

Theorem 16.1. *For any n, the permutation test based on the entire permutation group has Type-I error at most α for testing the null hypothesis $H_0 : \boldsymbol{\beta}_{\text{-}0} = \mathbf{0}$.*

Proof. Suppose H_0 holds. Let $\tau \in \mathcal{S}_n$. Then, by the permutation invariance property (16.3), we have

$$\begin{aligned}
&\mathbb{P}[T(\boldsymbol{X}, \boldsymbol{y}) > \mathbb{Q}_{1-\alpha}[\{T(\boldsymbol{X}, \pi\boldsymbol{y}) : \pi \in \mathcal{S}_n\}]] \\
&\quad = \mathbb{P}[T(\boldsymbol{X}, \tau(\tau^{-1}\boldsymbol{y})) > \mathbb{Q}_{1-\alpha}[\{T(\boldsymbol{X}, \pi\tau(\tau^{-1}\boldsymbol{y})) : \pi \in \mathcal{S}_n\}]] \\
&\quad = \mathbb{P}[T(\boldsymbol{X}, \tau\boldsymbol{y}) > \mathbb{Q}_{1-\alpha}[\{T(\boldsymbol{X}, \pi\tau\boldsymbol{y}) : \pi \in \mathcal{S}_n\}]] \\
&\quad = \mathbb{P}[T(\boldsymbol{X}, \tau\boldsymbol{y}) > \mathbb{Q}_{1-\alpha}[\{T(\boldsymbol{X}, \pi\boldsymbol{y}) : \pi \in \mathcal{S}_n\}]].
\end{aligned}$$

Therefore, the probability that $T(\boldsymbol{X}, \boldsymbol{y})$ exceeds the $(1-\alpha)$-quantile of the permutation distribution is the same as the probability that any other permuted test statistic exceeds the $(1-\alpha)$-quantile. Therefore, we have

$$\begin{aligned}
&\mathbb{P}[T(\boldsymbol{X}, \boldsymbol{y}) > \mathbb{Q}_{1-\alpha}[\{T(\boldsymbol{X}, \pi\boldsymbol{y}) : \pi \in \mathcal{S}_n\}]] \\
&\quad = \frac{1}{|\mathcal{S}_n|} \sum_{\tau \in \mathcal{S}_n} \mathbb{P}[T(\boldsymbol{X}, \tau\boldsymbol{y}) > \mathbb{Q}_{1-\alpha}[\{T(\boldsymbol{X}, \pi\boldsymbol{y}) : \pi \in \mathcal{S}_n\}]] \\
&\quad = \mathbb{E}\left[\frac{|\{\tau \in \mathcal{S}_n : T(\boldsymbol{X}, \tau\boldsymbol{y}) > \mathbb{Q}_{1-\alpha}[\{T(\boldsymbol{X}, \pi\boldsymbol{y}) : \pi \in \mathcal{S}_n\}\}|}{|\mathcal{S}_n|}\right] \\
&\quad \leq \alpha.
\end{aligned} \tag{16.4}$$

□

16.1.2 Permutation test based on a sample of the permutation group

The permutation test based on the entire permutation group is computationally infeasible for large n. Instead, we can use a random sample of the permutation group to approximate the null distribution of the test statistic.

Permutation test based on a sample of the permutation group

1. Compute the observed value of the test statistic $T(\boldsymbol{X}, \boldsymbol{y})$.
2. Draw a random sample $(\pi_1, \ldots, \pi_B)$ from $\mathcal{S}_n$.
3. For each $b = 1, \ldots, B$, compute the test statistic on the permuted data, $T(\boldsymbol{X}, \pi_b\boldsymbol{y})$.
4. Compute the quantile $\mathbb{Q}_{1-\alpha}[\{T(\boldsymbol{X}, \boldsymbol{y})\} \cup \{T(\boldsymbol{X}, \pi_b\boldsymbol{y}) : b = 1, \ldots, B\}]$.
5. Reject if $T(\boldsymbol{X}, \boldsymbol{y}) > \mathbb{Q}_{1-\alpha}[\{T(\boldsymbol{X}, \boldsymbol{y})\} \cup \{T(\boldsymbol{X}, \pi_b\boldsymbol{y}) : b = 1, \ldots, B\}]$.

This gives us not just an approximation to the permutation test based on the entire permutation group, but a finite-sample valid test in its own right. The inclusion of the original test statistic

$T(\boldsymbol{X}, \boldsymbol{y})$ in the quantile computation ensures that the test has finite-sample Type-I error control; see the exchangeability-based argument in the proof.

Theorem 16.2. *For any n, the permutation test based on a sample of the permutation group has Type-I error at most α for testing the null hypothesis $H_0 : \boldsymbol{\beta}_{\text{-}0} = \mathbf{0}$.*

Proof. Suppose H_0 holds. We claim that the $B + 1$ test statistics

$$\{T(\boldsymbol{X}, \boldsymbol{y})\} \cup \{T(\boldsymbol{X}, \pi_b \boldsymbol{y}) : b = 1, \ldots, B\}$$

are *exchangeable*, i.e., their joint distribution is independent of their ordering. To see that, let τ be a randomly sampled permutation from $\mathcal{S}_n$. Then, by the permutation invariance property (16.3), we have

$$\begin{aligned} &\{T(\boldsymbol{X}, \boldsymbol{y})\} \cup \{T(\boldsymbol{X}, \pi_b \boldsymbol{y}) : b = 1, \ldots, B\} \\ &\quad \overset{d}{=} \{T(\boldsymbol{X}, \tau \boldsymbol{y})\} \cup \{T(\boldsymbol{X}, \pi_b \tau \boldsymbol{y}) : b = 1, \ldots, B\}. \end{aligned}$$

It is not hard to see that $\{\tau, \pi_1 \tau, \ldots, \pi_B \tau\}$ is an i.i.d. sample from $\mathcal{S}_n$, from which the claimed exchangeability follows. From this exchangeability, we get that

$$\begin{aligned} &\mathbb{P}[T(\boldsymbol{X}, \boldsymbol{y}) > \mathbb{Q}_{1-\alpha}[\{T(\boldsymbol{X}, \boldsymbol{y})\} \cup \{T(\boldsymbol{X}, \pi_b \boldsymbol{y}) : b = 1, \ldots, B\}]] \\ &\quad = \mathbb{P}[T(\boldsymbol{X}, \pi_b \boldsymbol{y}) > \mathbb{Q}_{1-\alpha}[\{T(\boldsymbol{X}, \boldsymbol{y})\} \cup \{T(\boldsymbol{X}, \pi_b \boldsymbol{y}) : b = 1, \ldots, B\}]] \end{aligned}$$

for each $b = 1, \ldots, B$, from which Type-I error control follows by the same argument as in the derivation (16.4).

□

16.1.3 Obtaining p-values from permutation tests

In some cases, it is desirable to extract p-values from permutation tests, rather than just the decision to accept or reject at a fixed level α. The p-value is the smallest level at which the null hypothesis can be rejected, i.e., the probability under the permutation distribution of observing a test statistic at least as extreme as the observed test statistic. For permutation tests based on the full permutation group, the p-value can be computed as follows:

$$p = \frac{1}{|\mathcal{S}_n|} \sum_{\tau \in \mathcal{S}_n} \mathbb{I}\{T(\boldsymbol{X}, \tau \boldsymbol{y}) \geq T(\boldsymbol{X}, \boldsymbol{y})\}.$$

For permutation tests based on a sample of the permutation group, the p-value can be computed as

$$p = \frac{1}{B+1} \left(1 + \sum_{b=1}^{B} \mathbb{I}\{T(\boldsymbol{X}, \pi_b \boldsymbol{y}) \geq T(\boldsymbol{X}, \boldsymbol{y})\} \right). \tag{16.5}$$

> **Warning**
>
> A common mistake is to omit the "1+" term from the numerator and denominator of equation (16.5). These terms are essential for constructing a valid p-value. In particular, these terms prevent the p-value from being exactly zero.

16.2 Special case: Two-groups model

The most common application of the permutation test is to the two-groups model:

$$y_i = \beta_0 + \beta_1 x_{i,1} + \epsilon_i, \quad \text{where } x_{i,1} \in \{0, 1\}.$$

The goal here is to test whether the binary "treatment" variable has any effect on the response variable:

$$H_0 : \beta_1 = 0.$$

To make the two groups more explicit, we can write

$$\{y_i : x_{i,1} = 0\} \overset{\text{i.i.d.}}{\sim} G_0, \quad \{y_i : x_{i,1} = 1\} \overset{\text{i.i.d.}}{\sim} G_1,$$

and the null hypothesis can be reformulated as

$$H_0 : G_0 = G_1.$$

In this case, the permutation mechanism randomly reassigns observations to the two groups. A commonly used test statistic T used in conjunction with this test is the difference in means between the two groups. While the permutation test controls Type-I error exactly under the hypothesis that the two groups come from exactly the same distribution, we might want to test a weaker hypothesis that the two groups have the same mean. It turns out that, at least asymptotically, the permutation test controls Type-I error under this weaker null hypothesis if it is based on a *studentized* statistic, such as

$$T(\boldsymbol{X}, \boldsymbol{y}) = \frac{\bar{y}_1 - \bar{y}_0}{\sqrt{\frac{\hat{\sigma}_0^2}{n_0} + \frac{\hat{\sigma}_1^2}{n_1}}},$$

where $\bar{y}_0$ and $\bar{y}_1$ are the sample means of the two groups, $\hat{\sigma}_0^2$ and $\hat{\sigma}_1^2$ are the sample variances of the two groups, and n_0 and n_1 are the sample sizes of the two groups.

16.3 Permutation test versus bootstrap

The bootstrap and permutation are both resampling-based tests that use computation as a substitute for mathematical derivations of sampling distributions. Both methods have better finite-sample performance than their asymptotic counterparts. The bootstrap and the permutation test are typically considered primarily in the context of confidence interval construction and hypothesis testing, respectively, although the bootstrap can also be used for hypothesis testing in certain cases. The key difference is that the permutation test has valid Type-I error control in finite samples, while the bootstrap requires an asymptotic justification (even if the asymptotic convergence is faster than typical CLT-based asymptotics). Furthermore, the bootstrap is somewhat more versatile than the permutation test, as the latter is restricted to testing null hypotheses about all non-intercept coefficients.

Chapter 17

Robust estimation and inference

17.1 Drawback of squared error loss

Suppose that

$$y_i = \boldsymbol{x}_{i*}^T \boldsymbol{\beta} + \epsilon_i, \quad \epsilon_i \overset{\text{i.i.d.}}{\sim} G, \quad i = 1, \ldots, n, \tag{17.1}$$

for some distribution G. If the distribution G has heavy tails, then the residuals will contain outliers. Recall that the least squares estimate is defined as

$$\widehat{\boldsymbol{\beta}} \equiv \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} L(y_i - \boldsymbol{x}_{i*}^T \boldsymbol{\beta}), \quad \text{where} \quad L(d) \equiv \frac{1}{2} d^2.$$

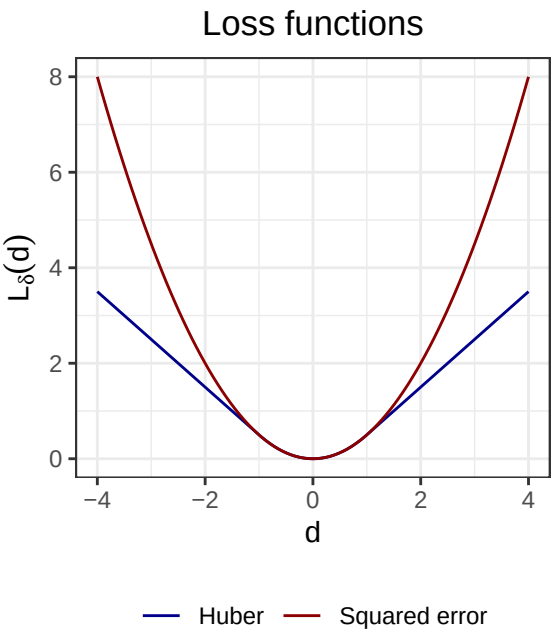
The squared error loss $L(d)$ is sensitive to outliers in the sense that a large value of $d_i \equiv y_i - \boldsymbol{x}_{i*}^T \boldsymbol{\beta}$ can have a significant impact on the loss function $L(d_i)$. The least squares estimate, as the minimizer of this loss function, is therefore sensitive to outliers.

17.2 The Huber loss

One way of addressing this challenge is to replace the squared error loss with a different loss that does not grow so quickly in $y_i - \boldsymbol{x}_{i*}^T \boldsymbol{\beta}$. A popular choice for such a loss function is the *Huber loss*:

$$L_\delta(d) = \begin{cases} \frac{1}{2} d^2, & \text{if } |d| \leq \delta; \\ \delta(|d| - \frac{1}{2}\delta), & \text{if } |d| > \delta. \end{cases}$$

This function is differentiable at the origin, like the squared error loss, but grows linearly as opposed to quadratically.



17.3 Scale estimation

The choice of $\delta > 0$ depends on the scale of the noise terms ϵ_i. Supposing that $\text{Var}[\epsilon_i] = \sigma^2$, a large residual is one where $|\epsilon_i/\sigma|$ is large. In this sense, δ should be on the same scale as σ. Of course, σ is unknown, so a first step towards obtaining a robust estimate is to estimate σ. While we would usually estimate σ based on the residuals from the least squares estimate, this approach is not robust to outliers. Instead, we can obtain a pilot estimate of the coefficients using the least absolute deviation (LAD) estimator, a scale-free and outlier-robust estimate:

$$\widehat{\boldsymbol{\beta}}^{\text{LAD}} \equiv \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} |y_i - \boldsymbol{x}_{i*}^T \boldsymbol{\beta}|.$$

Then, we can estimate σ from the residuals based on the LAD estimate. Since some of these residuals are outliers, it is better to avoid simply taking a sample variance. Instead, we can use the median absolute deviation (MAD) of the residuals, which is a robust estimate of the scale of the noise terms.

$$\widehat{\sigma} \equiv \frac{1}{0.675} \text{median} \left\{ |y_i - \boldsymbol{x}_{i*}^T \widehat{\boldsymbol{\beta}}^{\text{LAD}}| \right\}.$$

The purpose of the scaling factor of 0.675 is to connect the MAD to the standard deviation of the distribution of ϵ_i; it is derived based on the normal distribution.

> **Note**
>
> In principle, $\widehat{\boldsymbol{\beta}}^{\text{LAD}}$ could be used not just for estimation of σ but also for inference for $\boldsymbol{\beta}$ itself. However, the LAD estimator may be less efficient than the Huber estimator, so the latter estimator is usually preferred.

17.4 Huber estimation

With an estimate of σ in hand, we can use the Huber loss function to estimate $\boldsymbol{\beta}$:

$$\widehat{\boldsymbol{\beta}}^{\text{Huber}} \equiv \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} L_\delta \left(\frac{y_i - \boldsymbol{x}_{i*}^T \boldsymbol{\beta}}{\widehat{\sigma}} \right).$$

A common choice for δ is $\delta = 1.345$, which makes the Huber estimator 95% efficient relative to the least squares estimator under normality. The resulting $\widehat{\boldsymbol{\beta}}^{\text{Huber}}$ is an *M-estimator*. We can compute this estimator by taking a derivative of the objective and setting it to zero:

$$\sum_{i=1}^{n} L'_\delta \left(\frac{y_i - \boldsymbol{x}_{i*}^T \boldsymbol{\beta}}{\widehat{\sigma}} \right) \boldsymbol{x}_{i*} = 0.$$

Unlike least squares, this equation does not have a closed-form solution. However, it can be solved using an iterative algorithm. Under certain assumptions, the resulting estimator can be shown to be consistent.

17.5 Inference based on Huber estimates

We can construct hypothesis tests and confidence intervals using $\widehat{\boldsymbol{\beta}}^{\text{Huber}}$ based on the following result.

Theorem 17.1 (Asymptotic normality of Huber estimator (informal))**.** *Suppose the data $(\boldsymbol{X}, \boldsymbol{y})$ follow the model (17.1), with fixed design matrix $\boldsymbol{X}$. Then, if $\hat{\sigma}$ is a consistent estimator of σ and if the noise distribution G is symmetric, then*

$$\widehat{\boldsymbol{\beta}}^{Huber} \dot{\sim} N(\boldsymbol{\beta}, v(\boldsymbol{X}^T \boldsymbol{X})^{-1}), \quad \textit{where} \quad v \equiv \sigma^2 \frac{\mathbb{E}[L'_\delta(\epsilon_i/\sigma)^2]}{\mathbb{E}[L''_\delta(\epsilon_i/\sigma)]^2}.$$

Letting $\hat{\epsilon}_i \equiv y_i - \boldsymbol{x}_{i*}^T \widehat{\boldsymbol{\beta}}^{Huber}$, we can estimate v via

$$\widehat{v} \equiv \hat{\sigma}^2 \frac{\frac{1}{n}\sum_{i=1}^{n} L'_\delta(\hat{\epsilon}_i/\hat{\sigma})^2}{\left(\frac{1}{n}\sum_{i=1}^{n} L''_\delta(\hat{\epsilon}_i/\hat{\sigma})\right)^2}.$$

Under appropriate regularity conditions, $\widehat{v}$ is a consistent estimator of v, so that

$$\widehat{\boldsymbol{\beta}}^{Huber} \dot{\sim} N(\boldsymbol{\beta}, \hat{v}(\boldsymbol{X}^T \boldsymbol{X})^{-1}).$$

Chapter 18

R demo

We illustrate how to deal with heteroskedasticity, group-correlated errors, autocorrelated errors, and outliers in the following sections.

18.1 Heteroskedasticity

Next, let's look at another dataset, from the Current Population Survey (CPS).

```
library(readr)
library(ggplot2)
library(dplyr)
library(tibble)
library(tidyr)

cps_data <- read_tsv("data/cps2.tsv")
cps_data
```

```
# A tibble: 1,000 x 10
    wage  educ exper female black married union south fulltime metro
   <dbl> <dbl> <dbl>  <dbl> <dbl>   <dbl> <dbl> <dbl>    <dbl> <dbl>
 1  2.03    13     2      1     0       0     0     1        0     0
 2  2.07    12     7      0     0       0     0     0        0     1
 3  2.12    12    35      0     0       0     0     1        1     1
 4  2.54    16    20      1     0       0     0     1        1     1
 5  2.68    12    24      1     0       1     0     1        0     1
 6  3.09    13     4      0     0       0     0     1        0     1
 7  3.16    13     1      0     0       0     0     0        0     0
 8  3.17    12    22      1     0       1     0     1        0     1
 9  3.2     12    23      0     0       1     0     1        1     1
10  3.27    12     4      1     0       0     0     0        1     1
# i 990 more rows
```

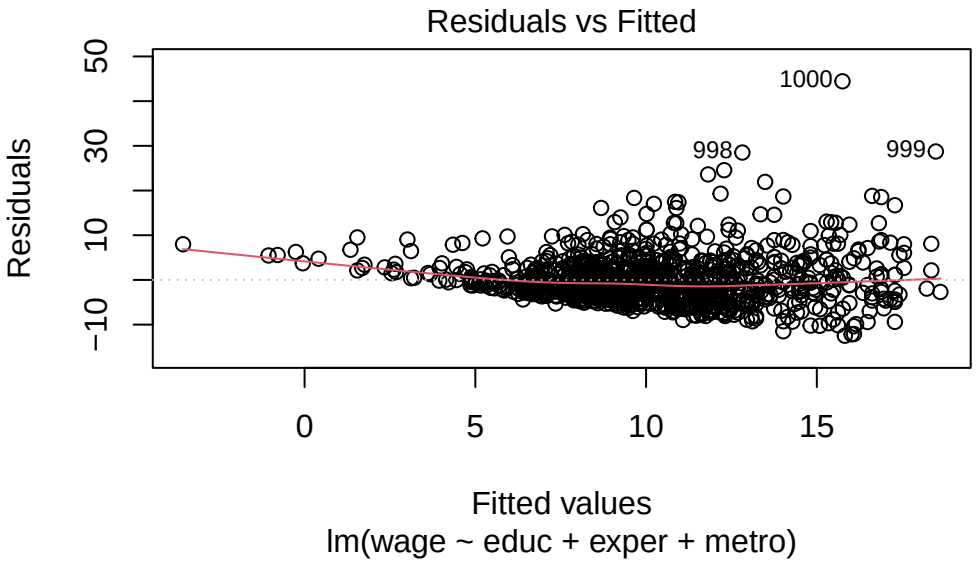
Suppose we want to regress `wage` on `educ`, `exper`, and `metro`.

```
lm_fit <- lm(wage ~ educ + exper + metro, data = cps_data)
```

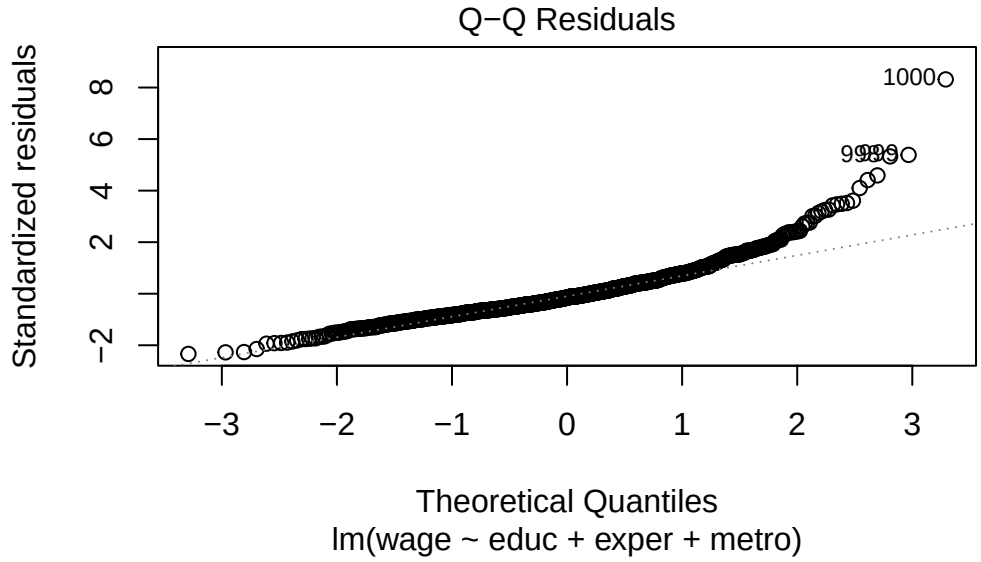
18.1.1 Diagnostics

Let's take a look at the standard linear model diagnostic plots built into R.

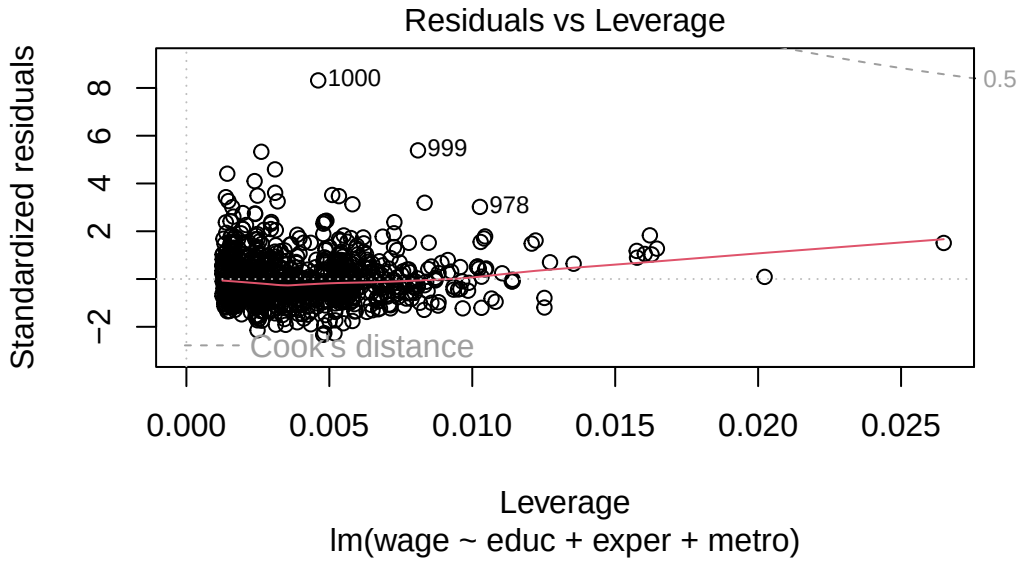
```
# residuals versus fitted
plot(lm_fit, which = 1)
```




```
# residual QQ plot
plot(lm_fit, which = 2)
```



```
# residuals versus leverage (with Cook's distance)
plot(lm_fit, which = 5)
```



The residuals versus fitted plot suggests significant heteroskedasticity, with variance growing as a function of the fitted value.

18.1.2 Sandwich standard errors

To get standard errors robust to this heteroskedasticity, we can use one of the robust estimators discussed in Section 13.2. Most of the robust standard error constructions discussed in that section are implemented in the R package `sandwich`.

```
library(sandwich)
```

For example, Huber-White's heteroskedasticity-consistent estimate $\widehat{\text{Var}}[\widehat{\boldsymbol{\beta}}]$ can be obtained via `vcovHC`:

```
HW_cov <- vcovHC(lm_fit)
HW_cov
```

```
             (Intercept)          educ         exper         metro
(Intercept)  1.484328645 -0.0967891868 -0.0096871141 -0.1218518012
educ        -0.096789187  0.0070467982  0.0004037764  0.0018334348
exper       -0.009687114  0.0004037764  0.0002517826  0.0008369831
metro       -0.121851801  0.0018334348  0.0008369831  0.1197713348
```

Compare this to the traditional estimate:

```
usual_cov <- vcovHC(lm_fit, type = "const")
usual_cov
```

```
             (Intercept)          educ         exper         metro
(Intercept)  1.157049852 -0.0671656102 -0.0070323974 -0.1287058354
educ        -0.067165610  0.0048945781  0.0001924359 -0.0018227782
exper       -0.007032397  0.0001924359  0.0002320022  0.0001471354
metro       -0.128705835 -0.0018227782  0.0001471354  0.1858394060
```

```
# extract the variance estimates from the diagonal
tibble(
  variable = rownames(usual_cov),
  usual_variance = sqrt(diag(usual_cov)),
  HW_variance = sqrt(diag(HW_cov))
)
```

```
# A tibble: 4 x 3
  variable    usual_variance HW_variance
  <chr>                <dbl>       <dbl>
1 (Intercept)         1.08        1.22
2 educ                0.0700      0.0839
3 exper               0.0152      0.0159
4 metro               0.431       0.346
```

Bootstrap standard errors are also implemented in `sandwich`:

```
# pairs bootstrap
bootstrap_cov <- vcovBS(lm_fit, type = "xy")
tibble(
  variable = rownames(usual_cov),
  usual_variance = diag(usual_cov),
  HW_variance = diag(HW_cov),
  bootstrap_variance = diag(bootstrap_cov)
)
```

```
# A tibble: 4 x 4
  variable    usual_variance HW_variance bootstrap_variance
  <chr>                <dbl>       <dbl>              <dbl>
1 (Intercept)       1.16        1.48               1.56
2 educ              0.00489     0.00705            0.00736
3 exper             0.000232    0.000252           0.000266
4 metro             0.186       0.120              0.111
```

The covariance estimate produced by `sandwich` can be easily integrated into linear model inference using the package `lmtest`.

```
library(lmtest)
```

```
Loading required package: zoo
```

```
Attaching package: 'zoo'
```

```
The following objects are masked from 'package:base':

    as.Date, as.Date.numeric
```

```r
# fit linear model as usual
lm_fit <- lm(wage ~ educ + exper + metro, data = cps_data)

# robust t-tests for coefficients
coeftest(lm_fit, vcov. = vcovHC)
```

```
t test of coefficients:

             Estimate Std. Error t value  Pr(>|t|)
(Intercept) -9.913984   1.218330 -8.1374 1.197e-15 ***
educ         1.233964   0.083945 14.6996 < 2.2e-16 ***
exper        0.133244   0.015868  8.3972 < 2.2e-16 ***
metro        1.524104   0.346080  4.4039 1.178e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# robust confidence intervals for coefficients
coefci(lm_fit, vcov. = vcovHC)
```

```
                 2.5 %     97.5 %
(Intercept) -12.3047729 -7.5231954
educ          1.0692342  1.3986938
exper         0.1021058  0.1643816
metro         0.8449747  2.2032337
```

```r
# robust F-test
lm_fit_partial <- lm(wage ~ educ, data = cps_data) # a partial model
waldtest(lm_fit_partial, lm_fit, vcov = vcovHC)
```

```
Wald test

Model 1: wage ~ educ
Model 2: wage ~ educ + exper + metro
  Res.Df Df      F    Pr(>F)
1    998
2    996  2 40.252 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

18.1.3 Bootstrap confidence intervals

One R package for performing bootstrap inference is `simpleboot`. Let's see how to get pairs bootstrap distributions for the coefficient estimates.

