STAT 9610 Lecture Notes

Eugene Katsevich

Fall 2022

Preface

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, and will be updated in subsequent iterations of STAT 9610.

Contents

Chapter 1

Linear models: Estimation

1.1 Introduction

See also Agresti 1.1

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:

$$g(\mu) = g(\mathbb{E}[y]) = \beta_0 x_0 + \cdots + \beta_{p-1} x_{p-1} = \eta. \tag{1.3}$$

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 (1.3) are called *generalized linear models* (GLMs). They specialize to linear models for identity link function, i.e. $g(\mu) = \mu$. The focus of this course are 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. This course is broken up into five units.

- **Chapter 1. Linear model: 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 (1.1) and (1.2).
- **Chapter 2. Linear model: 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.
- **Chapter 3. Linear model: 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.
- **Chapter 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*.
- **Chapter 5. GLMs: Special cases.** Looking more closely at the most important special cases of GLMs, including logistic regression and Poisson regression.

If time permits, we will cover further topics, including multiple testing (how to correct for multiplicity when testing many hypotheses—in GLMs or otherwise) and high-dimensional inference (how to carry out inference in situations where there are more predictors than samples).

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.

1.2 Types of predictors; interpreting linear model coefficients

See also Agresti 1.2

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*:

$$y = \beta_0 + \epsilon. \tag{1.4}$$

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:

$$y = \beta_0 + \beta_1 x_1 + \epsilon. \tag{1.5}$$

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 (1.5) 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 (1.5). 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 $x_j \equiv \mathbb{1}(w = w_j)$ for $j = 1, \ldots, C-1$ and consider the regression

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_{C-1} x_{C-1} + \epsilon. \tag{1.6}$$

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

$$y = \beta_0 + \beta_1 x_1 + \epsilon. \tag{1.7}$$

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 quantatitive 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

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \epsilon. \tag{1.8}$$

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 (1.5) would be nonzero but the coefficient in the multiple regression model (1.8) would be equal to zero. In this case, sex acts as a *confounder*.

Interactions. Note that the multiple regression model (1.8) 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

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 + \epsilon. \tag{1.9}$$

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.

1.3 Model matrices, model vectors spaces, and identifiability

See also Agresti 1.3-1.4

The matrix $\boldsymbol{X}$ is called the *model matrix* or the *design matrix*. Concatenating the linear model equations (1.1) and (1.2) across observations give 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 1.1).

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 $\boldsymbol{X}\boldsymbol{\beta} = \boldsymbol{X}\boldsymbol{\beta}'$.

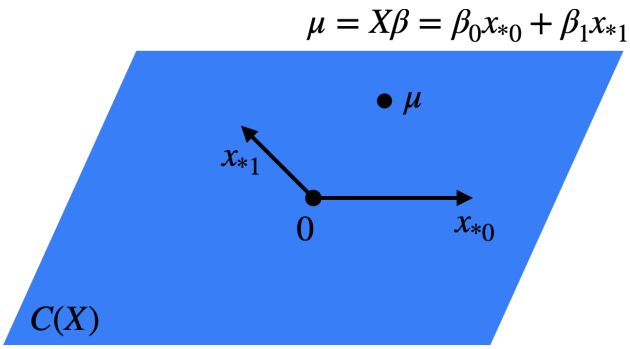


Figure 1.1: The model vector space.

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. $\boldsymbol{X\beta} \neq \boldsymbol{X\beta}'$ if $\boldsymbol{\beta} \neq \boldsymbol{\beta}'$. This is equivalent to the assumption that $\text{rank}(\boldsymbol{X}) = p$. 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, $\text{rank}(\boldsymbol{X}) = p$ 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 (1.6) 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.

1.4 Least squares estimation

See also Agresti 2.1.1, 2.7.1

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*:

$$\widehat{\boldsymbol{\beta}} \equiv \arg\min_{\boldsymbol{\beta}} \|\boldsymbol{y} - \boldsymbol{X\beta}\|^2. \tag{1.10}$$

The quantity

$$\|\boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}\|^2 = \|\boldsymbol{y} - \widehat{\boldsymbol{\mu}}\|^2 = \sum_{i=1}^{n} (y_i - \widehat{\mu}_i)^2 \tag{1.11}$$

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. 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

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

Letting $L(\boldsymbol{\beta}) = \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|^2$, we can do some calculus to derive that

$$\frac{\partial}{\partial \boldsymbol{\beta}} L(\boldsymbol{\beta}) = -\boldsymbol{X}^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}). \tag{1.12}$$

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

$$-\boldsymbol{X}^T(\boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}) = 0 \quad \Longleftrightarrow \quad \boldsymbol{X}^T\boldsymbol{X}\widehat{\boldsymbol{\beta}} = \boldsymbol{X}^T\boldsymbol{y}. \tag{1.13}$$

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

$$\widehat{\boldsymbol{\beta}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}. \tag{1.14}$$

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

$$\begin{aligned}
\mathrm{Var}[\widehat{\boldsymbol{\beta}}] &= \mathrm{Var}[(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}] \\
&= (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\mathrm{Var}[\boldsymbol{y}]\boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1} \\
&= (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T(\sigma^2\boldsymbol{I}_n)\boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1} \\
&= \sigma^2(\boldsymbol{X}^T\boldsymbol{X})^{-1}.
\end{aligned} \tag{1.15}$$

According to the Gauss-Markov theorm, this covariance matrix computed above is the smallest (in the sense of positive semidefinite matrices) among all linear unbiased estimates of $\boldsymbol{\beta}$.

1.5 Linear regression as orthogonal projection

See also Agresti 2.2, 2.3, 2.4.2, 2.4.3, 2.4.4

Let's think about the mapping $\boldsymbol{y} \mapsto \widehat{\boldsymbol{\mu}} = \boldsymbol{X}\widehat{\boldsymbol{\beta}} \in C(\boldsymbol{X})$. We claim that this mapping is an *orthogonal projection* (Figure 1.2). Geometrically it 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. One way of seeing this is to show that $\boldsymbol{v}^T(\boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}) = 0$ 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 (1.13).

To derive the projection matrix corresponding to this orthogonal projection, we write

$$\widehat{\boldsymbol{\mu}} = \boldsymbol{X}\widehat{\boldsymbol{\beta}} = \boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y} = \boldsymbol{H}\boldsymbol{y}, \tag{1.16}$$

where

$$\boldsymbol{H} \equiv \boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T \tag{1.17}$$

is called the *hat matrix*. This is the orthogonal projection matrix onto $C(\boldsymbol{X})$. Recall that a matrix $\boldsymbol{P}$ is an orthogonal projection onto a subspace $\boldsymbol{W}$ if 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 $\boldsymbol{P}\boldsymbol{v} = 0$. We can check for example the first of these conditions by noting that if $\boldsymbol{v} \in C(\boldsymbol{X})$, then $\boldsymbol{v} = \boldsymbol{X}\boldsymbol{\beta}$ for some $\boldsymbol{\beta} \in \mathbb{R}^p$. Therefore, we have

$$\boldsymbol{H}\boldsymbol{v} = \boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\beta} = \boldsymbol{X}\boldsymbol{\beta} = \boldsymbol{v}.$$

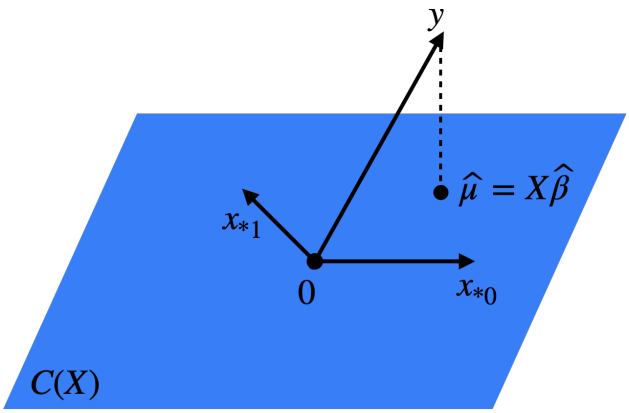


Figure 1.2: Least squares as orthogonal projection.

A simple example of $\boldsymbol{H}$ can be obtained by considering the intercept-only regression.

One consequence of this observation 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.

The orthogonality property of least squares, together with the Pythagorean theorem, leads to the following fundamental relationship. Let's say that $S \subset \{0, 1, \ldots, p-1\}$ is a subset of the predictors. First regress $\boldsymbol{y}$ on $\boldsymbol{X}$ to get $\widehat{\boldsymbol{\beta}}$ as usual. Then, we consider the *partial model matrix* $\boldsymbol{X}_{*S}$ obtained by selecting only the columns in S. Regression $\boldsymbol{y}$ on $\boldsymbol{X}_{*S}$ results in $\widehat{\boldsymbol{\beta}}_S$ (note: $\widehat{\boldsymbol{\beta}}_S$ is not necessarily obtained from $\widehat{\boldsymbol{\beta}}$ by extracting the coefficients corresponding to S). Now, consider the three points $\boldsymbol{y}, \boldsymbol{X}\widehat{\boldsymbol{\beta}}, \boldsymbol{X}_{*S}\widehat{\boldsymbol{\beta}}_S \in \mathbb{R}^n$. Since $\boldsymbol{X}\widehat{\boldsymbol{\beta}}$ and $\boldsymbol{X}_{*S}\widehat{\boldsymbol{\beta}}_S$ are both in $C(\boldsymbol{X})$, it follows by the orthogonal projection properpty that $\boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}$ is orthogonal to $\boldsymbol{X}\widehat{\boldsymbol{\beta}} - \boldsymbol{X}_{*S}\widehat{\boldsymbol{\beta}}_S$. In other words, these three points form a right triangle (Figure 1.3). By the Pythagorean theorem, we conclude that

$$\|\boldsymbol{y} - \boldsymbol{X}_{*S}\widehat{\boldsymbol{\beta}}_S\|^2 = \|\boldsymbol{X}\widehat{\boldsymbol{\beta}} - \boldsymbol{X}_{*S}\widehat{\boldsymbol{\beta}}_S\|^2 + \|\boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}\|^2. \tag{1.18}$$

We will rely on this fundamental relationship throughout this course.

For now, we can extract a few consequences of the relationship (1.18). As a starting point, consider the case when $S = \{0\}$, i.e. the partial model is the intercept-only model. In this case, $\boldsymbol{X}_{*S} = \mathbf{1}_n$ and $\widehat{\boldsymbol{\beta}}_{\boldsymbol{S}} = \bar{y}$. Therefore, equation (1.18) implies that

$$\|\boldsymbol{y} - \bar{y}\mathbf{1}_n\|^2 = \|\boldsymbol{X}\widehat{\boldsymbol{\beta}} - \bar{y}\mathbf{1}_n\|^2 + \|\boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}\|^2. \tag{1.19}$$

Equivalently, we can rewrite this equation as follows:

$$\text{SST} \equiv \sum_{i=1}^{n}(y_i - \bar{y})^2 = \sum_{i=1}^{n}(\widehat{\mu}_i - \bar{y})^2 + \sum_{i=1}^{n}(y_i - \widehat{\mu}_i)^2 \equiv \text{SSR} + \text{SSE}. \tag{1.20}$$

Figure 1.4 gives an interpretation of the ANOVA decomposition (1.20) in the case of the simple linear regression model (1.7).

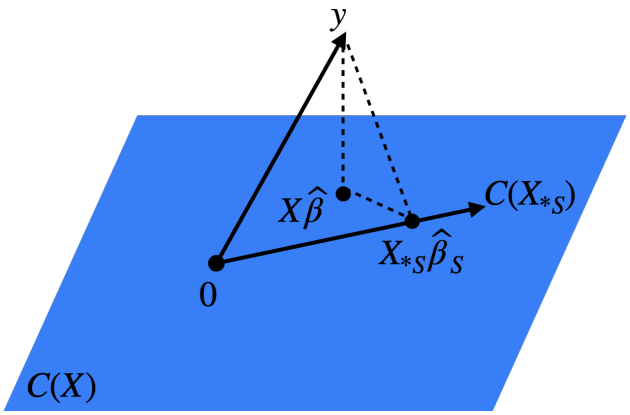


Figure 1.3: Pythagorean theorem for regression on a subset of predictors.

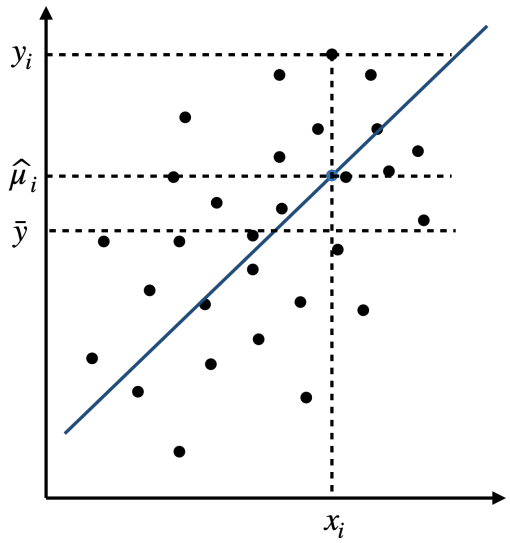
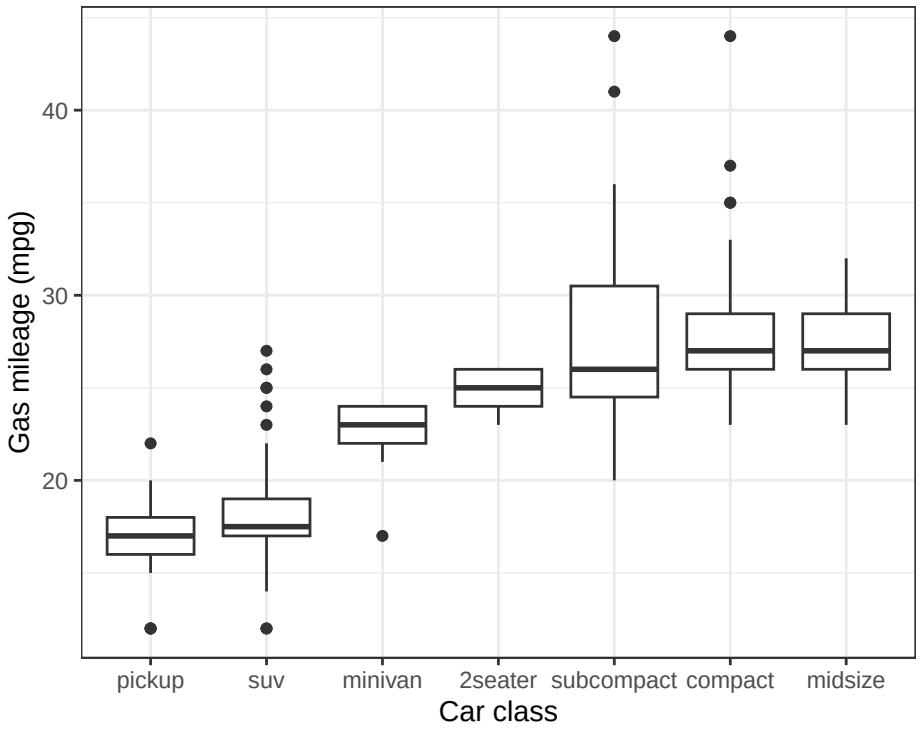


Figure 1.4: ANOVA decomposition for simple linear regression.

1.6 Correlation, multiple correlation, and R^2

See also Agresti 2.1.3, 2.4.6

ANOVA decomposition for C groups model. Let's consider the special case of the ANOVA decomposition (1.20) when the model matrix $\boldsymbol{X}$ represents a single categorical predictor w. In this case, each observation i is associated to 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:

$$\widehat{\mu}_i = \bar{y}_{c(i)}, \quad \text{where } \bar{y}_{c(i)} \equiv \frac{\sum_{i:c(i)=c} y_i}{|\{i : c(i) = c\}|}. \tag{1.21}$$

Therefore, we have

$$\text{SSR} = \sum_{i=1}^{n} (\widehat{\mu}_i - \bar{y})^2 = \sum_{i=1}^{n} (\bar{y}_{c(i)} - \bar{y})^2 \equiv \text{between-groups sum of squares (SSB)} \tag{1.22}$$

and

$$\text{SSE} = \sum_{i=1}^{n} (y_i - \widehat{\mu}_i)^2 = \sum_{i=1}^{n} (y_i - \bar{y}_{c(i)})^2 \equiv \text{within-groups sum of squares (SSW)}. \tag{1.23}$$

We therefore obtain the following corollary of the ANOVA decomposition (1.20):

$$\text{SST} = \text{SSB} + \text{SSW}. \tag{1.24}$$

R^2 definition and (multiple) correlation. The ANOVA decompositions (1.20) and (1.24) 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:

$$R^2 \equiv \frac{\text{SSR}}{\text{SST}} = \frac{\sum_{i=1}^{n} (\widehat{\mu}_i - \bar{y})^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2} = \frac{\|\boldsymbol{X}\widehat{\boldsymbol{\beta}} - \bar{y}\mathbf{1}_n\|^2}{\|\boldsymbol{y} - \bar{y}\mathbf{1}_n\|^2}. \tag{1.25}$$

By the decomposition (1.20), we have $R^2 \in [0, 1]$. The closer R^2 is to 1, the closely the data follow the fitted linear regression model. There is a connection between R^2 and correlation. To see this, let us first consider the case of the simple linear regression model with one predictor

$$y = \beta_0 + \beta_1 x_1 + \epsilon. \tag{1.26}$$

In this simple case, one can directly derive a formula for the fitted coefficients:

$$\widehat{\beta}_0 = \bar{y} - \widehat{\beta}_1 \bar{x}; \quad \widehat{\beta}_1 = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}. \tag{1.27}$$

Therefore,

$$\widehat{\boldsymbol{\mu}} - \bar{y}\mathbf{1}_n = \widehat{\beta}_0 \mathbf{1}_n + \widehat{\beta}_1 \boldsymbol{x}_{*1} - \bar{y}\mathbf{1}_n = \widehat{\beta}_1 (\boldsymbol{x}_{*1} - \bar{x}\mathbf{1}_n)$$

and thus

$$R^2 = \frac{\|\widehat{\boldsymbol{\mu}} - \bar{y}\mathbf{1}_n\|^2}{\|\boldsymbol{y} - \bar{y}\mathbf{1}_n\|^2} = \frac{\widehat{\beta}_1^2 \|\boldsymbol{x}_{*1} - \bar{x}\mathbf{1}_n\|^2}{\|\boldsymbol{y} - \bar{y}\mathbf{1}_n\|^2} = \left(\frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{(\sum_{i=1}^n (x_i - \bar{x})^2)^{1/2} (\sum_{i=1}^n (y_i - \bar{y})^2)^{1/2}} \right)^2 \equiv \rho_{xy}^2, \tag{1.28}$$

where ρ_{xy} is the sample correlation between x_1 and y. Therefore, in a simple linear regression, R^2 is the squared sample correlation between x_1 and y. For general regressions, one can derive that R^2 is the squared sample correlation between $\boldsymbol{X}\widehat{\boldsymbol{\beta}}$ and $\boldsymbol{y}$. For this reason, R^2 is sometimes called the *multiple correlation coefficient*.

Regression to the mean. Let's go back to the simple regression model (1.26), and let's take a closer look at $\widehat{\beta}_1$ in (1.27). Denoting by σ_x is the sample standard deviation of x_1 and σ_y is the sample standard deviation of y, we can rewrite $\widehat{\beta}_1$ as

$$\widehat{\beta}_1 = \frac{\sigma_y}{\sigma_x} \cdot \rho_{xy}. \tag{1.29}$$

Assuming that $\boldsymbol{x}_{*1}$ and $\boldsymbol{y}$ have been normalized to have the same sample standard deviation $\sigma_x = \sigma_y$, we find that the least squares coefficient $\widehat{\beta}_1$ is equal to the sample correlation ρ_{xy} between x and y. Since $|\rho_{xy}| < 1$ unless $\boldsymbol{x}_{*1}$ and $\boldsymbol{y}$ are perfectly correlated (by the Cauchy-Schwarz inequality), this means that

$$|\widehat{\mu}_i - \bar{y}| < |x_i - \bar{x}| \quad \text{for each } i. \tag{1.30}$$

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 next season.

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 including predictors $S \subset \{0, 1, \ldots, p-1\}$ and compare it to the model including all the predictors $\{0, 1, \ldots, p-1\}$. We can read off from the Pythagorean theorem (1.18) that

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

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

1.7 Collinearity, adjustment, and partial correlation

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 on what other predictors are in the model. This relationship is dictated by the extent to which the given predictor is correlated with the other predictors. In this section, we'll use some additional notation. Let $S \subset \{0, \dots, p-1\}$ be a group of predictors (we can assume without loss of generality that $S = \{0, \dots, s-1\}$ for some $1 \leq s < p$). Then, denote $\text{-}S \equiv \{0, \dots, p-1\} \setminus S$. Let $\widehat{\boldsymbol{\beta}}_S$ denote the least squares coefficients when regressing $\boldsymbol{y}$ on $\boldsymbol{X}_{*S}$ and let $\widehat{\boldsymbol{\beta}}_{S|\text{-}S}$ denote the least squares coefficients corresponding to S when regressing $\boldsymbol{y}$ on $\boldsymbol{X} = (\boldsymbol{X}_{*S}, \boldsymbol{X}_{*,\text{-}S})$.

Least squares estimates in the orthogonal case. The simplest case to analyze is when a groups of predictors $\boldsymbol{X}_{*S}$ is orthogonal to the rest of the predictors $\boldsymbol{X}_{*,\text{-}S}$ in the sense that

$$\boldsymbol{X}_{*S}^T \boldsymbol{X}_{*,\text{-}S} = \boldsymbol{0}. \tag{1.31}$$

In this case, we can derive the least squares coefficient vector $\widehat{\boldsymbol{\beta}} = (\widehat{\boldsymbol{\beta}}_{S|\text{-}S}, \widehat{\boldsymbol{\beta}}_{\text{-}S|S})$ from the normal equations:

$$\begin{aligned}
\begin{pmatrix} \widehat{\boldsymbol{\beta}}_{S|\text{-}S} \\ \widehat{\boldsymbol{\beta}}_{\text{-}S|S} \end{pmatrix} &= (\boldsymbol{X}^T \boldsymbol{X})^{-1} \boldsymbol{X}^T \boldsymbol{y} \\
&= \begin{pmatrix} \boldsymbol{X}_S^T \boldsymbol{X}_S & 0 \\ 0 & \boldsymbol{X}_{\text{-}S}^T \boldsymbol{X}_{\text{-}S} \end{pmatrix}^{-1} \begin{pmatrix} \boldsymbol{X}_S^T \\ \boldsymbol{X}_{\text{-}S}^T \end{pmatrix} \boldsymbol{y} \\
&= \begin{pmatrix} (\boldsymbol{X}_S^T \boldsymbol{X}_S)^{-1} \boldsymbol{X}_S^T \boldsymbol{y} \\ (\boldsymbol{X}_{\text{-}S}^T \boldsymbol{X}_{\text{-}S})^{-1} \boldsymbol{X}_{\text{-}S}^T \boldsymbol{y} \end{pmatrix} \\
&= \begin{pmatrix} \widehat{\boldsymbol{\beta}}_S \\ \widehat{\boldsymbol{\beta}}_{\text{-}S} \end{pmatrix}.
\end{aligned} \tag{1.32}$$

Therefore, the least squares coefficients when regressing $\boldsymbol{y}$ on $(\boldsymbol{X}_S, \boldsymbol{X}_{\text{-}S})$ are the same as those obtained from regressing $\boldsymbol{y}$ separately on $\boldsymbol{X}_S$ and $\boldsymbol{X}_{\text{-}S}$, i.e.

$$\widehat{\boldsymbol{\beta}}_{S|\text{-}S} = \widehat{\boldsymbol{\beta}}_S. \tag{1.33}$$

Least squares estimates via orthogonalization. Let's now focus our attention on a single predictor x_j. If this predictor is orthogonal to the remaining predictors, then the result (1.33) states that $\widehat{\beta}_{j|\text{-}j}$ can be obtained from simply regressing y on x_j. However, this is usually not the case. Usually, $\boldsymbol{x}_{*j}$ has a nonzero projection $\boldsymbol{X}_{*,\text{-}j}\widehat{\boldsymbol{\gamma}}$ onto $C(\boldsymbol{X}_{*,\text{-}j})$:

$$\boldsymbol{x}_{*j} = \boldsymbol{X}_{*,\text{-}j}\widehat{\boldsymbol{\gamma}} + \boldsymbol{x}_{*j}^{\perp}, \tag{1.34}$$

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 $C(\boldsymbol{X}_{*,\text{-}j})$.

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 (1.32) 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 $\boldsymbol{x}_{*j}^{\perp}$ (without intercept).

We can solve this univariate regression explicitly to obtain

$$\widehat{\beta}_{j|\text{-}j} = \frac{(\boldsymbol{x}_{*j}^{\perp})^T \boldsymbol{y}}{\|\boldsymbol{x}_{*j}^{\perp}\|^2}. \tag{1.35}$$

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

$$\widehat{\beta}_{j|\text{-}j} = \frac{(\boldsymbol{x}_{*j}^{\perp})^T(\boldsymbol{y} - \boldsymbol{X}_{*,\text{-}j}\widehat{\boldsymbol{\beta}}_{\text{-}j})}{\|\boldsymbol{x}_{*j}^{\perp}\|^2} = \frac{(\boldsymbol{x}_{*j} - \boldsymbol{X}_{*,\text{-}j}\widehat{\boldsymbol{\gamma}})^T(\boldsymbol{y} - \boldsymbol{X}_{*,\text{-}j}\widehat{\boldsymbol{\beta}}_{\text{-}j})}{\|\boldsymbol{x}_{*j} - \boldsymbol{X}_{*,\text{-}j}\widehat{\boldsymbol{\gamma}}\|^2}. \tag{1.36}$$

We can interpret this result as follows: 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.* A related quantity is the *partial correlation* between $\boldsymbol{x}_{*j}$ and $\boldsymbol{y}$ after controlling for $\boldsymbol{X}_{*,\text{-}j}$, defined as the correlation between $\boldsymbol{x}_{*j} - \boldsymbol{X}_{*,\text{-}j}\widehat{\boldsymbol{\gamma}}$ and $\boldsymbol{y} - \boldsymbol{X}_{*,\text{-}j}\widehat{\boldsymbol{\beta}}_{\text{-}j}$. We can then connect the least squares coefficient $\widehat{\beta}_j$ to this partial correlation in a similar spirit to equation (1.29).

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 (1.15). 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 (1.35) to calculate directly that

$$\text{Var}[\widehat{\beta}_{j|\text{-}j}] = \frac{\sigma^2}{\|\boldsymbol{x}_{*j}^{\perp}\|^2}. \tag{1.37}$$

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 $\boldsymbol{X}_{*,\text{-}j}$, 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

$$\text{Var}[\widehat{\beta}_j] = \frac{\sigma^2}{\|\boldsymbol{x}_{*j} - \bar{x}_j\mathbf{1}_n\|^2}. \tag{1.38}$$

Therefore, we can rewrite the variance (1.37) as

$$\text{Var}[\widehat{\beta}_{j|\text{-}j}] = \frac{\|\boldsymbol{x}_{*j} - \bar{x}_j\mathbf{1}_n\|^2}{\|\boldsymbol{x}_{*j} - \boldsymbol{X}_{*,\text{-}j}\widehat{\boldsymbol{\gamma}}\|^2} \cdot \text{Var}[\widehat{\beta}_j] = \frac{1}{1 - R_j^2} \cdot \text{Var}[\widehat{\beta}_j] \equiv \text{VIF}_j \cdot \text{Var}[\widehat{\beta}_j], \tag{1.39}$$

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 $\widehat{\beta}_{j|\text{-}j}$.

Aside: Average treatment effect estimation in causal inference. Suppose we'd like to study the effect of an exposure or treatment on a response y. 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 $y_i^{\text{obs}} = y_i(t_i)$. Based on the data $\{(t_i, y_i)\}_{i=1,\dots,n}$, 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 $y(1)$ and $y(0)$ for any unit. Additionally, usually in observational studies we have *confounding variables* $z_2, \dots, z_{p-1}$: variables that influence both the treatment assignment and the response. 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:

$$y(t) = \beta_0 + \beta_1 t + \beta_2 z_2 + \cdots + \beta_{p-1} z_{p-1} + \epsilon \quad \text{for } t \in \{0, 1\}, \quad \text{where } \epsilon \perp\!\!\!\perp t. \tag{1.40}$$

This assumption implies that the treatment effect is constant, and the response is a linear function of the treatment and observed confounders, and there is no unmeasured confounding. Note that

$$\tau \equiv \mathbb{E}[y(1) - y(0)] = \beta_1. \tag{1.41}$$

Furthermore,

$$y^{\text{obs}} = \beta_0 + \beta_1 t + \beta_2 z_2 + \cdots + \beta_{p-1} z_{p-1} + \epsilon \quad \text{for } t \in \{0, 1\}. \tag{1.42}$$

In this case, the average treatment effect τ is *identified* as the coefficient β_1 in the above regression, i.e. $\tau = \beta$. Therefore, the least squares estimate $\widehat{\beta}_1$ is an unbiased estimate of the average treatment effect. (Causal inference is beyond the scope of STAT 961; see STAT 921 instead.)

1.8 R demo

See also Agresti 2.6

The R demo will be based on the `ScotsRaces` data from the 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)."