```
library(simpleboot)
```

```
Simple Bootstrap Routines (1.1-8)
```

```
boot_out <- lm.boot(
  lm.object = lm_fit, # input the fit object from lm()
  R = 1000
) # R is the number of bootstrap replicates
perc(boot_out) # get the percentile 95% confidence intervals
```

```
      (Intercept)     educ     exper     metro
2.5%   -12.365466 1.075378 0.1034755 0.8985245
97.5%   -7.532934 1.407756 0.1642828 2.1715691
```

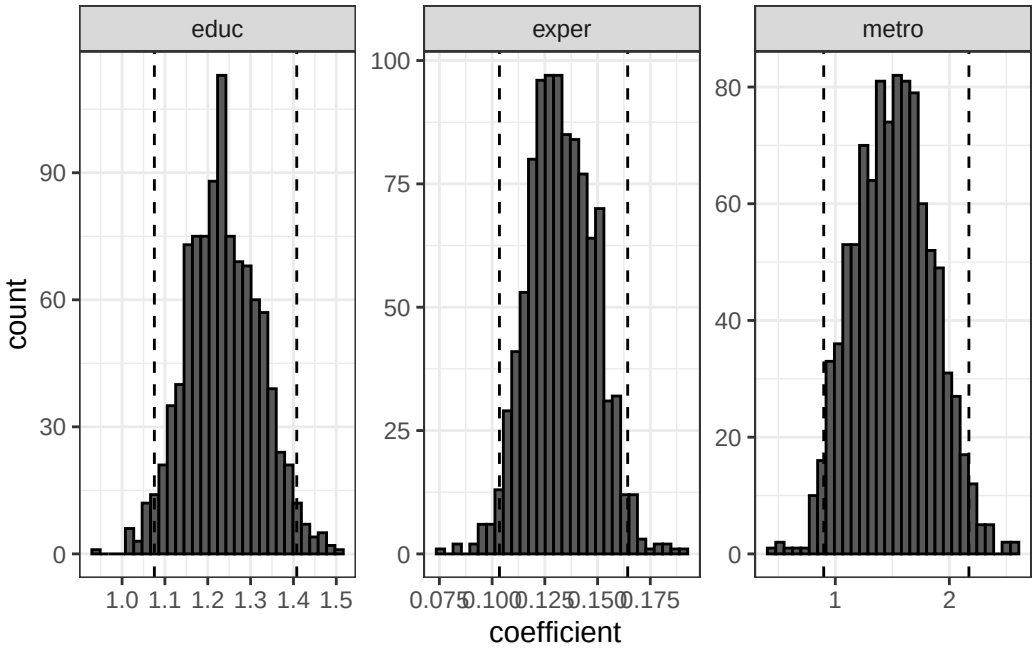
We can extract the resampling distributions for the coefficient estimates using the `samples` function:

```
samples(boot_out, name = "coef")[, 1:5]
```

```
                    1           2          3          4          5
(Intercept) -8.5183938 -10.1137042 -9.6244521 -9.6637688 -9.7635183
educ         1.1589410   1.2808242  1.2344501  1.2127978  1.2161110
exper        0.1090993   0.1097745  0.1196861  0.1243353  0.1467017
metro        1.5474578   1.9319631  1.6930226  1.8578930  1.3720004
```

We can plot these as follows:

```
boot_pctiles <- boot_out |>
  perc() |>
  t() |>
  as.data.frame() |>
  rownames_to_column(var = "var") |>
  filter(var != "(Intercept)")

samples(boot_out, name = "coef") |>
  as.data.frame() |>
  rownames_to_column(var = "var") |>
  filter(var != "(Intercept)") |>
  pivot_longer(-var, names_to = "resample", values_to = "coefficient") |>
  group_by(var) |>
  ggplot(aes(x = coefficient)) +
  geom_histogram(bins = 30, colour = "black") +
  geom_vline(aes(xintercept = `2.5%`), data = boot_pctiles, linetype = "dashed") +
  geom_vline(aes(xintercept = `97.5%`), data = boot_pctiles, linetype = "dashed") +
  facet_wrap(~var, scales = "free")
```



In this case, the bootstrap sampling distributions look roughly normal.

18.2 Group-correlated errors

Credit for this data example: https://www.r-bloggers.com/2021/05/clustered-standard-errors-with-r/.

Let's consider the `nslwork` data from the `webuse` package:

```
library(webuse)
nlswork_orig <- webuse("nlswork")
nlswork <- nlswork_orig |>
  filter(idcode <= 100) |>
  select(idcode, year, ln_wage, age, tenure, union) |>
  na.omit() |>
  mutate(
    union = as.integer(union),
    idcode = as.factor(idcode)
  )
nlswork
```

```
# A tibble: 386 x 6
   idcode  year ln_wage   age tenure union
   <fct>  <dbl>   <dbl> <dbl>  <dbl> <int>
 1 1         72    1.59    20  0.917     1
 2 1         77    1.78    25  1.5       0
 3 1         80    2.55    28  1.83      1
 4 1         83    2.42    31  0.667     1
 5 1         85    2.61    33  1.92      1
```

```
 6 1         87    2.54    35  3.92      1
 7 1         88    2.46    37  5.33      1
 8 2         71    1.36    19  0.25      0
 9 2         77    1.73    25  2.67      1
10 2         78    1.69    26  3.67      1
# i 376 more rows
```

The data comes from the US National Longitudinal Survey (NLS) and contains information about more than 4,000 young working women. We're interested in the relationship between wage (here as log-scaled GNP-adjusted wage) `ln_wage` and survey participant's current age, job tenure in years, and union membership as independent variables. It's a longitudinal survey, so subjects were asked repeatedly between 1968 and 1988, and each subject is identified by a unique idcode `idcode`. Here we restrict attention to the first 100 subjects and remove any rows with missing data.

Let's start by fitting a linear regression of the log wage on `age`, `tenure`, `union`, and the interaction between `tenure` and `union`:

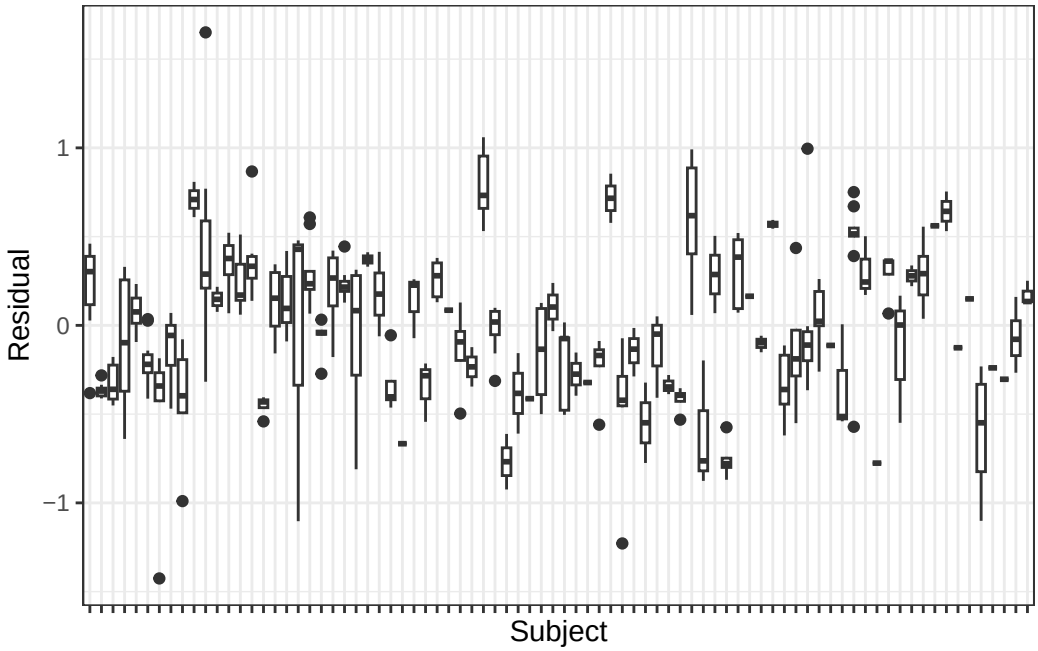
```
lm_fit <- lm(ln_wage ~ age + tenure + union + tenure:union, data = nlswork)
summary(lm_fit)
```

```
Call:
lm(formula = ln_wage ~ age + tenure + union + tenure:union, data = nlswork)

Residuals:
     Min       1Q   Median       3Q      Max
-1.42570 -0.28330  0.01694  0.27303  1.65052

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.379103   0.099658  13.838  < 2e-16 ***
age           0.013553   0.003388   4.000 7.60e-05 ***
tenure        0.022175   0.008051   2.754  0.00617 **
union         0.309936   0.070344   4.406 1.37e-05 ***
tenure:union -0.009629   0.012049  -0.799  0.42473
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4099 on 381 degrees of freedom
Multiple R-squared:  0.1811,    Adjusted R-squared:  0.1725
F-statistic: 21.07 on 4 and 381 DF,  p-value: 1.047e-15
```

Let's plot the residuals against the individuals:

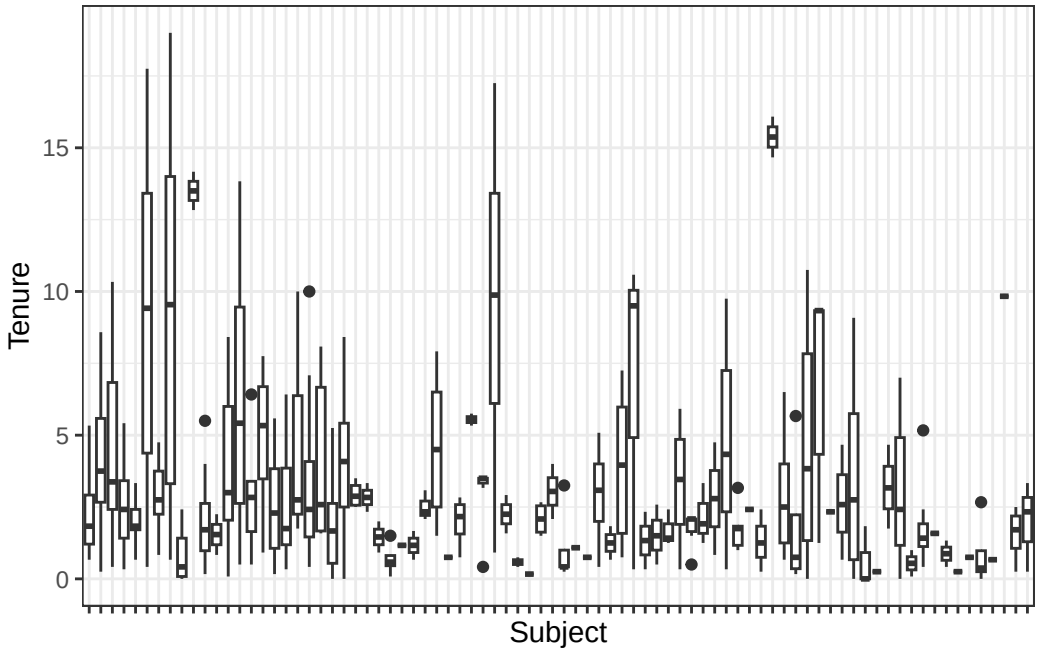
```
nlswork |>
  mutate(resid = lm_fit$residuals) |>
  ggplot(aes(x = idcode, y = resid)) +
  geom_boxplot() +
  labs(
```

```
    x = "Subject",
    y = "Residual"
  ) +
  theme(axis.text.x = element_blank())
```



Clearly, there is dependency among the residuals within subjects. Therefore, we have either model bias, or correlated errors, or both. To help assess whether we have model bias or not, we must check whether the variables of interest are correlated with the grouping variable `idcode`. We can check this with a plot, e.g., for the `tenure` variable:

```
nlswork |>
  ggplot(aes(x = idcode, y = tenure)) +
  geom_boxplot() +
  labs(
    x = "Subject",
    y = "Tenure"
  ) +
  theme(axis.text.x = element_blank())
```



Again, there seems to be nontrivial association between `tenure` and `idcode`. We can check this more formally with an ANOVA test:

```
summary(aov(tenure ~ idcode, data = nlswork))
```

```
             Df Sum Sq Mean Sq F value   Pr(>F)
idcode       81   2529  31.220   3.558 8.83e-16 ***
Residuals   304   2668   8.775
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

So, in this case, we do have model bias on our hands. We can address this using fixed effects for each subject.

```
lm_fit_FE <- lm(ln_wage ~ age + tenure + union + tenure:union + idcode, data = nlswork)
lm_fit_FE |>
  summary() |>
  coef() |>
  as.data.frame() |>
  rownames_to_column(var = "var") |>
  filter(!grepl("idcode", var)) |> # remove coefficients for fixed effects
  column_to_rownames(var = "var")
```

```
               Estimate  Std. Error   t value     Pr(>|t|)
(Intercept)  1.882478232 0.131411504 14.325064 8.022367e-36
age          0.005630809 0.003109803  1.810664 7.119315e-02
tenure       0.020756426 0.006964417  2.980353 3.114742e-03
union        0.174619394 0.060646038  2.879321 4.272027e-03
```

```
tenure:union 0.014974113 0.009548509  1.568215 1.178851e-01
```

Note the changes in the standard errors and p-values. Sometimes, we may have remaining correlation among residuals even after adding cluster fixed effects. Therefore, it is common practice to compute clustered (i.e., Liang-Zeger) standard errors in conjunction with cluster fixed effects. We can get clustered standard errors via the `vcovCL` function from `sandwich`:

```
LZ_cov <- vcovCL(lm_fit_FE, cluster = nlswork$idcode)
coeftest(lm_fit_FE, vcov. = LZ_cov)[, ] |>
  as.data.frame() |>
  rownames_to_column(var = "var") |>
  filter(!grepl("idcode", var)) |> # remove coefficients for fixed effects
  column_to_rownames(var = "var")
```

```
               Estimate  Std. Error    t value     Pr(>|t|)
(Intercept)  1.882478232 0.157611390 11.9437956 3.667970e-27
age          0.005630809 0.006339777  0.8881715 3.751601e-01
tenure       0.020756426 0.011149190  1.8616981 6.362342e-02
union        0.174619394 0.101970509  1.7124500 8.784708e-02
tenure:union 0.014974113 0.009646023  1.5523613 1.216301e-01
```

Again, note the changes in the standard errors and p-values.

18.3 Autocorrelated errors

Let's take a look at the `EuStockMarkets` data built into R, containing the daily closing prices of major European stock indices: Germany DAX (Ibis), Switzerland SMI, France CAC, and UK FTSE. Let's regress `DAX` on `FTSE` and take a look at the residuals:

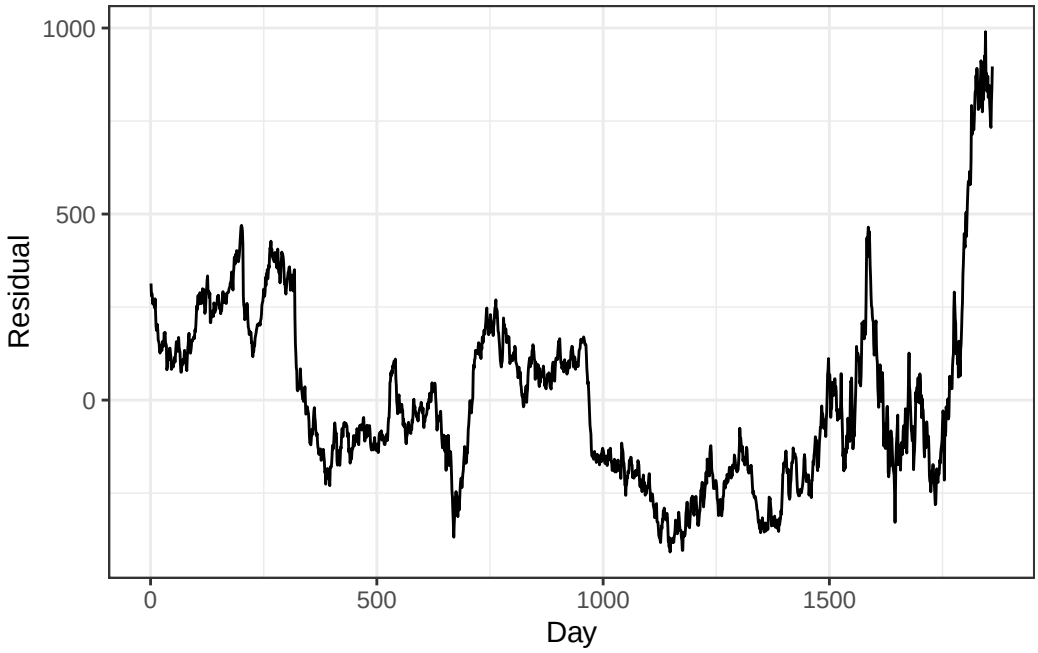
```
lm_fit <- lm(DAX ~ FTSE, data = EuStockMarkets)
summary(lm_fit)
```

```
Call:
lm(formula = DAX ~ FTSE, data = EuStockMarkets)

Residuals:
    Min      1Q  Median      3Q     Max
-408.43 -172.53  -45.71  137.68  989.96

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.331e+03  2.109e+01  -63.12   <2e-16 ***
FTSE         1.083e+00  5.705e-03  189.84   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 240.3 on 1858 degrees of freedom
Multiple R-squared:  0.951, Adjusted R-squared:  0.9509
```

```
F-statistic: 3.604e+04 on 1 and 1858 DF,  p-value: < 2.2e-16
```

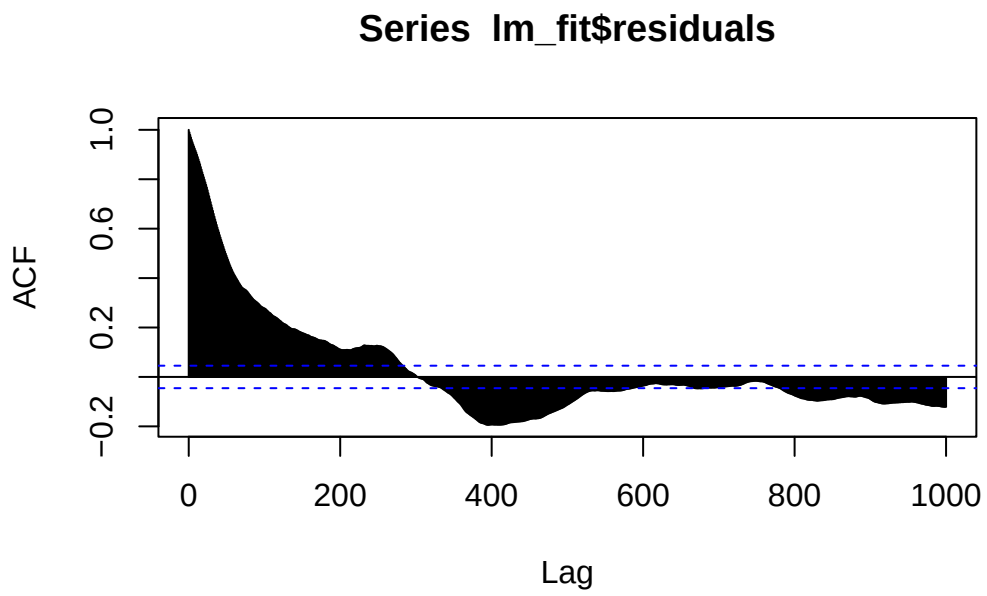
We find an extremely significant association between the two stock indices. But let's examine the residuals for autocorrelation:

```
EuStockMarkets |>
  as.data.frame() |>
  mutate(
    date = row_number(),
    resid = lm_fit$residuals
  ) |>
  ggplot(aes(x = date, y = resid)) +
  geom_line() +
  labs(
    x = "Day",
    y = "Residual"
  )
```



There is clearly some autocorrelation in the residuals. Let's quantify it using the autocorrelation function (`acf()` in R):

```
acf(lm_fit$residuals, lag.max = 1000)
```



We see that the autocorrelation gets into a reasonably low range around lag 200. We can then construct Newey-West standard errors based on this lag:

```
NW_cov <- NeweyWest(lm_fit)
coeftest(lm_fit, vcov. = NW_cov)
```

```
t test of coefficients:

             Estimate Std. Error t value Pr(>|t|)
(Intercept) -1331.2374  4398.3722 -0.3027   0.7622
FTSE            1.0831     1.4645  0.7396   0.4597
```

We see that the p-value for the association goes from `2e-16` to `0.46`, after accounting for autocorrelation.

18.4 Outliers

Let's take a look at the crime data from HW2:

```
# read crime data
crime_data <- read_tsv("data/Statewide_crime.dat")
```

```
Rows: 51 Columns: 6
-- Column specification --------------------------------------------------------
Delimiter: "\t"
chr (1): STATE
dbl (5): Violent, Murder, Metro, HighSchool, Poverty

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
  # read and transform population data
  population_data <- read_csv("data/state-populations.csv")
```

```
Rows: 52 Columns: 9
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (1): State
dbl (8): rank, Pop, Growth, Pop2018, Pop2010, growthSince2010, Percent, density

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
  population_data <- population_data |>
    filter(State != "Puerto Rico") |>
    select(State, Pop) |>
    rename(state_name = State, state_pop = Pop)

  # collate state abbreviations
  state_abbreviations <- tibble(
    state_name = state.name,
    state_abbrev = state.abb
  ) |>
    add_row(state_name = "District of Columbia", state_abbrev = "DC")

  # add CrimeRate to crime_data
  crime_data <- crime_data |>
    mutate(STATE = ifelse(STATE == "IO", "IA", STATE)) |>
    rename(state_abbrev = STATE) |>
    left_join(state_abbreviations, by = "state_abbrev") |>
    left_join(population_data, by = "state_name") |>
    mutate(CrimeRate = Violent / state_pop) |>
    select(state_abbrev, CrimeRate, Metro, HighSchool, Poverty)

  crime_data
```

```
# A tibble: 51 x 5
   state_abbrev CrimeRate Metro HighSchool Poverty
   <chr>            <dbl> <dbl>      <dbl>   <dbl>
 1 AK           0.000819   65.6       90.2     8
 2 AL           0.0000871  55.4       82.4    13.7
 3 AR           0.000150   52.5       79.2    12.1
 4 AZ           0.0000682  88.2       84.4    11.9
 5 CA           0.0000146  94.4       81.3    10.5
 6 CO           0.0000585  84.5       88.3     7.3
 7 CT           0.0000867  87.7       88.8     6.4
 8 DE           0.000664   80.1       86.5     5.8
 9 FL           0.0000333  89.3       85.9     9.7
```

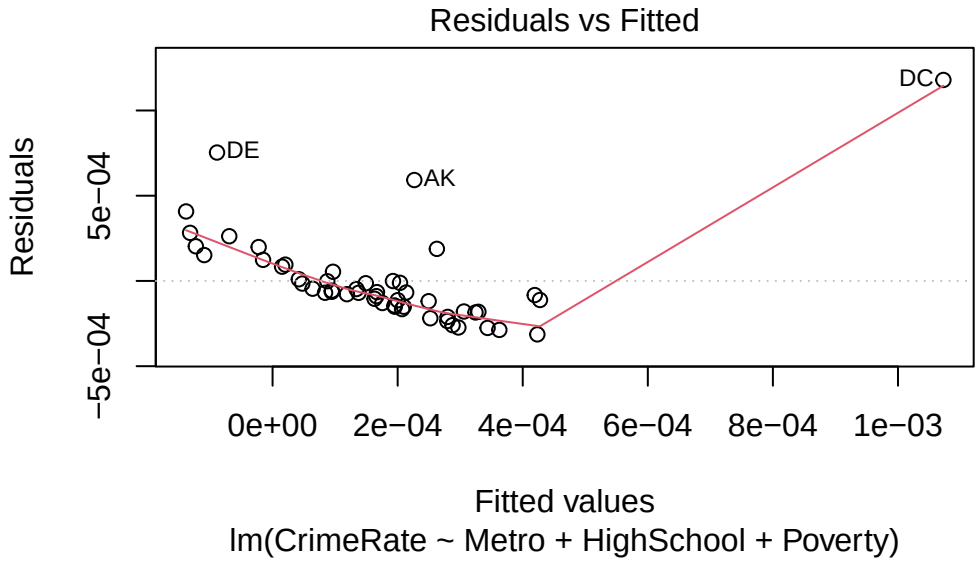
```
10 GA           0.0000419  71.6       85.2    10.8
# i 41 more rows
```

Let's fit the linear regression:

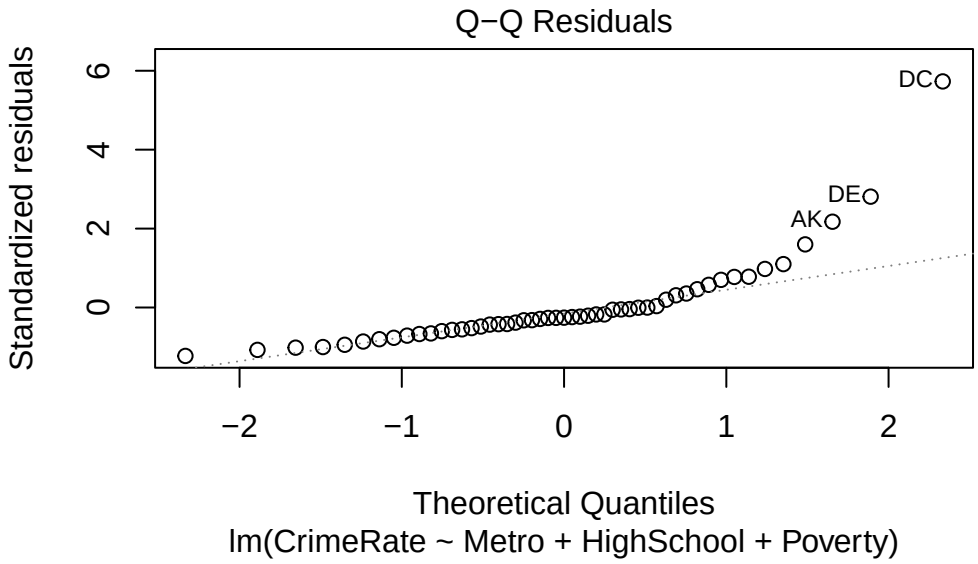
```
# note: we make the state abbreviations row names for better diagnostic plots
lm_fit <- lm(CrimeRate ~ Metro + HighSchool + Poverty, data = crime_data |> column_to_rownam
```

We can get the standard linear regression diagnostic plots as follows:

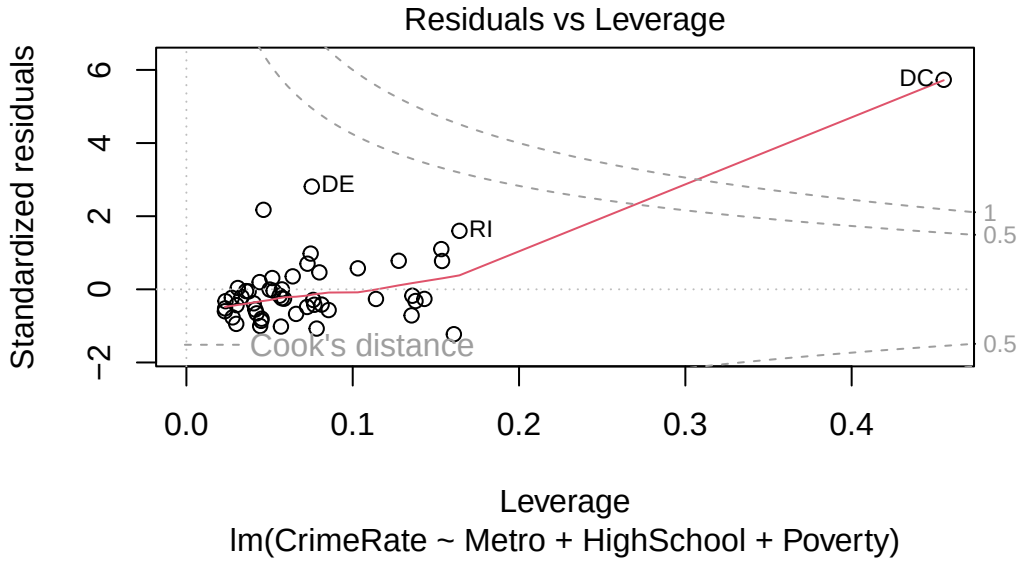
```
# residuals versus fitted
plot(lm_fit, which = 1)
```




```
# residual QQ plot
plot(lm_fit, which = 2)
```




```
# residuals versus leverage (with Cook's distance)
plot(lm_fit, which = 5)
```



The information underlying these diagnostic plots can be extracted as follows:

```
tibble(
  state = crime_data$state_abbrev,
  std_residual = rstandard(lm_fit),
  fitted_value = fitted.values(lm_fit),
  leverage = hatvalues(lm_fit),
  cooks_dist = cooks.distance(lm_fit)
)
```

```
# A tibble: 51 x 5
   state std_residual fitted_value leverage cooks_dist
   <chr>        <dbl>        <dbl>    <dbl>      <dbl>
 1 AK           2.17     0.000227    0.0463   0.0574
 2 AL          -0.422    0.000200    0.0769   0.00371
 3 AR           1.10    -0.000132    0.153    0.0547
 4 AZ          -1.02     0.000344    0.0568   0.0156
 5 CA          -0.264    0.0000839   0.114    0.00224
 6 CO          -0.383    0.000163    0.0405   0.00155
 7 CT          -0.175    0.000134    0.0561   0.000456
 8 DE           2.81    -0.0000888   0.0754   0.161
 9 FL          -0.804    0.000252    0.0452   0.00764
10 GA          -0.599    0.000207    0.0232   0.00213
# i 41 more rows
```

Clearly, DC is an outlier. We can either run a robust estimation procedure or redo the analysis without DC. Let's try both. First, we try robust regression using `rlm()` from the `MASS` package:

```
rlm_fit <- MASS::rlm(CrimeRate ~ Metro + HighSchool + Poverty, data = crime_data)
summary(rlm_fit)
```

```
Call: rlm(formula = CrimeRate ~ Metro + HighSchool + Poverty, data = crime_data)
Residuals:
       Min         1Q     Median         3Q        Max
-8.297e-05 -3.787e-05 -2.249e-05  4.407e-05  2.063e-03

Coefficients:
            Value   Std. Error t value
(Intercept) -0.0009  0.0004    -2.2562
Metro        0.0000  0.0000    -1.2963
HighSchool   0.0000  0.0000     2.6506
Poverty      0.0000  0.0000     2.7546

Residual standard error: 6.048e-05 on 47 degrees of freedom
```

For some reason, the p-values are not computed automatically. We can compute them ourselves instead:

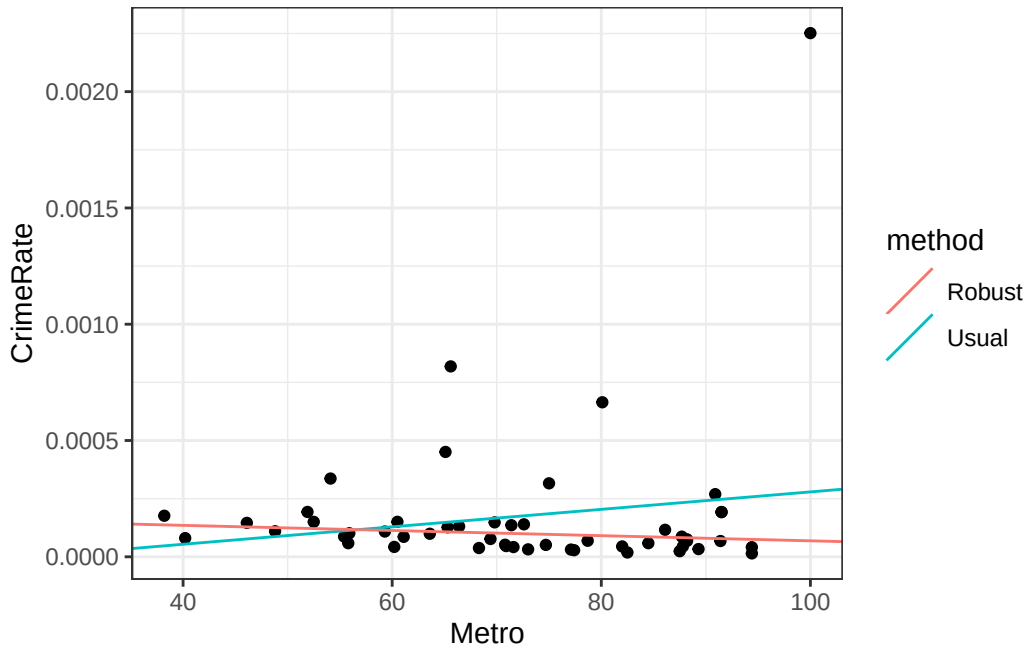
```
summary(rlm_fit)$coef |>
  as.data.frame() |>
  rename(Estimate = Value) |>
  mutate(`p value` = 2 * dnorm(-abs(`t value`)))
```

```
                Estimate   Std. Error   t value    p value
(Intercept) -8.538466e-04 3.784466e-04 -2.256188 0.06260042
Metro       -8.639252e-07 6.664623e-07 -1.296285 0.34439400
HighSchool   1.037849e-05 3.915573e-06  2.650568 0.02378865
Poverty      1.252839e-05 4.548172e-06  2.754600 0.01795833
```

To see the robust estimation action visually, let's consider a univariate example:

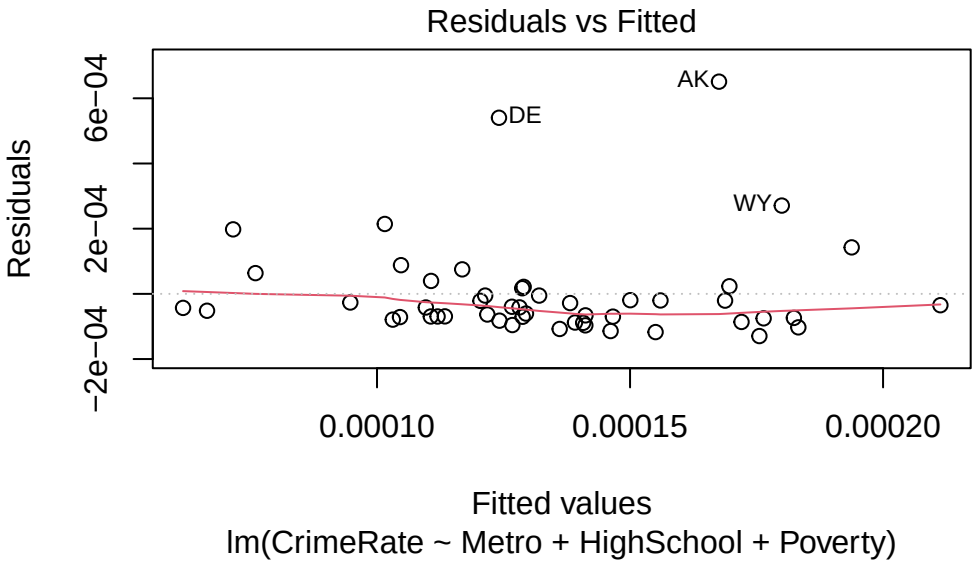
```
lm_fit <- lm(CrimeRate ~ Metro, data = crime_data)
rlm_fit <- MASS::rlm(CrimeRate ~ Metro, data = crime_data)

# collate the fits into a tibble
line_fits <- tibble(
  method = c("Usual", "Robust"),
  intercept = c(
    coef(lm_fit)["(Intercept)"],
    coef(rlm_fit)["(Intercept)"]
  ),
  slope = c(
    coef(lm_fit)["Metro"],
    coef(rlm_fit)["Metro"]
  )
)
```

```
# usual and robust univariate fits
# plot the fits
crime_data |>
  ggplot() +
  geom_point(aes(x = Metro, y = CrimeRate)) +
  geom_abline(aes(intercept = intercept, slope = slope, colour = method), data = line_fits)
```

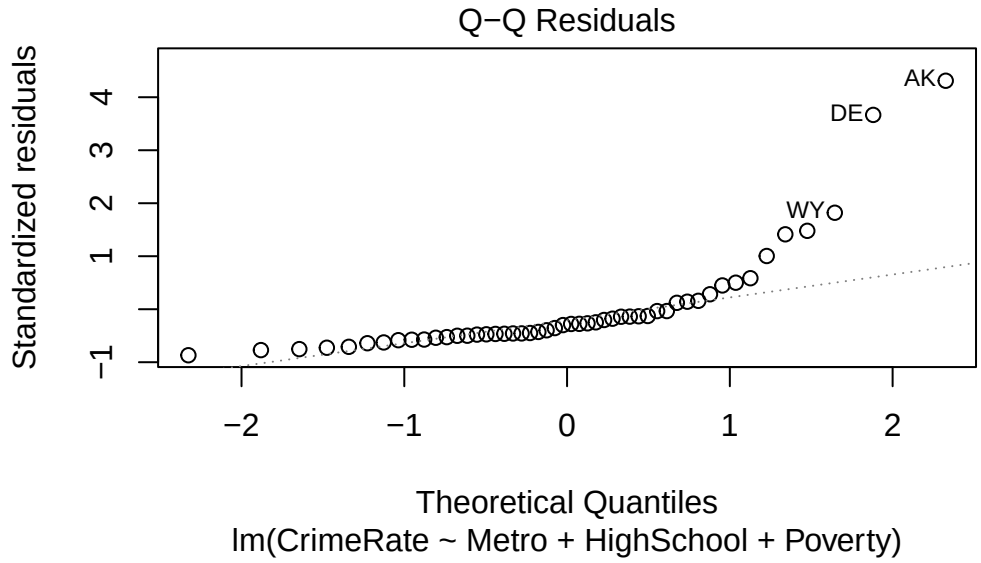


Next, let's try removing DC and running a usual linear regression.