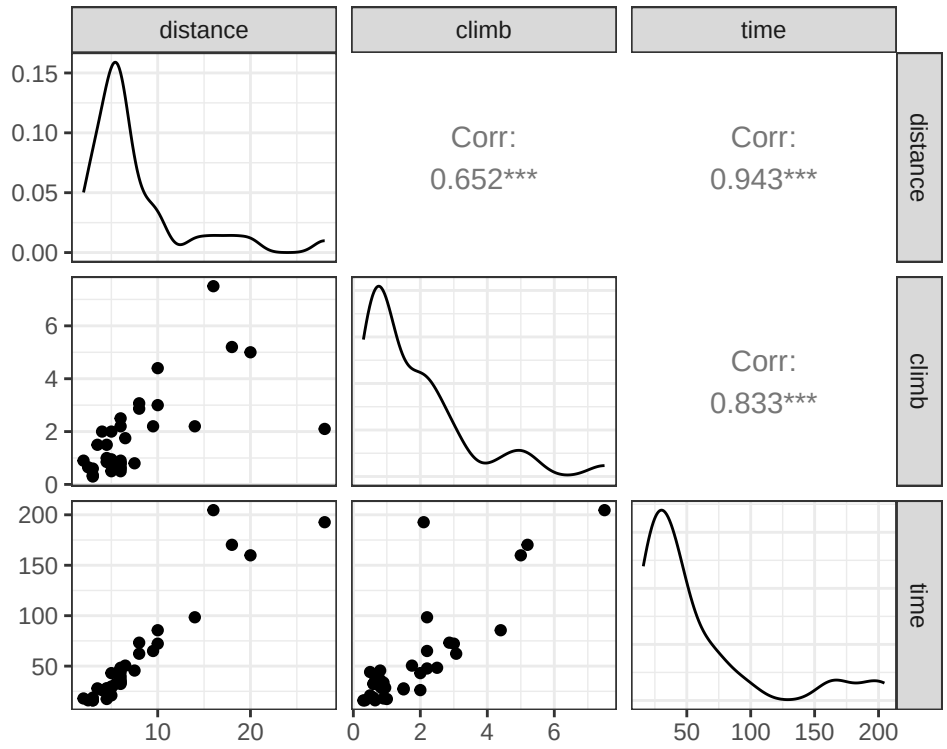
We will also familiarize ourselves with several important functions from the `tidyverse` packages, including the `ggplot2` package for data visualization and `dplyr` package for data manipulation.

```
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
```

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

## Rows:  35 Columns:  4
## - Column specification ----------------------------
## Delimiter:  "\t"
## chr (1):  race
## dbl (3):  distance, climb, time
##
## 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.

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
## # ... with 25 more rows
```

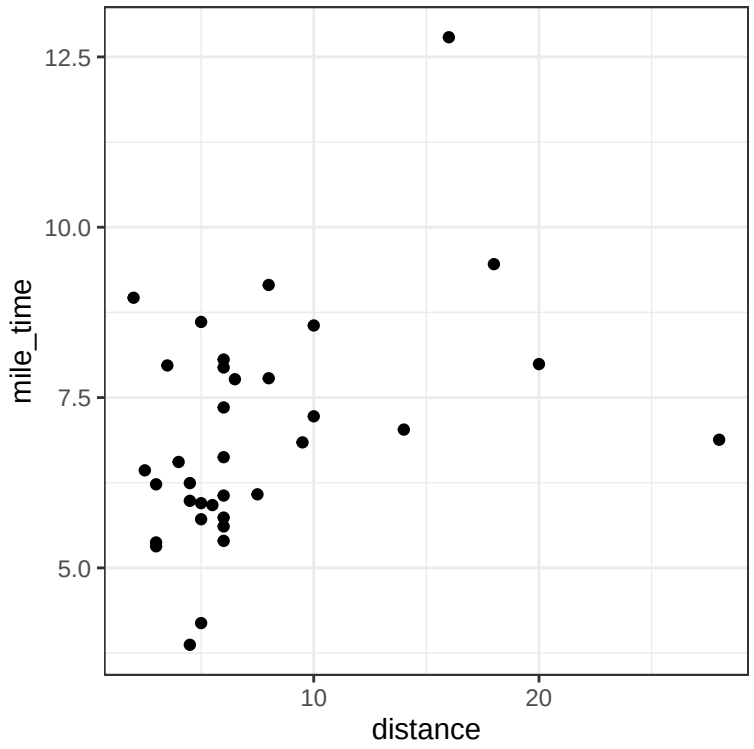
Exploration. Before modeling our data, let's first explore it.

```
# 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
```

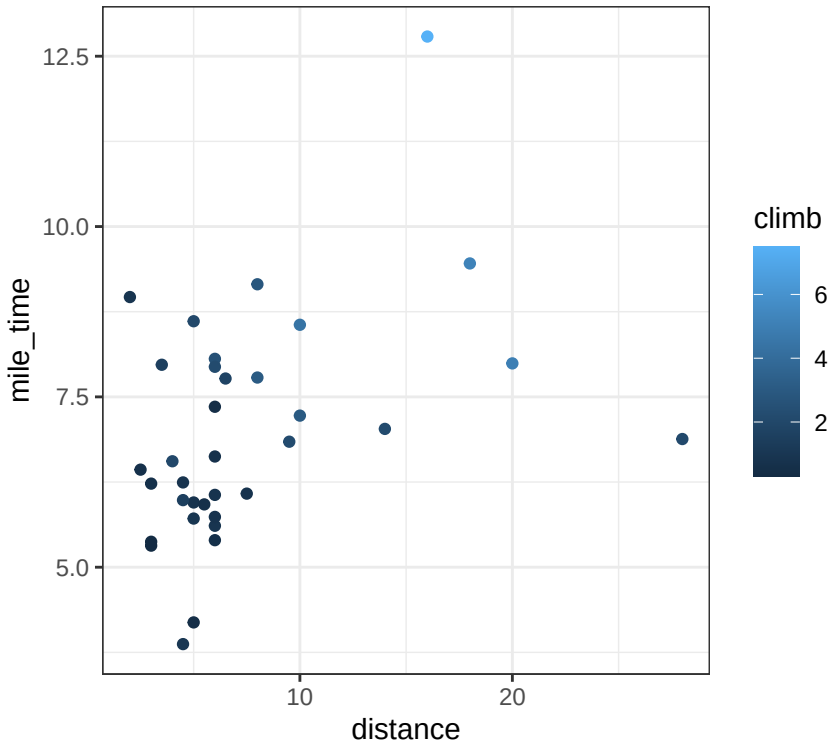



```
# mile time versus distance

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

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

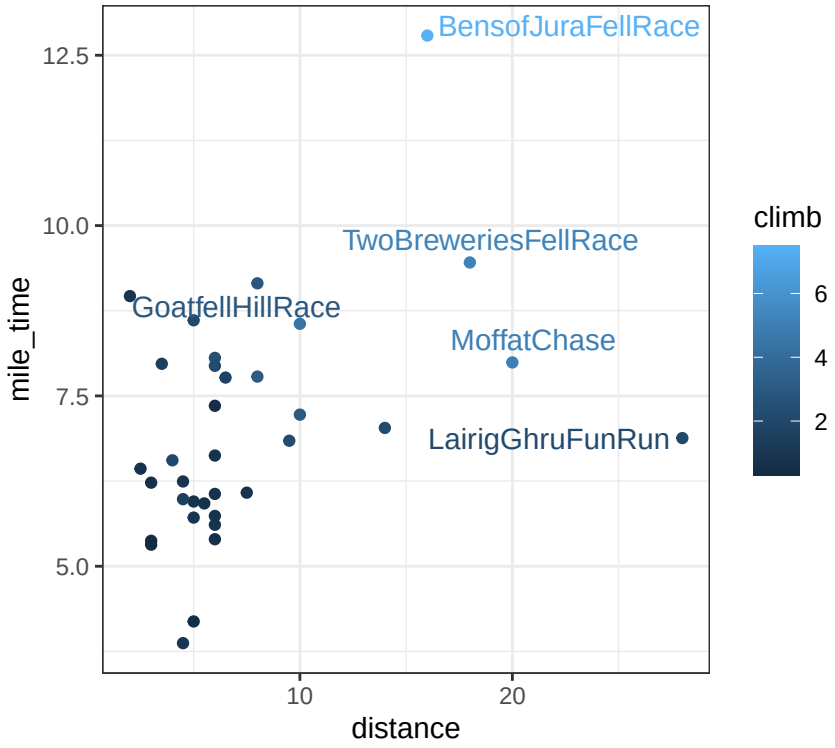
```
# plot mile time versus distance
scots_races %>%
  ggplot(aes(x = distance, y = mile_time)) +
  geom_point()
```



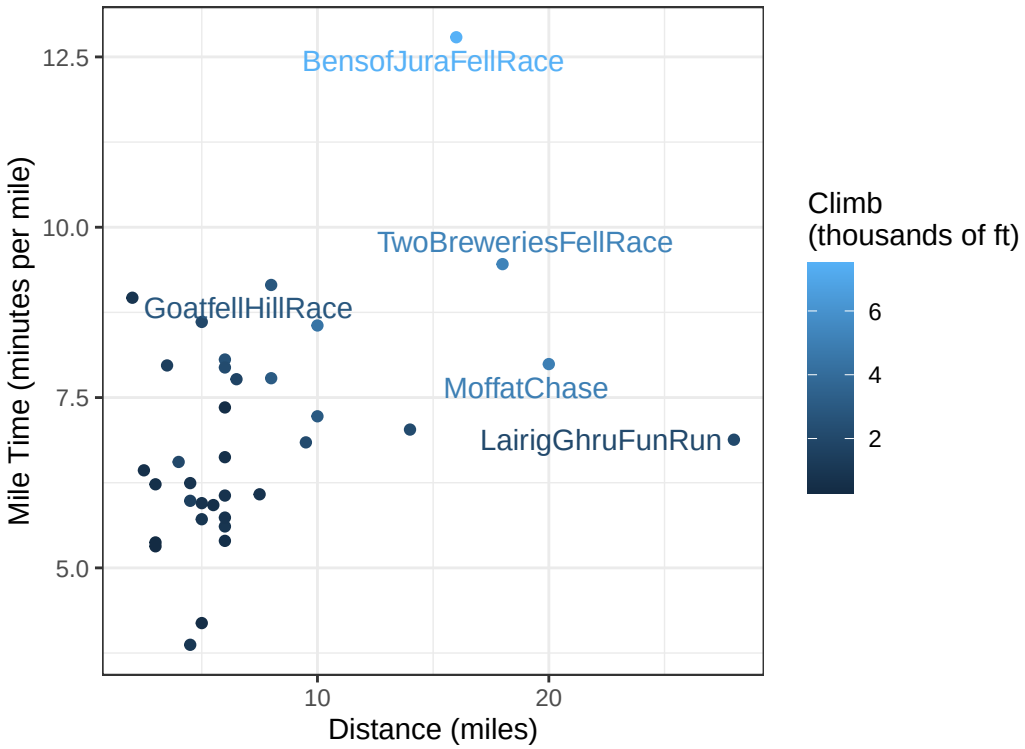

```
# add climb information as point color
scots_races %>%
  ggplot(aes(x = distance, y = mile_time, colour = climb)) +
  geom_point()
```



```
# 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)"
  )
```



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 (1.37).

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 (1.18) using the `anova` function. This function takes as arguments the partial model and the full model. For example, consider the partial model `time ~ distance`.

```
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:

```
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.