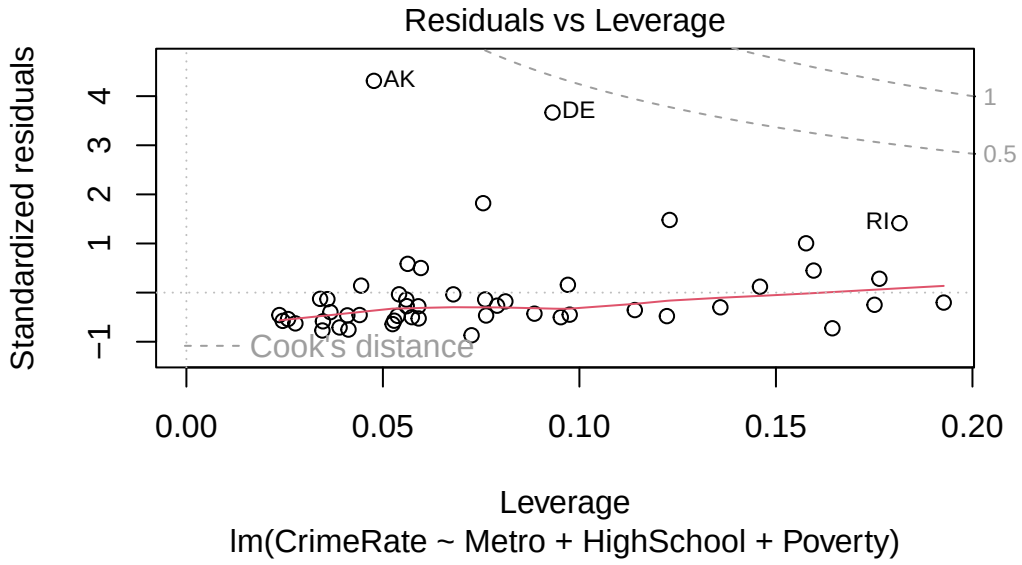
```r
lm_fit_no_dc <- lm(CrimeRate ~ Metro + HighSchool + Poverty,
  data = crime_data |>
    filter(state_abbrev != "DC") |>
    column_to_rownames(var = "state_abbrev")
)

# residuals versus fitted
plot(lm_fit_no_dc, which = 1)
```




```r
# residual QQ plot
plot(lm_fit_no_dc, which = 2)
```



```r
# residuals versus leverage (with Cook's distance)
plot(lm_fit_no_dc, which = 5)
```



Part IV

Generalized linear models: General theory

Chapters 1-3 focused on the most common class of models used in applications: linear models. Despite their versatility, linear models do not apply in all situations. In particular, they are not designed to deal with binary or count responses. In Chapter 4, we introduce *generalized linear models* (GLMs), a generalization of linear models that encompasses a wide variety of incredibly useful models, including logistic regression and Poisson regression.

We'll start Chapter 4 by introducing exponential dispersion models (Section Chapter 19), a generalization of the Gaussian distribution that serves as the backbone of GLMs. Then we formally define a GLM, demonstrating logistic regression and Poisson regression as special cases (Section Chapter 20). Next, we discuss maximum likelihood inference in GLMs (Section Chapter 21). Finally, we discuss how to carry out statistical inference in GLMs (Section Chapter 22).

Chapter 19

Exponential dispersion model (EDM) distributions

19.1 Definition

Let's start with the Gaussian distribution. If $y \sim N(\mu, \sigma^2)$, then it has the following density with respect to the Lebesgue measure ν on $\mathbb{R}$:

$$\begin{aligned} f_{\mu,\sigma^2}(y) &= \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(y-\mu)^2\right) \\ &= \exp\left(\frac{\mu y - \frac{1}{2}\mu^2}{\sigma^2}\right) \cdot \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}y^2\right). \end{aligned}$$

We can consider a more general class of densities with respect to any measure ν:

$$f_{\theta,\phi}(y) \equiv \exp\left(\frac{\theta y - \psi(\theta)}{\phi}\right) h(y,\phi), \quad \theta \in \Theta \subseteq \mathbb{R},\ \phi > 0. \tag{19.1}$$

Here θ is called the *natural parameter*, ψ is called the *log-partition function*, $\Theta \equiv \{\theta : \psi(\theta) < \infty\}$ is called the natural parameter space,[1] $\phi > 0$ is called the *dispersion parameter*, and h is called the *base density*. The distribution with density $f_{\theta,\phi}$ with respect to a measure ν on $\mathbb{R}$ is called an *exponential dispersion model* (EDM).[2] Sometimes, we parameterize this distribution using its mean and dispersion, writing

$$y \sim \text{EDM}(\mu, \phi).$$

When $\phi = 1$, the distribution becomes a *one-parameter natural exponential family* (see Figure 19.1).

[1] The Fisher information is the expectation of the Hessian, but for canonical links, the Hessian is non-random, so the two coincide.

[2] If you are not familiar with measure theory, you can view ν as specifying the support of a distribution (the set of values it can take). For example, for binary y, ν would indicate that y is supported on $\{0, 1\}$, and the "density" $f_{\theta,\phi}$ would be a probability mass function.

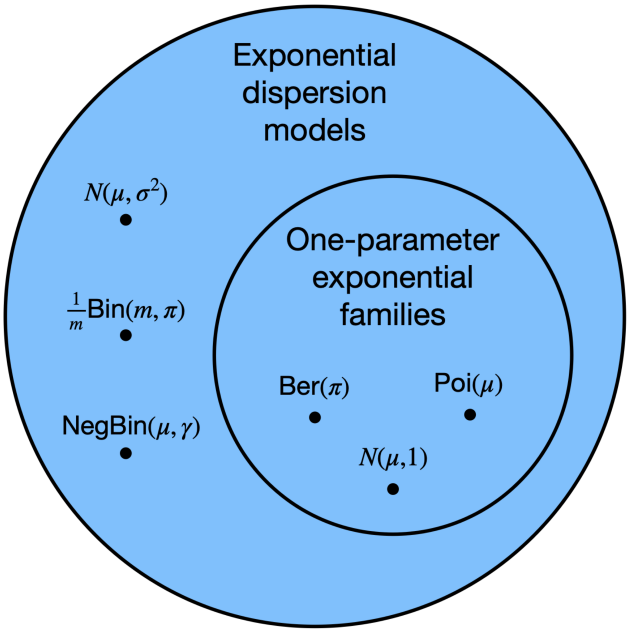


Figure 19.1: Relationship between exponential dispersion models and one-parameter exponential families.

The following proposition presents a useful property of EDMs, which facilitates inference by ruling out pathological cases.

Proposition 19.1. *The support of $y \sim EDM(\mu, \phi)$ remains fixed as (μ, ϕ) vary.*

19.2 Examples

19.2.1 Normal distribution

As derived above, $y \sim N(\mu, \sigma^2)$ is an EDM with

$$\theta = \mu, \quad \psi(\theta) = -\frac{1}{2}\theta^2, \quad \phi = \sigma^2, \quad h(y, \phi) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}y^2\right).$$

19.2.2 Bernoulli distribution

Suppose $y \sim \text{Ber}(\mu)$. Then, we have

$$f(y) = \mu^y(1-\mu)^{1-y} = \exp\left(y \log \frac{\mu}{1-\mu} + \log(1-\mu)\right).$$

Therefore, we have $\theta = \log \frac{\mu}{1-\mu}$, so that $\log(1-\mu) = -\log(1+e^\theta)$. It follows that

$$\theta = \log \frac{\mu}{1-\mu}, \quad \psi(\theta) = \log(1+e^\theta), \quad \phi = 1, \quad h(y) = 1.$$

Hence, the Bernoulli distribution is an EDM, as well as a one-parameter exponential family. Note that Ber(0) and Ber(1) are not included in this class of EDMs, because there is no $\theta \in \Theta = \mathbb{R}$ that gives rise to $\mu = 0$ or $\mu = 1$. Hence, $\mu \in (0, 1)$, and the support of any Bernoulli EDM is $\{0, 1\}$.

19.2.3 Binomial distribution

Consider the binomial proportion y: $my \sim \text{Bin}(m, \mu)$. We have

$$\begin{aligned} f(y) &= \binom{m}{my} \mu^{my} (1-\mu)^{m(1-y)} \\ &= \exp\left(m \left(y \log \frac{\mu}{1-\mu} + \log(1-\mu) \right) \right) \binom{m}{my}, \end{aligned}$$

so

$$\theta = \log \frac{\mu}{1-\mu}, \quad \psi(\theta) = \frac{e^\theta}{1+e^\theta}, \quad \phi = 1/m, \quad h(y, \phi) = \binom{m}{my}.$$

Note that $\text{Bin}(m, 0)$ and $\text{Bin}(m, 1)$ are not included in this class of EDMs, for the same reason as above. Hence, $\mu \in (0, 1)$, and the support of any binomial EDM is $\{0, \frac{1}{m}, \frac{2}{m}, \ldots, 1\}$.

19.2.4 Poisson distribution

Suppose $y \sim \text{Poi}(\mu)$. We have

$$f(y) = e^{-\mu} \frac{\mu^y}{y!} = \exp(y \log \mu - \mu) \frac{1}{y!}.$$

Therefore, we have $\theta = \log \mu$, so that $\mu = e^\theta$. It follows that

$$\theta = \log \mu, \quad \psi(\theta) = e^\theta, \quad \phi = 1, \quad h(y) = \frac{1}{y!}.$$

Hence, the Poisson distribution is an EDM, as well as a one-parameter exponential family. Note that Poi(0) is not included in this class of EDMs, because there is no $\theta \in \Theta = \mathbb{R}$ that gives rise to $\mu = 0$. Hence, $\mu \in (0, \infty)$, and the support of any Poisson EDM is $\mathbb{N}$.

Many other examples fall into this class, including the negative binomial, gamma, and inverse-Gaussian distributions. We will see at least some of these in the next chapter.

19.3 Moments of exponential dispersion model distributions

It turns out that the derivatives of the log-partition function ψ give the moments of y. Indeed, let's start with the relationship

$$\int f_{\theta,\phi}(y) d\nu(y) = \int \exp\left(\frac{\theta y - \psi(\theta)}{\phi} \right) h(y, \phi) d\nu(y) = 1.$$

Differentiating in θ and interchanging the derivative and the integral, we obtain

$$0 = \frac{d}{d\theta} \int f_{\theta,\phi}(y)dy = \int \frac{y - \dot{\psi}(\theta)}{\phi} f_{\theta,\phi}(y)dy,$$

from which it follows that

$$\dot{\psi}(\theta) = \int \dot{\psi}(\theta) f_{\theta,\phi}(y)dy = \int y f_{\theta,\phi}(y)dy = \mathbb{E}[y] \equiv \mu. \tag{19.2}$$

Thus, the first derivative of the log partition function is the mean of y. Differentiating again, we get

$$\begin{aligned}
\phi \cdot \ddot{\psi}(\theta) &= \phi \int \ddot{\psi}(\theta) f_{\theta,\phi}(y)d\nu(y) \\
&= \int (y - \dot{\psi}(\theta))^2 f_{\theta,\phi}(y)dy = \int (y-\mu)^2 f_{\theta,\phi}(y)d\nu(y) \\
&= \mathrm{Var}[y].
\end{aligned}$$

Thus, the second derivative of the log-partition function multiplied by the dispersion parameter is the variance of y. The following proposition summarizes these results.

Proposition 19.2 (EDM moments)**.** *If $y \sim EDM(\mu, \phi)$, then*

$$\mathbb{E}[y] = \dot{\psi}(\theta), \quad Var[y] = \phi \cdot \ddot{\psi}(\theta).$$

19.4 Relationships among the mean, variance, and natural parameter

19.4.1 Relationship between the mean and the natural parameter

The log-partition function ψ induces a connection (19.2) between the natural parameter θ and the mean μ. Because

$$\frac{d\mu}{d\theta} = \frac{d}{d\theta}\dot{\psi}(\theta) = \ddot{\psi}(\theta) = \frac{1}{\phi}\mathrm{Var}[y] > 0, \tag{19.3}$$

it follows that μ is a strictly increasing function of θ, so in particular the mapping between μ and θ is bijective. Therefore, we can think of equivalently parameterizing the distribution via μ or θ.

19.4.2 Relationship between the mean and variance

Note that the mean of an EDM, together with the dispersion parameter, determines its variance (since it determines the natural parameter θ). Define

$$V(\mu) \equiv \frac{d\mu}{d\theta},$$

so that $\text{Var}[y] = \phi V(\mu)$. For example, a Poisson random variable with mean μ has variance μ and a Bernoulli random variable with mean μ has $V(\mu) = \mu(1-\mu)$. The mean-variance relationship turns out to characterize the EDM, i.e. an EDM with mean equal to its variance is the Poisson distribution. For all EDMs except the normal distribution, the variance depends nontrivially on the mean. Therefore, heteroskedasticity is a natural feature of EDMs (rather than a pathology that needs to be corrected for).

19.5 The unit deviance

A key quantity in the analysis of normal linear regression models is $(y, \mu) \mapsto (y-\mu)^2$, which is a notion of distance between the parameter μ and the observation y. The *unit deviance* is a generalization of this quantity to EDMs. As a starting point, consider the log-likelihood of an EDM:

$$\ell(y, \mu) = \frac{\theta y - \psi(\theta)}{\phi} + \log h(y, \phi) = \frac{\theta(\mu) y - \psi(\theta(\mu))}{\phi} + \log h(y, \phi),$$

where $\theta(\mu) = \dot{\psi}^{-1}(\mu)$, recalling the relationship (19.2). The quantity $\ell(y, \mu)$ is larger to the extent that μ is a better fit for y. Furthermore, it is easy to verify that $\mu \mapsto \ell(y, \mu)$ is maximized by $\mu = y$. Motivated by this observation, we calculate that twice the log-likelihood ratio between $\mu = \mu$ and $\mu = y$ is

$$\begin{aligned} & 2(\ell(y, y) - \ell(y, \mu)) \\ &= \frac{2\{[\theta(y) y - \psi(\theta(y))] - [\theta(\mu) y - \psi(\theta(\mu))]\}}{\phi} \\ &\equiv \frac{d(y, \mu)}{\phi}. \end{aligned} \tag{19.4}$$

The quantity in the numerator is the *unit deviance* $d(y, \mu)$, defined as the dispersion ϕ times twice the log-likelihood ratio between y and μ. As we will see in Section 19.5.1, $d(y, \mu)$ generalizes the quantity $(y-\mu)^2$ for the normal distribution. The following proposition summarizes a few key properties of the unit deviance.

Proposition 19.3 (Unit deviance properties)**.** *When viewed as a function of μ, the unit deviance $d(y, \mu)$ is nonnegative function that achieves a unique global minimum of zero for $\mu = y$ and increases as μ moves away from y.*

Proof. Differentiating $d(y, \mu)$ in μ, we have

$$\frac{\partial d(y, \mu)}{\partial \mu} = \frac{\partial d(y, \mu)}{\partial \theta} \frac{\partial \theta}{\partial \mu} = \frac{\mu - y}{V(\mu)}.$$

Since $V(\mu) > 0$, this establishes that $d(y, \mu)$ decreases on $\mu \in (\infty, y)$ and then increases on (y, ∞), Therefore, $d(y, \mu) \geq d(y, y) = 0$ for all μ.

□

19.5.1 Example: Normal distribution

For the normal distribution, we have $\theta = \mu$ and $\psi(\theta) = \frac{1}{2}\theta^2$. Therefore,

$$\begin{aligned} d(y, \mu) &= 2\{[\theta(y) y - \psi(\theta(y))] - [\theta(\mu) y - \psi(\theta(\mu))]\} \\ &= 2\{[y^2 - \tfrac{1}{2} y^2] - [\mu y - \tfrac{1}{2}\mu^2]\} \\ &= (y-\mu)^2. \end{aligned}$$

Figure 19.2 displays an example of the normal unit deviance.

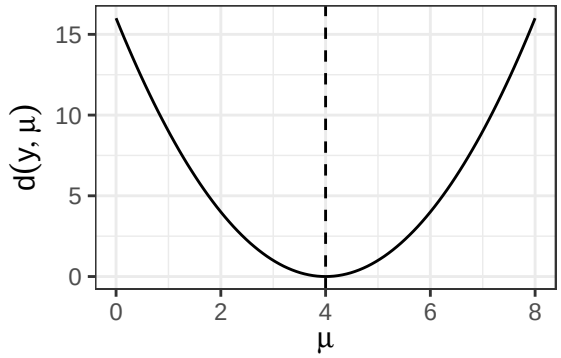


Figure 19.2: The normal unit deviance for $y = 4$.

19.5.2 Example: Poisson distribution

For the Poisson distribution, we have $\theta = \log \mu$ and $\psi(\theta) = e^\theta$, so

$$\begin{aligned} d(y, \mu) &= 2\{[\theta(y)y - \psi(\theta(y))] - [\theta(\mu)y - \psi(\theta(\mu))]\} \\ &= 2\{[y \log y - y] - [y \log \mu - \mu]\} \\ &= 2\left(y \log \frac{y}{\mu} - (y - \mu)\right). \end{aligned}$$

See Figure 19.3 for an example of the shape of this function.

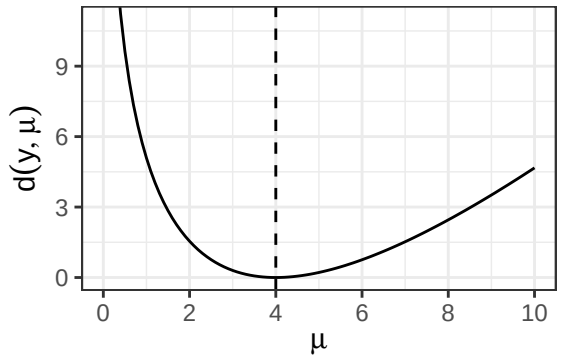


Figure 19.3: The Poisson unit deviance for $y = 4$.

Note that the Poisson deviance is asymmetric about $\mu = y$. This is a consequence of the nontrivial mean-variance relationship for the Poisson distribution. In particular, the Poisson distribution's variance grows with its mean. Therefore, an observation of $y = 4$ is less likely to have come from a Poisson distribution with mean $\mu = 2$ than from a Poisson distribution with mean $\mu = 6$.

19.6 Small-dispersion approximations to an EDM

If the dispersion ϕ is small, then that means that y is a fairly precise estimate of μ, similar to an average of multiple independent samples from a mean-μ distribution. Consider, for example, that

$\frac{1}{m}\text{Bin}(m, \mu)$ is the mean of m i.i.d. draws from $\text{Ber}(\mu)$. In this case, we can use either the normal approximation or the saddlepoint approximation to approximate the EDM density. For the sake of this section, we will abuse notation by denoting by $f_{\mu,\phi}$ the EDM with mean μ and dispersion ϕ.

19.6.1 The normal approximation

19.6.1.1 The approximation

For small values of ϕ, we can hope to approximate $f_{\mu,\phi}$ with a normal distribution. Recall that the mean and variance of this distribution are μ and $\phi \cdot V(\mu)$, respectively. The central limit theorem gives

$$\frac{y - \mu}{\sqrt{\phi \cdot V(\mu)}} \to_d N(0, 1) \quad \text{as} \quad \phi \to 0,$$

so

$$y \dot{\sim} N(\mu, \phi \cdot V(\mu)) \equiv \tilde{f}^{\text{normal}}_{\mu,\phi}.$$

For example, we have

$$\text{Poi}(\mu) \approx N(\mu, \mu).$$

For the normal EDM, note that the normal approximation is exact. One consequence of the normal approximation is

$$\frac{(y - \mu)^2}{\phi \cdot V(\mu)} \dot{\sim} \chi^2_1. \tag{19.5}$$

This fact will be useful to us as we carry out inference for GLMs.

19.6.1.2 Normal approximation accuracy

We have

$$\tilde{f}^{\text{normal}}_{\mu,\phi}(y) = f_{\mu,\phi}(y) + O(\sqrt{\phi}).$$

In practice, the rule of thumb for the applicability of this approximation to get statements like (19.5) is that

$$\tau \equiv \frac{\phi \cdot V(\mu)}{(\mu - \text{boundary})^2} \leq \frac{1}{5}.$$

Here, "boundary" represents the nearest boundary of the parameter space to μ. For example, if $y \sim \frac{1}{m}\text{Bin}(m, \mu)$, then we have

$$\begin{aligned}\tau &= \frac{\frac{1}{m} \cdot \mu \cdot (1-\mu)}{\min(\mu, 1-\mu)^2} \\ &= \frac{1}{m} \cdot \max\left(\frac{\mu}{1-\mu}, \frac{1-\mu}{\mu}\right) \\ &\approx \frac{1}{m} \cdot \max\left(\frac{1}{\mu}, \frac{1}{1-\mu}\right),\end{aligned}$$

so $\tau \leq 1/5$ roughly if $m\mu \leq 5$ and $m(1-\mu) \leq 5$. For Poisson distributions, we always have $\tau = 1$, but for some reason small-dispersion asymptotics still applies as $\mu \to \infty$ as opposed to $\tau \to 0$. The criterion $\tau \leq 1/5$ is satisfied when $\mu \leq 5$.

19.6.2 The saddlepoint approximation

19.6.2.1 The approximation

Another approximation to the EDM density is the saddlepoint approximation, which tends to be more accurate than the normal approximation. The reason the normal approximation may be inaccurate is that the quality of the central limit approximation degrades as one enters the tails of the distribution. In particular, the normal approximation to $f_{\mu,\phi}(y)$ may be poor if μ is far from y. The saddlepoint approximation is built on the observation that the EDM density for $f_{\mu,\phi}(y)$ can be written in terms of the density $f_{y,\phi}(y)$; the latter density is by definition evaluated at its mean. Indeed,

$$\begin{aligned}f_{\mu,\phi}(y) &\equiv \exp\left(\frac{\theta y - \psi(\theta)}{\phi}\right) h(y,\phi) \\ &= \exp\left(-\frac{d(y,\mu)}{2\phi}\right) \exp\left(\frac{\theta(y) y - \psi(\theta(y))}{\phi}\right) h(y,\phi) \\ &= \exp\left(-\frac{d(y,\mu)}{2\phi}\right) f_{y,\phi}(y).\end{aligned} \tag{19.6}$$

Now, we apply the central limit theorem to approximate $f_{y,\phi}(y)$:

$$f_{y,\phi}(y) \approx \frac{1}{\sqrt{2\pi\phi V(y)}}.$$

Substituting this approximation into (19.6), we obtain the *saddlepoint approximation*:

$$f_{\mu,\phi}(y) \approx \frac{1}{\sqrt{2\pi\phi V(y)}} \exp\left(-\frac{d(y,\mu)}{2\phi}\right) \equiv \widetilde{f}^{\text{saddle}}_{\mu,\phi}(y).$$

For the normal EDM, note that the normal approximation is exact. For the Poisson distribution, we get

$$\widetilde{f}^{\text{saddle}}_{\mu,\phi}(y) = \frac{1}{\sqrt{2\pi y}} \exp\left(-y \log\frac{y}{\mu} + (y-\mu)\right).$$

The approximation can be shown to lead to the following consequence:

$$\frac{d(y,\mu)}{\phi} \dot{\sim} \chi^2_1. \tag{19.7}$$

Here, we are using the unit deviance rather than the squared distance to measure the deviation of μ from y. This fact will be useful to us as we carry out inference for GLMs.

19.6.2.2 Saddlepoint approximation accuracy

We have still used a normal approximation, but this time we have used it to approximate $f_{y,\phi}(y)$ instead of $f_{\mu,\phi}(y)$. Since the normal approximation is applied to a distribution ($f_{y,\phi}$) at its mean, we expect it to be more accurate than a normal approximation applied to a distribution ($f_{\mu,\phi}$) at a point potentially far from its mean. The saddlepoint approximation yields an approximation to the density that is *multiplicative* rather than *additive*, and of order $O(\phi)$ rather than $O(\sqrt{\phi})$:

$$\tilde{f}^{\text{saddle}}_{\mu,\phi}(y) = f_{\mu,\phi}(y) \cdot (1 + O(\phi)).$$

In practice, the rule of thumb for the applicability of this approximation to get statements like (19.7) is that $\tau \leq 1/3$; the looser requirement on τ reflects the greater accuracy of the saddlepoint approximation. This translates to $m\mu \geq 3$ and $m(1-\mu) \geq 3$ for the binomial and $\mu \geq 3$ for the Poisson.

19.6.3 Comparing the two approximations

The saddlepoint approximation is more accurate than the normal approximation, as discussed above. However, the accuracy of the saddlepoint approximation relies on the assumption that the entire parametric form of the EDM is correctly specified. On the other hand, the accuracy of the normal distribution requires only that the first two moments of the EDM are correctly specified.

Chapter 20

GLM definition

In this class, the focus is on building models that tie a vector of predictors ($\boldsymbol{x}_{i*}$) to a response y_i. For linear regression, the mean of y was modeled as a linear combination of the predictors $\boldsymbol{x}_{i*}^T\boldsymbol{\beta}$: $\mu_i = \boldsymbol{x}_{i*}^T\boldsymbol{\beta}$. More generally, we might want to model *a function* of the mean $\eta_i = g(\mu_i)$ as a linear combination of the predictors; g is called the *link function* and η_i the *linear predictor*. Pairing a link function with an EDM gives us a *generalized linear model* (GLM):

20.1 Definition

We define $\{(y_i, \boldsymbol{x}_{i*})\}_{i=1}^n$ as following a generalized linear model based on the exponential dispersion model $f_{\theta,\phi}$, monotonic and differentiable link function g, offset terms $o_i \in \mathbb{R}$, and observation weights $w_i > 0$ if

$$y_i \overset{\text{ind}}{\sim} \text{EDM}(\mu_i, \phi_0/w_i), \quad \eta_i \equiv g(\mu_i) = o_i + \boldsymbol{x}_{i*}^T\boldsymbol{\beta}. \tag{20.1}$$

The offset terms o_i and observation weights w_i are both known in advance. The free parameters in a GLM are the coefficients $\boldsymbol{\beta}$ and, possibly, the parameter ϕ_0 controlling the dispersion. We will see examples where ϕ_0 is known (e.g. Poisson regression) and those where ϕ_0 is unknown (e.g. linear regression).

The "default" choice for the link function g is the *canonical link function*

$$g(\mu) = \dot{\psi}^{-1}(\mu),$$

which, given the relationship (19.2), gives $\eta = \dot{\psi}^{-1}(\mu) = \theta$, i.e. the linear predictor coincides with the natural parameter. As discussed in the context of equation (19.3), $\dot{\psi}^{-1}$ is a valid link function because it is monotonic and differentiable. Canonical link functions are very commonly used with GLMs because they lead to various nice properties that general GLMs do not enjoy (e.g. concave log-likelihood).

20.2 Examples

20.2.1 Example: Linear regression model

The linear regression model is a special case of a GLM, with $\phi_0 = \sigma^2$ (unknown), $w_i = 1$, $o_i = 0$, and identity (canonical) link function:

$$y_i \overset{\text{ind}}{\sim} N(\mu_i, \sigma^2); \quad \eta_i = \mu_i = \boldsymbol{x}_{i*}^T \boldsymbol{\beta}.$$

20.2.2 Example: Weighted linear regression model

If each observation y_i is the mean of m_i independent repeated observations, then we get a weighted linear regression model, with $\phi_0 = \sigma^2$ (unknown), $w_i = m_i$, $o_i = 0$, and identity (canonical) link function:

$$y_i \overset{\text{ind}}{\sim} N(\mu_i, \tfrac{\sigma^2}{m_i}); \quad \eta_i = \mu_i = \boldsymbol{x}_{i*}^T \boldsymbol{\beta}.$$

20.2.3 Example: Ungrouped logistic regression model

The *ungrouped logistic regression model* is the GLM based on the Bernoulli EDM with $\phi_0 = 1$ (known), $w_i = 1$, $o_i = 0$, and the canonical link function:

$$y_i \overset{\text{ind}}{\sim} \text{Ber}(\mu_i); \quad \eta_i = \theta_i = \log \frac{\mu_i}{1 - \mu_i} = \boldsymbol{x}_{i*}^T \boldsymbol{\beta}.$$

Thus the canonical link function for logistic regression is the *logistic link function* $g(\mu) = \log \frac{\mu}{1-\mu}$.

20.2.4 Example: Grouped logistic regression model

Suppose y_i is a binomial proportion based on m_i trials. The *grouped logistic regression model* is the GLM based on the binomial EDM with $\phi_0 = 1$ (known), $w_i = 1/m_i$, $o_i = 0$, and the canonical link function:

$$m_i y_i \sim \text{Bin}(m_i, \mu_i); \quad \eta_i = \log \frac{\mu_i}{1 - \mu_i} = o_i + \boldsymbol{x}_{i*}^T \boldsymbol{\beta}.$$

Note that a binomial proportion y_i based on m_i trials and a success probability of μ_i can be equivalently represented as m_i independent Bernoulli draws with the same success probability μ_i. Therefore, any grouped logistic regression model can be equivalently represented as an ungrouped logistic regression model with $\sum_{i=1}^n m_i$ observations. We will see that, despite this equivalence, grouped logistic regression models have some useful properties that ungrouped logistic regression models do not.

20.2.5 Example: Poisson regression model

Poisson regression is the Poisson EDM with $\phi_0 = 1$ (known), $w_i = 1$, $o_i = 0$, and the canonical link function:

$$y_i \overset{\text{ind}}{\sim} \text{Poi}(\mu_i); \quad \eta_i = \theta_i = \log \mu_i = \boldsymbol{x}_{i*}^T \boldsymbol{\beta}.$$

Thus the canonical link function for Poisson regression is the *log link function* $g(\mu) = \log \mu$.

Chapter 21

Parameter estimation

21.1 The GLM likelihood, score, and Fisher information

The log-likelihood of a GLM is:

$$\log \mathcal{L}(\boldsymbol{\beta}) = \sum_{i=1}^{n} \frac{\theta_i y_i - \psi(\theta_i)}{\phi_0 / w_i} + \sum_{i=1}^{n} \log h(y_i, \phi_0 / w_i). \tag{21.1}$$

Let's differentiate this with respect to $\boldsymbol{\beta}$, using the chain rule:

$$\begin{aligned}
\frac{\partial \log \mathcal{L}(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} &= \frac{\partial \log \mathcal{L}(\boldsymbol{\beta})}{\partial \boldsymbol{\theta}} \frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\mu}} \frac{\partial \boldsymbol{\mu}}{\partial \boldsymbol{\eta}} \frac{\partial \boldsymbol{\eta}}{\partial \boldsymbol{\beta}} \\
&= (\boldsymbol{y} - \boldsymbol{\mu})^T \mathrm{diag}(\phi_0 / w_i)^{-1} \cdot \mathrm{diag}(\ddot{\psi}(\theta_i))^{-1} \cdot \mathrm{diag}\left(\frac{\partial \mu_i}{\partial \eta_i}\right) \cdot \boldsymbol{X} \\
&= \frac{1}{\phi_0} (\boldsymbol{y} - \boldsymbol{\mu})^T \mathrm{diag}\left(\frac{w_i}{V(\mu_i)(d\eta_i / d\mu_i)^2}\right) \cdot \mathrm{diag}\left(\frac{\partial \eta_i}{\partial \mu_i}\right) \cdot \boldsymbol{X} \\
&\equiv \frac{1}{\phi_0} (\boldsymbol{y} - \boldsymbol{\mu})^T \boldsymbol{W} \boldsymbol{M} \boldsymbol{X}.
\end{aligned}$$

Here, $\boldsymbol{W} \equiv \mathrm{diag}(W_i)$ is a diagonal matrix of *working weights* and $\boldsymbol{M} \equiv \mathrm{diag}\left(\frac{\partial \eta_i}{\partial \mu_i}\right) = \mathrm{diag}(g'(\mu_i))$ is a diagonal matrix of link derivatives. Transposing, we get the score vector:

$$\boldsymbol{U}(\boldsymbol{\beta}) = \frac{1}{\phi_0} \boldsymbol{X}^T \boldsymbol{M} \boldsymbol{W} (\boldsymbol{y} - \boldsymbol{\mu}). \tag{21.2}$$

To get the Fisher information matrix, note first that:

$$\mathrm{Var}[\boldsymbol{y}] = \mathrm{diag}\left(\phi_0 \frac{V(\mu_i)}{w_i}\right) = \phi_0 \boldsymbol{W}^{-1} \boldsymbol{M}^{-2} \tag{21.3}$$

we can compute the covariance matrix of the score vector:

$$\begin{aligned} \boldsymbol{I}(\boldsymbol{\beta}) = \mathrm{Var}[\boldsymbol{U}(\boldsymbol{\beta})] &= \frac{1}{\phi_0^2}\boldsymbol{X}^T\boldsymbol{M}\boldsymbol{W}\mathrm{Var}[\boldsymbol{y}]\boldsymbol{M}\boldsymbol{W}\boldsymbol{X} \\ &= \frac{1}{\phi_0^2}\boldsymbol{X}^T\boldsymbol{M}\boldsymbol{W}\phi_0\boldsymbol{W}^{-1}\boldsymbol{M}^{-2}\boldsymbol{M}\boldsymbol{W}\boldsymbol{X} \\ &= \frac{1}{\phi_0}\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X}. \end{aligned} \tag{21.4}$$

21.2 Maximum likelihood estimation of $\boldsymbol{\beta}$

To estimate $\boldsymbol{\beta}$, we can set the score vector to zero:

$$\frac{1}{\phi_0}\boldsymbol{X}^T\widehat{\boldsymbol{M}}\widehat{\boldsymbol{W}}(\boldsymbol{y}-\widehat{\boldsymbol{\mu}}) = 0 \quad \Longleftrightarrow \quad \boldsymbol{X}^T\mathrm{diag}\left(\frac{w_i}{V(\widehat{\mu}_i)g'(\widehat{\mu}_i)}\right)(\boldsymbol{y}-\widehat{\boldsymbol{\mu}}) = 0.$$

These equations are called the *normal equations*. Unfortunately, unlike least squares, the normal equations cannot be solved analytically for $\widehat{\boldsymbol{\beta}}$. They are solved numerically instead; see Section 21.3. Note that ϕ_0 cancels from the normal equations, and therefore the coefficients $\boldsymbol{\beta}$ can be estimated without estimating the dispersion. Recall that we have seen this phenomenon for least squares. Also note that the normal equations simplify when the canonical link function is used, so that $\eta_i = \theta_i$. Assuming additionally that $w_i = 1$, we get:

$$\widehat{\boldsymbol{M}}\widehat{\boldsymbol{W}} = \mathrm{diag}\left(\frac{\widehat{\partial\mu_i/\partial\theta_i}}{V(\widehat{\mu}_i)}\right) = \frac{\ddot{\psi}(\widehat{\theta}_i)}{\ddot{\psi}(\widehat{\theta}_i)} = 1,$$

so the normal equations reduce to:

$$\boldsymbol{X}^T(\boldsymbol{y}-\widehat{\boldsymbol{\mu}}) = 0. \tag{21.5}$$

We recognize these as the normal equation for linear regression. Since both ungrouped logistic regression and Poisson regression also use canonical links and have unit weights, the simplified normal equations (21.5) apply to the latter regressions as well.

In the linear regression case, we interpreted the normal equations (21.5) as an orthogonality statement: $\boldsymbol{y}-\widehat{\boldsymbol{\mu}} \perp C(\boldsymbol{X})$. In the case of GLMs, the set $C(\boldsymbol{X}) \equiv \{\boldsymbol{\mu} = \mathbb{E}[\boldsymbol{y}] : \boldsymbol{\beta} \in \mathbb{R}^p\}$ is no longer a linear space. In fact, it is a nonlinear transformation of the column space of $\boldsymbol{X}$ (a p-dimensional manifold in $\mathbb{R}^n$):

$$C(\boldsymbol{X}) \equiv \{\boldsymbol{\mu} = \mathbb{E}[\boldsymbol{y}] : \boldsymbol{\beta} \in \mathbb{R}^p\} = \{g^{-1}(\boldsymbol{X}\boldsymbol{\beta}) : \boldsymbol{\beta} \in \mathbb{R}^p\}.$$

Therefore, we cannot view the mapping $\boldsymbol{y} \mapsto \widehat{\boldsymbol{\mu}}$ as a linear projection. Nevertheless, it is possible to interpret $\widehat{\boldsymbol{\mu}}$ as the "closest" point (in some sense) to $\boldsymbol{y}$ in $C(\boldsymbol{X})$. To see this, recall the deviance form of the EDM density (19.6). Taking a logarithm and summing over $i = 1, \ldots, n$, we find the following expression for the negative log likelihood:

$$
\begin{aligned}
-\log \mathcal{L}(\boldsymbol{\beta}) &= \sum_{i=1}^{n} \frac{d(y_i, \mu_i)}{2\phi_i} + C \\
&= \frac{\sum_{i=1}^{n} w_i d(y_i, \mu_i)}{2\phi_0} + C \\
&\equiv \frac{D(\boldsymbol{y}, \boldsymbol{\mu})}{2\phi_0} + C \\
&\equiv \frac{1}{2} D^*(\boldsymbol{y}, \boldsymbol{\mu}) + C.
\end{aligned}
\tag{21.6}
$$

$D(\boldsymbol{y}, \boldsymbol{\mu})$ is called the *deviance* or the *total deviance*, and it can be interpreted as a kind of distance between the mean vector $\boldsymbol{\mu}$ and the observation vector $\boldsymbol{y}$. For example, in the linear model case, $D(\boldsymbol{y}, \boldsymbol{\mu}) = \|\boldsymbol{y} - \boldsymbol{\mu}\|^2$. The quantity $D^*(\boldsymbol{y}, \boldsymbol{\mu})$ is called the *scaled deviance.* In the linear model case, $D^*(\boldsymbol{y}, \boldsymbol{\mu}) = \frac{\|\boldsymbol{y}-\boldsymbol{\mu}\|^2}{\sigma^2}$. Therefore, maximizing the GLM log likelihood is equivalent to minimizing the deviance:

$$
\widehat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \; D(\boldsymbol{y}, \boldsymbol{\mu}(\boldsymbol{\beta})), \quad \text{so that} \quad \widehat{\boldsymbol{\mu}} = \arg\min_{\boldsymbol{\mu} \in C(\boldsymbol{X})} \; D(\boldsymbol{y}, \boldsymbol{\mu}).
$$

21.3 Iteratively reweighted least squares

21.3.1 Log-concavity of GLM likelihood

Before talking about maximizing the GLM log-likelihood, we investigate the concavity of this function. We claim that, in the case when the canonical link is used, $\log \mathcal{L}(\boldsymbol{\beta})$ is a concave function of $\boldsymbol{\beta}$, which implies that this function is "easy to optimize", i.e., has no local maxima.

Proposition 21.1. ***Proposition:*** *If g is the canonical link function, then the function* $\log \mathcal{L}(\boldsymbol{\beta})$ *defined in 21.1 is concave in $\boldsymbol{\beta}$.*

Proof. It suffices to show that ψ is a convex function since then $\log \mathcal{L}(\boldsymbol{\beta})$ would be the sum of a linear function of $\boldsymbol{\beta}$ and the composition of a concave function with a linear function. To verify that ψ is convex, it suffices to recall that $\ddot{\psi}(\theta) = \frac{1}{\phi}\mathrm{Var}_\theta[y] > 0$.

□

Proposition 21.1 gives us confidence that an iterative algorithm will converge to the global maximum of the likelihood. We present such an iterative algorithm next.

21.3.2 Newton-Raphson

We can maximize the log-likelihood (21.1) via the Newton-Raphson algorithm, which involves the gradient and Hessian of the function we would like to maximize. We derive the Newton-Raphson algorithm for canonical GLMs. In this case, the gradient is the score vector (21.2), while the Hessian is the Fisher information (21.4).[1] The Newton-Raphson iteration is therefore:

[1]The Fisher information is the expectation of the Hessian, but for canonical links, the Hessian is non-random, so the two coincide.

$$\begin{aligned}\widehat{\boldsymbol{\beta}}^{(t+1)} &= \widehat{\boldsymbol{\beta}}^{(t)} - (\nabla^2_{\boldsymbol{\beta}} \log \mathcal{L}(\widehat{\boldsymbol{\beta}}^{(t)}))^{-1} \nabla_{\boldsymbol{\beta}} \log \mathcal{L}(\widehat{\boldsymbol{\beta}}^{(t)}) \\ &= \widehat{\boldsymbol{\beta}}^{(t)} + (\boldsymbol{X}^T \widehat{\boldsymbol{W}}^{(t)} \boldsymbol{X})^{-1} \boldsymbol{X}^T (\boldsymbol{y} - \widehat{\boldsymbol{\mu}}^{(t)}).\end{aligned} \tag{21.7}$$

See Figure 21.1.

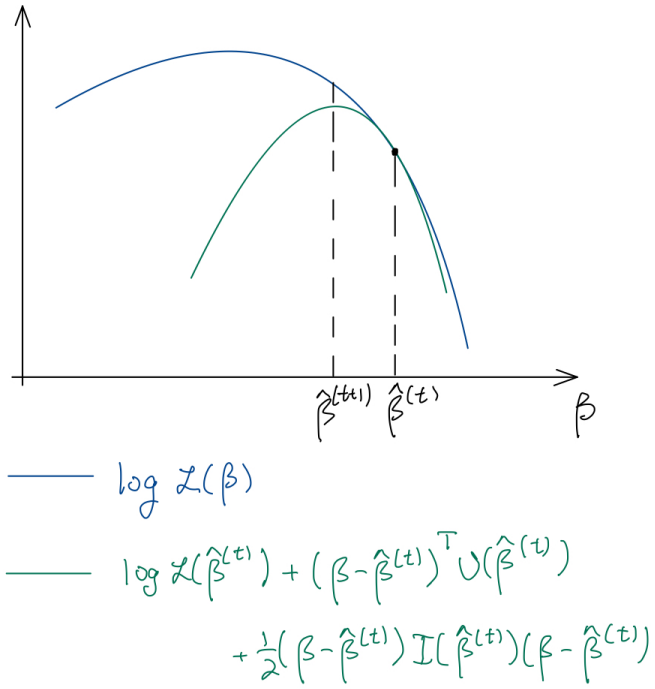


Figure 21.1: Newton-Raphson iteratively approximates the log likelihood via a quadratic function and maximizing that function.

21.3.3 Iteratively reweighted least squares (IRLS)

A nice interpretation of the Newton-Raphson algorithm is as a sequence of weighted least squares fits, known as the iteratively reweighted least squares (IRLS) algorithm. Suppose that we have a current estimate $\widehat{\boldsymbol{\beta}}^{(t)}$, and suppose we are looking for a vector $\boldsymbol{\beta}$ near $\widehat{\boldsymbol{\beta}}^{(t)}$ that fits the model even better. We have:

$$\begin{aligned}\mathbb{E}_{\beta}[\boldsymbol{y}] &= g^{-1}(\boldsymbol{X}\boldsymbol{\beta}) \\ &\approx g^{-1}(\boldsymbol{X}\widehat{\boldsymbol{\beta}}^{(t)}) + \operatorname{diag}(\partial \mu_i / \partial \eta_i)(\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}^{(t)}) \\ &= \widehat{\boldsymbol{\mu}}^{(t)} + (\widehat{\boldsymbol{M}}^{(t)})^{-1}(\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}^{(t)})\end{aligned}$$

and

$$\operatorname{Var}_{\beta}[\boldsymbol{y}] \approx \phi_0 (\widehat{\boldsymbol{W}}^{(t)})^{-1} (\widehat{\boldsymbol{M}}^{(t)})^{-2} = \phi_0 \widehat{\boldsymbol{W}}^{(t)},$$

recalling equation (21.3). Thus, up to the first two moments, near $\boldsymbol{\beta} = \widehat{\boldsymbol{\beta}}^{(t)}$ the distribution of $\boldsymbol{y}$ is approximately:

$$\begin{aligned} \boldsymbol{y} = \widehat{\boldsymbol{\mu}}^{(t)} + (\widehat{\boldsymbol{M}}^{(t)})^{-1}(\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}^{(t)}) + \boldsymbol{\epsilon}, \\ \boldsymbol{\epsilon} \sim N(\boldsymbol{0}, \phi_0 \widehat{\boldsymbol{W}}^{(t)}), \end{aligned}$$

or, equivalently:

$$\begin{aligned} \boldsymbol{z}^{(t)} \equiv \widehat{\boldsymbol{M}}^{(t)}(\boldsymbol{y} - \widehat{\boldsymbol{\mu}}^{(t)}) + \boldsymbol{X}\widehat{\boldsymbol{\beta}}^{(t)} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}', \\ \boldsymbol{\epsilon}' \sim N(\boldsymbol{0}, \phi_0 (\widehat{\boldsymbol{W}}^{(t)})^{-1}). \end{aligned} \tag{21.8}$$

The regression of the *adjusted response variable* $\boldsymbol{z}^{(t)}$ on $\boldsymbol{X}$ leaves us with a weighted linear regression (hence the name *working weights* for W_i), whose maximum likelihood estimate is:

$$\widehat{\boldsymbol{\beta}}^{(t+1)} = (\boldsymbol{X}^T \widehat{\boldsymbol{W}}^{(t)} \boldsymbol{X})^{-1} \boldsymbol{X}^T \widehat{\boldsymbol{W}}^{(t)} \boldsymbol{z}^{(t)}, \tag{21.9}$$

which we define as our next iterate. It's easy to verify that the IRLS iteration (21.9) is equivalent to the Newton-Raphson iteration (21.7). Note that we have derived these algorithms for canonical links; they each can be derived for non-canonical links but need not be equivalent in this more general case.

21.4 Estimation of ϕ_0 and GLM residuals

While sometimes the parameter ϕ_0 is known (e.g., for binomial or Poisson GLMs), in other cases ϕ_0 must be estimated (e.g., for the normal linear model). Recall from the linear model that we estimated $\sigma^2 = \phi_0$ by taking the sum of the squares of the residuals: $\widehat{\sigma}^2 = \frac{1}{n-p}\|\boldsymbol{y} - \widehat{\boldsymbol{\mu}}\|^2$. However, it's unclear in the GLM context exactly how to define a residual. In fact, there are two common ways of doing so, called *deviance residuals* and *Pearson residuals*. Deviance residuals are defined in terms of the unit deviance:

$$r_i^D \equiv \text{sign}(y_i - \widehat{\mu}_i)\sqrt{w_i d(y_i, \widehat{\mu}_i)}.$$

On the other hand, Pearson residuals are defined as variance-normalized residuals:

$$r_i^P \equiv \frac{y_i - \widehat{\mu}_i}{\sqrt{V(\widehat{\mu}_i)/w_i}}.$$

These residuals can be viewed as residuals from the (converged) weighted linear regression model (21.8). In the normal case, these residuals coincide, but in the general case, they do not. Based on these two notions of GLM residuals, we can define two estimators of ϕ_0. One, based on the deviance residuals, is the *mean deviance estimator of dispersion*:

$$\widetilde{\phi}_0^D \equiv \frac{1}{n-p}\|r^D\|^2 \equiv \frac{1}{n-p}\sum_{i=1}^n w_i d(y_i, \widehat{\mu}_i) \equiv \frac{1}{n-p} D(\boldsymbol{y}; \widehat{\boldsymbol{\mu}});$$

recall that the total deviance $D(\boldsymbol{y}; \widehat{\boldsymbol{\mu}})$ is a generalization of the residual sum of squares. The other, based on the Pearson residuals, is called the *Pearson estimator of dispersion*:

$$\begin{aligned} \widetilde{\phi}_0^P &\equiv \frac{1}{n-p} X^2 \\ &\equiv \frac{1}{n-p} \|r^P\|^2 \\ &\equiv \frac{1}{n-p} \sum_{i=1}^{n} w_i \frac{(y_i - \widehat{\mu}_i)^2}{V(\mu_i)}. \end{aligned} \tag{21.10}$$

X^2 is known as the Pearson X^2 statistic. The deviance-based estimator can be more accurate than the Pearson estimator under small-dispersion asymptotics. However, the Pearson estimator is more robust when only the first two moments of the EDM model are correct and in the absence of small-dispersion asymptotics. For these reasons, the Pearson estimator is generally preferred.

Chapter 22

Inference in GLMs

22.1 Preliminaries

22.1.1 Inferential goals

There are two types of inferential goals: hypothesis testing and confidence interval/region construction.

22.1.1.1 Hypothesis testing

1. **Single coefficient**: $H_0 : \beta_j = \beta_j^0$ versus $H_1 : \beta_j \neq \beta_j^0$ for some $\beta_j^0 \in \mathbb{R}$.
2. **Group of coefficients**: $H_0 : \boldsymbol{\beta}_S = \boldsymbol{\beta}_S^0$ versus $H_1 : \boldsymbol{\beta}_S \neq \boldsymbol{\beta}_S^0$ for some $S \subset \{0, \ldots, p-1\}$ and some $\boldsymbol{\beta}_S^0 \in \mathbb{R}^{|S|}$.
3. **Goodness of fit**: The goodness of fit null hypothesis is that the GLM (20.1) is correctly specified. Consider the *saturated model*:

$$y_i \overset{\text{ind}}{\sim} \text{EDM}(\mu_i, \phi_0/w_i) \quad \text{for} \quad i = 1, \ldots, n. \tag{22.1}$$

Let

$$\mathcal{M}^{\text{GLM}} \equiv \{\boldsymbol{\mu} : g(\mu_i) = \boldsymbol{x}_{i*}^T \boldsymbol{\beta} + o_i \text{ for some } \boldsymbol{\beta} \in \mathbb{R}^p\}$$

be the set of mean vectors consistent with the GLM. Then, the goodness of fit testing problem is $H_0 : \boldsymbol{\mu} \in \mathcal{M}^{\text{GLM}}$ versus $H_1 : \boldsymbol{\mu} \notin \mathcal{M}^{\text{GLM}}$.

22.1.1.2 Confidence interval/region construction

1. **Confidence interval for a single coefficient**: Here, the goal is to produce a confidence interval $\text{CI}(\beta_j)$ for a coefficient β_j.
2. **Confidence region for a group of coefficients**: Here, the goal is to produce a confidence region $\text{CR}(\boldsymbol{\beta}_S)$ for a group of coefficients $\boldsymbol{\beta}_S$.
3. **Confidence interval for a fitted value**: In GLMs, fitted values can either be considered for parameters on the linear scale ($\eta_i = \boldsymbol{x}_{i*}^T \boldsymbol{\beta} + o_i$) or the mean scale ($\mu_i = g^{-1}(\boldsymbol{x}_{i*}^T \boldsymbol{\beta} + o_i)$). The goal, then, is to produce confidence intervals $\text{CI}(\eta_i)$ or $\text{CI}(\mu_i)$ for η_i or μ_i, respectively.

22.1.2 Inferential tools

Inference in GLMs is based on asymptotic likelihood theory. These asymptotics can be based on *large-sample asymptotics* or *small-dispersion asymptotics*. Large-sample asymptotics are applicable for testing hypotheses and estimating parameters within models where the number of parameters is fixed while the sample size grows. Small-dispersion asymptotics are applicable for testing hypotheses and estimating parameters within models where the dispersion is small, regardless of the sample size. Large-sample asymptotics apply to testing and estimating coefficients in GLMs (20.1) with a fixed number of parameters as the sample size grows, but not to testing goodness of fit. Indeed, goodness-of-fit tests refer to the saturated model (22.1), whose number of parameters grows with n. Small-dispersion asymptotics, on the other hand, apply to goodness-of-fit testing.

Hypothesis tests (and, by inversion, confidence intervals) can be constructed in three asymptotically equivalent ways: Wald tests, likelihood ratio tests (LRT), and score tests. These tests can be justified using either large-sample or small-dispersion asymptotics, depending on the context. Despite their asymptotic equivalence, in finite samples, some tests may be preferable to others (though for normal linear models, these tests are equivalent in finite samples as well). See Figure 22.1.

Inference method	Testing individual coefficients	Testing groups of coefficients	Testing goodness of fit	Confidence intervals for individual coefficients	Confidence regions	Confidence intervals for fitted values	Accurate in small samples
Wald	✓	✓	✗	✓	✓	✓	✗
Likelihood ratio	✓	✓	✓	✓*	✗	✗	✓
Score	✓	✓	✓	✓*	✗	✗	✓

★ Computationally slower

Figure 22.1: A comparison of the three asymptotic methods for GLM inference.

22.2 Wald inference

Wald inference is based on the following asymptotic normality statement:

$$\widehat{\boldsymbol{\beta}} \dot{\sim} N(\boldsymbol{\beta}, \boldsymbol{I}^{-1}(\boldsymbol{\beta})) = N(\boldsymbol{\beta}, \phi_0(\boldsymbol{X}^T\boldsymbol{W}(\boldsymbol{\beta})\boldsymbol{X})^{-1}), \tag{22.2}$$

recalling our derivation of the Fisher information from equation (21.4). This normal approximation can be justified via the central limit theorem in the context of *large-sample asymptotics* or *small-dispersion asymptotics*. Wald inference is easy to carry out, and for this reason, it is considered the default type of inference. However, as we will see in Unit 5, it also tends to be the least accurate in small samples. Furthermore, Wald tests are usually not applied for testing goodness of fit.

22.2.1 Wald test for $\beta_j = \beta_j^0$ (known ϕ_0)

Based on the Wald approximation (22.2), under the null hypothesis, we have:

$$\begin{aligned}\widehat{\beta}_j &\dot{\sim} N(\beta_j^0, \phi_0[(\boldsymbol{X}^T\boldsymbol{W}(\boldsymbol{\beta})\boldsymbol{X})^{-1}]_{jj}) \\ &\approx N(\beta_j^0, \phi_0[(\boldsymbol{X}^T\boldsymbol{W}(\widehat{\boldsymbol{\beta}})\boldsymbol{X})^{-1}]_{jj}) \\ &\equiv N(\beta_j^0, \mathrm{SE}(\widehat{\beta}_j)^2),\end{aligned}$$

where we have used a plug-in estimator of the variance. This leads us to the Wald z-test:

$$\phi(\boldsymbol{X}, \boldsymbol{y}) \equiv 1\left(\left|\frac{\widehat{\beta}_j - \beta_j^0}{\mathrm{SE}(\widehat{\beta}_j)}\right| > z_{1-\alpha/2}\right).$$

Since a one-dimensional parameter is being tested, we can make the test one-sided if desired.

22.2.2 Wald test for $\boldsymbol{\beta}_S = \boldsymbol{\beta}_S^0$ (known ϕ_0)

Extending the reasoning above, we have under the null hypothesis that:

$$\widehat{\boldsymbol{\beta}}_S \dot{\sim} N(\boldsymbol{\beta}_S^0, \phi_0[(\boldsymbol{X}^T\boldsymbol{W}(\boldsymbol{\beta})\boldsymbol{X})^{-1}]_{S,S}) \approx N(\boldsymbol{\beta}_S^0, \phi_0[(\boldsymbol{X}^T\boldsymbol{W}(\widehat{\boldsymbol{\beta}})\boldsymbol{X})^{-1}]_{S,S}),$$

and therefore:

$$\frac{1}{\phi_0}(\widehat{\boldsymbol{\beta}}_S - \boldsymbol{\beta}_S^0)^T\left([(\boldsymbol{X}^T\boldsymbol{W}(\widehat{\boldsymbol{\beta}})\boldsymbol{X})^{-1}]_{S,S}\right)^{-1}(\widehat{\boldsymbol{\beta}}_S - \boldsymbol{\beta}_S^0) \dot{\sim} \chi^2_{|S|}.$$

Hence, we have the Wald χ^2 test:

$$\begin{aligned}&\phi(\boldsymbol{X}, \boldsymbol{y}) \\ &\equiv 1\left(\frac{1}{\phi_0}(\widehat{\boldsymbol{\beta}}_S - \boldsymbol{\beta}_S^0)^T\left([(\boldsymbol{X}^T\boldsymbol{W}(\widehat{\boldsymbol{\beta}})\boldsymbol{X})^{-1}]_{S,S}\right)^{-1}(\widehat{\boldsymbol{\beta}}_S - \boldsymbol{\beta}_S^0) > \chi^2_{|S|}(1-\alpha)\right).\end{aligned}$$

22.2.3 Wald confidence interval for β_j (known ϕ_0)

Inverting the Wald test for β_j, we get a Wald confidence interval:

$$\mathrm{CI}(\beta_j) \equiv \widehat{\beta}_j \pm z_{1-\alpha/2} \cdot \mathrm{SE}(\widehat{\beta}_j), \tag{22.3}$$

where

$$\mathrm{SE}(\widehat{\beta}_j) \equiv \sqrt{\phi_0[(\boldsymbol{X}^T\boldsymbol{W}(\widehat{\boldsymbol{\beta}})\boldsymbol{X})^{-1}]_{jj}}.$$

22.2.4 Wald confidence region for $\boldsymbol{\beta}_S$ (known ϕ_0)

By inverting the test of $H_0 : \boldsymbol{\beta}_S = \boldsymbol{\beta}_S^0$, we get the Wald confidence region:

$$\mathrm{CR}(\boldsymbol{\beta}_S) \equiv \left\{ \boldsymbol{\beta}_S : \frac{1}{\phi_0}(\widehat{\boldsymbol{\beta}}_S - \boldsymbol{\beta}_S)^T \left([(\boldsymbol{X}^T \boldsymbol{W}(\widehat{\boldsymbol{\beta}})\boldsymbol{X})^{-1}]_{S,S} \right)^{-1} (\widehat{\boldsymbol{\beta}}_S - \boldsymbol{\beta}_S) \leq \chi^2_{|S|}(1-\alpha) \right\}.$$

If $S = \{0, 1, \ldots, p-1\}$, we are left with:

$$\mathrm{CR}(\boldsymbol{\beta}_S) \equiv \left\{ \boldsymbol{\beta} : \frac{1}{\phi_0}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta})^T \boldsymbol{X}^T \boldsymbol{W}(\widehat{\boldsymbol{\beta}})\boldsymbol{X}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}) \leq \chi^2_p(1-\alpha) \right\}.$$

22.2.5 Wald confidence intervals for η_i and μ_i (known ϕ_0)

Given the Wald approximation (22.2), we have:

$$\widehat{\eta}_i \equiv o_i + \boldsymbol{x}_{i*}^T \widehat{\boldsymbol{\beta}} \dot{\sim} N(\eta_i, \phi_0 \cdot \boldsymbol{x}_{i*}^T (\boldsymbol{X}^T \boldsymbol{W}(\widehat{\boldsymbol{\beta}})\boldsymbol{X})^{-1} \boldsymbol{x}_{i*}) \equiv N(\eta_i, \mathrm{SE}(\hat{\eta}_i)^2).$$

Hence, the Wald interval for η_i is:

$$\mathrm{CI}(\eta_i) \equiv o_i + \boldsymbol{x}_{i*}^T \widehat{\boldsymbol{\beta}} \pm z_{1-\alpha/2} \cdot \mathrm{SE}(\hat{\eta}_i),$$

where

$$\mathrm{SE}(\hat{\eta}_i) \equiv \sqrt{\phi_0 \boldsymbol{x}_{i*}^T (\boldsymbol{X}^T \boldsymbol{W}(\widehat{\boldsymbol{\beta}})\boldsymbol{X})^{-1} \boldsymbol{x}_{i*}}.$$

A confidence interval for $\mu_i \equiv \mathbb{E}_\beta[y_i] = g^{-1}(\eta_i)$ can be obtained by applying the monotonic function g^{-1} to the endpoints of the confidence interval for η_i. Note that the resulting confidence interval may be asymmetric. We can get a symmetric interval by applying the delta method, but this interval would be less accurate because it involves the delta method approximation in addition to the Wald approximation.

22.2.6 Wald inference when ϕ_0 is unknown

When ϕ_0 is unknown, we need to plug in an estimate $\widetilde{\phi}_0$ (e.g. the deviance-based or Pearson-based estimate). Now our standard errors are

$$\mathrm{SE}(\widehat{\beta}_j) \equiv \sqrt{\widetilde{\phi}_0 \cdot [(\boldsymbol{X}^T \boldsymbol{W}(\widehat{\boldsymbol{\beta}})\boldsymbol{X})^{-1}]_{jj}},$$

and our test statistic for $H_0 : \beta_j = \beta_j^0$ is

$$\frac{\widehat{\beta}_j - \beta_j^0}{\sqrt{\widetilde{\phi}_0}\sqrt{[(\boldsymbol{X}^T \boldsymbol{W}(\widehat{\boldsymbol{\beta}})\boldsymbol{X})^{-1}]_{jj}}}.$$

Unlike linear regression, it is not the case in general that $\widehat{\boldsymbol{\beta}}$ and $\widetilde{\phi}_0$ are independent. Nevertheless, they are *asymptotically independent*. Therefore, the above statistic is *approximately* distributed as t_{n-p}. Hence, the test for $H_0 : \beta_j = \beta_j^0$ is:

$$\phi(\boldsymbol{X}, \boldsymbol{y}) \equiv 1\left(\left|\frac{\widehat{\beta}_j - \beta_j^0}{\mathrm{SE}(\widehat{\beta}_j)}\right| > t_{n-p}(1-\alpha/2)\right).$$

Likewise, we would replace $z_{1-\alpha}$ by $t_{n-p}(1-\alpha/2)$ for all tests and confidence intervals concerning univariate quantities. For multivariate quantities, we will get approximate F distributions instead of approximate χ^2 distributions. For example:

$$\frac{\frac{1}{|S|}(\widehat{\boldsymbol{\beta}}_S - \boldsymbol{\beta}_S^0)^T\left([(\boldsymbol{X}^T\boldsymbol{W}(\widehat{\boldsymbol{\beta}})\boldsymbol{X})^{-1}]_{S,S}\right)^{-1}(\widehat{\boldsymbol{\beta}}_S - \boldsymbol{\beta}_S^0)}{\widetilde{\phi}_0} \dot{\sim} F_{|S|,n-p}.$$

22.3 Likelihood ratio inference

22.3.1 Testing one or more coefficients (ϕ_0 known)

Let $\ell(\boldsymbol{y}, \boldsymbol{\mu}) = -\frac{D(\boldsymbol{y},\boldsymbol{\mu})}{2\phi_0} + C$ be the GLM log-likelihood (recall equation (21.6)). Let $H_0 : \boldsymbol{\beta}_S = \boldsymbol{\beta}_S^0$ be a null hypothesis about some subset of variables $S \subset \{0, 1, \ldots, p-1\}$, and let $\widehat{\boldsymbol{\mu}}_{\text{-}S}$ be the maximum likelihood estimate under the null hypothesis. Likelihood ratio inference is based on the following asymptotic chi-square distribution:

$$2(\ell(\boldsymbol{y}, \widehat{\boldsymbol{\mu}}) - \ell(\boldsymbol{y}, \widehat{\boldsymbol{\mu}}_{\text{-}S})) = \frac{D(\boldsymbol{y}, \widehat{\boldsymbol{\mu}}_{\text{-}S}) - D(\boldsymbol{y}, \widehat{\boldsymbol{\mu}})}{\phi_0} \dot{\sim} \chi^2_{|S|}. \tag{22.4}$$

This approximation holds either in large samples (large-sample asymptotics) or in small samples but with small dispersion (small-dispersion asymptotics). The latter has to do with the fact that under small-dispersion asymptotics,

$$\frac{d(y_i, \mu_i)}{\phi_0/w_i} \dot{\sim} \chi^2_1,$$

so

$$\frac{D(\boldsymbol{y}, \boldsymbol{\mu})}{\phi_0} = \sum_{i=1}^{n} \frac{d(y_i, \mu_i)}{\phi_0/w_i} \dot{\sim} \chi^2_n.$$

Suppose we wish to test the null hypothesis $H_0 : \boldsymbol{\beta}_S = \boldsymbol{\beta}_S^0$. Then, based on the approximation (22.4), we can define the likelihood ratio test:

$$\phi(\boldsymbol{X}, \boldsymbol{y}) \equiv 1\left(\frac{D(\boldsymbol{y}, \widehat{\boldsymbol{\mu}}_{\text{-}S}) - D(\boldsymbol{y}, \widehat{\boldsymbol{\mu}})}{\phi_0} > \chi^2_{|S|}(1-\alpha)\right).$$

22.3.2 Confidence interval for a single coefficient

We can obtain a confidence interval for β_j by inverting the likelihood ratio test. Let $\widehat{\boldsymbol{\mu}}_{\text{-}j}(\beta_j^0)$ be the fitted mean vector under the constraint $\beta_j = \beta_j^0$. Then, inverting the likelihood ratio test gives us the confidence interval:

$$\text{CI}(\beta_j) \equiv \left\{ \beta_j : \frac{D(\boldsymbol{y}, \widehat{\boldsymbol{\mu}}_{\text{-}j}(\beta_j)) - D(\boldsymbol{y}, \widehat{\boldsymbol{\mu}})}{\phi_0} \leq \chi^2_{|S|}(1-\alpha) \right\}.$$

Likelihood ratio-based confidence intervals tend to be more accurate than Wald intervals, especially when the parameter is near the edge of the parameter space, but they require more computation because $\widehat{\boldsymbol{\mu}}_{\text{-}j}(\beta_j)$ must be computed on a large grid of β_j values. If we wanted to create *confidence regions* for groups of parameters, this would become computationally intensive due to the curse of dimensionality.