Adjustment and collinearity. We can also test the adjustment formula (1.35) 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`:

```
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 (1.39), 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?

Chapter 2

Linear models: Inference

We now understand the least squares estimator $\widehat{\boldsymbol{\beta}}$ from geometric and algebraic points of view. In Chapter 2, we will 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:

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \quad \text{where } \boldsymbol{\epsilon} \sim N(\boldsymbol{0}, \sigma^2 \boldsymbol{I}_n). \tag{2.1}$$

2.1 Building blocks for linear model inference

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 2.1.1), the distributions of linear regression estimates and residuals (Section 2.1.2), and estimation of the noise variance σ^2 (Section 2.1.3).

2.1.1 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 covariate matrix $\boldsymbol{\Sigma}$ if it has probability density

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

These random vectors have lots of special properties, including:

- (Linear transformation) If $\boldsymbol{w} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, then $\boldsymbol{A}\boldsymbol{w} + \boldsymbol{b} \sim N(\boldsymbol{A}\boldsymbol{\mu} + \boldsymbol{b}, \boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^T)$.
- (Independence) If $\begin{pmatrix} \boldsymbol{w}_1 \\ \boldsymbol{w}_2 \end{pmatrix} \sim N\left(\begin{pmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{pmatrix}, \begin{pmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{12}^T & \Sigma_{22} \end{pmatrix}\right)$, 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

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

2.1.2 The distributions of linear regression estimates and residuals

The most important distributional result in linear regression is that

$$\widehat{\boldsymbol{\beta}} \sim N(\boldsymbol{\beta}, \sigma^2(\boldsymbol{X}^T\boldsymbol{X})^{-1}). \tag{2.2}$$

Indeed, by the linear transformation property of the multivariate normal distribution,

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

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

$$\begin{aligned}\begin{pmatrix}\widehat{\boldsymbol{\mu}} \\ \widehat{\boldsymbol{\epsilon}}\end{pmatrix} = \begin{pmatrix}\boldsymbol{H}\boldsymbol{y} \\ (\boldsymbol{I}-\boldsymbol{H})\boldsymbol{y}\end{pmatrix} = \begin{pmatrix}\boldsymbol{H} \\ \boldsymbol{I}-\boldsymbol{H}\end{pmatrix}\boldsymbol{y} &\sim N\left(\begin{pmatrix}\boldsymbol{H} \\ \boldsymbol{I}-\boldsymbol{H}\end{pmatrix}\boldsymbol{X}\boldsymbol{\beta}, \begin{pmatrix}\boldsymbol{H} \\ \boldsymbol{I}-\boldsymbol{H}\end{pmatrix}\cdot\sigma^2\boldsymbol{I}\begin{pmatrix}\boldsymbol{H} & \boldsymbol{I}-\boldsymbol{H}\end{pmatrix}\right) \\ &= N\left(\begin{pmatrix}\boldsymbol{X}\boldsymbol{\beta} \\ \boldsymbol{0}\end{pmatrix}, \begin{pmatrix}\sigma^2\boldsymbol{H} & \boldsymbol{0} \\ \boldsymbol{0} & \sigma^2(\boldsymbol{I}-\boldsymbol{H})\end{pmatrix}\right).\end{aligned} \tag{2.3}$$

In other words,

$$\widehat{\boldsymbol{\mu}} \sim N(\boldsymbol{X}\boldsymbol{\beta}, \sigma^2\boldsymbol{H}) \quad \text{and} \quad \widehat{\boldsymbol{\epsilon}} \sim N(\boldsymbol{0}, \sigma^2(\boldsymbol{I}-\boldsymbol{H})), \quad \text{with} \quad \widehat{\boldsymbol{\mu}} \perp\!\!\!\perp \widehat{\boldsymbol{\epsilon}}. \tag{2.4}$$

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

$$\widehat{\boldsymbol{\beta}} \perp\!\!\!\perp \widehat{\boldsymbol{\epsilon}}. \tag{2.5}$$

2.1.3 Estimation of the noise variance σ^2

We can't quite do inference for $\boldsymbol{\beta}$ based on the distributional result (2.2) 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:

Lemma 2.1.1. *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 $\boldsymbol{P} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T$ 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 2.1.1 and equation (2.4) we have

$$\|\widehat{\boldsymbol{\epsilon}}\|^2 \sim \sigma^2\chi^2_{n-p}. \tag{2.6}$$

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

$$\widehat{\sigma}^2 \equiv \frac{1}{n-p}\|\widehat{\boldsymbol{\epsilon}}\|^2 \tag{2.7}$$

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 $\boldsymbol{\epsilon}$ onto the lower-dimensional subspace $C(\boldsymbol{X})^\perp$. To see this, note that

$$\widehat{\boldsymbol{\epsilon}} = (\boldsymbol{I}-\boldsymbol{H})\boldsymbol{y} = (\boldsymbol{I}-\boldsymbol{H})(\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}) = (\boldsymbol{I}-\boldsymbol{H})\boldsymbol{\epsilon}. \tag{2.8}$$

2.2 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 hypothesis in Sections 2.2.1 and 2.2.2, and then discuss the power of these tests in Section 2.3.

2.2.1 Testing a one-dimensional parameter

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

$$H_0 : \beta_j = 0. \tag{2.9}$$

According to (2.2), we have $\widehat{\beta}_j \sim N(0, \sigma^2/s_j^2)$, where, as we learned in Chapter 1,

$$s_j^2 \equiv [(\boldsymbol{X}^T\boldsymbol{X})^{-1}_{jj}]^{-1} = \|\boldsymbol{x}_{*j}^{\perp}\|^2. \tag{2.10}$$

Therefore,

$$\frac{\widehat{\beta}_j}{\sigma/s_j} \sim N(0, 1), \tag{2.11}$$

and we are tempted to define a level α test of the null hypothesis (2.9) based on this normal distribution. While this is infeasible since we don't know σ^2, we can substitute in the unbiased estimate (2.7) derived in Section 2.1.3. Then,

$$\mathrm{SE}_j \equiv \frac{\widehat{\sigma}}{s_j} \quad \text{is the standard error of } \widehat{\beta}_j, \tag{2.12}$$

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

$$t_j \equiv \frac{\widehat{\beta}_j}{\mathrm{SE}_j} = \frac{\widehat{\beta}_j}{\sqrt{\frac{1}{n-p}\|\widehat{\boldsymbol{\epsilon}}\|^2/s_j}}. \tag{2.13}$$

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

$$t_j = \frac{\frac{\widehat{\beta}}{\sigma/s_j}}{\sqrt{\frac{\sigma^{-2}\|\widehat{\boldsymbol{\epsilon}}\|^2}{n-p}}}. \tag{2.14}$$

Recalling the independence of $\widehat{\boldsymbol{\beta}}$ and $\widehat{\boldsymbol{\epsilon}}$ (2.5), the scaled chi square distribution of $\|\widehat{\boldsymbol{\epsilon}}\|^2$ (2.6), the standard normal distribution of $\frac{\widehat{\beta}}{\sigma/s_j}$ (2.11), we find that

$$\text{under } H_0 : \beta_j = 0, \quad t_j \sim \frac{N(0,1)}{\sqrt{\frac{1}{n-p}\chi^2_{n-p}}}, \quad \text{with numerator and denominator independent.} \tag{2.15}$$

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

$$\phi_t(\boldsymbol{X}, \boldsymbol{y}) = \mathbb{1}(|t_j| > t_{n-p}(1-\alpha/2)), \tag{2.16}$$

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,

$$\frac{1}{n-p}\chi^2_{n-p} \xrightarrow{P} 1 \quad \text{as} \quad n-p \to \infty, \tag{2.17}$$

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:

$$\phi_t(\boldsymbol{X}, \boldsymbol{y}) \approx \phi_z(\boldsymbol{X}, \boldsymbol{y}) \equiv \mathbb{1}(|t_j| > z(1-\alpha/2)), \tag{2.18}$$

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.

Example: One-sample model. Consider the intercept-only linear regression model $y = \beta_0 + \epsilon$, and let's apply 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

$$\mathrm{SE}_0^2 = \frac{\widehat{\sigma}^2}{n}, \quad \text{where} \quad \widehat{\sigma}^2 = \frac{1}{n-1}\|\boldsymbol{y} - \bar{y}\mathbf{1}_n\|^2. \tag{2.19}$$

Hence, we obtain the t statistic

$$t = \frac{\widehat{\beta}_0}{\mathrm{SE}_0} = \frac{\sqrt{n}\bar{y}}{\sqrt{\frac{1}{n-1}\|\boldsymbol{y} - \bar{y}\mathbf{1}_n\|^2}}. \tag{2.20}$$

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

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

$$y_i \sim \begin{cases} N(\beta_0, \sigma^2) & \text{for } x_i = 0; \\ N(\beta_0 + \beta_1, \sigma^2) & \text{for } x_i = 1. \end{cases} \tag{2.21}$$

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's define

$$\bar{y}_0 \equiv \frac{1}{n_0}\sum_{i:x_i=0} y_i, \quad \bar{y}_1 \equiv \frac{1}{n_1}\sum_{i:x_i=1} y_i, \quad \text{where} \quad n_0 = |\{i : x_i = 0\}| \text{ and } n_1 = |\{i : x_i = 1\}|. \tag{2.22}$$

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

$$s_1^2 \equiv \|\boldsymbol{x}_{*1}^{\perp}\|^2 = \|\boldsymbol{x}_{*1} - \frac{n_1}{n}\mathbf{1}\|^2 = n_1\frac{n_0^2}{n^2} + n_0\frac{n_1^2}{n^2} = \frac{n_0 n_1}{n} = \frac{1}{\frac{1}{n_0} + \frac{1}{n_1}} \tag{2.23}$$

and

$$\widehat{\sigma}^2 = \frac{1}{n-2}\left(\sum_{i:x_i=0}(y_i - \bar{y}_0)^2 + \sum_{i:x_i=1}(y_i - \bar{y}_1)^2\right). \tag{2.24}$$

Therefore, we arrive at a t-statistic of

$$t = \frac{\sqrt{\frac{1}{\frac{1}{n_0}+\frac{1}{n_1}}}(\bar{y}_1 - \bar{y}_0)}{\sqrt{\frac{1}{n-2}\left(\sum_{i:x_i=0}(y_i-\bar{y}_0)^2 + \sum_{i:x_i=1}(y_i-\bar{y}_1)^2\right)}}. \tag{2.25}$$

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

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

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

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

$$\frac{\boldsymbol{c}^T\widehat{\boldsymbol{\beta}}}{\widehat{\sigma}\sqrt{\boldsymbol{c}^T(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{c}}} \sim t_{n-p}, \tag{2.26}$$

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 jth coordinate equal to 1 and all others equal to zero.

2.2.2 Testing a multi-dimensional parameter

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

$$H_0 : \boldsymbol{y} = \boldsymbol{X}_{*,-S}\boldsymbol{\beta}_{-S} + \boldsymbol{\epsilon} \quad \text{versus} \quad H_1 : \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}. \tag{2.27}$$

To test this hypothesis, let us fit least squares coefficients $\widehat{\boldsymbol{\beta}}_{-S}$ and $\widehat{\boldsymbol{\beta}}$ for the partial model as well as the full model. If the partial model fits well, then the residuals $\boldsymbol{y} - \boldsymbol{X}_{*,-S}\widehat{\boldsymbol{\beta}}_{-\boldsymbol{S}}$ 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:

$$\|\boldsymbol{y} - \boldsymbol{X}_{*,-S}\widehat{\boldsymbol{\beta}}_{-\boldsymbol{S}}\|^2 = \|\boldsymbol{X}\widehat{\boldsymbol{\beta}} - \boldsymbol{X}_{*,-S}\widehat{\boldsymbol{\beta}}_{-\boldsymbol{S}}\|^2 + \|\boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}\|^2. \tag{2.28}$$

Let's consider the ratio

$$\frac{\|\boldsymbol{y} - \boldsymbol{X}_{*,-S}\widehat{\boldsymbol{\beta}}_{-\boldsymbol{S}}\|^2 - \|\boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}\|^2}{\|\boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}\|^2} = \frac{\|\boldsymbol{X}\widehat{\boldsymbol{\beta}} - \boldsymbol{X}_{*,-S}\widehat{\boldsymbol{\beta}}_{-S}\|^2}{\|\boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}\|^2}, \tag{2.29}$$

which is the relative increase in the residual sum of squares when going from the full model to the partial model. Let us rewrite this ratio in terms of projection matrices. Let $\boldsymbol{H}$ be the projection matrix for the full model, and let $\boldsymbol{H}_{-S}$ be the projection matrix for the partial model. Note that $\boldsymbol{H} - \boldsymbol{H}_{-S}$ is the projection matrix onto the $|S|$-dimensional space $C(\boldsymbol{X}) \cap C(\boldsymbol{X}_{-S})^{\perp}$ (Figure 2.1).

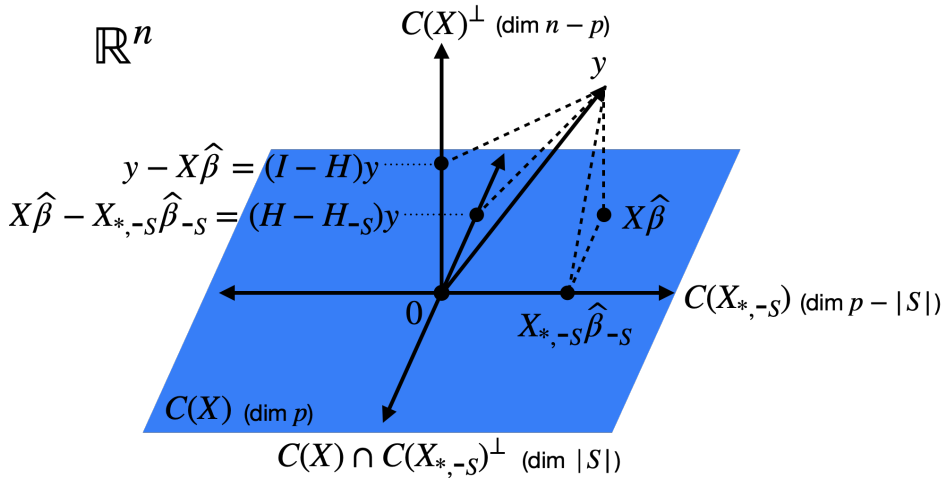


Figure 2.1: Geometry of the F-test. Orthogonality relationships stem from $C(\boldsymbol{X}_{*,S}) \perp C(\boldsymbol{X}) \cap C(\boldsymbol{X}_{*,S})^\perp \perp C(\boldsymbol{X})^\perp$.

We have

$$\frac{\|\boldsymbol{X}\widehat{\boldsymbol{\beta}} - \boldsymbol{X}_{*,S}\widehat{\boldsymbol{\beta}}_{S}\|^2}{\|\boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}\|^2} = \frac{\|(\boldsymbol{H} - \boldsymbol{H}_{S})\boldsymbol{y}\|^2}{\|(\boldsymbol{I} - \boldsymbol{H})\boldsymbol{y}\|^2}, \tag{2.30}$$

so the numerator and denominator are the squared norms of the projections of y onto $C(\boldsymbol{X}) \cap C(\boldsymbol{X}_{*,S})^\perp$ and $C(\boldsymbol{X})^\perp$, respectively (Figure 2.1). Under the null hypothesis, we have $\boldsymbol{y} = \boldsymbol{X}_{*,S}\boldsymbol{\beta}_{S} + \boldsymbol{\epsilon}$, and

$$(\boldsymbol{H} - \boldsymbol{H}_{S})\boldsymbol{X}_{*,S}\boldsymbol{\beta}_{S} = (\boldsymbol{I} - \boldsymbol{H})\boldsymbol{X}_{*,S}\boldsymbol{\beta}_{S} = 0 \tag{2.31}$$

because $\boldsymbol{X}_{*,S}\boldsymbol{\beta}_{S} \in C(\boldsymbol{X}_{*,S}) \perp C(\boldsymbol{X}) \cap C(\boldsymbol{X}_{*,S})^T \perp C(\boldsymbol{X})^\perp$. It follows that

$$\frac{\|(\boldsymbol{H} - \boldsymbol{H}_{S})\boldsymbol{y}\|^2}{\|(\boldsymbol{I} - \boldsymbol{H})\boldsymbol{y}\|^2} = \frac{\|(\boldsymbol{H} - \boldsymbol{H}_{S})\boldsymbol{\epsilon}\|^2}{\|(\boldsymbol{I} - \boldsymbol{H})\boldsymbol{\epsilon}\|^2}. \tag{2.32}$$

Since the projection matrices in the numerator and denominator project onto orthogonal subspaces, we have $(\boldsymbol{H} - \boldsymbol{H}_{S})\boldsymbol{\epsilon} \perp\!\!\!\perp (\boldsymbol{I} - \boldsymbol{H})\boldsymbol{\epsilon}$, with $\|(\boldsymbol{H} - \boldsymbol{H}_{S})\boldsymbol{\epsilon}\|^2 \sim \sigma^2\chi^2_{|S|}$ 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

$$F \equiv \frac{(\|\boldsymbol{y} - \boldsymbol{X}_{*,S}\widehat{\boldsymbol{\beta}}_{S}\|^2 - \|\boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}\|^2)/|S|}{\|\boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}\|^2/(n-p)}. \tag{2.33}$$

We have derived that under the null hypothesis,

$$F \sim \frac{\chi^2_{|S|}/|S|}{\chi^2_{n-p}/(n-p)}, \quad \text{with numerator and denominator independent.} \tag{2.34}$$

This distribution is called the F-distribution with $|S|$ and $n-p$ degrees of freedom, and 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

$$\phi_F(\boldsymbol{X}, \boldsymbol{y}) \equiv \mathbb{1}(F > F_{|S|,n-p}(1-\alpha)). \tag{2.35}$$

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 -$S = \{0\}$. The corresponding F statistic is

$$F \equiv \frac{(\|\boldsymbol{y} - \bar{y}\mathbf{1}\|^2 - \|\boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}\|^2)/(p-1)}{\|\boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}\|^2/(n-p)}, \tag{2.36}$$

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

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

$$\sum_{i=1}^{n}(y_i - \bar{y})^2 = \sum_{i=1}^{n}(\bar{y}_{c(i)} - \bar{y})^2 + \sum_{i=1}^{n}(y_i - \bar{y}_{c(i)})^2 = \text{SSB} + \text{SSW}. \tag{2.37}$$

The F-statistic in this case becomes

$$F = \frac{\sum_{i=1}^{n}(\bar{y}_{c(i)} - \bar{y})^2/(C-1)}{\sum_{i=1}^{n}(y_i - \bar{y}_{c(i)})^2/(n-C)} = \frac{\text{SSB}/(C-1)}{\text{SSW}/(n-C)}, \tag{2.38}$$

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

2.3 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.

The power of a t-test. 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,

$$t = \frac{\widehat{\beta}_j}{\text{SE}_j} = \frac{\beta_j}{\text{SE}_j} + \frac{\widehat{\beta}_j - \beta_j}{\text{SE}_j} \dot{\sim} N\left(\frac{\beta_j s_j}{\sigma}, 1\right). \tag{2.39}$$

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

$$\mathbb{E}[\phi_t] = \mathbb{P}[\phi_t = 1] \approx \mathbb{P}[|t| > z_{1-\alpha/2}] \approx \mathbb{P}\left[\left|N\left(\frac{\beta_j s_j}{\sigma}, 1\right)\right| > z_{1-\alpha/2}\right]. \tag{2.40}$$

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

$$\boldsymbol{x}_{*j}^{\perp} \approx \boldsymbol{x}_{*j} - \mathbb{E}[\boldsymbol{x}_{*j}|\boldsymbol{X}_{*,\text{-}j}], \tag{2.41}$$

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

$$s_j^2 \equiv \|\boldsymbol{x}_{*j}^{\perp}\|^2 \approx n\mathbb{E}[(x_j - \mathbb{E}[x_j|\boldsymbol{x}_{\text{-}j}])^2] = n\mathbb{E}[\text{Var}[x_j|\boldsymbol{x}_{\text{-}j}]]. \tag{2.42}$$

Hence, we can rewrite the alternative distribution (2.39) as

$$t \dot{\sim} N\left(\frac{\beta_j \cdot \sqrt{n} \cdot \sqrt{\mathbb{E}[\text{Var}[x_j|\boldsymbol{x}_{\text{-}j}]]}}{\sigma}, 1\right). \tag{2.43}$$

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.

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

$$F = \frac{\|\boldsymbol{X}\widehat{\boldsymbol{\beta}} - \boldsymbol{X}_{*,\text{-}S}\widehat{\boldsymbol{\beta}}_{-S}\|^2/|S|}{\|\boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}\|^2/|n-p|} = \frac{\|(\boldsymbol{H} - \boldsymbol{H}_{\text{-}S})\boldsymbol{y}\|^2/|S|}{\|(\boldsymbol{I} - \boldsymbol{H})\boldsymbol{y}\|^2/|n-p|} \approx \frac{\|(\boldsymbol{H} - \boldsymbol{H}_{\text{-}S})\boldsymbol{y}\|^2/|S|}{\sigma^2}. \tag{2.44}$$

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

Definition 2.3.1. *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 non-central chi-square random variable with d degrees of freedom and noncentrality parameter $\|\boldsymbol{\mu}\|^2$ and denote this distribution by $\chi_d^2(\|\boldsymbol{\mu}\|^2)$.*

It can be shown that if $\boldsymbol{P}$ is a projection matrix and $\boldsymbol{y} = \boldsymbol{\mu} + \boldsymbol{\epsilon}$, then $\frac{1}{\sigma^2}\|\boldsymbol{P}\boldsymbol{y}\|^2 \sim \chi^2_{\text{tr}(\boldsymbol{P})}(\frac{1}{\sigma^2}\|\boldsymbol{P}\boldsymbol{\mu}\|^2)$. It therefore follows that

$$F \approx \frac{\|(\boldsymbol{H} - \boldsymbol{H}_{\text{-}S})\boldsymbol{y}\|^2/|S|}{\sigma^2} \sim \frac{1}{|S|}\chi^2_{|S|}(\|(\boldsymbol{H} - \boldsymbol{H}_{\text{-}S})\boldsymbol{X}\boldsymbol{\beta}\|^2) = \frac{1}{|S|}\chi^2_{|S|}\left(\frac{1}{\sigma^2}\|\boldsymbol{X}_{*,S}^{\perp}\boldsymbol{\beta}_S\|^2\right). \tag{2.45}$$

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

$$\|\boldsymbol{X}_{*,S}^{\perp}\boldsymbol{\beta}_S\|^2 \approx n\boldsymbol{\beta}_S^T\mathbb{E}[\text{Var}[\boldsymbol{x}_S|\boldsymbol{x}_{\text{-}S}]]\boldsymbol{\beta}_S. \tag{2.46}$$

Therefore,

$$F \dot{\sim} \frac{1}{|S|}\chi^2_{|S|}\left(\frac{n\beta_S^T\mathbb{E}[\text{Var}[\boldsymbol{x}_S|\boldsymbol{x}_{\text{-}S}]]\boldsymbol{\beta}_S}{\sigma^2}\right), \tag{2.47}$$

which is similar in spirit to equation (2.43).

Power 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

$$\text{true underlying model: } y = \beta_0 x_0 + \beta_1 x_1 + \epsilon. \tag{2.48}$$

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

$$\text{model 0: } y = \beta_0 x_0 + \epsilon \quad \text{versus} \quad \text{model 1: } y = \beta_0 x_0 + \beta_1 x_1 + \epsilon. \tag{2.49}$$

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. 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. Similar reasoning can be applied to the F-test.

2.4 Confidence and prediction intervals

See also Agresti 3.3

In addition to hypothesis testing, we often want to construct confidence intervals for the coefficients.

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

$$\frac{\widehat{\beta}_j - \beta_j}{\widehat{\sigma}/s_j} \sim t_{n-p}. \tag{2.50}$$

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

$$\begin{aligned}
1 - \alpha = \mathbb{P}[|t_{n-p}| \leq t_{n-p}(1-\alpha/2)] &= \mathbb{P}\left[\left|\frac{\widehat{\beta}_j - \beta_j}{\widehat{\sigma}/s_j}\right| \leq t_{n-p}(1-\alpha/2)\right] \\
&= \mathbb{P}\left[\beta_j \in \left[\widehat{\beta}_j - \frac{\widehat{\sigma}}{s_j} t_{n-p}(1-\alpha/2), \widehat{\beta}_j + \frac{\widehat{\sigma}}{s_j} t_{n-p}(1-\alpha/2)\right]\right] \\
&\equiv \mathbb{P}\left[\beta_j \in \left[\widehat{\beta}_j - \text{SE}(\widehat{\beta}_j) t_{n-p}(1-\alpha/2), \widehat{\beta}_j + \text{SE}(\widehat{\beta}_j) t_{n-p}(1-\alpha/2)\right]\right] \\
&\equiv \mathbb{P}[\beta_j \in I_j].
\end{aligned} \tag{2.51}$$

The confidence interval I_j defined above therefore has $1 - \alpha$ coverage.

Confidence interval for $\mathbb{E}[y|\boldsymbol{x}_0]$. Suppose now that we have a new predictor vector $\boldsymbol{x}_0 \in \mathbb{R}^p$. The mean of the response for this predictor vector is $\mathbb{E}[y|\boldsymbol{x}_0] = \boldsymbol{x}_0^T\boldsymbol{\beta}$. Plugging in $\boldsymbol{x}_0$ for $\boldsymbol{c}$ in the relation (2.26), we obtain

$$\frac{\boldsymbol{x}_0^T\widehat{\boldsymbol{\beta}} - \boldsymbol{x}_0^T\boldsymbol{\beta}}{\widehat{\sigma}\sqrt{\boldsymbol{x}_0^T(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{x}_0}} \sim t_{n-p}.$$

From this we can derive that

$$\boldsymbol{x}_0^T\widehat{\boldsymbol{\beta}} \pm \widehat{\sigma}\sqrt{\boldsymbol{x}_0^T(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{x}_0} \cdot t_{n-p}(1-\alpha/2) \equiv \boldsymbol{x}_0^T\widehat{\boldsymbol{\beta}} \pm \mathrm{SE}(\boldsymbol{x}_0^T\widehat{\boldsymbol{\beta}}) \cdot t_{n-p}(1-\alpha/2) \tag{2.52}$$

is a $1-\alpha$ confidence interval for $\boldsymbol{x}_0^T\boldsymbol{\beta}$. We see that the width of this confidence interval depends on $\boldsymbol{x}_0$ through the quantity $\boldsymbol{x}_0^T(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{x}_0$. Let's give this quantity a closer look, in the case when the regression contains an intercept, i.e. $\boldsymbol{x}_{*,0} = \boldsymbol{1}$. Then, we have $\boldsymbol{x}_0 = (1, \boldsymbol{x}_{0,\text{-}0}^T)$. Then, defining $\bar{x} \in \mathbb{R}^{p-1}$ as the vector of column-wise means of $\boldsymbol{X}_{*,\text{-}0}$, we can rewrite the regression as

$$y = \beta_0 + \boldsymbol{x}_{\text{-}0}^T\boldsymbol{\beta}_{\text{-}0} + \epsilon \equiv \beta_0' + (\boldsymbol{x}_{\text{-}0} - \bar{x})^T\boldsymbol{\beta}_{\text{-}0} + \epsilon. \tag{2.53}$$

Therefore, we seek a prediction interval for $\boldsymbol{x}_0^T\boldsymbol{\beta} = \beta_0' + (\boldsymbol{x}_{0,\text{-}0} - \bar{x})^T\boldsymbol{\beta}_{\text{-}0}$. With this reformulation, we can compute

$$\begin{aligned}\boldsymbol{x}_0^T(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{x}_0 &= (1 \ (\boldsymbol{x}_{0,\text{-}0} - \bar{x})^T)\begin{pmatrix}\boldsymbol{1}^T\boldsymbol{1} & 0\\ 0 & \boldsymbol{X}_{*,\text{-}0}^T\boldsymbol{X}_{*,\text{-}0}\end{pmatrix}^{-1}\begin{pmatrix}1\\ \boldsymbol{x}_{0,\text{-}0} - \bar{x}\end{pmatrix}\\ &= \frac{1}{n} + (\boldsymbol{x}_{0,\text{-}0} - \bar{x})^T(\boldsymbol{X}_{*,\text{-}0}^T\boldsymbol{X}_{*,\text{-}0})^{-1}(\boldsymbol{x}_{0,\text{-}0} - \bar{x}).\end{aligned} \tag{2.54}$$

Hence, we see that this quantity grows larger as $\boldsymbol{x}_{0,\text{-}0} - \bar{x}$ grows larger, and achieves its minimum when $\boldsymbol{x}_{0,\text{-}0} = \bar{x}$. Let's look at the special case when $p = 2$, so there is just one predictor except the intercept. Then, we have $\boldsymbol{X}_{*,\text{-}0} = \boldsymbol{x}_{*,1} - \bar{x}_1$, so

$$\boldsymbol{x}_0^T(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{x}_0 = \frac{1}{n} + \frac{(x_{01} - \bar{x}_1)^2}{\|\boldsymbol{x}_{*,1} - \bar{x}_1\|^2}. \tag{2.55}$$

Prediction interval for $y|\boldsymbol{x}_0$. 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 y_0 of y for $\boldsymbol{x} = \boldsymbol{x}_0$, i.e. a *prediction interval*. Note that

$$y_0 - \boldsymbol{x}_0^T\widehat{\beta} = \boldsymbol{x}_0^T\beta + \epsilon_0 - \boldsymbol{x}_0^T\widehat{\beta} = \epsilon_0 + \boldsymbol{x}_0^T(\beta - \widehat{\beta}) \sim N(0, \sigma^2 + \sigma^2\boldsymbol{x}_0^T(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{x}_0). \tag{2.56}$$

Therefore, we have

$$\frac{y_0 - \boldsymbol{x}_0^T\widehat{\beta}}{\widehat{\sigma}\sqrt{1 + \boldsymbol{x}_0^T(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{x}_0}} \sim t_{n-p}, \tag{2.57}$$

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

$$\boldsymbol{x}_0^T\widehat{\boldsymbol{\beta}} \pm \widehat{\sigma}\sqrt{1 + \boldsymbol{x}_0^T(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{x}_0} \cdot t_{n-p}(1-\alpha/2) \equiv \boldsymbol{x}_0^T\widehat{\boldsymbol{\beta}} \pm \mathrm{SE}(\boldsymbol{x}_0^T\widehat{\boldsymbol{\beta}}) \cdot t_{n-p}(1-\alpha/2). \tag{2.58}$$

Simultaneous intervals. Note that the intervals in the preceding sections have *pointwise coverage*. For example, we have

$$\mathbb{P}[\beta_j \in \mathrm{CI}(\beta_j)] \geq 1-\alpha \quad \text{for each } j. \tag{2.59}$$

or

$$\mathbb{P}[\boldsymbol{x}_0^T\boldsymbol{\beta} \in \mathrm{CI}(\boldsymbol{x}_0^T\boldsymbol{\beta})] \geq 1-\alpha \quad \text{for each } \boldsymbol{x}_0. \tag{2.60}$$

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

$$\mathbb{P}[\beta_j \in \mathrm{CI}(\beta_j) \text{ for each } j] \geq 1-\alpha \tag{2.61}$$

or

$$\mathbb{P}[\boldsymbol{x}_0^T\boldsymbol{\beta} \in \mathrm{CI}(\boldsymbol{x}_0^T\boldsymbol{\beta}) \text{ for each } \boldsymbol{x}_0] \geq 1-\alpha. \tag{2.62}$$

Simultaneous confidence intervals are possible to construct as well. As a starting point, note that

$$\frac{\frac{1}{p}\|\boldsymbol{X}\widehat{\boldsymbol{\beta}}-\boldsymbol{X}\boldsymbol{\beta}\|^2}{\widehat{\sigma}^2} \sim F_{p,n-p}. \tag{2.63}$$

Hence, we have

$$\mathbb{P}[\|\boldsymbol{X}\widehat{\boldsymbol{\beta}}-\boldsymbol{X}\boldsymbol{\beta}\|^2 \leq p\widehat{\sigma}^2 F_{p,n-p}(1-\alpha)] \geq 1-\alpha. \tag{2.64}$$

Hence, the region

$$\mathrm{CR}(\boldsymbol{\beta}) \equiv \{\boldsymbol{\beta} : (\widehat{\boldsymbol{\beta}}-\boldsymbol{\beta})^T\boldsymbol{X}^T\boldsymbol{X}(\widehat{\boldsymbol{\beta}}-\boldsymbol{\beta}) \leq p\widehat{\sigma}^2 F_{p,n-p}(1-\alpha)\} \subseteq \mathbb{R}^p \tag{2.65}$$

is a $1-\alpha$ confidence region for the vector $\boldsymbol{\beta}$:

$$\mathbb{P}[\boldsymbol{\beta} \in \mathrm{CR}(\boldsymbol{\beta})] \geq 1-\alpha. \tag{2.66}$$

It's easy to see that $\mathrm{CR}(\boldsymbol{\beta})$ is an ellipse centered at $\widehat{\boldsymbol{\beta}}$. Since the confidence region is for the entire vector $\boldsymbol{\beta}$, we can define simultaneous confidence intervals for each coordinate as follows:

$$\mathrm{CI}(\beta_j) \equiv \{\beta_j : \boldsymbol{\beta} \in \mathrm{CR}(\boldsymbol{\beta})\}. \tag{2.67}$$

Then, these confidence intervals will satisfy the simultaneous coverage property (2.61). We will obtain a more explicit expression for $\mathrm{CI}(\beta_j)$ shortly.

Similarly, we may define the simultaneous confidence regions

$$\mathrm{CI}(\boldsymbol{x}_0^T\boldsymbol{\beta}) \equiv \{\boldsymbol{x}_0^T\boldsymbol{\beta} : \boldsymbol{\beta} \in \mathrm{CR}(\boldsymbol{\beta})\}. \tag{2.68}$$

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

$$\begin{aligned}(\boldsymbol{x}_0^T\widehat{\boldsymbol{\beta}}-\boldsymbol{x}_0^T\boldsymbol{\beta})^2 = \|\boldsymbol{x}_0^T(\widehat{\boldsymbol{\beta}}-\boldsymbol{\beta})\|^2 &= \|(\boldsymbol{\Sigma}^{-1/2}\boldsymbol{x}_0)^T\boldsymbol{\Sigma}^{1/2}(\widehat{\boldsymbol{\beta}}-\boldsymbol{\beta})\|^2 \\ &\leq \|(\boldsymbol{\Sigma}^{-1/2}\boldsymbol{x}_0)\|^2\|\boldsymbol{\Sigma}^{1/2}(\widehat{\boldsymbol{\beta}}-\boldsymbol{\beta})\|^2 \leq \boldsymbol{x}_0^T\boldsymbol{\Sigma}^{-1}\boldsymbol{x}_0 p\widehat{\sigma}^2 F_{p,n-p}(1-\alpha),\end{aligned} \tag{2.69}$$

i.e.

$$\boldsymbol{x}_0^T\boldsymbol{\beta} \in \boldsymbol{x}_0^T\widehat{\boldsymbol{\beta}} \pm \widehat{\sigma}\sqrt{\boldsymbol{x}_0^T(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{x}_0}\sqrt{pF_{p,n-p}(1-\alpha)} \equiv \boldsymbol{x}_0^T\widehat{\boldsymbol{\beta}} \pm \mathrm{SE}(\boldsymbol{x}_0^T\widehat{\boldsymbol{\beta}}) \cdot \sqrt{pF_{p,n-p}(1-\alpha)}. \tag{2.70}$$

Defining the above interval as $\mathrm{CI}(\boldsymbol{x}_0^T\boldsymbol{\beta})$ gives us the simultaneous coverage property (2.62). Comparing to equation (2.58), we see that the simultaneous interval is the pointwise interval expanded by a factor of $\sqrt{pF_{p,n-p}(1-\alpha)}/t_{n-p}(1-\alpha/2)$. Specializing to the case $\boldsymbol{x}_0 \equiv \boldsymbol{e}_j$, we get an expression for the simultaneous intervals for each coordinate:

$$\mathrm{CI}(\beta_j) \equiv \widehat{\beta}_j \pm \widehat{\sigma}\sqrt{(\boldsymbol{X}^T\boldsymbol{X})^{-1}_{jj}}\sqrt{pF_{p,n-p}(1-\alpha)} \equiv \mathrm{SE}(\widehat{\beta}_j)\sqrt{pF_{p,n-p}(1-\alpha)}, \tag{2.71}$$

which again is the pointwise interval (2.51) expanded by a factor of $\sqrt{pF_{p,n-p}(1-\alpha)}/t_{n-p}(1-\alpha/2)$. These simultaneous intervals are called *Working-Hotelling intervals*.

2.5 Practical considerations

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.

Correlation versus causation, and Simpson's paradox. We need to be very careful when interpreting linear regression coefficients, which can be sensitive to the choice of other predictors to include. You can get misleading conclusions if you omit important variables from the regression. A special case of this is *Simpson's paradox*, where an important discrete variable is omitted. Consider the example in Figure 2.2.

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

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."

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 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.

2.6 R demo

See also Agresti 3.4.1, 3.4.3

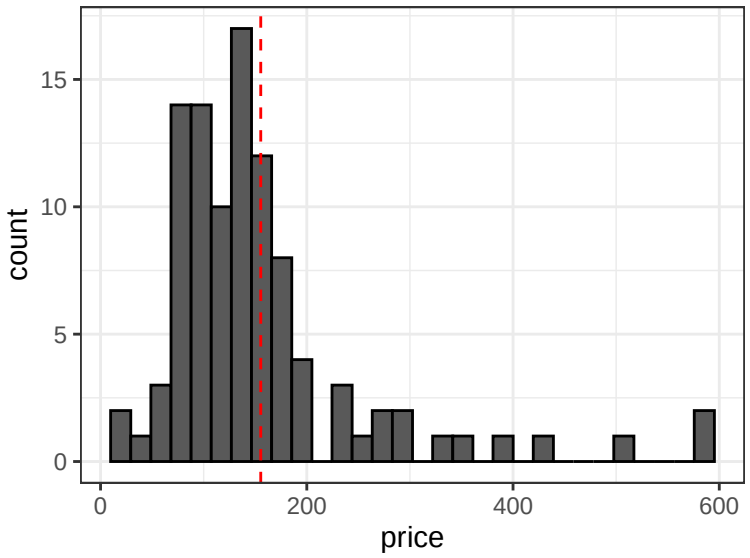
Let's put into practice what we've learned in this chapter by analyzing data about house prices.