22.3.3 Goodness of fit testing (ϕ_0 known)

For ϕ_0 known, we can also construct a goodness of fit test. To this end, we compare the deviances of the GLM and saturated model:

$$\frac{D(\boldsymbol{y}, \widehat{\boldsymbol{\mu}}) - D(\boldsymbol{y}, \boldsymbol{y})}{\phi_0} = \frac{D(\boldsymbol{y}; \widehat{\boldsymbol{\mu}})}{\phi_0} \mathrel{\dot\sim} \chi^2_{n-p}.$$

Note that the goodness of fit test is a significance test with respect to the saturated model (22.1), which has n free parameters. Therefore, the number of free parameters increases with the sample size, so large-sample asymptotics cannot justify this test. Instead, we must rely on small-dispersion asymptotics. In particular, the likelihood ratio test relies on the saddlepoint approximation for small dispersions. To verify whether the saddlepoint approximation is accurate, we can apply the rules of thumb from Section 19.6.2.2 for each observation y_i, when it is drawn from the distribution fitted under the GLM (rather than the saturated model). For instance, we can check that $m\hat{\mu}_i \geq 3$ and $m(1-\hat{\mu}_i) \geq 3$ in the case of grouped logistic regression of $\hat{\mu}_i \geq 3$ for Poisson regression. Here, $\hat{\mu}_i$ are the fitted means under the GLM.

22.3.4 Likelihood ratio inference for ϕ_0 unknown

If ϕ_0 is unknown, we can estimate it as discussed above and construct an F-statistic as follows:

$$F \equiv \frac{(D(\boldsymbol{y}; \widehat{\boldsymbol{\mu}}_{\text{-}S}) - D(\boldsymbol{y}; \widehat{\boldsymbol{\mu}}))/|S|}{\widetilde{\phi}_0}.$$

In normal linear model theory, the null distribution of F is *exactly* $F_{|S|,n-p}$. For GLMs, the null distribution of F is *approximately* $F_{|S|,n-p}$. We can use this F distribution to construct hypothesis tests for groups of coefficients, or invert it to get a confidence interval for a single coefficient. We cannot construct a goodness of fit test in the case that ϕ_0 is unknown because the residual degrees of freedom would be used up to estimate ϕ_0 rather than to carry out inference.

22.4 Score-based inference

Score-based inference can be used for the same set of inferential tasks as likelihood ratio inference.

22.4.1 Testing multiple coefficients (ϕ_0 known)

Let $H_0 : \boldsymbol{\beta}_S = \boldsymbol{\beta}_S^0$ be a null hypothesis about a subset of variables S, and let $\widehat{\boldsymbol{\beta}}^0$ be the maximum likelihood estimate under this null hypothesis. In particular, let $\widehat{\boldsymbol{\beta}}_S^0 \equiv \boldsymbol{\beta}_S^0$ and

$$\widehat{\boldsymbol{\beta}}_{\text{-}S}^0 = \arg\max_{\boldsymbol{\beta}_{\text{-}S}} \ell(\boldsymbol{\beta}_S^0, \boldsymbol{\beta}_{\text{-}S}). \tag{22.5}$$

Let us partition the score vector $\boldsymbol{U}(\boldsymbol{\beta}) = \frac{\partial \ell(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} \equiv (\boldsymbol{U}_S(\boldsymbol{\beta}), \boldsymbol{U}_{\text{-}S}(\boldsymbol{\beta}))$. We have

$$\begin{aligned} \boldsymbol{U}(\boldsymbol{\beta}) &= \begin{pmatrix} \boldsymbol{U}_S(\boldsymbol{\beta}) \\ \boldsymbol{U}_{\text{-}S}(\boldsymbol{\beta}) \end{pmatrix} \\ &\sim N\left(0, \boldsymbol{I}(\boldsymbol{\beta})\right) = N\left(0, \begin{bmatrix} \boldsymbol{I}_{S,S}(\boldsymbol{\beta}) & \boldsymbol{I}_{S,\text{-}S}(\boldsymbol{\beta}) \\ \boldsymbol{I}_{\text{-}S,S}(\boldsymbol{\beta}) & \boldsymbol{I}_{\text{-}S,\text{-}S}(\boldsymbol{\beta}) \end{bmatrix}\right). \end{aligned} \tag{22.6}$$

This approximation can be justified either by small-dispersion asymptotics or large-sample asymptotics (both based on the central limit theorem). The score test statistic is based on plugging in the null estimate $\widehat{\boldsymbol{\beta}}^0$ into the score vector and extracting the components corresponding to S:

$$\boldsymbol{U}_S(\widehat{\boldsymbol{\beta}}^0).$$

This vector does not have the distribution obtained from the coordinates S of equation (22.6) because an estimate is plugged in. Instead, we can derive the distribution of this vector by conditioning on $\boldsymbol{U}_{\text{-}S}(\widehat{\boldsymbol{\beta}}^0) = 0$:

$$\begin{aligned} \boldsymbol{U}_S(\widehat{\boldsymbol{\beta}}^0) &\overset{d}{\approx} \mathcal{L}(\boldsymbol{U}_S(\boldsymbol{\beta}) \mid \boldsymbol{U}_{\text{-}S}(\boldsymbol{\beta}) = 0)|_{\boldsymbol{\beta}=\widehat{\boldsymbol{\beta}}^0} \\ &\dot{\sim} N\left(0, \boldsymbol{I}_{S,S}(\widehat{\boldsymbol{\beta}}^0) - \boldsymbol{I}_{S,\text{-}S}(\widehat{\boldsymbol{\beta}}^0)\boldsymbol{I}_{\text{-}S,\text{-}S}(\widehat{\boldsymbol{\beta}}^0)^{-1}\boldsymbol{I}_{\text{-}S,S}(\widehat{\boldsymbol{\beta}}^0)\right). \end{aligned} \tag{22.7}$$

The second line is obtained from (22.6) using the formula for a conditional distribution of a multivariate normal distribution. Therefore, the score test is based on the following chi-square approximation:

$$\boldsymbol{U}_S(\widehat{\boldsymbol{\beta}}^0)^T \left[\boldsymbol{I}_{S,S}(\widehat{\boldsymbol{\beta}}^0) - \boldsymbol{I}_{S,\text{-}S}(\widehat{\boldsymbol{\beta}}^0)\boldsymbol{I}_{\text{-}S,\text{-}S}(\widehat{\boldsymbol{\beta}}^0)^{-1}\boldsymbol{I}_{\text{-}S,S}(\widehat{\boldsymbol{\beta}}^0)\right]^{-1} \boldsymbol{U}_S(\widehat{\boldsymbol{\beta}}^0) \dot{\sim} \chi^2_{|S|}.$$

Recalling the expressions for the score (21.2) and Fisher information matrix (21.4) in GLMs, we derive the score statistic

$$\begin{aligned} &T^2_{\text{score}}(\boldsymbol{X}, \boldsymbol{y}) \equiv \\ &\frac{1}{\phi_0}(\boldsymbol{y} - \widehat{\boldsymbol{\mu}}^0)^T \widehat{\boldsymbol{W}}^0 \widehat{\boldsymbol{M}}^0 \boldsymbol{X}_{*,S} \times \\ &\quad \left[\boldsymbol{X}_{*,S}^T \widehat{\boldsymbol{W}}^0 \boldsymbol{X}_{*,S} - \boldsymbol{X}_{*,S}^T \widehat{\boldsymbol{W}}^0 \boldsymbol{X}_{*,\text{-}S} (\boldsymbol{X}_{*,\text{-}S}^T \widehat{\boldsymbol{W}}^0 \boldsymbol{X}_{*,\text{-}S})^{-1} \boldsymbol{X}_{*,\text{-}S}^T \widehat{\boldsymbol{W}}^0 \boldsymbol{X}_{*,S}\right]^{-1} \times \\ &\quad \boldsymbol{X}_{*,S}^T \widehat{\boldsymbol{M}}^0 \widehat{\boldsymbol{W}}^0 (\boldsymbol{y} - \widehat{\boldsymbol{\mu}}^0). \end{aligned}$$

The score test is therefore

$$\phi(\boldsymbol{X}, \boldsymbol{y}) \equiv 1(T^2_{\text{score}}(\boldsymbol{X}, \boldsymbol{y}) > \chi^2_{|S|}(1 - \alpha)).$$

An equivalent formulation of the score test can be derived by noting that

$$\left[I_{S,S}(\widehat{\boldsymbol{\beta}}^0) - \boldsymbol{I}_{S,\text{-}S}(\widehat{\boldsymbol{\beta}}^0)\boldsymbol{I}_{\text{-}S,\text{-}S}(\widehat{\boldsymbol{\beta}}^0)^{-1}\boldsymbol{I}_{\text{-}S,S}(\widehat{\boldsymbol{\beta}}^0)\right]^{-1} = [\boldsymbol{I}(\widehat{\boldsymbol{\beta}}^0)^{-1}]_{S,S}.$$

Hence, we have

$$\begin{aligned}
&\boldsymbol{U}_S(\widehat{\boldsymbol{\beta}}^0)^T \left[\boldsymbol{I}_{S,S}(\widehat{\boldsymbol{\beta}}^0) - \boldsymbol{I}_{S,\text{-}S}(\widehat{\boldsymbol{\beta}}^0)\boldsymbol{I}_{\text{-}S,\text{-}S}(\widehat{\boldsymbol{\beta}}^0)^{-1}\boldsymbol{I}_{\text{-}S,S}(\widehat{\boldsymbol{\beta}}^0)\right]^{-1} \boldsymbol{U}_S(\widehat{\boldsymbol{\beta}}^0) \\
&= \boldsymbol{U}_S(\widehat{\boldsymbol{\beta}}^0)^T [\boldsymbol{I}(\widehat{\boldsymbol{\beta}}^0)^{-1}]_{S,S} \boldsymbol{U}_S(\widehat{\boldsymbol{\beta}}^0) \\
&= \boldsymbol{U}(\widehat{\boldsymbol{\beta}}^0)^T \boldsymbol{I}(\widehat{\boldsymbol{\beta}}^0)^{-1} \boldsymbol{U}(\widehat{\boldsymbol{\beta}}^0),
\end{aligned}$$

where the last step used the fact that $\boldsymbol{U}_{\text{-}S}(\widehat{\boldsymbol{\beta}}^0) = 0$. Specializing to GLMs, we find that the score test statistic can be written as

$$\begin{aligned}
&T^2_{\text{score}}(\boldsymbol{X}, \boldsymbol{y}) = \\
&\frac{1}{\phi_0}(\boldsymbol{y} - \widehat{\boldsymbol{\mu}}^0)\widehat{\boldsymbol{W}}^0\widehat{\boldsymbol{M}}^0\boldsymbol{X}(\boldsymbol{X}^T\widehat{\boldsymbol{W}}^0\boldsymbol{X})^{-1}\boldsymbol{X}^T\widehat{\boldsymbol{M}}^0\widehat{\boldsymbol{W}}^0(\boldsymbol{y} - \widehat{\boldsymbol{\mu}}^0).
\end{aligned} \tag{22.8}$$

22.4.2 Testing a single coefficient (ϕ_0 known)

If $S = \{j\}$, the normal approximation (22.7) specializes to

$$T_{\text{score}} \dot{\sim} N(0, 1),$$

where

$$T_{\text{score}} = \frac{\boldsymbol{x}_{*,j}^T\widehat{\boldsymbol{M}}^0\widehat{\boldsymbol{W}}^0(\boldsymbol{y} - \widehat{\boldsymbol{\mu}}^0)}{\sqrt{\phi_0(\boldsymbol{x}_{*,j}^T\widehat{\boldsymbol{W}}^0\boldsymbol{x}_{*,j} - \boldsymbol{x}_{*,j}^T\widehat{\boldsymbol{W}}^0\boldsymbol{X}_{*,\text{-}j}(\boldsymbol{X}_{*,\text{-}j}^T\widehat{\boldsymbol{W}}^0\boldsymbol{X}_{*,\text{-}j})^{-1}\boldsymbol{X}_{*,\text{-}j}^T\widehat{\boldsymbol{W}}^0\boldsymbol{x}_{*,j})}}.$$

Unlike its multivariate counterparts, we can construct not just a two-sided test but also one-sided tests based on this normal approximation. For example, below is a right-sided score test for $H_0 : \beta_j = \beta_j^0$:

$$\phi(\boldsymbol{X}, \boldsymbol{y}) = 1\left(T_{\text{score}} > z_{1-\alpha}\right).$$

The nice thing about the score test is that the model need only be fit under the null hypothesis. Therefore, computation can be recycled if $\boldsymbol{X}_{*,\text{-}j}$ is a standard set of control variables and there are several options for $\boldsymbol{x}_{*,j}$ to test.

22.4.3 Confidence interval for a single coefficient (ϕ_0 known)

Just as with the likelihood ratio test, it is possible to invert a score test for a single coefficient to obtain a confidence interval. It is uncommon to invert a multivariate test to obtain a confidence region for multiple coordinates of $\boldsymbol{\beta}$, given the computationally expensive search across a grid of possible $\boldsymbol{\beta}$ values.

22.4.4 Goodness of fit testing (ϕ_0 known)

We can view goodness of fit testing as testing a hypothesis about the coefficients in the following augmented GLM:

$$y_i \sim \text{EDM}(\mu_i, \phi_i); \quad g(\mu_i) = \boldsymbol{x}_{i*}^T\boldsymbol{\beta} + \boldsymbol{z}_{i*}^T\boldsymbol{\gamma},$$

where $\boldsymbol{Z} \in \mathbb{R}^{n\times(n-p)}$ is a matrix of extra variables so that the columns of the augmented model matrix $\widetilde{\boldsymbol{X}} \equiv [\boldsymbol{X}, \boldsymbol{Z}]$ form a basis for $\mathbb{R}^n$. Then, the goodness of fit null hypothesis is $H_0 : \boldsymbol{\gamma} = \boldsymbol{0}$, and the alternative hypothesis is the saturated model. To test this hypothesis, we use the score test statistic in the form of equation (22.8) with the augmented model matrix $\widetilde{\boldsymbol{X}}$ in place of $\boldsymbol{X}$ and the GLM fits $\widehat{\boldsymbol{\mu}}, \widehat{\boldsymbol{W}}, \widehat{\boldsymbol{M}}$ in place of $\widehat{\boldsymbol{\mu}}^0, \widehat{\boldsymbol{W}}^0, \widehat{\boldsymbol{M}}^0$ to reflect that the full original GLM is being fit under the null hypothesis. Therefore, we get the score statistic

$$T^2_{\text{score}} = \frac{1}{\phi_0}(\boldsymbol{y} - \widehat{\boldsymbol{\mu}})\widehat{\boldsymbol{W}}\widehat{\boldsymbol{M}}\widetilde{\boldsymbol{X}}(\widetilde{\boldsymbol{X}}^T\widehat{\boldsymbol{W}}\widetilde{\boldsymbol{X}})^{-1}\widetilde{\boldsymbol{X}}^T\widehat{\boldsymbol{M}}\widehat{\boldsymbol{W}}(\boldsymbol{y} - \widehat{\boldsymbol{\mu}}).$$

Now, note that the matrix $\widetilde{\boldsymbol{X}}$ is a full-rank square matrix, and therefore it is invertible. Hence, we can simplify the statistic as follows:

$$\begin{aligned}
T^2_{\text{score}} &= \frac{1}{\phi_0}(\boldsymbol{y} - \widehat{\boldsymbol{\mu}})\widehat{\boldsymbol{W}}\widehat{\boldsymbol{M}}\widetilde{\boldsymbol{X}}\widetilde{\boldsymbol{X}}^{-1}\widehat{\boldsymbol{W}}^{-1}((\widetilde{\boldsymbol{X}})^T)^{-1}\widetilde{\boldsymbol{X}}^T\widehat{\boldsymbol{M}}\widehat{\boldsymbol{W}}(\boldsymbol{y} - \widehat{\boldsymbol{\mu}}) \\
&= \frac{1}{\phi_0}(\boldsymbol{y} - \widehat{\boldsymbol{\mu}})\widehat{\boldsymbol{W}}\widehat{\boldsymbol{M}}\widehat{\boldsymbol{W}}^{-1}\widehat{\boldsymbol{M}}\widehat{\boldsymbol{W}}(\boldsymbol{y} - \widehat{\boldsymbol{\mu}}) \\
&= \frac{1}{\phi_0}(\boldsymbol{y} - \widehat{\boldsymbol{\mu}})^T\widehat{\boldsymbol{W}}\widehat{\boldsymbol{M}}^2(\boldsymbol{y} - \widehat{\boldsymbol{\mu}}) \\
&= \frac{1}{\phi_0}(\boldsymbol{y} - \widehat{\boldsymbol{\mu}})^T\text{diag}\left(\frac{w_i}{V(\widehat{\mu}_i)}\right)(\boldsymbol{y} - \widehat{\boldsymbol{\mu}}) \\
&= \frac{1}{\phi_0}\sum_{i=1}^{n}\frac{w_i(y_i - \widehat{\mu}_i)^2}{V(\widehat{\mu}_i)} \\
&\equiv \frac{1}{\phi_0}X^2,
\end{aligned}$$

where X^2 is the Pearson chi-square statistic. Therefore, the score test for goodness of fit is:

$$\phi(\boldsymbol{X}, \boldsymbol{y}) \equiv 1\left(X^2 > \phi_0\chi^2_{n-p}(1-\alpha)\right).$$

In the context of contingency table analysis (see the next chapter), this test reduces to the Pearson chi-square test of independence between two categorical variables. This test was proposed in 1900; it was only pointed out about a century later that this is a score test (Smyth 2003).

As with the likelihood ratio test for goodness of fit, the score test for goodness of fit must be justified by small-dispersion asymptotics. In particular, the score test for goodness of fit relies on the central limit theorem for small dispersions. To verify whether this approximation is accurate, we can apply the rules of thumb from Section 19.6.1.2 for each observation y_i, when it is drawn from the distribution fitted under the GLM (rather than the saturated model). For instance, we can check that $m\hat{\mu}_i \geq 5$ and $m(1-\hat{\mu}_i) \geq 5$ in the case of grouped logistic regression of $\hat{\mu}_i \geq 5$ for Poisson regression. Here, $\hat{\mu}_i$ are the fitted means under the GLM.

22.4.5 Score test inference for ϕ_0 unknown

Score test inference for one or more coefficients $\boldsymbol{\beta}_S$ can be achieved by replacing ϕ_0 with one of its estimators and replacing the normal and chi-square distributions with t and F distributions, respectively. For example, the score test for a single coefficient β_j is:

$$\phi(\boldsymbol{X}, \boldsymbol{y}) = 1\left(T_{\text{score}} > t_{n-p}(1-\alpha)\right),$$

where

$$T_{\text{score}} = \frac{\boldsymbol{x}_{*,j}^T \widehat{\boldsymbol{M}}^0 \widehat{\boldsymbol{W}}^0 (\boldsymbol{y} - \widehat{\boldsymbol{\mu}}^0)}{\sqrt{\tilde{\phi}_0 (\boldsymbol{x}_{*,j}^T \widehat{\boldsymbol{W}}^0 \boldsymbol{x}_{*,j} - \boldsymbol{x}_{*,j}^T \widehat{\boldsymbol{W}}^0 \boldsymbol{X}_{*,-j} (\boldsymbol{X}_{*,-j}^T \widehat{\boldsymbol{W}}^0 \boldsymbol{X}_{*,-j})^{-1} \boldsymbol{X}_{*,-j}^T \widehat{\boldsymbol{W}}^0 \boldsymbol{x}_{*,j})}}.$$

The t and F distributions are not exact in finite samples, but are better approximations than the normal and chi-square distributions. The score test for goodness of fit is not applicable in the case when ϕ_0 is unknown, similarly to the likelihood ratio test. Indeed, note the relationship between the Pearson goodness of fit test, which rejects when $\frac{1}{\phi_0} X^2 > \chi^2_{n-p}(1-\alpha)$, and the Pearson estimator of the dispersion parameter: $\widetilde{\phi}_0 \equiv \frac{X^2}{n-p}$. If we try to plug in the Pearson estimator for the dispersion into the Pearson goodness of fit test, we end up with a test statistic deterministically equal to $n - p$. This reflects the fact that the residual degrees of freedom can either be used to estimate the dispersion or to test goodness of fit; they cannot be used for both.

Chapter 23

R demo

23.1 Crime data

Let's revisit the crime data from Homework 2, this time fitting a logistic regression to it.

```
library(readr)
library(dplyr)
library(ggplot2)

# read crime data
crime_data <- read_tsv("data/Statewide_crime.dat")

# read and transform population data
population_data <- read_csv("data/state-populations.csv")
population_data <- population_data |>
  filter(State != "Puerto Rico") |>
  select(State, Pop) |>
  rename(state_name = State, state_pop = Pop)

# collate state abbreviations
state_abbreviations <- tibble(
  state_name = state.name,
  state_abbrev = state.abb
) |>
  add_row(state_name = "District of Columbia", state_abbrev = "DC")

# add CrimeRate to crime_data
crime_data <- crime_data |>
  mutate(STATE = ifelse(STATE == "IO", "IA", STATE)) |>
  rename(state_abbrev = STATE) |>
  filter(state_abbrev != "DC") |> # remove outlier
  left_join(state_abbreviations, by = "state_abbrev") |>
  left_join(population_data, by = "state_name") |>
```

```
    mutate(CrimeRate = Violent / state_pop) |>
    select(state_abbrev, CrimeRate, Metro, HighSchool, Poverty, state_pop)

  crime_data
```

```
# A tibble: 50 x 6
   state_abbrev CrimeRate Metro HighSchool Poverty state_pop
   <chr>            <dbl> <dbl>      <dbl>   <dbl>     <dbl>
 1 AK           0.000819   65.6       90.2     8      724357
 2 AL           0.0000871  55.4       82.4    13.7   4934193
 3 AR           0.000150   52.5       79.2    12.1   3033946
 4 AZ           0.0000682  88.2       84.4    11.9   7520103
 5 CA           0.0000146  94.4       81.3    10.5  39613493
 6 CO           0.0000585  84.5       88.3     7.3   5893634
 7 CT           0.0000867  87.7       88.8     6.4   3552821
 8 DE           0.000664   80.1       86.5     5.8    990334
 9 FL           0.0000333  89.3       85.9     9.7  21944577
10 GA           0.0000419  71.6       85.2    10.8  10830007
# i 40 more rows
```

We can fit a GLM using the `glm` command, specifying as additional arguments the observation weights as well as the exponential dispersion model. In this case, the weights are the state populations and the family is binomial:

```
  glm_fit <- glm(CrimeRate ~ Metro + HighSchool + Poverty,
    weights = state_pop,
    family = "binomial",
    data = crime_data
  )
```

We can print the summary table as usual:

```
  summary(glm_fit)
```

```
Call:
glm(formula = CrimeRate ~ Metro + HighSchool + Poverty, family = "binomial",
    data = crime_data, weights = state_pop)

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.609e+01  3.520e-01  -45.72   <2e-16 ***
Metro       -2.586e-02  5.727e-04  -45.15   <2e-16 ***
HighSchool   9.106e-02  3.450e-03   26.39   <2e-16 ***
Poverty      6.077e-02  4.852e-03   12.53   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 15590  on 49  degrees of freedom
Residual deviance: 11742  on 46  degrees of freedom
AIC: 12136

Number of Fisher Scoring iterations: 5
```

Amazingly, everything is very significant! This is because the weights for each observation (the state populations) are very high, effectively making the sample size very high. But frankly, this is a bit suspicious. Glancing at the bottom of the regression summary, we see a residual deviance of 11742 on 46 degrees of freedom. This part of the summary refers to the deviance-based goodness of fit test. Under the null hypothesis that the model fits well, we expect that the residual deviance has a distribution of χ^2_{46}, which has a mean of 46.

Let's formally check the goodness of fit. We can extract the residual deviance and residual degrees of freedom from the GLM fit:

```
glm_fit$deviance
```

```
[1] 11742.28
```

```
glm_fit$df.residual
```

```
[1] 46
```

We can then compute the chi-square p-value:

```
# compute based on residual deviance from fit object
pchisq(glm_fit$deviance,
  df = glm_fit$df.residual,
  lower.tail = FALSE
)
```

```
[1] 0
```

```
# compute residual deviance as sum of squares of residuals
pchisq(sum(resid(glm_fit, "deviance")^2),
  df = glm_fit$df.residual,
  lower.tail = FALSE
)
```

```
[1] 0
```

Wow, we get a p-value of zero! Let's try doing a score-based (i.e., Pearson) goodness of fit test:

```
pchisq(sum(resid(glm_fit, "pearson")^2),
  df = glm_fit$df.residual,
  lower.tail = FALSE
```

```
  )
```

```
[1] 0
```

Also zero. So we need to immediately stop using this model for inference about these data, since it fits the data very poorly. We will discuss how to build a better model for the crime data in the next unit. For now, we turn to analyzing a different dataset.

23.2 Noisy miner data

Credit: Generalized Linear Models With Examples in R textbook.

Let's consider the noisy miner dataset. Noisy miners are a small but aggressive native Australian bird. We want to know how the number of these birds observed in a patch of land depends on various factors of that patch of land.

```
  library(GLMsData)
  data("nminer")
  nminer |> as_tibble()
```

```
# A tibble: 31 x 8
   Miners  Eucs  Area Grazed Shrubs Bulokes Timber Minerab
    <int> <int> <int>  <int>  <int>   <int>  <int>   <int>
 1      0     2    22      0      1     120     16       0
 2      0    10    11      0      1      67     25       0
 3      1    16    51      0      1      85     13       3
 4      1    20    22      0      1      45     12       2
 5      1    19     4      0      1     160     14       8
 6      1    18    61      0      1      75      6       1
 7      1    12    16      0      1     100     12       8
 8      1    16    14      0      1     321     15       5
 9      0     3     5      0      1     275      8       0
10      1    12     6      1      0     227     10       4
# i 21 more rows
```

Since the response is a count, we can model it as a Poisson random variable. Let's fit that GLM:

```
  glm_fit <- glm(Minerab ~ . - Miners, family = "poisson", data = nminer)
  summary(glm_fit)
```

```
Call:
glm(formula = Minerab ~ . - Miners, family = "poisson", data = nminer)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.886345   0.875737  -1.012    0.311
Eucs         0.129309   0.021757   5.943 2.79e-09 ***
Area        -0.028736   0.013241  -2.170    0.030 *
```

```
Grazed       0.140831   0.364622   0.386    0.699
Shrubs       0.335828   0.375059   0.895    0.371
Bulokes      0.001469   0.001773   0.828    0.408
Timber      -0.006781   0.009074  -0.747    0.455
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 150.545  on 30  degrees of freedom
Residual deviance:  54.254  on 24  degrees of freedom
AIC: 122.41

Number of Fisher Scoring iterations: 6
```

We exclude `Miners` because this is just a binarized version of the response variable. Things look a bit better on the GOF front:

```
pchisq(sum(resid(glm_fit, "deviance")^2),
  df = glm_fit$df.residual,
  lower.tail = FALSE
)
```

```
[1] 0.000394186
```

```
pchisq(sum(resid(glm_fit, "pearson")^2),
  df = glm_fit$df.residual,
  lower.tail = FALSE
)
```

```
[1] 0.0001185197
```

Still, there is some model misspecification, but for now, we still proceed with the rest of the analysis.

The standard errors shown in the summary are based on the Wald test. We can get Wald confidence intervals based on these standard errors by using the formula:

```
glm_fit |>
  summary() |>
  coef() |>
  as.data.frame() |>
  transmute(`2.5 %` = Estimate + qnorm(0.025)*`Std. Error`,
            `97.5 %` = Estimate + qnorm(0.025)*`Std. Error`)
```

```
                 2.5 %       97.5 %
(Intercept) -2.602757559 -2.602757559
Eucs         0.086666177  0.086666177
Area        -0.054686818 -0.054686818
Grazed      -0.573814583 -0.573814583
```

```
Shrubs      -0.399274191 -0.399274191
Bulokes     -0.002007061 -0.002007061
Timber      -0.024565751 -0.024565751
```

Or, we can simply use `confint.default()`:

```
confint.default(glm_fit)
```

```
                  2.5 %       97.5 %
(Intercept) -2.602757559  0.830066560
Eucs         0.086666177  0.171951888
Area        -0.054686818 -0.002784651
Grazed      -0.573814583  0.855476296
Shrubs      -0.399274191  1.070929206
Bulokes     -0.002007061  0.004944760
Timber      -0.024565751  0.011002885
```

Or, we might want LRT-based confidence intervals, which are given by `confint()`:

```
confint(glm_fit)
```

```
Waiting for profiling to be done...
```

```
                 2.5 %       97.5 %
(Intercept) -2.63176754  0.812111327
Eucs         0.08782624  0.173336323
Area        -0.05658079 -0.004456166
Grazed      -0.57858596  0.855903871
Shrubs      -0.38600748  1.090319407
Bulokes     -0.00214123  0.004838901
Timber      -0.02483241  0.010820749
```

In this case, the two sets of confidence intervals seem fairly similar.