```
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
## # ... with 90 more rows
```

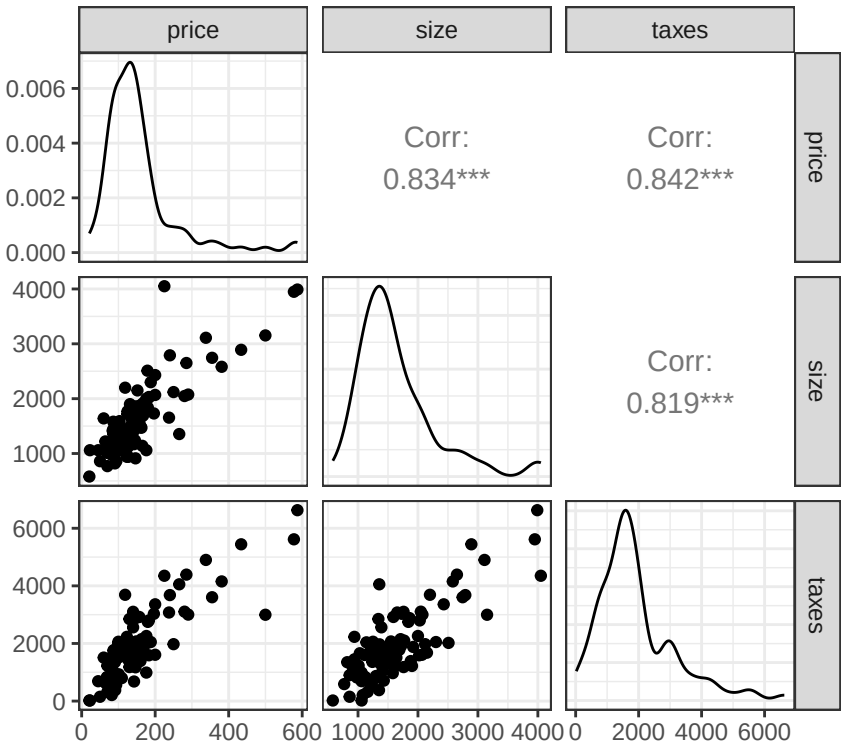
Exploration. Let's first do a bit of exploration:

```
# 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"
  )
```

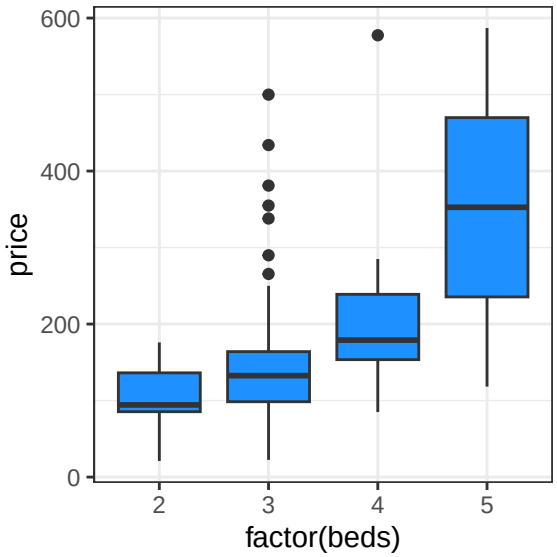



```
# compare median and mean price
houses_data %>%
  summarise(
    mean_price = mean(price),
    median_price = median(price)
  )

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

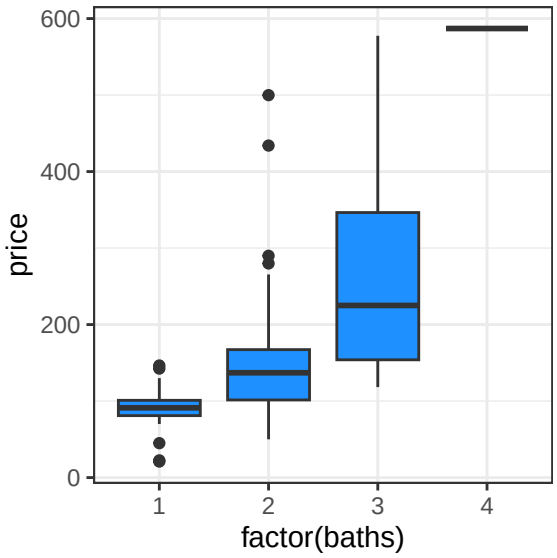
```
# create a pairs plot of continuous variables
houses_data %>%
  select(price, size, taxes) %>%
  ggpairs()
```



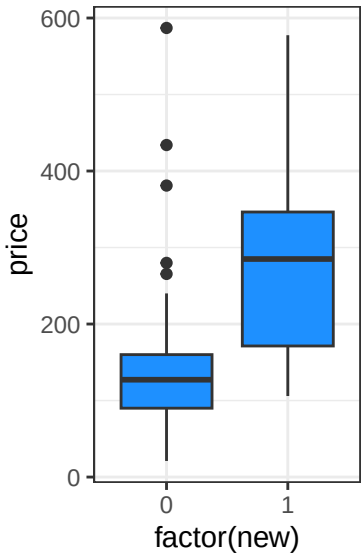

```
# see how price relates to beds
houses_data %>%
  ggplot(aes(x = factor(beds), y = price)) +
  geom_boxplot(fill = "dodgerblue")
```




```
# 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")
```



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.

```
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.

```
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 adjust 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

# 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.

```
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.

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 (2.71):

```
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
```

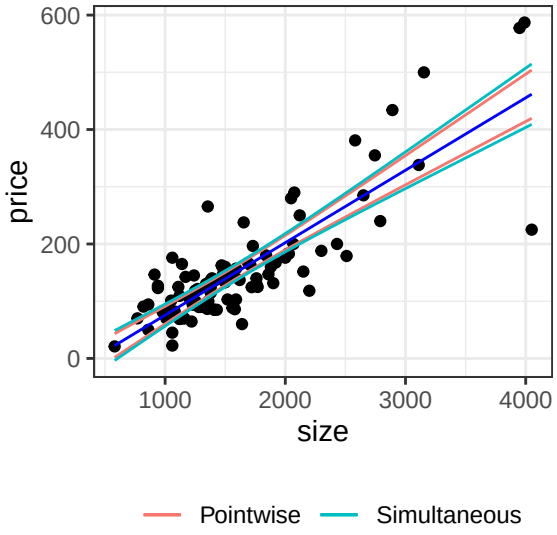
Now we can construct the simultaneous confidence intervals via the formula (2.70):

```
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.
## # ... with 90 more rows
```

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

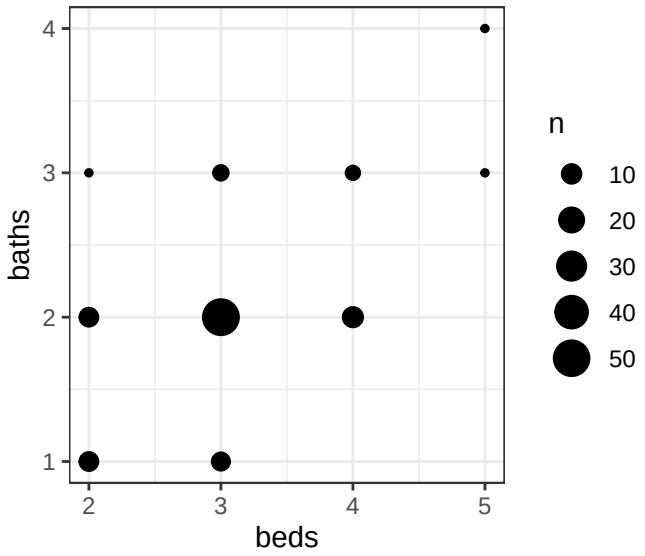
```
# to produce confidence intervals for fits in general, use the predict() function
n <- nrow(houses_data)
p <- 2
alpha <- 0.05
lm_fit <- lm(price ~ size, data = houses_data)
predictions <- predict(lm_fit, se.fit = TRUE)
t_quantile <- qt(1 - alpha / 2, df = n - p)
f_quantile <- qf(1 - alpha, df1 = 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")
```



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
```

```
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`:

```
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 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.

```
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.

Chapter 3

Linear models: Misspecification

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

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \quad \text{where } \boldsymbol{\epsilon} \sim N(\boldsymbol{0}, \sigma^2 \boldsymbol{I}_n). \tag{3.1}$$

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

- Non-normality (Section 3.1): $\boldsymbol{\epsilon} \sim (0, \sigma^2 \boldsymbol{I}_n)$ but not $N(0, \sigma^2 \boldsymbol{I}_n)$.
- Heteroskedastic and/or correlated errors (Section 3.2): $\boldsymbol{\epsilon} \sim (0, \boldsymbol{\Sigma})$, 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 3.3): It is not the case that $\mathbb{E}[\boldsymbol{y}] = \boldsymbol{X}\boldsymbol{\beta}$ for some $\boldsymbol{\beta} \in \mathbb{R}^p$.
- Outliers (Section 3.4): 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 (Sections 3.1-3.4). We conclude with an R demo (Section 3.5).

3.1 Non-normality

3.1.1 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.

3.1.2 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.

3.1.3 Detection

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

$$\widehat{\epsilon}_i = y_i - \boldsymbol{x}_{i*}^T \widehat{\boldsymbol{\beta}}. \tag{3.2}$$

Recall from Chapter 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

$$r_i = \frac{\widehat{\epsilon}_i}{\widehat{\sigma}\sqrt{1-h_i}}. \tag{3.3}$$

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 (see Figure 3.1).

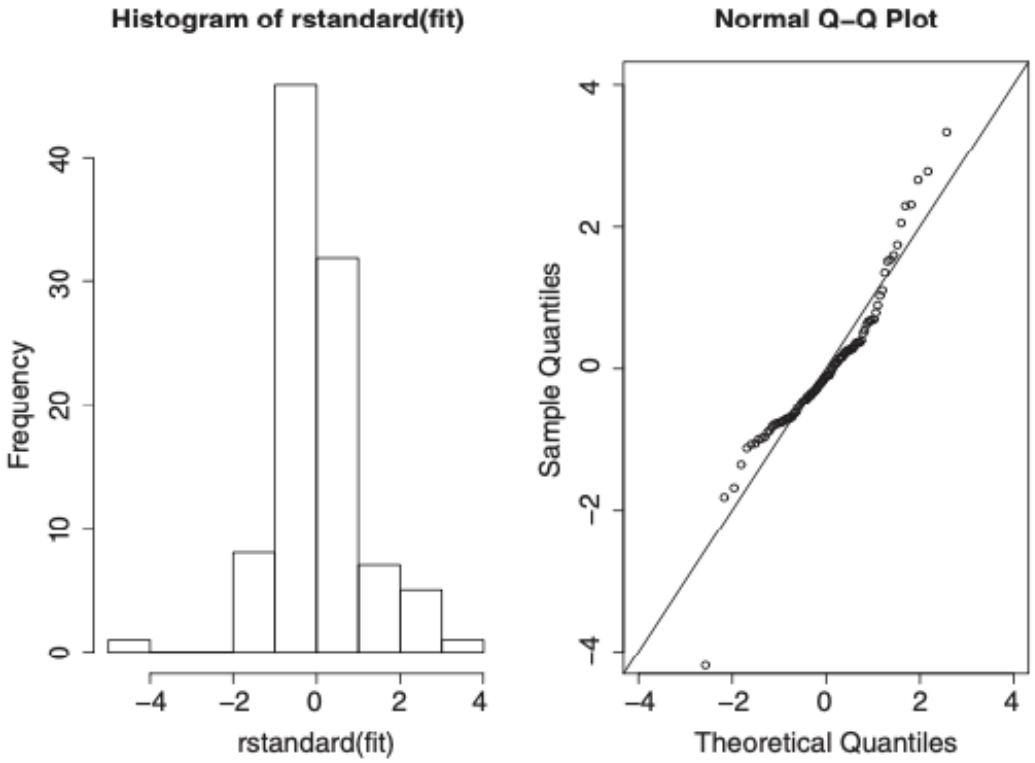


Figure 3.1: Histogram and normal QQ plot of standardized residuals.

3.1.4 Fixes

As mentioned in Section 3.1.2, 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.

Standard errors via the residual bootstrap. The *bootstrap* is one way of carrying out robust inference. 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 test statistic by simulating from this approximate data-generating distribution. This approach, pioneered by Brad Efron in 1979, replaces mathematical derivations with computation. The bootstrap is extremely flexible, and can be adapted to apply in a variety of settings.

Suppose that $y_i = \boldsymbol{x}_{i*}^T\boldsymbol{\beta} + \epsilon_i$, where $\epsilon_i \overset{\text{i.i.d.}}{\sim} F$ for some distribution F. Then, the data-generating distribution is specified by $(\boldsymbol{\beta}, F)$, which we approximate by substituting $\widehat{\boldsymbol{\beta}}$ for $\boldsymbol{\beta}$ and the empirical distribution of the residuals $\widehat{\epsilon}_i$ (call it $\widehat{F}$) for F. We can then sample new response vectors based on this approximate data-generating distribution:

$$y_i^b = \boldsymbol{x}_{i*}^T\widehat{\boldsymbol{\beta}} + \epsilon_i^b, \quad \epsilon_i^b \overset{\text{i.i.d.}}{\sim} \widehat{F} \quad \text{for } b = 1, \ldots, B. \tag{3.4}$$

Note that i.i.d. sampling ϵ_i^b from $\widehat{F}$ amounts to sampling $(\epsilon_1^b, \ldots, \epsilon_n^b)$ with replacement from $(\widehat{\epsilon}_1, \ldots, \widehat{\epsilon}_n)$. Then, as with the parametric bootstrap, we fit a least squares coefficient vector $\widehat{\boldsymbol{\beta}}^b$ to $(\boldsymbol{X}, \boldsymbol{y}^b)$ for each b and obtain standard errors by treating $\{\widehat{\boldsymbol{\beta}}^b\}_{b=1}^B$ as though it were the sampling distribution of $\widehat{\boldsymbol{\beta}}$.

Hypothesis testing via the residual bootstrap. While the bootstrap is commonly associated with the construction of standard errors, it can also be used directly for hypothesis testing. Suppose we wish to test the linear regression null hypothesis $H_0 : \boldsymbol{\beta}_S = \boldsymbol{0}$ for some $S \subseteq \{1, \ldots, p-1\}$ (which recall we cannot do using a permutation test). We compute some test statistic $T(\boldsymbol{X}, \boldsymbol{y})$ measuring the significance of $\boldsymbol{\beta}_S$ (e.g. an F-statistic but it could be anything else). Then, we can use a variant of the residual bootstrap. We fit the least squares estimate $\widehat{\boldsymbol{\beta}}$ as usual and extract the residuals $\widehat{\epsilon}_i \equiv y_i - \boldsymbol{x}_{i*}^T\widehat{\boldsymbol{\beta}}$ and their empirical distribution $\widehat{F}$. Then, placing ourselves under the null hypothesis, we generate new samples $\boldsymbol{y}^b$ from the null distribution analogously to the usual residual bootstrap (3.4):

$$y_i^b = \boldsymbol{x}_{i,\text{-}S}^T\widehat{\boldsymbol{\beta}}_{\text{-}S} + \epsilon_i^b, \quad \epsilon_i^b \overset{\text{i.i.d.}}{\sim} \widehat{F} \quad \text{for } b = 1, \ldots, B. \tag{3.5}$$

We can then build a null distribution by recomputing $T(\boldsymbol{X}, \boldsymbol{y}^b)$ for each b and then define the bootstrap-based p-value

$$p^{\text{boot}} \equiv \frac{1}{B+1}\left(1 + \sum_{b=1}^{B} \mathbb{1}(T(\boldsymbol{X}, \boldsymbol{y}^b) \geq T(\boldsymbol{X}, \boldsymbol{y}))\right). \tag{3.6}$$

3.2 Heteroskedastic and correlated errors

3.2.1 Origin

Origins of heteroskedasticity. 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 3.2), whereas for a linear model the variance of y stays constant as the mean of y varies.

Origins of correlated errors. 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×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 sample that are in the same group or spatially/temporally close to one another.

3.2.2 Consequences

All normal linear model inference from Chapter 2 hinges on the assumption that $\boldsymbol{\epsilon} \sim N(\mathbf{0}, \sigma^2 \boldsymbol{I})$. If instead of $\sigma^2 I$ 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 3.2 for intuition.

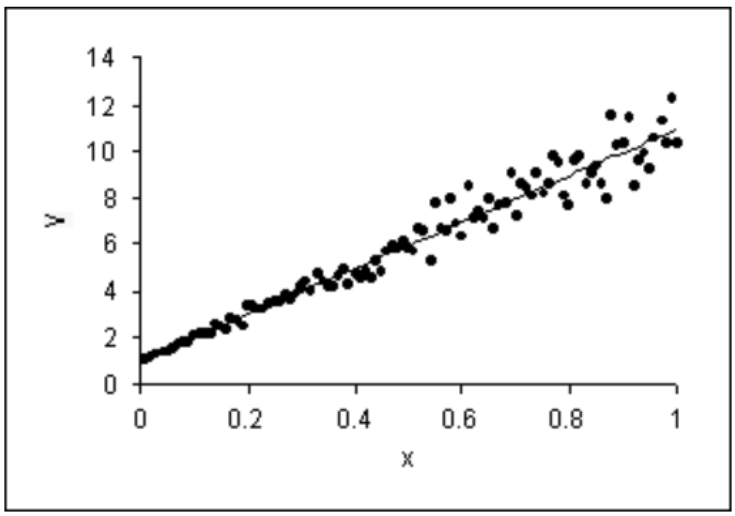


Figure 3.2: Heteroskedasticity in a simple bivariate linear model (image source).

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.

3.2.3 Detection

Heteroskedasticity is usually assessed via the *residual plot* (Figure 3.3). In this plot, the standardized residuals r_i (3.3) 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 3.3(a)). A common sign of heteroskedasticity is the fan shape where variance increases as a function of $\widehat{\mu}$ (Figure 3.3(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

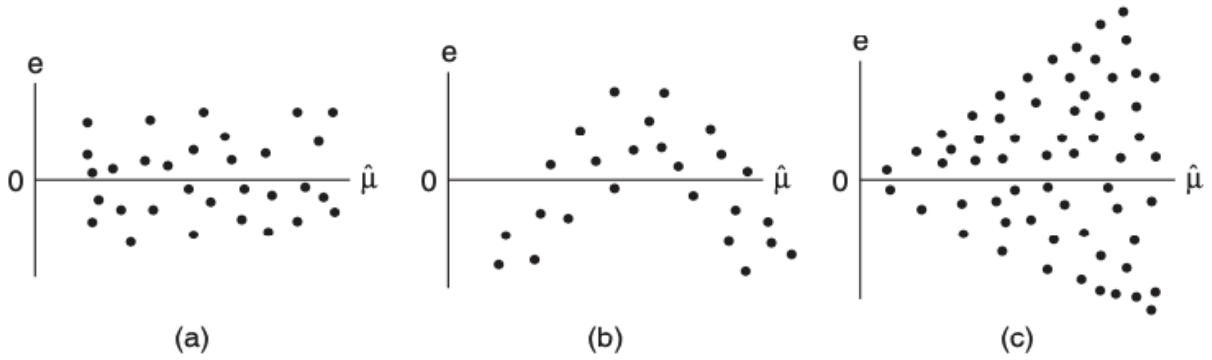


Figure 3.3: Residuals plotted against linear-model fitted values that reflect (a) model adequacy, (b) quadratic rather than linear relationship, and (c) nonconstant variance(image source: Agresti Figure 2.8).

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.

3.2.4 Fixes

Broadly speaking, approaches to fixing heteroskedastic or correlated errors can be divided into (1) those based on estimating $\boldsymbol{\Sigma}$ and (2) those based on resampling. Methods based on estimating $\boldsymbol{\Sigma}$ can use this estimate to either (i) build a better estimate $\widehat{\boldsymbol{\beta}}$ or (ii) build better standard errors for the least squares estimate. Resampling methods include the bootstrap (for estimation) and the permutation test (for testing).

3.2.4.1 Methods based on estimating $\boldsymbol{\Sigma}$

Methods that build a better estimate of $\widehat{\boldsymbol{\beta}}$. Suppose $\boldsymbol{y} \sim N(\boldsymbol{X}\boldsymbol{\beta}, \Sigma)$. This is a *generalized least squares* problem for which inference can be carried out. The generalized least squares estimate is $\widehat{\boldsymbol{\beta}} = (\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{y}$, which is distributed as $\widehat{\boldsymbol{\beta}} \sim N(\boldsymbol{\beta}, (\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X})^{-1})$. This is the best linear unbiased estimate of $\boldsymbol{\beta}$, recovering efficiency. We can carry out inference based on the latter distributional result analogously to how we did so in Chapter 2. The issue, of course, is that we usually do not know $\boldsymbol{\Sigma}$. Therefore, we can consider the following approach: (1) estimate $\widehat{\boldsymbol{\beta}}$ using OLS, (2) use this estimate to get an estimate $\widehat{\boldsymbol{\Sigma}}$ of $\boldsymbol{\Sigma}$, (3) use $\widehat{\boldsymbol{\Sigma}}$ to get a (hopefully) more efficient estimator

$$\widehat{\boldsymbol{\beta}}^{\text{FGLS}} \equiv (\boldsymbol{X}^T\widehat{\boldsymbol{\Sigma}}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^T\widehat{\boldsymbol{\Sigma}}^{-1}\boldsymbol{y}. \tag{3.7}$$

This is called the *feasible generalized least squares estimate* (FGLS), to contrast it with the infeasible estimate that assumes $\boldsymbol{\Sigma}$ is known exactly. The procedure above can be iterated until convergence. To estimate $\widehat{\boldsymbol{\Sigma}}$, we usually need to make some parametric assumptions. For example, in the case of grouped structure, we might assume a *random effects model*. In the case of a temporal structure, we might assume an *AR(1) model*.

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.

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

$$\text{Var}[\widehat{\boldsymbol{\beta}}] = (\boldsymbol{X}^T\boldsymbol{X})^{-1}(\boldsymbol{X}^T\boldsymbol{\Sigma}\boldsymbol{X})(\boldsymbol{X}^T\boldsymbol{X})^{-1}. \tag{3.8}$$

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 between 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

$$\widehat{\text{Var}}[\widehat{\boldsymbol{\beta}}] \equiv (\boldsymbol{X}^T\boldsymbol{X})^{-1}(\boldsymbol{X}^T\widehat{\boldsymbol{\Sigma}}\boldsymbol{X})(\boldsymbol{X}^T\boldsymbol{X})^{-1}. \tag{3.9}$$

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).

Huber-White standard errors. Now, suppose $\boldsymbol{\Sigma} = \text{diag}(\sigma_1^2, \dots, \sigma_n^2)$ for some variances $\sigma_1^2, \dots, \sigma_n^2 > 0$. The Huber-White sandwich estimator is defined by (3.8), with

$$\widehat{\boldsymbol{\Sigma}} \equiv \text{diag}(\widehat{\sigma}_1^2, \dots, \widehat{\sigma}_n^2), \quad \text{where} \quad \widehat{\sigma}_i^2 = (y_i - \boldsymbol{x}_{i*}^T\widehat{\boldsymbol{\beta}})^2. \tag{3.10}$$

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.

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 $c(i) \in \{1, \dots, C\}$. 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 $\widehat{\boldsymbol{\epsilon}} = (\widehat{\boldsymbol{\epsilon}}_1, \dots, \widehat{\boldsymbol{\epsilon}}_C)$. Then, the true covariance matrix is $\boldsymbol{\Sigma} = \text{block-diag}(\boldsymbol{\Sigma}_1, \dots, \boldsymbol{\Sigma}_C)$ for some positive definite $\boldsymbol{\Sigma}_1, \dots, \boldsymbol{\Sigma}_C$. The Liang-Zeger estimator is then defined by (3.8), with

$$\widehat{\boldsymbol{\Sigma}} \equiv \text{block-diag}(\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{1}}, \dots, \widehat{\boldsymbol{\Sigma}}_{\boldsymbol{C}}), \quad \text{where} \quad \widehat{\boldsymbol{\Sigma}}_{\boldsymbol{c}} \equiv \widehat{\boldsymbol{\epsilon}}_c\widehat{\boldsymbol{\epsilon}}_c^T. \tag{3.11}$$

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 sometimes referred to as "clustered standard errors."

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 $|i1 - i2| \le L$. Then, a natural extension of the Huber-White estimate of $\boldsymbol{\Sigma}$ is $\widehat{\boldsymbol{\Sigma}}_{i1,i2} = \widehat{\epsilon}_{i1}\widehat{\epsilon}_{i2}$ for each pair $(i1, i2)$ such that $|i1 - i2| \le L$. Unfortunately, this is not guaranteed to give a positive semidefinite matrix $\widehat{\boldsymbol{\Sigma}}$. Therefore, Newey and West proposed a slightly modified estimator:

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

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 $\widehat{\boldsymbol{\Sigma}}$.

3.2.4.2 Resampling methods

Pairs bootstrap. The residual bootstrap corrects for non-normality, but not heteroskedasticity or correlated errors, since it assumes that the noise terms are i.i.d. from some distribution.

Weakening the assumptions further, let's assume only that $(\boldsymbol{x}_{i*}, y_i) \overset{\text{i.i.d.}}{\sim} F$ for some joint distribution F. We then resample our observations by sampling with replacement from the original observations.

Note that, unlike the parametric or residual bootstrap, the pairs bootstrap treats the predictors $\boldsymbol{X}$ as random rather than fixed. 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.

The pairs bootstrap has several variants that help it overcome correlated errors, in addition to heteroskedasticity. The *cluster bootstrap* is applicable in the case when errors have a clustered/grouped structure. In this case, we sample entire clusters of observations, with replacement, from the original set of clusters. The *moving blocks bootstrap* is applicable in the case of spatially or temporally structured errors. In this variant of the pairs bootstrap, we resample spatially or temporally adjacent blocks of observations together to preserve their joint correlation structure.

Permutation tests. Unlike the residual bootstrap, the pairs bootstrap cannot accommodate hypothesis testing. If we would like resampling-based hypothesis tests in the presence of heteroskedasticity, we can consider permutation tests instead. Permutation tests are an easy way of testing the null hypothesis of independence between two random variables (or vectors). For our purposes, suppose that $(\boldsymbol{x}_{i*}, y_i)$ are drawn i.i.d. from some joint distribution F (as opposed to the usual assumption that $\boldsymbol{X}$ is fixed). Then, consider the null hypothesis

$$H_0 : \boldsymbol{x} \perp\!\!\!\perp y. \tag{3.12}$$

This null hypothesis is related to the null hypothesis $H_0 : \boldsymbol{\beta}_{\text{-}0} = 0$ in a linear regression, as formalized by the following lemma.

Lemma 3.2.1. *Suppose $\boldsymbol{x} \in \mathbb{R}^{p-1}$ has a nondegenerate distribution $F_{\boldsymbol{x}}$ in the sense that there does not exist a vector $c \in \mathbb{R}^{p-1}$ such that $\boldsymbol{c}^T\boldsymbol{x}$ is deterministic. Suppose also that $F_{y|\boldsymbol{x}}$ is a distribution such that $\mathbb{E}[y|\boldsymbol{x}] = \beta_0 + \boldsymbol{x}^T\boldsymbol{\beta}_{\text{-}0}$ and that the distribution $F_{y|\boldsymbol{x}}$ is specified by its mean. Then,*

$$\boldsymbol{x} \perp\!\!\!\perp y \quad \iff \quad \boldsymbol{\beta}_{\text{-}0} = \boldsymbol{0}. \tag{3.13}$$

Proof. If $\boldsymbol{\beta}_{\text{-}0} = \boldsymbol{0}$, then $\mathbb{E}[y|\boldsymbol{x}] = \beta_0$. Therefore, the mean of y does not depend on $\boldsymbol{x}$. By the assumption on $F_{y|\boldsymbol{x}}$, it follows that the entire distribution $F_{y|\boldsymbol{x}}$ does not depend on $\boldsymbol{x}$, i.e. $y \perp\!\!\!\perp \boldsymbol{x}$. If $\boldsymbol{\beta}_{\text{-}0} \neq \boldsymbol{0}$, then $\mathbb{E}[y|\boldsymbol{x}] = \beta_0 + \boldsymbol{x}^T\boldsymbol{\beta}_{\text{-}0}$, which by assumption is non-constant. Since $\mathbb{E}[y|\boldsymbol{x}]$ depends on $\boldsymbol{x}$, it follows that y is not independent of $\boldsymbol{x}$. □

Therefore, any valid independence test automatically gives a non-normality-robust and heteroskedasticity-robust test of $H_0 : \boldsymbol{\beta}_{\text{-}0} = \boldsymbol{0}$ in a linear regression.

Now, suppose we have n i.i.d. samples $(\boldsymbol{x}_{i*}, y_i)$ from F. Under the independence null hypothesis (3.12), the distribution of the data is unchanged if we permute the response variables y_i. Formally,

let $\boldsymbol{y}_{()}$ be the order statistics of the response variable, let S_n be the permutation group on $\{1, \dots, n\}$, and let $\boldsymbol{y}_\tau$ denote the permutation of $\boldsymbol{y}$ by $\tau \in S_n$. Then,

$$\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{y}_{()} \sim \frac{1}{n!} \sum_{\tau \in S_n} \delta(\boldsymbol{y}_\tau). \tag{3.14}$$

Now, let $T(\boldsymbol{X}, \boldsymbol{y})$ be any test statistic measuring the association between $\boldsymbol{y}$ and $\boldsymbol{X}$, e.g. a linear regression F-statistic. Then, the above distributional result implies that

$$T(\boldsymbol{X}, \boldsymbol{y})|\boldsymbol{X}, \boldsymbol{y}_{()} \sim \frac{1}{n!} \sum_{\tau \in S_n} \delta(T(\boldsymbol{X}, \boldsymbol{y}_\tau)). \tag{3.15}$$

Hence, we can compute the null distribution of T by repeatedly permuting the response $\boldsymbol{y}$ and recomputing $T(\boldsymbol{X}, \boldsymbol{y}_\tau)$. This gives rise to the permutation p-value

$$p^{\text{perm}} \equiv \frac{1}{n!} \sum_{\tau \in S_n} \mathbb{1}(T(\boldsymbol{X}, \boldsymbol{y}_\tau) \geq T(\boldsymbol{X}, \boldsymbol{y})). \tag{3.16}$$

The uniform distribution of $T(\boldsymbol{X}, \boldsymbol{y})|\boldsymbol{X}, \boldsymbol{y}_{()}$ implies that

$$\mathbb{P}[p^{\text{perm}} \leq t|\boldsymbol{X}, \boldsymbol{y}_{()}] \leq t \quad \implies \quad \mathbb{P}[p^{\text{perm}} \leq t] = \mathbb{E}[\mathbb{P}[p^{\text{perm}} \leq t|\boldsymbol{X}, \boldsymbol{y}_{()}]] \leq t \quad \text{for all } t \in [0, 1].. \tag{3.17}$$

In practice, p^{perm} is approximated by independently sampling B permutations $\tau_1, \dots, \tau_B$ from the uniform distribution over S_n. Letting τ_0 be the identity permutation, it follows that

$$\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{y} \in \{\boldsymbol{y}_{\tau_0}, \dots, \boldsymbol{y}_{\tau_B}\} \sim \frac{1}{B+1} \sum_{b=0}^{B} \delta(\boldsymbol{y}_{\tau_b}). \tag{3.18}$$

Similar logic as above leads to the approximate permutation p-value

$$\widehat{p}^{\text{perm}} \equiv \frac{1}{B+1} \sum_{b=0}^{B} \mathbb{1}(T(\boldsymbol{X}, \boldsymbol{y}_{\tau_b}) \geq T(\boldsymbol{X}, \boldsymbol{y})) = \frac{1}{B+1} \left(1 + \sum_{b=1}^{B} \mathbb{1}(T(\boldsymbol{X}, \boldsymbol{y}_{\tau_b}) \geq T(\boldsymbol{X}, \boldsymbol{y}))\right). \tag{3.19}$$

Although $\widehat{p}^{\text{perm}}$ can be viewed as an approximation to $\boldsymbol{p}^{\text{perm}}$, it is also stochastically larger than the uniform distribution in finite samples:

$$\mathbb{P}[\widehat{p}^{\text{perm}} \leq t] \leq t \quad \text{for all } t \in [0, 1]. \tag{3.20}$$

Warning: A common mistake is to omit the "1+" in the numerator and denominator of the definition (3.19). The resulting p-value is *not valid* in the sense of equation (3.20).

Example. A common application of the permutation test is testing for equality of distributions in the two-sample problem, where the permutation test amounts to generating a null distribution for any test statistic (e.g. a difference in means) by pooling together the two samples and randomly reassigning the classes of the samples.

Strengths and weaknesses. The strength of the permutation test is that it is valid under almost no assumptions on the data-generating process. Its main weakness is that it is not applicable to the hypothesis $H_0 : \beta_S = 0$ for any group of predictors $S \neq \{1, \dots, p-1\}$. Intuitively, this would require a fancy kind of permutation that breaks the association between $\boldsymbol{y}$ and $\boldsymbol{X}_{*,S}$ while preserving the association between $\boldsymbol{X}_{*,S}$ and $\boldsymbol{X}_{*,\text{-}S}$. This amounts to a test of *conditional* independence, which requires more assumptions on the joint distribution $F_{\boldsymbol{x},y}$ than an independence test. Another weakness of a permutation test is that it is computationally expensive, although in the 21st century this is not a huge issue.

3.3 Model bias

3.3.1 Origin

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

$$\text{assumed model: } \boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}; \quad \text{actual model: } \boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{Z}\boldsymbol{\gamma} + \boldsymbol{\epsilon}. \tag{3.21}$$

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:

$$\text{assumed model: } \mathbb{E}[\boldsymbol{y}] = \boldsymbol{X}\boldsymbol{\beta}; \quad \text{actual model: } g(\mathbb{E}[\boldsymbol{y}]) = \boldsymbol{X}\boldsymbol{\beta}. \tag{3.22}$$

3.3.2 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 (3.21) 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 Chapter 2, this can lead to misleading conclusions.

3.3.3 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 3.3(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:

$$H_0 : y_i = \boldsymbol{x}_{i*}^T\boldsymbol{\beta} + \epsilon_i \quad \text{versus} \quad H_1 : y_i = \beta_{c(i)} + \epsilon_i. \tag{3.23}$$

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*.

3.3.4 Fixes

To fix model bias in the case (3.21), 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 (3.22), 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 Chapter 4.

3.4 Outliers

3.4.1 Origin

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

3.4.2 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.

3.4.3 Detection

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

$$\text{leverage of observation } i \equiv \text{corr}^2(y_i, \widehat{\mu}_i)^2. \tag{3.24}$$

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

$$\text{leverage of observation } i = h_{ii}, \tag{3.25}$$

which is the ith diagonal element of the hat matrix $\boldsymbol{H}$. This is related to the fact that $\text{Var}[\widehat{\epsilon}_i] = \sigma^2(1 - h_{ii})$. 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 $y_i = \beta_0 + \beta_1 x_i + \epsilon_i$,

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

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*

$$D_i = \frac{\sum_{i'=1}^{n}(\widehat{\mu}_{i'} - \widehat{\mu}_{i'}^{-i})^2}{p\widehat{\sigma}^2}, \tag{3.26}$$

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 Cooks distance greater than one.

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

$$D_i = \left(\frac{y_i - \widehat{\mu}_i}{\widehat{\sigma}\sqrt{1 - h_{ii}}}\right)^2 \cdot \frac{h_{ii}}{p(1 - h_{ii})}. \tag{3.27}$$

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.

3.4.4 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 Section 3.4.4.1.

3.4.4.1 Robust estimation

The squared error loss $\sum_{i=1}^n (y_i - \boldsymbol{x}_{i*}^T\boldsymbol{\beta})^2$ is sensitive to outliers in the sense that a large value of $y_i - \boldsymbol{x}_{i*}^T\boldsymbol{\beta}$ can have a significant impact on the loss function. The least squares estimate, as the minimizer of this loss function, is therefore sensitive to outliers. One way of addressing this challenge is to replace the squared error loss by 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:

$$L_\delta(y_i - \boldsymbol{x}_{i*}^T\boldsymbol{\beta}) = \begin{cases} \frac{1}{2}(y_i - \boldsymbol{x}_{i*}^T\boldsymbol{\beta})^2, & \text{if } |y_i - \boldsymbol{x}_{i*}^T\boldsymbol{\beta}| \le \delta; \\ \delta(|y_i - \boldsymbol{x}_{i*}^T\boldsymbol{\beta}| - \delta), & \text{if } |y_i - \boldsymbol{x}_{i*}^T\boldsymbol{\beta}| > \delta. \end{cases} \tag{3.28}$$

This function is differentiable, like the squared error loss, but grows linearly as opposed to quadratically. We can then define

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

This is an *M-estimator*; it is consistent and has an asymptotic normal distribution that can be used for inference.

3.5 R demo

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

3.5.1 Heteroskedasticity

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