Now, we can get prediction intervals, either on the linear predictor scale or on the mean scale:

```
pred_linear <- predict(glm_fit, newdata = nminer[31,], se.fit = TRUE)
pred_mean <- predict(glm_fit, newdata = nminer[31,], type = "response", se.fit = TRUE)

pred_linear
```

```
$fit
       31
0.6556799

$se.fit
[1] 0.2635664

$residual.scale
```

```
[1] 1
```

```
  pred_mean
```

```
$fit
      31
1.926452

$se.fit
       31
0.5077481

$residual.scale
[1] 1
```

```
  log(pred_mean$fit)
```

```
       31
0.6556799
```

We see that the prediction on the linear predictor scale is exactly the logarithm of the prediction on the mean scale. However, the standard error given on the mean scale uses the delta method. We prefer to directly transform the confidence interval from the linear scale using the inverse link function (in this case, the exponential):

```
  # using delta method
  c(pred_mean$fit + qnorm(0.025)*pred_mean$se.fit,
    pred_mean$fit + qnorm(0.975)*pred_mean$se.fit)
```

```
       31        31
0.9312839 2.9216197
```

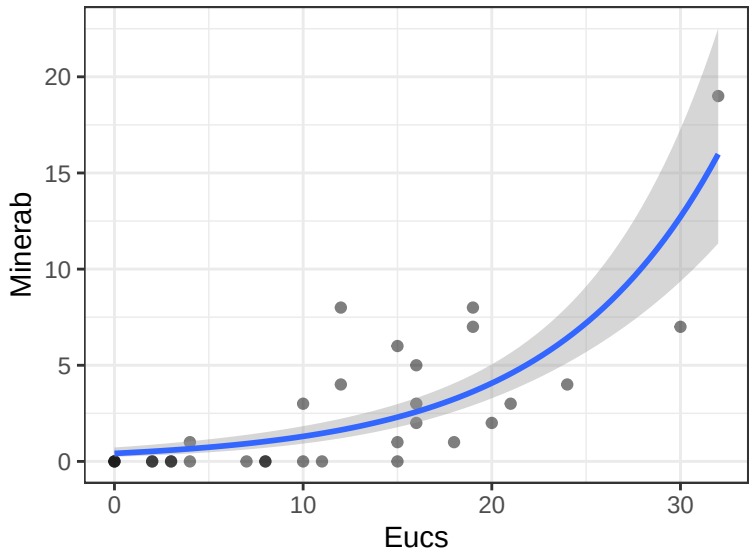
```
  # using transformation
  exp(c(pred_linear$fit + qnorm(0.025)*pred_linear$se.fit,
        pred_linear$fit + qnorm(0.975)*pred_linear$se.fit))
```

```
      31       31
1.149238 3.229285
```

In this case, the intervals obtained are somewhat different. We can plot confidence intervals for the fit in a univariate case (e.g., regressing `Minerab` on `Eucs`) using `geom_smooth()`:

```
  nminer |>
    ggplot(aes(x = Eucs, y = Minerab)) +
    geom_point(alpha = 0.5) +
    geom_smooth(method = "glm",
                method.args = list(family = "poisson"))
```

```
`geom_smooth()` using formula = 'y ~ x'
```



We can also test the coefficients in the model. The Wald tests for individual coefficients were already given by the regression summary above. We might want to carry out likelihood ratio tests for individual coefficients instead. For example, let's do this for `Eucs`:

```
glm_fit_partial <- glm(Minerab ~ . - Miners - Eucs, family = "poisson", data = nminer)
anova(glm_fit_partial, glm_fit, test = "LRT")
```

```
Analysis of Deviance Table

Model 1: Minerab ~ (Miners + Eucs + Area + Grazed + Shrubs + Bulokes +
    Timber) - Miners - Eucs
Model 2: Minerab ~ (Miners + Eucs + Area + Grazed + Shrubs + Bulokes +
    Timber) - Miners
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1        25     95.513
2        24     54.254  1   41.259 1.333e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The `Eucs` variable is quite significant! We can manually carry out the LRT as a sanity check:

```
deviance_partial <- glm_fit_partial$deviance
deviance_full <- glm_fit$deviance
lrt_stat <- deviance_partial - deviance_full
p_value <- pchisq(lrt_stat, df = 1, lower.tail = FALSE)
tibble(lrt_stat, p_value)
```

```
# A tibble: 1 x 2
  lrt_stat  p_value
```

```
     <dbl>    <dbl>
1     41.3 1.33e-10
```

We can test groups of variables using the likelihood ratio test as well.

Part V

Generalized linear models: Special cases

Chapter 4 developed a general theory for GLMs. In Chapter 5, we specialize this theory to several important cases, including logistic regression and Poisson regression.

Chapter 24

Logistic regression

24.1 Model definition and interpretation

24.1.1 Model definition

Recall from Chapter 4 that the logistic regression model is

$$m_i y_i \overset{\text{ind}}{\sim} \text{Bin}(m_i, \pi_i); \quad \text{logit}(\pi_i) = \log \frac{\pi_i}{1-\pi_i} = \boldsymbol{x}_{i*}^T \boldsymbol{\beta}.$$

Here we use the canonical logit link function, although other link functions are possible. We also set the offsets to 0. The interpretation of the parameter β_j is that a unit increase in x_j—other predictors held constant—is associated with an (additive) increase of β_j on the log-odds scale or a multiplicative increase of e^{β_j} on the odds scale. Note that logistic regression data come in two formats: *ungrouped* and *grouped*. For ungrouped data, we have $m_1 = \cdots = m_n = 1$, so $y_i \in \{0, 1\}$ are Bernoulli random variables. For grouped data, we can have several independent Bernoulli observations per predictor $\boldsymbol{x}_{i*}$, which give rise to binomial proportions $y_i \in [0, 1]$. This happens most often when all the predictors are discrete. You can always convert grouped data into ungrouped data, but not necessarily vice versa. We'll discuss below that the grouped and ungrouped formulations of logistic regression have the same MLE and standard errors but different deviances.

24.1.2 Generative model equivalent

Consider the following generative model for $(\boldsymbol{x}, y) \in \mathbb{R}^{p-1} \times \{0, 1\}$:

$$y \sim \text{Ber}(\pi); \quad \boldsymbol{x}|y \sim \begin{cases} N(\boldsymbol{\mu}_0, \boldsymbol{V}) & \text{if } y = 0 \\ N(\boldsymbol{\mu}_1, \boldsymbol{V}) & \text{if } y = 1 \end{cases}.$$

Then, we can derive that $y|\boldsymbol{x}$ follows a logistic regression model (called a *discriminative* model because it conditions on $\boldsymbol{x}$). Indeed,

$$\begin{aligned}
\text{logit}(p(y=1|\boldsymbol{x})) &= \log \frac{p(y=1)p(\boldsymbol{x}|y=1)}{p(y=0)p(\boldsymbol{x}|y=0)} \\
&= \log \frac{\pi \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_1)^T \boldsymbol{V}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_1)\right)}{(1-\pi)\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_0)^T \boldsymbol{V}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_0)\right)} \\
&= \beta_0 + \boldsymbol{x}^T \boldsymbol{V}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) \\
&\equiv \beta_0 + \boldsymbol{x}^T \boldsymbol{\beta}_{-0}.
\end{aligned}$$

This is another natural route to motivating the logistic regression model.

24.1.3 Special case: 2×2 contingency table

Suppose that $x \in \{0, 1\}$, and consider the logistic regression model $\text{logit}(\pi_i) = \beta_0 + \beta_1 x_i$. For example, suppose that $x \in \{0, 1\}$ encodes treatment (1) and control (0) in a clinical trial, and $y_i \in \{0, 1\}$ encodes success (1) and failure (0). We make n observations of (x_i, y_i) in this ungrouped setup. The parameter e^{β_1} can be interpreted as the *odds ratio*:

$$e^{\beta_1} = \frac{\mathbb{P}[y=1|x=1]/\mathbb{P}[y=0|x=1]}{\mathbb{P}[y=1|x=0]/\mathbb{P}[y=0|x=0]}.$$

This parameter is the multiple by which the odds of success increase when going from control to treatment. We can summarize such data via the 2×2 *contingency table* (Table 24.1). A grouped version of this data would be $\{(x_1, y_1) = (0, 7/24), (x_2, y_2) = (1, 9/21)\}$. The null hypothesis $H_0 : \beta_1 = 0 \iff H_0 : e^{\beta_1} = 1$ states that the success probability in both rows of the table is the same.

Table 24.1: An example of a 2×2 contingency table.

	Success	Failure	Total
Treatment	9	12	21
Control	7	17	24
Total	16	29	45

24.2 Logistic regression with case-control studies

In a prospective study (e.g. a clinical trial), we assign treatment or control (i.e., x) to individuals, and then observe a binary outcome (i.e., y). Sometimes, the outcome y takes a long time to measure or has a highly imbalanced distribution in the population (e.g., the development of lung cancer). In this case, an appealing study design is the *retrospective study*, where individuals are sampled based on their *response values* (e.g., presence of lung cancer) rather than their treatment/exposure status (e.g., smoking). It turns out that a logistic regression model is appropriate for such retrospective study designs as well.

Indeed, suppose that $y|\boldsymbol{x}$ follows a logistic regression model. Let's try to figure out the distribution of $y|\boldsymbol{x}$ in the retrospectively gathered sample. Letting $z \in \{0, 1\}$ denote the indicator that an

observation is sampled, define $\rho_1 \equiv \mathbb{P}[z = 1|y = 1]$ and $\rho_0 \equiv \mathbb{P}[z = 1|y = 0]$, and assume that $\mathbb{P}[z = 1, y, \boldsymbol{x}] = \mathbb{P}[z = 1|y]$. The latter assumption states that the predictors $\boldsymbol{x}$ were not used in the retrospective sampling process. Then,

$$\begin{aligned}\text{logit}(\mathbb{P}[y = 1|z = 1, \boldsymbol{x}]) &= \log \frac{\rho_1 \mathbb{P}[y = 1|\boldsymbol{x}]}{\rho_0 \mathbb{P}[y = 0|\boldsymbol{x}]} \\ &= \log \frac{\rho_1}{\rho_0} + \text{logit}(\mathbb{P}[y = 1|\boldsymbol{x}]) \\ &= \left(\log \frac{\rho_1}{\rho_0} + \beta_0\right) + \boldsymbol{x}^T \boldsymbol{\beta}_{-0}.\end{aligned}$$

Thus, conditioning on retrospective sampling changes only the intercept term, but preserves the coefficients of $\boldsymbol{x}$. Therefore, we can carry out inference for $\boldsymbol{\beta}_{-0}$ in the same way regardless of whether the study design is prospective or retrospective.

24.3 Estimation and inference

24.3.1 Score and Fisher information

Recall from Chapter 4 that

$$\boldsymbol{U}(\boldsymbol{\beta}) = \frac{1}{\phi_0} \boldsymbol{X}^T \boldsymbol{M} \boldsymbol{W} (\boldsymbol{y} - \boldsymbol{\mu}) \quad \text{and} \quad \boldsymbol{I}(\boldsymbol{\beta}) = \frac{1}{\phi_0} \boldsymbol{X}^T \boldsymbol{W} \boldsymbol{X},$$

where

$$\begin{aligned}\boldsymbol{W} &\equiv \text{diag}\left(\frac{w_i}{V(\mu_i)(d\eta_i/d\mu_i)^2}\right), \\ \boldsymbol{M} &\equiv \text{diag}\left(\frac{\partial \eta_i}{\partial \mu_i}\right).\end{aligned}$$

Since logistic regression uses a canonical link function, we get the following simplifications:

$$\begin{aligned}\boldsymbol{W} &= \text{diag}\left(w_i V(\mu_i)\right) = \text{diag}\left(m_i \pi_i (1 - \pi_i)\right), \\ \boldsymbol{M} &= \text{diag}\left(\frac{1}{\pi_i(1 - \pi_i)}\right).\end{aligned}$$

Here we have substituted the notation $\boldsymbol{\pi}$ for $\boldsymbol{\mu}$, and recall that for logistic regression, $\phi_0 = 1$, $w_i = m_i$, and $V(\pi_i) = \pi_i(1 - \pi_i)$. Therefore, the score equations for logistic regression are

$$0 = \boldsymbol{X}^T \text{diag}\left(m_i\right)(\boldsymbol{y} - \widehat{\boldsymbol{\mu}}) \quad \Longleftrightarrow \quad \sum_{i=1}^{n} m_i x_{ij} (y_i - \widehat{\pi}_i) = 0, \tag{24.1}$$

for $j = 0, \ldots, p - 1$. We can solve these equations using IRLS. The Fisher information is

$$\boldsymbol{I}(\boldsymbol{\beta}) = \boldsymbol{X}^T \text{diag}\left(m_i \pi_i (1 - \pi_i)\right) \boldsymbol{X}.$$

24.3.2 Wald inference

Using the results in the previous paragraph, we can carry out Wald inference based on the normal approximation

$$\widehat{\boldsymbol{\beta}} \dot{\sim} N\left(\boldsymbol{\beta}, \left(\boldsymbol{X}^T \text{diag}(m_i \widehat{\pi}_i (1-\widehat{\pi}_i)) \boldsymbol{X}\right)^{-1}\right).$$

This approximation holds for $\sum_{i=1}^n m_i \to \infty$.

24.3.2.1 Example: 2×2 contingency table

Suppose we have a 2×2 contingency table. The grouped logistic regression formulation of these data is

$$y_0 \sim \frac{1}{m_0}\text{Bin}(m_0, \pi_0); \quad y_1 \sim \frac{1}{m_1}\text{Bin}(m_1, \pi_1); \quad \text{logit}(\pi_i) = \beta_0 + \beta_1 x_i.$$

In this case, we have $n = p = 2$, so the grouped logistic regression model is saturated. Therefore, we have

$$\hat{\pi}_0 = y_0, \quad \text{and} \quad \hat{\pi}_1 = y_1, \quad \text{so} \quad \hat{\beta}_1 = \log \frac{\hat{\pi}_1/(1-\hat{\pi}_1)}{\hat{\pi}_0/(1-\hat{\pi}_0)} = \log \frac{y_1/(1-y_1)}{y_0/(1-y_0)}.$$

The squared Wald standard error for $\hat{\beta}_1$ is

$$\begin{aligned}
\text{SE}^2(\widehat{\beta}_1) &\equiv \left[\left(\boldsymbol{X}^T \text{diag}(m_i \widehat{\pi}_i (1-\widehat{\pi}_i)) \boldsymbol{X}\right)^{-1}\right]_{22} \\
&= \left[\left(\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^T \begin{pmatrix} m_0 y_0(1-y_0) & 0 \\ 0 & m_1 y_1(1-y_1) \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}\right)^{-1}\right]_{22} \\
&= \left[\left(\begin{pmatrix} m_0 y_0(1-y_0) + m_1 y_1(1-y_1) & m_1 y_1(1-y_1) \\ m_1 y_1(1-y_1) & m_1 y_1(1-y_1) \end{pmatrix}\right)^{-1}\right]_{22} \\
&= \frac{m_0 y_0(1-y_0) + m_1 y_1(1-y_1)}{m_0 y_0(1-y_0) \cdot m_1 y_1(1-y_1)} \\
&= \frac{1}{m_0 y_0(1-y_0)} + \frac{1}{m_1 y_1(1-y_1)}.
\end{aligned}$$

Therefore, the Wald test for $H_0 : \beta_1 = 0$ rejects if

$$\left|\frac{\hat{\beta}_1}{\text{SE}(\hat{\beta}_1)}\right| = \left|\frac{\log \frac{y_1/(1-y_1)}{y_0/(1-y_0)}}{\sqrt{\frac{1}{m_0 y_0(1-y_0)} + \frac{1}{m_1 y_1(1-y_1)}}}\right| > z_{1-\alpha/2}.$$

24.3.2.2 Hauck-Donner effect

Unfortunately, Wald inference in finite samples does not always perform very well. The Wald test above is known to be conservative if one or more of the mean parameters (in this case, π_i) tends to the edge of the parameter space (in this case, $\pi_i \to 0$ or $\pi_i \to 1$). This is called the *Hauck-Donner effect*. As an example, consider testing $H_0 : \beta_0 = 0$ in the intercept-only model

$$my \sim \text{Bin}(m, \pi); \quad \text{logit}(\pi) = \beta_0.$$

The Wald test statistic is $z \equiv \widehat{\beta}/\text{SE} = \text{logit}(y)\sqrt{my(1-y)}$. This test statistic actually tends to *decrease* as $y \to 1$ (see Figure 24.1), since the standard error grows faster than the estimate itself. So the test statistic becomes less significant as we go further away from the null! A similar situation arises in the 2×2 contingency table example above, where the Wald test for $H_0 : \beta_1 = 0$ becomes less significant as $y_0 \to 0$ and $y_1 \to 1$. As a limiting case of this, the Wald test is undefined if $y_0 = 0$ and $y_1 = 1$. This situation is a special case of *perfect separability* in logistic regression: when a hyperplane in covariate space separates observations with $y_i = 0$ from those with $y_i = 1$. Some of the maximum likelihood coefficient estimates are infinite in this case, causing the Wald test to be undefined since it uses these coefficient estimates as test statistics.

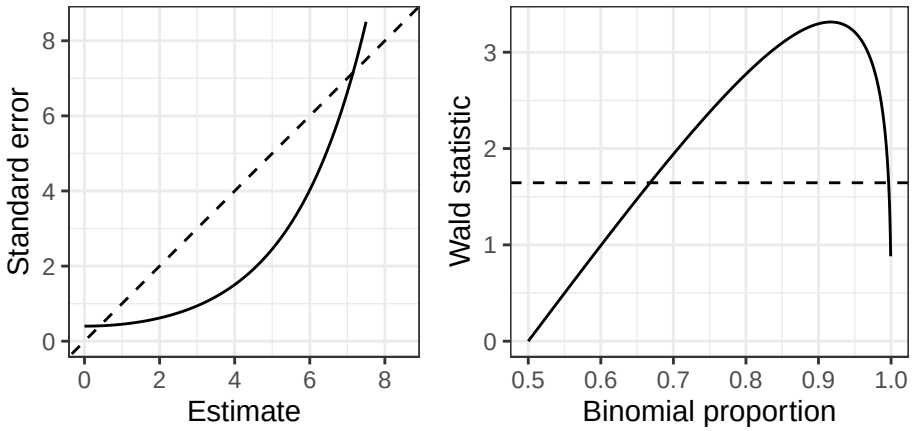


Figure 24.1: The Hauck-Donner effect: The Wald statistic for testing $H_0 : \pi = 0.5$ within the model $my \sim \text{Bin}(m, \pi)$ decreases as the proportion y approaches 1. Here, $m = 25$.

24.3.3 Likelihood ratio inference

24.3.3.1 The Bernoulli and binomial deviance

Let's first compute the deviance of a Bernoulli or binomial model. These deviances are the same because these two models have the same natural parameter and log-partition function. Recalling the definition of the unit deviance (19.4), we have

$$\begin{aligned} d(y, \mu) &\equiv 2\{[\theta(y)y - \psi(\theta(y))] - [\theta(\mu)y - \psi(\theta(\mu))]\} \\ &= 2\{[y \log \tfrac{y}{1-y} + \log(1-y)] - [y \log \tfrac{\pi}{1-\pi} + \log(1-\pi)]\} \\ &= 2\left(y \log \frac{y}{\pi} + (1-y) \log \frac{1-y}{1-\pi}\right). \end{aligned}$$

The total deviance, therefore, is

$$\begin{aligned} D(\boldsymbol{y}, \widehat{\boldsymbol{\pi}}) &\equiv \sum_{i=1}^{n} w_i d(y_i, \widehat{\pi}_i) \\ &= 2 \sum_{i=1}^{n} \left(m_i y_i \log \frac{y_i}{\widehat{\pi}_i} + m_i (1 - y_i) \log \frac{1 - y_i}{1 - \widehat{\pi}_i} \right). \end{aligned} \tag{24.2}$$

24.3.3.2 Comparing the deviances of grouped and ungrouped logistic regression models

Let us pause to compare the total deviances of grouped and ungrouped logistic regression models. Consider the following grouped and ungrouped models:

$$y_i^{\text{grp}} \overset{\text{ind}}{\sim} \frac{1}{m_i} \text{Bin}(m_i, \pi_i) \quad \text{and} \quad y_{ik}^{\text{ungrp}} \overset{\text{ind}}{\sim} \text{Ber}(\pi_i), \quad k = 1, \ldots, m_i,$$

where

$$\text{logit}(\pi_i) = \boldsymbol{x}_{i*}^T \boldsymbol{\beta}.$$

The relationship between the grouped and ungrouped observations is that

$$y_i^{\text{grp}} = \frac{1}{m_i} \sum_{k=1}^{m_i} y_{ik}^{\text{ungrp}} \equiv \bar{y}_i^{\text{ungrp}}.$$

Since the grouped and ungrouped logistic regression models have the same likelihoods, it follows that they have the same maximum likelihood estimates $\widehat{\boldsymbol{\beta}}$ and $\widehat{\boldsymbol{\pi}}$. However, the total deviances of the two models are different. The total deviance of the grouped model can be derived from equation (24.2):

$$\begin{aligned} &D(\boldsymbol{y}^{\text{grp}}, \widehat{\boldsymbol{\pi}}) \\ &\quad = 2 \sum_{i=1}^{n} \left(m_i y_i^{\text{grp}} \log \frac{y_i^{\text{grp}}}{\widehat{\pi}_i} + m_i (1 - y_i^{\text{grp}}) \log \frac{1 - y_i^{\text{grp}}}{1 - \widehat{\pi}_i} \right). \end{aligned} \tag{24.3}$$

On the other hand, the total deviance of the ungrouped model is

$$\begin{aligned} &D(\boldsymbol{y}^{\text{ungrp}}, \widehat{\boldsymbol{\pi}}) \\ &= 2 \sum_{i=1}^{n} \sum_{k=1}^{m_i} \left(y_{ik}^{\text{ungrp}} \log \frac{y_{ik}^{\text{ungrp}}}{\widehat{\pi}_i} + (1 - y_{ik}^{\text{ungrp}}) \log \frac{1 - y_{ik}^{\text{ungrp}}}{1 - \widehat{\pi}_i} \right) \\ &= 2 \sum_{i=1}^{n} \sum_{k=1}^{m_i} \left(y_{ik}^{\text{ungrp}} \log \frac{1}{\widehat{\pi}_i} + (1 - y_{ik}^{\text{ungrp}}) \log \frac{1}{1 - \widehat{\pi}_i} \right) \\ &= 2 \sum_{i=1}^{n} \left(m_i y_i^{\text{grp}} \log \frac{1}{\widehat{\pi}_i} + m_i (1 - y_i^{\text{grp}}) \log \frac{1}{1 - \widehat{\pi}_i} \right). \end{aligned} \tag{24.4}$$

In the second line, we used the fact that $y \log y \to 0$ and $(1 - y) \log(1 - y) \to 0$ as $y \to 0$ or $y \to 1$. Comparing the grouped (24.3) and ungrouped (24.4) total deviances, we see that these are given

by related, but different expressions. Because small dispersion asymptotics applies to the grouped model but not the ungrouped model, we have that under small-dispersion asymptotics,

$$D(\boldsymbol{y}^{\text{grp}}, \hat{\boldsymbol{\pi}}) \dot{\sim} \chi^2_{n-p} \quad \text{but} \quad D(\boldsymbol{y}^{\text{ungrp}}, \hat{\boldsymbol{\pi}}) \not\sim \chi^2_{n-p}.$$

24.3.3.3 Likelihood ratio inference for one or more coefficients

Letting $\widehat{\boldsymbol{\pi}}_0$ and $\widehat{\boldsymbol{\pi}}_1$ be the MLEs from two nested models, we can then express the likelihood ratio statistic as

$$D(\boldsymbol{y}, \widehat{\boldsymbol{\pi}}_0) - D(\boldsymbol{y}, \widehat{\boldsymbol{\pi}}_1) = 2\sum_{i=1}^{n} \left(m_i y_i \log \frac{\widehat{\pi}_{i1}}{\widehat{\pi}_{i0}} + m_i(1-y_i) \log \frac{1-\widehat{\pi}_{i1}}{1-\widehat{\pi}_{i0}} \right).$$

Note that this expression holds for grouped or ungrouped logistic regression models. We can then construct a likelihood ratio test in the usual way. Likelihood ratio inference can be justified by either large-sample or small-dispersion asymptotics.

24.3.3.4 Goodness of fit testing

In grouped logistic regression, we can also use the likelihood ratio test to test goodness of fit. To do so, we compare the total deviance of the fitted model (24.2) to a chi-squared quantile. In particular, the deviance-based goodness of fit test rejects when:

$$\begin{aligned} &D(\boldsymbol{y}, \hat{\boldsymbol{\pi}}) = \\ &2\sum_{i=1}^{n} \left(m_i y_i \log \frac{y_i}{\widehat{\pi}_i} + m_i(1-y_i) \log \frac{1-y_i}{1-\widehat{\pi}_i} \right) > \chi^2_{n-p}(1-\alpha). \end{aligned} \tag{24.5}$$

This test is justified by small-dispersion asymptotics based on the saddlepoint approximation, which is decent when $\min(m_i\widehat{\pi}_i, (1-m_i)\widehat{\pi}_i) \geq 3$ for each i.

24.3.3.5 Example: 2×2 table

Let us revisit the example of the 2×2 table model, within which we would like to test $H_0 : \beta_1 = 0$. Note that we can view this as a goodness of fit test of the intercept-only model in a grouped logistic regression model since the alternative model is the saturated model (it has two observations and two parameters). To compute the likelihood ratio statistic, we first need to fit the intercept-only model. The score equations (24.1) reduce to:

$$m_0(y_0 - \hat{\pi}) + m_1(y_1 - \hat{\pi}) = 0 \quad \Longrightarrow \quad \hat{\pi}_0 = \hat{\pi}_1 = \hat{\pi} = \frac{m_0 y_0 + m_1 y_1}{m_0 + m_1}.$$

Therefore, the deviance-based test of $H_0 : \beta_1 = 0$ rejects when:

$$\begin{aligned} D(\boldsymbol{y}, \widehat{\boldsymbol{\pi}}) &= 2\sum_{i=1}^{n}\left(m_i y_i \log\frac{y_i}{\widehat{\pi}_i} + m_i(1-y_i)\log\frac{1-y_i}{1-\widehat{\pi}_i}\right) \\ &= 2\left(m_0 y_0 \log\frac{y_0}{\hat{\pi}} + m_0(1-y_0)\log\frac{1-y_0}{1-\hat{\pi}}\right) + \\ &\quad 2\left(m_1 y_1 \log\frac{y_1}{\hat{\pi}} + m_1(1-y_1)\log\frac{1-y_1}{1-\hat{\pi}}\right) \\ &> \chi^2_1(1-\alpha). \end{aligned}$$

Likelihood ratio inference can give nontrivial conclusions in cases when Wald inference cannot, e.g. in the case of perfect separability. In the above example, suppose $y_0 = 0$ and $y_1 = 1$, giving perfect separability. Then, we can use the fact that $y \log y \to 0$ and $(1-y)\log(1-y) \to 0$ as $y \to 0$ or $y \to 1$ to see that:

$$\begin{aligned} D(\boldsymbol{y}, \widehat{\boldsymbol{\pi}}) &= 2\left(m_0 \log\frac{1}{1-\hat{\pi}} + m_1 \log\frac{1}{\hat{\pi}}\right) \\ &= 2\left(m_0 \log\frac{m_0+m_1}{m_0} + m_1 \log\frac{m_0+m_1}{m_1}\right). \end{aligned} \tag{24.6}$$

This gives us a finite value, which we can compare to $\chi^2_1(1-\alpha)$ to test $H_0 : \beta_1 = 0$. Even though the likelihood ratio statistic is still defined, we do still have to be careful because the data may suggest that the parameters are too close to the boundary of the parameter space. However, the rate at which the test breaks down as the parameters approach this boundary is slower than the rate at which the Wald test breaks down.

24.3.4 Score-based inference

Here we present only the score-based goodness-of-fit test. Recalling Section 22.4.4, the score statistic for goodness of fit is Pearson's X^2 statistic:

$$X^2 = \sum_{i=1}^{n} \frac{w_i(y_i - \widehat{\mu}_i)^2}{V(\widehat{\mu}_i)} = \sum_{i=1}^{n} \frac{m_i(y_i - \widehat{\pi}_i)^2}{\widehat{\pi}_i(1-\widehat{\pi}_i)}. \tag{24.7}$$

This test is justified by small-dispersion asymptotics based on the central limit theorem, which is decent when $\min(m_i\pi_i, (1-m_i)\pi_i) \geq 5$ for each i.

24.3.5 Fisher's exact test

As an alternative to asymptotic tests for logistic regression, in the case of 2×2 tables, there is an *exact* test of $H_0 : \beta_1 = 0$. Suppose we have:

$$s_1 = m_1 y_1 \sim \text{Bin}(m_1, \pi_1) \quad \text{and} \quad s_2 = m_2 y_2 \sim \text{Bin}(m_2, \pi_2). \tag{24.8}$$

The trick is to conduct inference *conditional on* $s_1 + s_2$. Note that under $H_0 : \pi_1 = \pi_2$, we have:

$$
\begin{aligned}
&\mathbb{P}[s_1 = t | s_1 + s_2 = v] \\
&\quad = \mathbb{P}[s_1 = t | s_1 + s_2 = v] \\
&\quad = \frac{\mathbb{P}[s_1 = t, s_2 = v - t]}{\mathbb{P}[s_1 + s_2 = v]} \\
&\quad = \frac{\binom{m_1}{t}\pi^t(1-\pi)^{m_1-t}\binom{m_2}{v-t}\pi^{v-t}(1-\pi)^{m_2-(v-t)}}{\binom{m_1+m_2}{v}\pi^v(1-\pi)^{m_1+m_2-v}} \\
&\quad = \frac{\binom{m_1}{t}\binom{m_2}{v-t}}{\binom{m_1+m_2}{v}}.
\end{aligned}
\tag{24.9}
$$

Therefore, a finite-sample p-value to test $H_0 : \pi_1 = \pi_2$ versus $H_1 : \pi_1 > \pi_2$ is $\mathbb{P}[s_1 \geq t | s_1 + s_2]$, which can be computed exactly based on the formula above.

Chapter 25

Poisson regression

25.1 Model definition and interpretation

The Poisson regression model (with offsets) is:

$$y_i \overset{\text{ind}}{\sim} \text{Poi}(\mu_i); \quad \log \mu_i = o_i + \boldsymbol{x}_{i*}^T \boldsymbol{\beta}. \tag{25.1}$$

Because the log of the mean is linear in the predictors, Poisson regression models are often called *loglinear models*. To interpret the coefficients, note that a unit increase in x_j (while keeping the other variables fixed) is associated with an increase in the predicted mean by a factor of e^{β_j}.

25.2 Example: Modeling rates

One cool feature of the Poisson model is that rates can be easily modeled with the help of offsets. Let's say that the count y_i is collected over the course of a time interval of length t_i, or a spatial region with area t_i, or a population of size t_i, etc. Then, it is meaningful to model:

$$y_i \overset{\text{ind}}{\sim} \text{Poi}(\pi_i t_i), \quad \log \pi_i = \boldsymbol{x}_{i*}^T \boldsymbol{\beta},$$

where π_i represents the rate of events per day / per square mile / per capita, etc. In other words:

$$y_i \overset{\text{ind}}{\sim} \text{Poi}(\mu_i), \quad \log \mu_i = \log t_i + \boldsymbol{x}_{i*}^T \boldsymbol{\beta},$$

which is exactly equation (25.1) with offsets $o_i = \log t_i$. For example, in single-cell RNA-sequencing, y_i is the number of reads aligning to a gene in cell i and t_i is the total number of reads measured in the cell, a quantity called the *sequencing depth*. We might use a Poisson regression model to carry out a *differential expression analysis* between two cell types.

25.3 Estimation and inference

25.3.1 Score, Fisher information, and Wald inference

We found in Chapter @ch-glm-theory that the score and Fisher information for Poisson regression are:

$$\boldsymbol{U}(\boldsymbol{\beta}) = \boldsymbol{X}^T(\boldsymbol{y} - \boldsymbol{\mu}),$$

and:

$$\boldsymbol{I}(\boldsymbol{\beta}) = \boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X} = \boldsymbol{X}^T\text{diag}(V(\mu_i))\boldsymbol{X} = \boldsymbol{X}^T\text{diag}(\mu_i)\boldsymbol{X},$$

respectively. Hence, the normal equations for the MLE are:

$$\boldsymbol{X}^T(\boldsymbol{y} - \hat{\boldsymbol{\mu}}).$$

Wald inference is based on the approximation:

$$\hat{\boldsymbol{\beta}} \dot{\sim} N(\boldsymbol{\beta}, (\boldsymbol{X}^T\text{diag}(\hat{\mu}_i)\boldsymbol{X})^{-1}).$$

25.3.2 Likelihood ratio inference

For likelihood ratio inference, we first derive the total deviance. The unit deviance of a Poisson distribution is:

$$d(y, \mu) = y \log \frac{y}{\mu} - (y - \mu).$$

Hence, the total deviance is:

$$D(\boldsymbol{y}, \boldsymbol{\mu}) = \sum_{i=1}^{n} d(y_i, \mu_i) = \sum_{i=1}^{n} \left(y_i \log \frac{y_i}{\mu_i} - (y_i - \mu_i) \right).$$

The residual deviance is then:

$$D(\boldsymbol{y}, \hat{\boldsymbol{\mu}}) = \sum_{i=1}^{n} \left(y_i \log \frac{y_i}{\hat{\mu}_i} - (y_i - \hat{\mu}_i) \right) = \sum_{i=1}^{n} y_i \log \frac{y_i}{\hat{\mu}_i}.$$

The last equality holds for any model containing the intercept, since by the normal equations we have $\sum_{i=1}^{n}(y_i - \hat{\mu}_i) = \mathbf{1}^T(\boldsymbol{y} - \hat{\boldsymbol{\mu}}) = 0$. We can carry out a likelihood ratio test for $H_0 : \boldsymbol{\beta}_S = \mathbf{0}$ via:

$$D(\boldsymbol{y}, \hat{\boldsymbol{\mu}}^0) - D(\boldsymbol{y}, \hat{\boldsymbol{\mu}}) = \sum_{i=1}^{n} y_i \log \frac{\hat{\mu}_i}{\hat{\mu}_i^0} \dot{\sim} \chi^2_{|S|}.$$

We can carry out a goodness-of-fit test via:

$$D(\boldsymbol{y}, \hat{\boldsymbol{\mu}}) = \sum_{i=1}^{n} y_i \log \frac{y_i}{\hat{\mu}_i} \dot{\sim} \chi^2_{n-p}.$$

25.3.3 Score-based inference

Recalling Section 22.4.2, the score test for $H_0 : \beta_j = 0$ is based on the approximation

$$\frac{\boldsymbol{x}_{*,j}^T(\boldsymbol{y} - \widehat{\boldsymbol{\mu}}^0)}{\sqrt{\boldsymbol{x}_{*,j}^T \widehat{\boldsymbol{W}}^0 \boldsymbol{x}_{*,j} - \boldsymbol{x}_{*,j}^T \widehat{\boldsymbol{W}}^0 \boldsymbol{X}_{*,\text{-}j} (\boldsymbol{X}_{*,\text{-}j}^T \widehat{\boldsymbol{W}}^0 \boldsymbol{X}_{*,\text{-}j})^{-1} \boldsymbol{X}_{*,\text{-}j}^T \widehat{\boldsymbol{W}}^0 \boldsymbol{x}_{*,j}}} \dot{\sim} N(0,1),$$

where

$$\widehat{\boldsymbol{W}}^0 = \text{diag}(\widehat{\boldsymbol{\mu}}^0).$$

On the other hand, the score test for goodness-of-fit is based on the Pearson X^2 statistic:

$$X^2 \equiv \sum_{i=1}^{n} \frac{(y_i - \hat{\mu}_i)^2}{\hat{\mu}_i} \dot{\sim} \chi^2_{n-p}.$$

25.4 Relationship between Poisson and multinomial distributions

Suppose that $y_i \overset{\text{ind}}{\sim} \text{Poi}(\mu_i)$ for $i = 1, \ldots, n$. Then,

$$\begin{aligned}
\mathbb{P}\left[y_1 = m_1, \ldots, y_n = m_n \middle| \sum_i y_i = m\right] &= \frac{\mathbb{P}[y_1 = m_1, \ldots, y_n = m_n]}{\mathbb{P}[\sum_i y_i = m]} \\
&= \frac{\prod_{i=1}^n e^{-\mu_i} \frac{\mu_i^{m_i}}{m_i!}}{e^{-\sum_i \mu_i} \frac{(\sum_i \mu_i)^m}{m!}} \\
&= \binom{m}{m_1, \ldots, m_n} \prod_{i=1}^n \pi_i^{m_i},
\end{aligned}$$

where

$$\pi_i \equiv \frac{\mu_i}{\sum_{i'=1}^n \mu_{i'}}.$$

We recognize the last expression in the previous display as the probability mass function of the multinomial distribution with parameters $(\pi_1, \ldots, \pi_n)$ summing to one. In words, the joint distribution of a set of independent Poisson distributions conditional on their sum is a multinomial distribution.

25.5 Example: 2×2 contingency tables

25.5.1 Poisson model for 2×2 contingency tables

Let's say that we have two binary random variables $x_1, x_2 \in \{0, 1\}$ with joint distribution $\mathbb{P}(x_1 = j, x_2 = k) = \pi_{jk}$ for $j, k \in \{0, 1\}$. We collect a total of n samples from this joint distribution and

summarize the counts in a 2×2 table, where y_{jk} is the number of times we observed $(x_1, x_2) = (j, k)$, so that:

$$(y_{00}, y_{01}, y_{10}, y_{11})|n \sim \text{Mult}(n, (\pi_{00}, \pi_{01}, \pi_{10}, \pi_{11})).$$

Our primary question is whether these two random variables are independent, i.e.

$$\pi_{jk} = \pi_{j+}\pi_{+k}, \tag{25.2}$$

where

$$\pi_{j+} \equiv \mathbb{P}[x_1 = j] = \pi_{j1} + \pi_{j2}; \quad \pi_{+k} \equiv \mathbb{P}[x_2 = k] = \pi_{1k} + \pi_{2k}.$$

We can express this equivalently as:

$$\pi_{00}\pi_{11} = \pi_{01}\pi_{10}.$$

In other words, we can express the independence hypothesis concisely as:

$$H_0 : \frac{\pi_{11}\pi_{00}}{\pi_{10}\pi_{01}} = 1. \tag{25.3}$$

Let's arbitrarily assume that, additionally, $n \sim \text{Poi}(\mu_{++})$. Then, by the relationship between Poisson and multinomial distributions, we have:

$$y_{jk} \overset{\text{ind}}{\sim} \text{Poi}(\mu_{++}\pi_{jk}).$$

Let $i \in \{1, 2, 3, 4\}$ index the four pairs

$$(x_1, x_2) \in \{(0, 0), (0, 1), (1, 0), (1, 1)\},$$

so that for $i = 1, \ldots, 4$, we have

$$y_i \overset{\text{ind}}{\sim} \text{Poi}(\mu_i); \quad \log \mu_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_{12} x_{i1} x_{i2}, \tag{25.4}$$

where:

$$\begin{aligned} \beta_0 = \log \mu_{++} + \log \pi_{00}; \quad \beta_1 = \log \frac{\pi_{10}}{\pi_{00}}; \\ \beta_2 = \log \frac{\pi_{01}}{\pi_{00}}; \quad \beta_{12} = \log \frac{\pi_{11}\pi_{00}}{\pi_{10}\pi_{01}}. \end{aligned} \tag{25.5}$$

Note that the independence hypothesis (25.3) reduces to the hypothesis $H_0 : \beta_{12} = 0$:

$$H_0 : \frac{\pi_{11}\pi_{00}}{\pi_{10}\pi_{01}} = 1 \quad \iff \quad H_0 : \beta_{12} = 0.$$

So the presence of an interaction in the Poisson regression is equivalent to a lack of independence between x_1 and x_2. We can test the latter hypothesis using our standard tools for Poisson regression.

For example, we can use the Pearson X^2 goodness-of-fit test. To apply this test, we must find the fitted means under the null hypothesis model:

$$y_i \overset{\text{ind}}{\sim} \text{Poi}(\mu_i); \quad \log \mu_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2}, \quad i = 1, \ldots, 4. \tag{25.6}$$

The normal equations give us the following:

$$\begin{aligned} y_{++} &\equiv \sum_{j,k=0}^{1} y_{jk} = \sum_{j,k=0}^{1} \hat{\mu}_{jk} \equiv \hat{\mu}_{++}; \\ y_{+1} &\equiv \sum_{j=0}^{1} y_{j1} = \sum_{j=0}^{1} \hat{\mu}_{j1} \equiv \hat{\mu}_{+1}; \\ y_{1+} &\equiv \sum_{k=0}^{1} y_{1k} = \sum_{k=0}^{1} \hat{\mu}_{1k} \equiv \hat{\mu}_{1+}. \end{aligned}$$

By combining these equations, we arrive at:

$$\hat{\mu}_{++} = y_{++}; \quad \hat{\mu}_{j+} = y_{j+} \text{ for all } j \in \{0, 1\}; \quad \hat{\mu}_{+k} = y_{+k} \text{ for all } k \in \{0, 1\}.$$

Therefore, the fitted means under the null hypothesis model 25.6 are:

$$\hat{\mu}_{jk} = \hat{\mu}_{++}\hat{\pi}_{jk} = \hat{\mu}_{++}\hat{\pi}_{j+}\hat{\pi}_{+k} = y_{++}\frac{y_{j+}}{y_{++}}\frac{y_{+k}}{y_{++}} = \frac{y_{j+}y_{+k}}{y_{++}}.$$

Hence, we have:

$$X^2 = \sum_{j,k=0}^{1} \frac{(y_{jk} - \widehat{\mu}_{jk})^2}{\widehat{\mu}_{jk}} = \sum_{j,k=0}^{1} \frac{(y_{jk} - y_{j+}y_{+k}/y_{++})^2}{y_{j+}y_{+k}/y_{++}}.$$

Alternatively, we can use the likelihood ratio test, which gives:

$$G^2 = 2\sum_{j,k=0}^{1} y_{jk} \log \frac{y_{jk}}{\widehat{\mu}_{jk}} = 2\sum_{j,k=0}^{1} y_{jk} \log \frac{y_{jk}}{y_{j+}y_{+k}/y_{++}}.$$

We would compare both X^2 and G^2 to a χ^2_1 distribution.

25.5.2 Inference is the same regardless of conditioning on margins

Now, our data might actually have been collected such that $n \sim \text{Poi}(\mu_{++})$, or maybe n was fixed in advance. Is the Poisson inference proposed above actually valid in the latter case? In fact, it is! To see this, let us consider the log likelihoods of the two models:

$$p_{\boldsymbol{\mu}}(\boldsymbol{y}) = p_{\mu_{++}}(y_{++} = n) p_{\boldsymbol{\pi}}(\boldsymbol{y} | y_{++} = n),$$

so:

$$\begin{aligned}\log p_{\boldsymbol{\mu}}(\boldsymbol{y}) &= \log p_{\mu_{++}}(y_{++} = n) + \log p_{\boldsymbol{\pi}}(\boldsymbol{y}|y_{++} = n) \\ &= C + \log p_{\boldsymbol{\pi}}(\boldsymbol{y}|y_{++} = n).\end{aligned}$$

In other words, the log-likelihoods of the Poisson and multinomial models, as a function of $\boldsymbol{\pi}$, differ from each other by a constant. Therefore, any likelihood-based inference in these models is equivalent. The same argument shows that conditioning on the row or column totals (as opposed to the overall total) also yields the same exact inference. Therefore, regardless of the sampling mechanism, we can always conduct an independence test in a 2×2 table via a Poisson regression.

25.5.3 Equivalence among Poisson and logistic regressions

We've talked about two ways to view a 2×2 contingency table. In the logistic regression view, we thought about one variable as a predictor and the other as a response, seeking to test whether the predictor has an impact on the response. In the Poisson regression view, we thought about the two variables symmetrically, seeking to test independence. It turns out that these two perspectives are equivalent. Recall that we have derived in equations 25.4 and 25.5 that $x_1 \perp\!\!\!\perp x_2$ if and only if $\beta_{12} = 0$ in the Poisson regression:

$$\log y_i \overset{\text{ind}}{\sim} \text{Poi}(\mu_i), \quad \log \mu_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_{12} x_{i1} x_{i2}, \quad i = 1, \ldots, 4.$$

However, we have:

$$\begin{aligned}\beta_{12} &= \log \frac{\pi_{11}\pi_{00}}{\pi_{01}\pi_{10}} \\ &= \log \frac{\pi_{11}/\pi_{01}}{\pi_{01}/\pi_{00}} = \log \frac{\mathbb{P}[x_2 = 1 \mid x_1 = 1]/\mathbb{P}[x_2 = 0 \mid x_1 = 1]}{\mathbb{P}[x_2 = 1 \mid x_1 = 0]/\mathbb{P}[x_2 = 0 \mid x_1 = 0]}.\end{aligned}$$

Recalling the logistic regression of x_2 on x_1:

$$\text{logit } \mathbb{P}[x_2 = 1 \mid x_1] = \tilde{\beta}_0 + \tilde{\beta}_1 x_1, \tag{25.7}$$

and that $\tilde{\beta}_1$ is the log odds ratio, we conclude that:

$$\beta_{12} = \tilde{\beta}_1,$$

so $x_1 \perp\!\!\!\perp x_2$ if and only if $\tilde{\beta}_1 = 0$. Due to the equivalence between Poisson and multinomial distributions, the hypothesis tests and confidence intervals for the log odds ratio β_{12} (or $\tilde{\beta}_1$) obtained from Poisson and logistic regressions will be the same.

25.6 Example: Poisson models for $J \times K$ contingency tables

Suppose now that $x_1 \in \{1, \ldots, J\}$ and $x_2 \in \{1, \ldots, K\}$. Then, we denote $\mathbb{P}[x_1 = j, x_2 = k] = \pi_{jk}$. We still are interested in testing for independence between j and k, which amounts to a goodness-of-fit test for the Poisson model:

$$y_{jk} \overset{\text{ind}}{\sim} \text{Poi}(\mu_{jk}); \quad \log \mu_{jk} = \beta_0 + \beta_j^1 + \beta_k^2.$$

The score (Pearson) and deviance-based goodness-of-fit statistics for this test are:

$$X^2 = \sum_{j=1}^{J} \sum_{k=1}^{K} \frac{(y_{jk} - \widehat{\mu}_{jk})^2}{\widehat{\mu}_{jk}} \quad \text{and} \quad G^2 = 2 \sum_{j=1}^{J} \sum_{k=1}^{K} y_{jk} \log \frac{y_{jk}}{\widehat{\mu}_{jk}},$$

where $\widehat{\mu}_{jk} = \widehat{y}_{++} \frac{y_{j+}}{y_{++}} \frac{y_{+k}}{y_{++}}$. Like with the 2×2 case, the test is the same regardless of whether we condition on the row sums, column sums, total count, or if we do not condition at all. The degrees of freedom in the full model is JK, while the degrees of freedom in the partial model is $J + K - 1$, so the degrees of freedom for the goodness-of-fit test is $JK - J - K + 1 = (J - 1)(K - 1)$. Pearson erroneously concluded that the test had $JK - 1$ degrees of freedom, which, when Fisher corrected it, created a lot of animosity between these two statisticians.

25.7 Example: Poisson models for $J \times K \times L$ contingency tables

These ideas can be extended to multi-way tables, for example, three-way tables. If we have $x_1 \in \{1, \ldots, J\}$, $x_2 \in \{1, \ldots, K\}$, $x_3 \in \{1, \ldots, L\}$, then we might be interested in testing several kinds of null hypotheses:

- Mutual independence: $H_0 : x_1 \perp\!\!\!\perp x_2 \perp\!\!\!\perp x_3$.
- Joint independence: $H_0 : x_1 \perp\!\!\!\perp (x_2, x_3)$.
- Conditional independence: $H_0 : x_1 \perp\!\!\!\perp x_2 \mid x_3$.

These three null hypotheses can be shown to be equivalent to the Poisson regression model:

$$y_{jkl} \overset{\text{ind}}{\sim} \text{Poi}(\mu_{jkl}),$$

where:

$$\log \mu_{jkl} = \beta_0 + \beta_j^1 + \beta_k^2 + \beta_l^3 \quad \text{(mutual independence)};$$

$$\log \mu_{jkl} = \beta_0 + \beta_j^1 + \beta_k^2 + \beta_l^3 + \beta_{kl}^{2,3} \quad \text{(joint independence)};$$

$$\log \mu_{jkl} = \beta_0 + \beta_j^1 + \beta_k^2 + \beta_l^3 + \beta_{jl}^{1,3} + \beta_l^{2,3} \quad \text{(conditional independence)}.$$

Chapter 26

Negative binomial regression

26.1 Overdispersion

A pervasive issue with Poisson regression is *overdispersion*: that the variances of observations are greater than the corresponding means. A common cause of overdispersion is omitted variable bias. Suppose that $y \sim \text{Poi}(\mu)$, where $\log \mu = \beta_0 + \beta_1 x_1 + \beta_2 x_2$. However, we omitted variable x_2 and are considering the GLM based on $\log \mu = \beta_0 + \beta_1 x_1$. If $\beta_2 \neq 0$ and x_2 is correlated with x_1, then we have a confounding issue. Let's consider the more benign situation that x_2 is independent of x_1. Then, we have

$$\mathbb{E}[y|x_1] = \mathbb{E}[\mathbb{E}[y|x_1, x_2]|x_1] = \mathbb{E}[e^{\beta_0+\beta_1 x_1+\beta_2 x_2}|x_1] = e^{\beta_0+\beta_1 x_1}\mathbb{E}[e^{\beta_2 x_2}] = e^{\beta_0'+\beta_1 x_1}. \tag{26.1}$$

So in the model for the mean of y, the impact of omitted variable x_2 seems only to have impacted the intercept. Let's consider the variance of y:

$$\text{Var}[y|x_1] = \mathbb{E}[\text{Var}[y|x_1, x_2]|x_1] + \text{Var}[\mathbb{E}[y|x_1, x_2]|x_1] = e^{\beta_0'+\beta_1 x_1} + e^{2(\beta_0'+\beta_1 x_1)}\text{Var}[e^{\beta_2 x_2}] > e^{\beta_0'+\beta_1 x_1} = \mathbb{E}[y|x_1]. \tag{26.2}$$

So indeed, the variance is larger than what we would have expected under the Poisson model.

26.2 Hierarchical Poisson regression

Let's say that $y|\boldsymbol{x} \sim \text{Poi}(\lambda)$, where $\lambda|\boldsymbol{x}$ is random due to the fluctuations of the omitted variables. A common distribution used to model nonnegative random variables is the *gamma* distribution $\Gamma(\mu, k)$, parameterized by a mean $\mu > 0$ and a *shape* $k > 0$. This distribution has probability density function

$$f(\lambda; k, \mu) = \frac{(k/\mu)^k}{\Gamma(k)} e^{-k\lambda/\mu}\lambda^{k-1}, \tag{26.3}$$

with mean and variance given by

$$\mathbb{E}[\lambda] = \mu; \quad \mathrm{Var}[\lambda] = \mu^2/k. \tag{26.4}$$

Therefore, it makes sense to augment the Poisson regression model as follows:

$$\lambda|\boldsymbol{x} \sim \Gamma(\mu, k), \quad \log \mu = \boldsymbol{x}^T\boldsymbol{\beta}, \quad y|\lambda \sim \mathrm{Poi}(\lambda). \tag{26.5}$$

26.3 Negative binomial distribution

A simpler way to write the hierarchical model (26.5) would be to marginalize out λ. Doing so leaves us with a count distribution called the *negative binomial distribution*:

$$\lambda \sim \Gamma(\mu, k),\ y|\lambda \sim \mathrm{Poi}(\lambda) \quad \Longrightarrow \quad y \sim \mathrm{NegBin}(\mu, k). \tag{26.6}$$

The negative binomial probability mass function is

$$p(y; \mu, k) = \int_0^\infty \frac{(k/\mu)^k}{\Gamma(k)} e^{-k\lambda/\mu} \lambda^{k-1} e^{-\lambda} \frac{\lambda^y}{y!} d\lambda = \frac{\Gamma(y+k)}{\Gamma(k)\Gamma(y+1)} \left(\frac{\mu}{\mu+k}\right)^y \left(\frac{k}{\mu+k}\right)^k. \tag{26.7}$$

This random variable has mean and variance given by

$$\mathbb{E}[y] = \mathbb{E}[\lambda] = \mu \quad \text{and} \quad \mathrm{Var}[y] = \mathbb{E}[\lambda] + \mathrm{Var}[\lambda] = \mu + \frac{\mu^2}{k}. \tag{26.8}$$

As we send $k \to \infty$, the distribution of λ tends to a point mass and the negative binomial distribution tends to $\mathrm{Poi}(\mu)$.

26.4 Negative binomial as exponential dispersion model

Let us see whether we can express the negative binomial model as an exponential dispersion model. First, let us write out the probability mass function:

$$p(y; \mu, k) = \exp\left(y \log \frac{\mu}{\mu+k} - k \log \frac{\mu+k}{k}\right) \frac{\Gamma(y+k)}{\Gamma(k)\Gamma(y+1)}. \tag{26.9}$$

Unfortunately, we run into difficulties expressing this probability mass function in EDM form, because there is not a neat decoupling between the natural parameter and the dispersion parameter. Indeed, for unknown k, the negative binomial model is *not* an EDM. However, we can still express the negative binomial model as an EDM (in fact, a one-parameter exponential family) if we treat k as known. In particular, we can read off that

$$\theta = \log \frac{\mu}{\mu+k}, \quad \psi(\theta) = k \log \frac{\mu+k}{k} = -k \log(1 - e^\theta). \tag{26.10}$$

An alternate parameterization of the negative binomial model is via $\gamma = 1/k$. With this parameterization, we have

$$\mathbb{E}[y] = \mu \quad \text{and} \quad \text{Var}[y] = \mu + \gamma\mu^2. \tag{26.11}$$

Here, γ acts as a kind of dispersion parameter, as the variance of y grows with γ. Note that the relationship between $\text{Var}[y]$ and γ is not exactly proportional, as it is in EDMs. Nevertheless, the γ parameter is often called the negative binomial *dispersion*. Note that setting $\gamma = 0$ recovers the Poisson distribution.

26.5 Negative binomial regression

Let's revisit the hierarchical model 26.5, writing it more succinctly in terms of the negative binomial distribution:

$$y_i \overset{\text{ind}}{\sim} \text{NegBin}(\mu_i, \gamma), \quad \log \mu_i = \boldsymbol{x}^T\boldsymbol{\beta}. \tag{26.12}$$

Notice that we typically assume that all observations share the same dispersion parameter γ. Reading off from equation (26.10), we see that the canonical link function for the negative binomial distribution is $\mu \mapsto \log \frac{\mu}{\mu+k}$. However, typically for negative binomial regression we use the log link $g(\mu) = \log \mu$ instead. This is the link of Poisson regression, and leads to more interpretable coefficient estimates. This is our first example of a non-canonical link!

26.6 Score and Fisher information

Recall from Chapter 4 that

$$\boldsymbol{U}(\boldsymbol{\beta}) = \frac{1}{\phi_0}\boldsymbol{X}^T\boldsymbol{M}\boldsymbol{W}(\boldsymbol{y} - \boldsymbol{\mu}) \quad \text{and} \quad \boldsymbol{I}(\boldsymbol{\beta}) = \frac{1}{\phi_0}\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X}, \tag{26.13}$$

where

$$\boldsymbol{W} \equiv \text{diag}\left(\frac{w_i}{V(\mu_i)(d\eta_i/d\mu_i)^2}\right) \quad \text{and} \quad \boldsymbol{M} \equiv \text{diag}\left(\frac{\partial \eta_i}{\partial \mu_i}\right). \tag{26.14}$$

In our case, we have

$$w_i = 1; \quad V(\mu_i) = \mu_i + \gamma\mu_i^2; \quad \frac{\partial \eta_i}{\partial \mu_i} = \frac{1}{\mu_i}. \tag{26.15}$$

Putting this together, we find that

$$\boldsymbol{W} = \text{diag}\left(\frac{\mu_i}{1+\gamma\mu_i}\right); \quad \boldsymbol{M} = \text{diag}\left(\frac{1}{1+\gamma\mu_i}\right). \tag{26.16}$$

26.7 Estimation in negative binomial regression

Negative binomial regression is an EDM when γ is known, but typically the dispersion parameter is unknown. Note that there is a dependency in ψ on k (i.e. on γ), which complicates things. It means that the estimate $\widehat{\boldsymbol{\beta}}$ depends on the parameter γ (this does not happen, for example, in the normal linear model case).[1] Therefore, estimation in negative binomial regression is typically an iterative procedure, where at each step $\boldsymbol{\beta}$ is estimated for the current value of γ and then γ is estimated based on the updated value of $\boldsymbol{\beta}$. Let's discuss each of these tasks in turn. Given a value of $\widehat{\gamma}$, we have the normal equations:

$$\boldsymbol{X}^T \text{diag}\left(\frac{1}{1+\widehat{\gamma}\widehat{\mu}_i}\right)(\boldsymbol{y}-\widehat{\boldsymbol{\mu}}) = 0. \tag{26.17}$$

This reduces to the Poisson normal equations when $\widehat{\gamma} = 0$. Solving these equations for a fixed value of $\widehat{\gamma}$ can be done via IRLS, as usual. Estimating γ for a fixed value of $\widehat{\boldsymbol{\beta}}$ can be done in several ways, including setting to zero the derivative of the likelihood with respect to γ. This results in a nonlinear equation (not given here) that can be solved iteratively.

26.8 Wald inference

Wald inference is based on

$$\widehat{\text{Var}}[\widehat{\boldsymbol{\beta}}] = (\boldsymbol{X}^T\widehat{\boldsymbol{W}}\boldsymbol{X})^{-1}, \quad \text{where} \quad \widehat{\boldsymbol{W}} = \text{diag}\left(\frac{\widehat{\mu}_i}{1+\widehat{\gamma}\widehat{\mu}_i}\right). \tag{26.18}$$

26.9 Likelihood ratio test inference

The negative binomial deviance is

$$D(\boldsymbol{y};\widehat{\boldsymbol{\mu}}) = 2\sum_{i=1}^{n}\left(y_i \log\frac{y_i}{\widehat{\mu}_i} - \left(y_i + \frac{1}{\widehat{\gamma}}\right)\log\frac{1+\widehat{\gamma}y_i}{1+\widehat{\gamma}\widehat{\mu}_i}\right). \tag{26.19}$$

We can use this for comparing nested models, **but not for goodness of fit testing!** The issue is that we have estimated the parameter γ, whereas goodness of fit tests are applicable only when the dispersion parameter is known.

26.10 Testing for overdispersion

It is reasonable to want to test for overdispersion, i.e., to test the null hypothesis $H_0 : \gamma = 0$. This is somewhat of a tricky task because $\gamma = 0$ is at the edge of the parameter space. We can do so using a likelihood ratio test. As it turns out, the likelihood ratio statistic T^{LRT} has asymptotic null distribution

$$T^{\text{LRT}} \equiv 2(\ell^{\text{NB}} - \ell^{\text{Poi}}) \dot{\sim} \frac{1}{2}\delta_0 + \frac{1}{2}\chi^2_1. \tag{26.20}$$

[1] Having said that, the dependency between $\widehat{\boldsymbol{\beta}}$ and $\widehat{\gamma}$ is weak, as the two are asymptotically independent parameters.

Here, δ_0 is the delta mass at zero. The reason for this is that, under the null, we can view the estimated dispersion parameter as being symmetrically distributed around 0. However, since the dispersion parameter is nonnegative, this means it gets rounded up to 0 with probability 1/2. Therefore, the likelihood ratio test for $H_0 : \gamma = 0$ rejects when

$$T^{\text{LRT}} > \chi^2_1(1 - 2\alpha). \tag{26.21}$$

Note that the above test for overdispersion can be viewed as a goodness of fit test for the Poisson GLM. It is different from the usual GLM goodness of fit tests, because the saturated model against which the latter tests stays in the Poisson family. Nevertheless, significant results in standard goodness of fit tests for Poisson GLMs are often an indication of overdispersion. Or, they may indicate omitted variable bias (e.g., you forgot to include an interaction), so it's somewhat tricky.

26.11 Overdispersion in logistic regression

Note that overdispersion is potentially an issue not only in Poisson regression models but in logistic regression models as well. Dealing with overdispersion in the latter case is more tricky, because the analog of the negative binomial model (the beta-binomial model) is not an exponential family. An alternate route to dealing with overdispersion is quasi-likelihood modeling, but this topic is beyond the scope of the course.

Chapter 27

R demo

27.1 Contingency table analysis

Let's take a look at the UC Berkeley admissions data:

```
library(readr)
library(dplyr)
library(ggplot2)
library(tibble)
library(tidyr)

ucb_data <- UCBAdmissions |> as_tibble()
ucb_data
```

```
# A tibble: 24 x 4
   Admit    Gender Dept      n
   <chr>    <chr>  <chr> <dbl>
 1 Admitted Male   A       512
 2 Rejected Male   A       313
 3 Admitted Female A        89
 4 Rejected Female A        19
 5 Admitted Male   B       353
 6 Rejected Male   B       207
 7 Admitted Female B        17
 8 Rejected Female B         8
 9 Admitted Male   C       120
10 Rejected Male   C       205
# i 14 more rows
```

It contains data on applicants to graduate school at Berkeley for the six largest departments in 1973 classified by admission and sex. Let's see whether there is an association between `Gender` and `Admit`. Let's first aggregate over department:

```
ucb_data_agg <- ucb_data |>
  group_by(Admit, Gender) |>
  summarise(n = sum(n), .groups = "drop")
ucb_data_agg
```

```
# A tibble: 4 x 3
  Admit    Gender     n
  <chr>    <chr>  <dbl>
1 Admitted Female   557
2 Admitted Male    1198
3 Rejected Female  1278
4 Rejected Male    1493
```

Let's see what the admissions rates are by gender:

```
ucb_data_agg |>
  group_by(Gender) |>
  summarise(`Admission rate` = sum(n*(Admit == "Admitted"))/sum(n))
```

```
# A tibble: 2 x 2
  Gender `Admission rate`
  <chr>             <dbl>
1 Female            0.304
2 Male              0.445
```

This suggests that men have substantially higher admission rates than women. Let's see if we can confirm this using either a Fisher's exact test or a Pearson chi-square test.

```
# first convert to 2x2 table format
admit_vs_gender <- ucb_data_agg |>
  pivot_wider(names_from = Gender, values_from = n) |>
  column_to_rownames(var = "Admit")
admit_vs_gender
```

```
         Female Male
Admitted    557 1198
Rejected   1278 1493
```

```
# Fisher exact test (note that the direction of the effect can be deduced)
fisher.test(admit_vs_gender)
```

```
    Fisher's Exact Test for Count Data

data:  admit_vs_gender
p-value < 2.2e-16
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
```

```
 0.4781839 0.6167675
sample estimates:
odds ratio
 0.5432254
```

```
# Chi-square test
chisq.test(admit_vs_gender)
```

```
    Pearson's Chi-squared test with Yates' continuity correction

data:  admit_vs_gender
X-squared = 91.61, df = 1, p-value < 2.2e-16
```

As a sanity check, let's run the Poisson regression underlying the chi-square test above.

```
pois_fit <- glm(n ~ Admit + Gender + Admit*Gender,
                family = "poisson",
                data = ucb_data_agg)
summary(pois_fit)
```

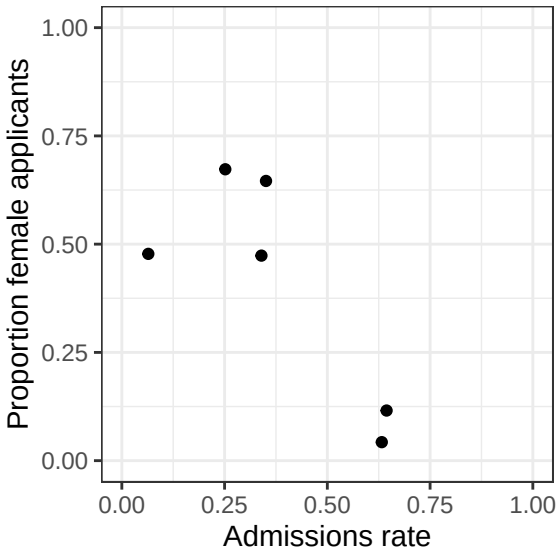
```
Call:
glm(formula = n ~ Admit + Gender + Admit * Gender, family = "poisson",
    data = ucb_data_agg)

Coefficients:
                         Estimate Std. Error z value Pr(>|z|)
(Intercept)               6.32257    0.04237 149.218   <2e-16 ***
AdmitRejected             0.83049    0.05077  16.357   <2e-16 ***
GenderMale                0.76584    0.05128  14.933   <2e-16 ***
AdmitRejected:GenderMale -0.61035    0.06389  -9.553   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance:  4.8635e+02  on 3  degrees of freedom
Residual deviance: -3.4062e-13  on 0  degrees of freedom
AIC: 43.225

Number of Fisher Scoring iterations: 2
```

Based on all of these tests, there seems to be a very substantial difference in admissions rates based on gender. That is not good.

But perhaps, women tend to apply to more selective departments? Let's look into this:

```
ucb_data |>
  group_by(Dept) |>
```

```
  summarise(admissions_rate = sum(n*(Admit == "Admitted"))/sum(n),
            prop_female_applicants = sum(n*(Gender == "Female"))/sum(n)) |>
  ggplot(aes(x = admissions_rate, y = prop_female_applicants)) +
  geom_point() +
  scale_x_continuous(limits = c(0, 1)) +
  scale_y_continuous(limits = c(0, 1)) +
  labs(x = "Admissions rate",
       y = "Proportion female applicants")
```



Indeed, it does seem that female applicants typically applied to more selective departments! This suggests that it is very important to control for department when evaluating the association between admissions and gender. To do this, we can run a test for conditional independence in the $J \times K \times L$ table:

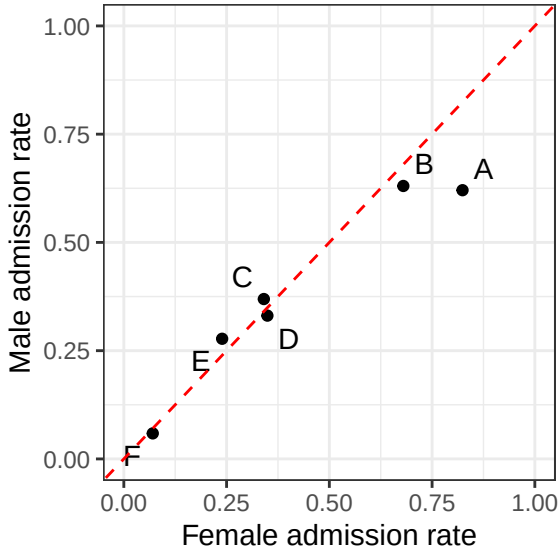
```
pois_fit <- glm(n ~ Admit + Dept + Gender + Admit:Dept + Gender:Dept,
                family = "poisson",
                data = ucb_data)
pchisq(sum(resid(pois_fit, "pearson")^2),
  df = pois_fit$df.residual,
  lower.tail = FALSE
)
```

```
[1] 0.002840164
```

Still we find a significant effect! But what is the direction of the effect? The chi-square test does not tell us. We can simply compute the admissions rates by department and plot them:

```
ucb_data |>
  group_by(Dept, Gender) |>
  summarise(`Admission rate` = sum(n*(Admit == "Admitted"))/sum(n),
```

```
            .groups = "drop") |>
  pivot_wider(names_from = Gender, values_from = `Admission rate`) |>
  ggplot(aes(x = Female, y = Male, label = Dept)) +
  geom_point() +
  ggrepel::geom_text_repel() +
  geom_abline(color = "red", linetype = "dashed") +
  scale_x_continuous(limits = c(0, 1)) +
  scale_y_continuous(limits = c(0, 1)) +
  labs(x = "Female admission rate",
       y = "Male admission rate")
```



Now the difference doesn't seem so huge, with most departments close to even and with department A heavily skewed towards admitting women!

27.2 Revisiting the crime data, again

```
library(tidyverse)
```

Here we are again, face to face with the crime data, with one last chance to get the analysis right. Let's load and preprocess it, as before.

```
# read crime data
crime_data <- read_tsv("data/Statewide_crime.dat")

# read and transform population data
population_data <- read_csv("data/state-populations.csv")
population_data <- population_data |>
  filter(State != "Puerto Rico") |>
```

```
    select(State, Pop) |>
    rename(state_name = State, state_pop = Pop)

  # collate state abbreviations
  state_abbreviations <- tibble(
    state_name = state.name,
    state_abbrev = state.abb
  ) |>
    add_row(state_name = "District of Columbia", state_abbrev = "DC")

  # add CrimeRate to crime_data
  crime_data <- crime_data |>
    mutate(STATE = ifelse(STATE == "IO", "IA", STATE)) |>
    rename(state_abbrev = STATE) |>
    filter(state_abbrev != "DC") |> # remove outlier
    left_join(state_abbreviations, by = "state_abbrev") |>
    left_join(population_data, by = "state_name") |>
    select(state_abbrev, Violent, Metro, HighSchool, Poverty, state_pop)

  crime_data
```

```
# A tibble: 50 x 6
   state_abbrev Violent Metro HighSchool Poverty state_pop
   <chr>          <dbl> <dbl>      <dbl>   <dbl>     <dbl>
 1 AK               593  65.6       90.2     8      724357
 2 AL               430  55.4       82.4    13.7   4934193
 3 AR               456  52.5       79.2    12.1   3033946
 4 AZ               513  88.2       84.4    11.9   7520103
 5 CA               579  94.4       81.3    10.5  39613493
 6 CO               345  84.5       88.3     7.3   5893634
 7 CT               308  87.7       88.8     6.4   3552821
 8 DE               658  80.1       86.5     5.8    990334
 9 FL               730  89.3       85.9     9.7  21944577
10 GA               454  71.6       85.2    10.8  10830007
# i 40 more rows
```

Let's recall the logistic regression we ran on these data in Chapter 4:

```
  bin_fit <- glm(Violent / state_pop ~ Metro + HighSchool + Poverty,
    weights = state_pop,
    family = "binomial",
    data = crime_data
  )
```

We had found very poor results from the goodness of fit test for this model. We have therefore omitted some important variables and/or we have serious overdispersion on our hands.

We haven't discussed in any detail how to deal with overdispersion in logistic regression models, so

let's try a Poisson model instead. The natural way to model rates using Poisson distributions is via offsets:

```
pois_fit <- glm(Violent ~ Metro + HighSchool + Poverty + offset(log(state_pop)),
  family = "poisson",
  data = crime_data
)
summary(pois_fit)
```

```
Call:
glm(formula = Violent ~ Metro + HighSchool + Poverty + offset(log(state_pop)),
    family = "poisson", data = crime_data)

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.609e+01  3.520e-01  -45.72   <2e-16 ***
Metro       -2.585e-02  5.727e-04  -45.15   <2e-16 ***
HighSchool   9.106e-02  3.450e-03   26.39   <2e-16 ***
Poverty      6.077e-02  4.852e-03   12.53   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 15589  on 49  degrees of freedom
Residual deviance: 11741  on 46  degrees of freedom
AIC: 12135

Number of Fisher Scoring iterations: 5
```

Again, everything is significant, and again, the regression summary shows that we have a huge residual deviance. This was to be expected, given that $\text{Bin}(m, \pi) \approx \text{Poi}(m\pi)$ for large m and small π. So, the natural thing to try is a negative binomial regression! Negative binomial regression is not implemented in the regular `glm` package, but `glm.nb()` from the `MASS` package is a dedicated function for this task. Let's see what we get:

```
nb_fit <- MASS::glm.nb(Violent ~ Metro + HighSchool + Poverty + offset(log(state_pop)),
  data = crime_data
)
summary(nb_fit)
```

```
Call:
MASS::glm.nb(formula = Violent ~ Metro + HighSchool + Poverty +
    offset(log(state_pop)), data = crime_data, init.theta = 1.467747388,
    link = log)

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
```

```
(Intercept) -10.254088   5.273418  -1.944   0.0518 .
Metro        -0.012188   0.008518  -1.431   0.1525
HighSchool    0.028052   0.052482   0.535   0.5930
Poverty      -0.026852   0.068449  -0.392   0.6948
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Negative Binomial(1.4677) family taken to be 1)

    Null deviance: 59.516  on 49  degrees of freedom
Residual deviance: 55.487  on 46  degrees of freedom
AIC: 732.58

Number of Fisher Scoring iterations: 1

              Theta:  1.468
          Std. Err.:  0.268

 2 x log-likelihood:  -722.575
```

Aha! Things are not looking so significant anymore! And the residual deviance is not as huge! Although, we must be careful! The residual deviance no longer has the usual χ^2 distribution because of the estimated dispersion parameter. So we don't really have an easy goodness of fit test. The estimated value of γ (confusingly called θ in the summary) is significantly different from zero, indicating overdispersion. Let's formally test for overdispersion using the nonstandard likelihood ratio test discussed above:

```
T_LRT <- 2 * (as.numeric(logLik(nb_fit)) - as.numeric(logLik(pois_fit)))
p_LRT <- pchisq(T_LRT, df = 1, lower.tail = FALSE)/2
p_LRT
```

```
[1] 0
```

So at the very least the NB model fits much better than the Poisson model. Let's do some inference based on this model. For example, we can get Wald confidence intervals:

```
confint.default(nb_fit)
```

```
                  2.5 %      97.5 %
(Intercept) -20.58979658 0.081620714
Metro        -0.02888413 0.004507747
HighSchool   -0.07481066 0.130915138
Poverty      -0.16100973 0.107305015
```

Or we can get LRT-based (i.e. profile) confidence intervals:

```
confint(nb_fit)
```

```
Waiting for profiling to be done...
```

```
                  2.5 %       97.5 %
(Intercept) -19.20209590 -0.860399348
Metro        -0.03153902  0.006365841
HighSchool   -0.06265118  0.115318303
Poverty      -0.13930110  0.085200541
```

Or we can get confidence intervals for the predicted means:

```
predict(nb_fit,
  newdata = crime_data |> column_to_rownames(var = "state_abbrev"),
  type = "response",
  se.fit = TRUE
)
```

```
$fit
       AK        AL        AR        AZ        CA        CO        CT        DE
 116.1520  617.7064  375.4895  700.6931 3257.5300  725.1538  436.7863  127.2572
       FL        GA        HI        ID        IL        IN        IA        KS
2232.2308 1301.2937  157.1416  263.8572 1379.1847  954.3366  546.5503  439.0649
       KY        LA        MA        MD        ME        MI        MN        MO
 541.5706  391.6745  747.7454  737.0032  274.2879 1322.9956  970.4078  871.2829
       MS        MT        NC        ND        NE        NH        NJ        NM
 380.6756  199.4947 1313.0904  134.8128  305.0634  261.1975  966.9940  204.3311
       NV        NY        OH        OK        OR        PA        RI        SC
 327.7316 1926.3861 1477.1713  495.9711  517.8397 1600.0813   96.3565  684.9102
       SD        TN        TX        UT        VA        VT        WA        WI
 160.9225  867.0224 2423.0647  416.6648 1244.5168  148.1635 1012.1932  892.0644
       WV        WY
 226.4515  100.1906

$se.fit
       AK        AL        AR        AZ        CA        CO        CT        DE
 21.00552 143.65071 130.44272 165.08459 910.57769 121.34777  85.53768  32.15169
       FL        GA        HI        ID        IL        IN        IA        KS
427.89514 173.04544  31.73873  40.28262 239.43324 147.21049 104.05752  68.82044
       KY        LA        MA        MD        ME        MI        MN        MO
133.28938 129.40665 150.23524 158.93816  92.04222 171.28409 216.32477 110.88843
       MS        MT        NC        ND        NE        NH        NJ        NM
138.28105  65.60335 379.90855  26.74061  69.62560  66.73731 220.88371  59.26953
       NV        NY        OH        OK        OR        PA        RI        SC
 64.30971 387.25204 241.24541  95.44911  81.97419 220.42078  33.97964 119.45174
       SD        TN        TX        UT        VA        VT        WA        WI
 41.50215 169.68896 738.95321 107.62725 209.14651  51.32810 191.75629 137.35158
       WV        WY
 71.55328  22.79279

$residual.scale
[1] 1
```

We can carry out some hypothesis tests as well, e.g. to test $H_0 : \beta_{\text{Metro}} = 0$:

```
nb_fit_partial <- MASS::glm.nb(Violent ~ HighSchool + Poverty + offset(log(state_pop)),
  data = crime_data
)
anova_fit <- anova(nb_fit_partial, nb_fit)
anova_fit
```

```
Likelihood ratio tests of Negative Binomial Models

Response: Violent
                                                    Model    theta Resid. df
1         HighSchool + Poverty + offset(log(state_pop)) 1.428675        47
2 Metro + HighSchool + Poverty + offset(log(state_pop)) 1.467747        46
     2 x log-lik.   Test    df LR stat.   Pr(Chi)
1       -724.1882
2       -722.5753 1 vs 2     1 1.612878 0.2040877
```

Part VI

Multiple testing

Chapter 28

Introduction

Consider the problem of assessing which variables in a GLM have nonzero coefficients. In the preceding chapters, we have described a variety of tests for obtaining p-values for each coefficient. Given this set of p-values (call them $p_1, \ldots, p_m$), we must determine which variables to deem significant. As it turns out, this task is a nontrivial one for several reasons, the first of which is the *multiplicity problem*.

28.1 The multiplicity problem

When R prints a regression summary, it adds stars to variables based on their p-values. Variables with p-values below 0.05 get one star, those with p-values below 0.01 get two stars, and those with p-values below 0.001 get three stars. A natural strategy for selecting significant variables is to choose those with one or more stars. However, the issue with this strategy is that even null variables (those with coefficients of zero) will sometimes have small p-values by chance (Figure 28.1). The more total variables we are testing, the more of them will have small p-values by chance. This is the *multiplicity problem*.

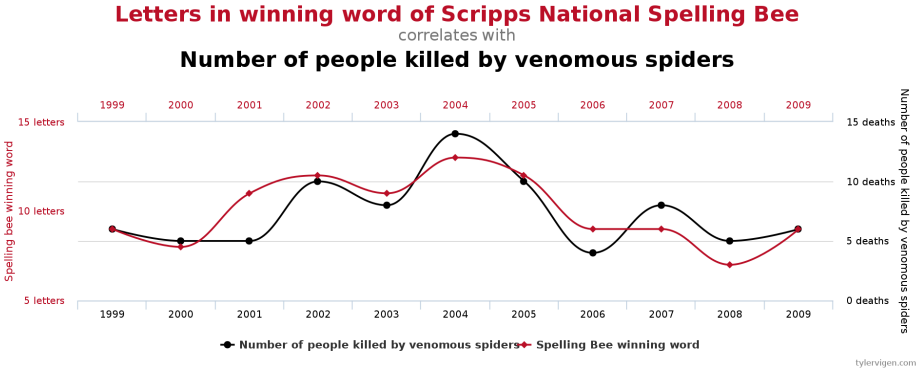


Figure 28.1: A spurious correlation resulting from data snooping.

To quantify this issue, consider the case when all m variables under consideration are null. Then, the chance that any one of them has a p-value below 0.05 is 0.05. So, the expected number of variables with one or more stars is $0.05m$. For example, if we have 100 variables, then we expect to see 5 variables with stars on average, even though none of the variables are actually relevant to the

response! The growth of the quantity $0.05m$ with m confirms that the multiplicity problem grows more severe as the number of hypotheses tested increases.

Another way of thinking about the multiplicity problem is in the context of *selection bias*. The process of scanning across all variables and selecting those with small p-values is a *selection event*. Once the selection event has occurred, one must consider the null distribution of a p-value *conditionally on the fact that it was selected*. Since the selection event favors small p-values, the null distribution of a p-value conditional on selection is no longer uniform; it becomes skewed toward zero. Interpreting p-values (and their accompanying stars) "at face value" is misleading because it ignores the crucial selection step. Other terms for this include "data snooping" and "p-hacking."

The multiplicity problem is not limited to regression. In the next two sections, we develop some definitions to describe the multiplicity problem more formally and generally.

28.2 Global testing and multiple testing

Suppose we have m null hypotheses $H_{01}, \ldots, H_{0m}$. Let \$p_1, ..., p_m be the corresponding p-values.

Definition 28.1. A p-value p_j for a null hypothesis H_{0j} is *valid* if

$$\mathbb{P}_{H_{0j}}[p_j \leq t] \leq t \quad \text{for all } t \in [0, 1]. \tag{28.1}$$

This definition covers the uniform distribution, as well as distributions that are stochastically larger than uniform. Distributions of the latter kind are often obtained from resampling-based tests, such as permutation tests. In the remainder of this chapter, we will assume that all p-values are valid.

Given a set of p-values, there are several inferential goals potentially of interest. These can be subdivided first into *global testing* and *multiple testing*.

Definition 28.2. A *global testing procedure* is a test of the *global null hypothesis*

$$H_0 \equiv \bigcap_{j=1}^{m} H_{0j}.$$

In other words, it is a function $\phi : (p_1, \ldots, p_m) \mapsto [0, 1]$. A global test has level α if it controls the Type-I error at this level:

$$\mathbb{E}_{H_0}[\phi(p_1, \ldots, p_m)] \leq \alpha. \tag{28.2}$$

A global testing procedure determines whether *any* of the null hypotheses can be rejected. In regression modeling, a global test would be a test of the hypothesis $H_0 : \beta_1 = \cdots = \beta_m = 0$.

Definition 28.3. A *multiple testing procedure* is a mapping from the set of p-values to a set of hypotheses to reject:

$$\mathcal{M} : (p_1, \ldots, p_m) \mapsto \hat{S} \subseteq \{1, \ldots, m\}.$$

A multiple testing procedure determines *which* of the null hypotheses can be rejected. In regression modeling, a multiple testing procedure would be a method for selecting which variables have nonzero coefficients, the problem we discussed in the beginning of this section.

28.3 Multiple testing goals

Let us define

$$\mathcal{H}_0 \equiv \{j \in \{1, \ldots, m\} : H_{0j} \text{ is true}\} \quad \text{and} \quad \mathcal{H}_1 \equiv \{j \in \{1, \ldots, m\} : H_{0j} \text{ is false}\}.$$

In other words, $\mathcal{H}_0$ is the set of indices of the true null hypotheses, and $\mathcal{H}_1$ is the set of indices of the false null hypotheses. There are two primary notions of Type-I error that multiple testing procedures seek to control: the *family-wise error rate* (FWER) and the *false discovery rate* (FDR).

28.3.1 Definitions of Type-I error rate and power

Definition 28.4. The family-wise error rate (FWER) of a multiple testing procedure $\mathcal{M} : (p_1, \ldots, p_m) \mapsto \hat{S}$ is the probability that it makes any false rejections:

$$\text{FWER}(\mathcal{M}) \equiv \mathbb{P}[\hat{S} \cap \mathcal{H}_0 \neq \varnothing].$$

A multiple testing procedure controls the FWER at level α if

$$\text{FWER}(\mathcal{M}) \leq \alpha.$$

Definition 28.5. The false discovery proportion (FDP) of a rejection set $\hat{S}$ is the proportion of these rejections that are false:

$$\text{FDP}(\hat{S}) \equiv \frac{|\hat{S} \cap \mathcal{H}_0|}{|\hat{S}|}, \quad \text{where} \quad \frac{0}{0} \equiv 0.$$

The false discovery rate (FDR) of a multiple testing procedure $\mathcal{M} : (p_1, \ldots, p_m) \mapsto \hat{S}$ is its expected false discovery proportion:

$$\text{FDR}(\mathcal{M}) \equiv \mathbb{E}[\text{FDP}(\hat{S})] = \mathbb{E}\left[\frac{|\hat{S} \cap \mathcal{H}_0|}{|\hat{S}|}\right]. \tag{28.3}$$

A multiple testing procedure controls the FDR at level q if

$$\text{FDR}(\mathcal{M}) \leq q.$$

Regardless of what error rate a multiple testing procedure is intended to control, we would like it to have high *power*:

$$\text{power}(\mathcal{M}) \equiv \mathbb{E}\left[\frac{|\hat{S} \cap \mathcal{H}_1|}{|\mathcal{H}_1|}\right].$$

28.3.2 Relationship between the FWER and FDR

Note that the FWER is a probability, while the FDR is an expected proportion. This distinction is highlighted by using the different symbols α and q for the nominal FWER and FDR levels, respectively. The FWER is a more stringent error rate than the FDR, because it can only be low if *no* false discoveries are made most of the time; the FDR can be low if false discoveries are a small proportion of the total number of discoveries most of the time.

Proposition 28.1. *For any multiple testing procedure $\mathcal{M}$, we have FDR($\mathcal{M}$) $\leq$ FWER($\mathcal{M}$). Therefore, a multiple testing procedure controlling the FWER at level α also controls the FDR at level α.*

Proof.

$$\text{FDR} \equiv \mathbb{E}\left[\frac{|\hat{S} \cap \mathcal{H}_0|}{|\hat{S}|}\right] \leq \mathbb{E}\left[1(|\hat{S} \cap \mathcal{H}_0| > 0)\right] \equiv \text{FWER}.$$

□

The FWER was the error rate of choice in the 20th century, when limitations on data collection permitted only small handfuls of hypotheses to be tested. In the 21st century, the internet and other new technologies permitted much larger-scale collection of data, leading to much larger sets of hypotheses being tested (e.g., tens of thousands). In this context, the less stringent FDR rate became more popular. In many cases, an initial large-scale FDR-controlling procedure is viewed as an *exploratory analysis*, whose goal is to nominate a smaller number of hypotheses for confirmation with follow-up experiments. The purpose of controlling the FDR in this context is to limit resources wasted on following up false leads.

Chapter 29

Global testing

Recall that a global test is a test of the intersection null hypothesis $H_0 \equiv \cap_{j=1}^m H_{0j}$. Let us first examine the naive global test, which rejects if any of the p-values are below α:

$$\phi_{\text{naive}}(p_1, \ldots, p_m) = 1\left(p_j \leq \alpha \text{ for some } j = 1, \ldots, m\right). \tag{29.1}$$

This test does not control the Type-I error. In fact, assuming the input p-values are independent, we have

$$\mathbb{E}_{H_0}[\phi_{\text{naive}}(p_1, \ldots, p_m)] = 1 - (1-\alpha)^m \to 1 \quad \text{as} \quad m \to \infty.$$

This is a manifestation of the multiplicity problem discussed before. In this section, we will discuss two ways of adjusting for multiplicity in the context of global testing:

- Bonferroni test: Powerful against few strong signals.
- Fisher combination test: Powerful against many weak signals.

Each test is listed with the alternative against which it is powerful. Note that in the context of global testing and multiple testing, the alternative is a multivariate object. The main difference between the Bonferroni test and the Fisher combination test is how the signal (i.e., deviation from the null) is spread across the m hypotheses being tested. If the signal is highly concentrated in a few non-null hypotheses, then the Bonferroni test is better. If the signal is spread out over many non-null hypotheses, then the Fisher combination test is better.

29.1 Bonferroni global test (Bonferroni, 1936 and Dunn, 1961)

29.1.1 Test definition and validity

The motivation for the Bonferroni global test is to find the strongest signal among the p-values and reject the global null if this signal is strong enough. It makes sense that such a strategy would be powerful against sparse alternatives. We define the Bonferroni test via

$$\phi(p_1, \ldots, p_m) \equiv 1\left(\min_{1 \leq j \leq m} p_j \leq \alpha/m\right).$$

The Bonferroni global test rejects if any of the p-values cross the *multiplicity-adjusted* or *Bonferroni-adjusted* significance threshold of α/m. This test can be viewed as a modified version of the naive

test (29.1), but with the significance threshold α adjusted downward to α/m. The more hypotheses we test, the more stringent the significance threshold must be.

Proposition 29.1. *The Bonferroni test controls the FWER at level α for any joint dependence structure among the p-values.*

Proof. We can verify the Type-I error control of the Bonferroni test via a union bound:

$$\mathbb{P}_{H_0}\left[\min_{1\le j\le m} p_j \le \alpha/m\right] \le \sum_{j=1}^{m} \mathbb{P}_{H_{0j}}\left[p_j \le \alpha/m\right] = m \cdot \alpha/m = \alpha.$$

□

29.1.2 The impact of p-value dependence

While the Bonferroni global test is valid for arbitrary p-value dependence structures, the underlying union bound may be loose for certain dependence structures. In particular, the Bonferroni bound derived above is tightest for independent p-values. Intuitively, the smallest p-value has the highest chance of being small if each p-value has its own independent source of randomness. Mathematically, let us compute the Type-I error of the Bonferroni global test under independence:

$$\mathbb{P}_{H_0}\left[\min_{1\le j\le m} p_j \le \alpha/m\right] = 1 - (1-\alpha/m)^m \approx \alpha.$$

Therefore, the Bonferroni test exhausts essentially its entire level under independence. On the other hand, under perfect dependence (i.e., $p_1 = \cdots = p_m$ almost surely), the Bonferroni test is quite conservative:

$$\mathbb{P}_{H_0}\left[\min_{1\le j\le m} p_j \le \alpha/m\right] = \mathbb{P}_{H_{01}}\left[p_1 \le \alpha/m\right] = \alpha/m.$$

In this special case, the level is m times lower than it should be, because no multiplicity adjustment is needed if the p-values are perfectly dependent.

29.2 Fisher combination test (Fisher, 1925)

If, on the other hand, we expect the signal to be spread out over many non-null hypotheses, the valuable evidence against the alternative is missed if only the minimum p-value is considered. In such circumstances, the Fisher combination test may be more powerful than the Bonferroni global test.

29.2.1 Test definition and validity

The Fisher combination test is based on the observation that

$$\text{if } p \sim U[0,1], \quad \text{then} \quad -2\log p \sim \chi^2_2.$$

Therefore, if $p_1, \ldots, p_m$ are independent uniform random variables, then we have

$$-2\sum_{j=1}^{m} \log p_j \sim \chi^2_{2m}.$$

This leads to the Fisher combination test:

$$\phi(p_1, \ldots, p_m) \equiv 1\left(-2\sum_{j=1}^{m} \log p_j \geq \chi^2_{2m}(1-\alpha)\right). \tag{29.2}$$

Proposition 29.2. *The Fisher combination test controls Type-I error at level α (28.2) if the p-values are independent.*

Proof. Under the null, the p-values are stochastically larger than uniform (29.2). Therefore, $-2\sum_{j=1}^{m} \log p_j$ is stochastically larger than χ^2_{2m}, from which the conclusion follows.

□

29.2.2 Discussion

The Fisher exact test has a similar flavor to another chi-squared test. Suppose $X_j \sim N(\mu_j, 1)$, and we would like to test $H_j : \mu_j = 0$. Under the global null, we have

$$\text{if } X_1, \ldots, X_m \overset{\text{i.i.d.}}{\sim} N(0,1), \text{ then } \sum_{j=1}^{m} X_j^2 \sim \chi^2_m. \tag{29.3}$$

It turns out that the tests based on (29.2) and (29.3) are quite similar. This helps us build intuition for what the Fisher combination test is doing: it's averaging the strengths of the signal across hypotheses.

The independence assumption of the Fisher combination test makes it significantly less broadly applicable than the Bonferroni global test. However, one common application of the Fisher combination test is *meta-analysis*: the combination of information across multiple studies of the same hypothesis (or very related hypotheses). In this setting, the p-values are independent across studies, and the Fisher combination test is a natural choice because the strength of the signal is roughly the same across studies since they are studying very related hypotheses.

Chapter 30

Multiple testing

Here we present one method each for FWER and FDR control.

30.1 The Bonferroni procedure for FWER control

We discussed the Bonferroni test for the global null. This test can be extended to an FWER-controlling procedure:

$$\hat{S} \equiv \{j : p_j \leq \alpha/m\} \tag{30.1}$$

Proposition 30.1. *The Bonferroni procedure controls the FWER at level α for arbitrary p-value dependence structures.*

Proof. We have

$$\mathbb{P}[\hat{S} \cap \mathcal{H}_0 \neq \varnothing] = \mathbb{P}\left[\min_{j \in \mathcal{H}_0} p_j \leq \alpha/m\right] \leq \sum_{j \in \mathcal{H}_0} \mathbb{P}[p_j \leq \alpha/m] = \frac{|\mathcal{H}_0|}{m}\alpha \leq \alpha.$$

This completes the proof.

□

Note that the FWER is actually controlled at the level $\frac{|\mathcal{H}_0|}{m}\alpha \leq \alpha$, making the Bonferroni test conservative to the extent that $|\mathcal{H}_0| < m$. The null proportion $\frac{|\mathcal{H}_0|}{m}$ has such an effect on the performance of many multiple testing procedures. Not all global tests can be extended to FWER-controlling procedures in this way. For example, the Fisher combination test does not single out any of the hypotheses, as it only aggregates the p-values. By contrast, the Bonferroni test searches for p-values that are individually very small, allowing it to double as an FWER-controlling procedure.

30.2 The Benjamini-Hochberg procedure for FDR control

Designing procedures with FDR control, as well as verifying the latter property, is typically harder than for FWER control. It is harder to decouple the effects of the individual hypotheses, as the

denominator $|S|$ in the FDR definition (28.3) couples them together. Both the FDR criterion and the most popular FDR-controlling procedure were proposed by Benjamini and Hochberg in 1995.

30.2.1 Procedure

To define the BH procedure, consider thresholding the p-values at $t \in [0,1]$. We would expect $\mathbb{E}[|\{j : p_j \le t\} \cap \mathcal{H}_0|] = |\mathcal{H}_0|t$ false discoveries among $\{j : p_j \le t\}$. Since $|\mathcal{H}_0|$ is unknown, we can bound it from above by mt. This leads to the FDP estimate:

$$\widehat{\mathrm{FDP}}(t) \equiv \frac{mt}{|\{j : p_j \le t\}|} \tag{30.2}$$

The BH procedure is then defined via:

$$\hat{S} \equiv \{j : p_j \le \widehat{t}\}, \quad \text{where} \quad \widehat{t} = \max\{t \in [0,1] : \widehat{\mathrm{FDP}}(t) \le q\} \tag{30.3}$$

In words, we choose the most liberal p-value threshold for which the estimated FDP is below the nominal level q. Note that the set over which the above maximum is taken is always nonempty because it at least contains 0: $\widehat{\mathrm{FDP}}(0) = \frac{0}{0} \equiv 0$.

30.2.2 FDR control under independence

Benjamini and Hochberg established that their procedure controls the FDR if the p-values are independent. Here we present an alternative argument due to Storey, Taylor, and Siegmund (2004).

Proposition 30.2. *The BH procedure controls the FDR at level q assuming that the p-values are independent.*

Proof. We have

$$\begin{aligned}\mathrm{FDR} = \mathbb{E}\left[\mathrm{FDP}(\widehat{t})\right] &= \mathbb{E}\left[\frac{|\{j \in \mathcal{H}_0 : p_j \le \widehat{t}\}|}{|\{j : p_j \le \widehat{t}\}|}\right] \\ &= \mathbb{E}\left[\frac{|\{j \in \mathcal{H}_0 : p_j \le \widehat{t}\}|}{m\widehat{t}}\widehat{\mathrm{FDP}}(\widehat{t})\right] \le q \cdot \mathbb{E}\left[\frac{|\{j \in \mathcal{H}_0 : p_j \le \widehat{t}\}|}{m\widehat{t}}\right].\end{aligned}$$

To prove that the last expectation is bounded above by 1, note that

$$M(t) \equiv \frac{|\{j \in \mathcal{H}_0 : p_j \le t\}|}{mt} \tag{30.4}$$

is a backwards martingale with respect to the filtration

$$\mathcal{F}_t = \sigma(\{p_j : j \in \mathcal{H}_1\}, |\{j \in \mathcal{H}_0 : p_j \le t'\}| \text{ for } t' \ge t), \tag{30.5}$$

with t running backwards from 1 to 0. Indeed, for $s < t$ we have

$$\mathbb{E}[M(s)|\mathcal{F}_t] = \mathbb{E}\left[\frac{|\{j \in \mathcal{H}_0 : p_j \leq s\}|}{ms}\middle| \mathcal{F}_t\right] = \frac{\frac{s}{t}|\{j \in \mathcal{H}_0 : p_j \leq t\}|}{ms} = \frac{|\{j \in \mathcal{H}_0 : p_j \leq t\}|}{mt} = M(t).$$

The threshold $\widehat{t}$ is a stopping time with respect to this filtration, so by the optional stopping theorem, we have

$$\mathbb{E}\left[\frac{|\{j \in \mathcal{H}_0 : p_j \leq \widehat{t}\}|}{m\widehat{t}}\right] = \mathbb{E}[M(\widehat{t})] \leq \mathbb{E}[M(1)] = \frac{|\mathcal{H}_0|}{m} \leq 1.$$

This completes the proof.

□

30.2.3 FDR control under dependence

Under dependence, the BH procedure's FDR can be bounded by a multiple of the nominal FDR level.

Proposition 30.3. *The BH procedure controls the FDR at level $q(1 + \frac{1}{2} + \cdots + \frac{1}{m})$ regardless of the p-value dependency structure.*

Proof. We have

$$\begin{aligned}
\mathrm{FDP}(\hat{S}) &= \sum_{k=1}^{m} \frac{|\hat{S} \cap \mathcal{H}_0|}{k} 1(|\hat{S}| = k) \\
&= \sum_{k=1}^{m} \sum_{j \in \mathcal{H}_0} \frac{1}{k} 1(j \in \hat{S}, |\hat{S}| = k) \\
&= \sum_{k=1}^{m} \sum_{j \in \mathcal{H}_0} \frac{1}{k} 1\left(p_j \leq \frac{qk}{m}, |\hat{S}| = k\right) \\
&\leq \sum_{j \in \mathcal{H}_0} \sum_{l=1}^{m} \frac{1}{l} 1\left(p_j \in \left[\frac{q(l-1)}{m}, \frac{ql}{m}\right]\right).
\end{aligned}$$

It follows that

$$\mathrm{FDR} = \mathbb{E}[\mathrm{FDP}(\hat{S})] \leq \frac{|\mathcal{H}_0|}{m} q\left(1 + \frac{1}{2} + \cdots + \frac{1}{m}\right).$$

This completes the proof.

□

30.3 Additional topics

30.3.1 Weighted multiple testing procedures

Sometimes, we may have more prior evidence against certain null hypotheses than others, which we wish to incorporate in the global testing or multiple testing procedure to boost power. A common approach to doing so is to *weight* the p-values. Letting $w_1, \ldots, w_m$ be p-value weights averaging to 1, define *weighted p-values* $\tilde{p}_j$ via:

$$\tilde{p}_j \equiv \frac{p_j}{w_j} \tag{30.6}$$

Note that p-values corresponding to hypotheses with large (small) weights will be made more (less) significant. We can then attempt to apply the above global testing and multiple testing procedures on the weighted p-values $\tilde{p}_j$ rather than the original p-values p_j. As it turns out, in many cases these weighted procedures retain the Type-I error guarantees of their unweighted counterparts.

Proposition 30.4. *The weighted variants of the Bonferroni global test, the Bonferroni FWER procedure, and the BH FDR procedure all control their respective Type-I error rates under the same conditions as their unweighted counterparts (arbitrary dependence for the Bonferroni procedures and independence for BH).*

Proof. Here, we prove the statement just for the Bonferroni global test; the remaining proofs are left as exercises. The weighted Bonferroni global test is as follows:

$$\phi(p_1, \ldots, p_m) \equiv 1\left(\min_{1 \le j \le m} \frac{p_j}{w_j} \le \frac{\alpha}{m}\right).$$

It follows that

$$\mathbb{E}_{H_0}[\phi(p_1, \ldots, p_m)] \le \sum_{j=1}^{m} \frac{\alpha}{m} w_j = \alpha.$$

The last equality follows from the fact that the weights w_j average to 1 by assumption.

This completes the proof.

□