```
library(tidyverse)
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
## # ... with 990 more rows
```

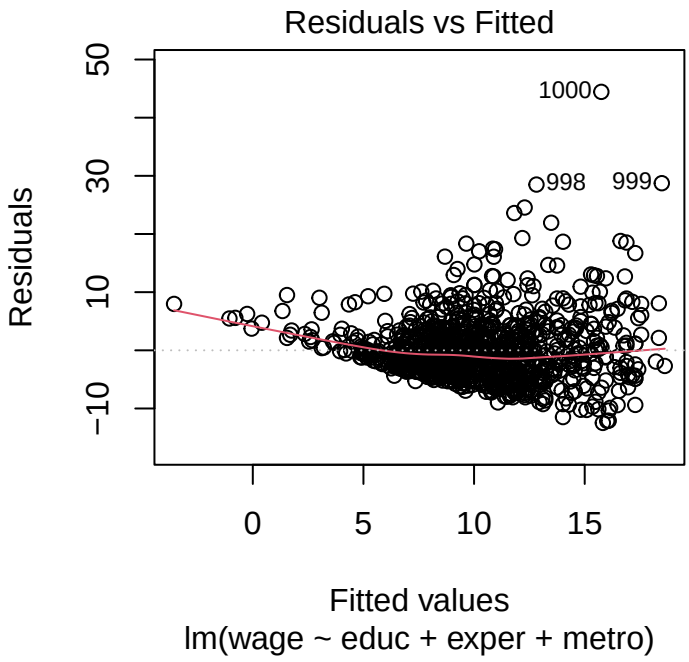
Suppose we want to regress `wage` on `educ`, `exper`, and `metro`.

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

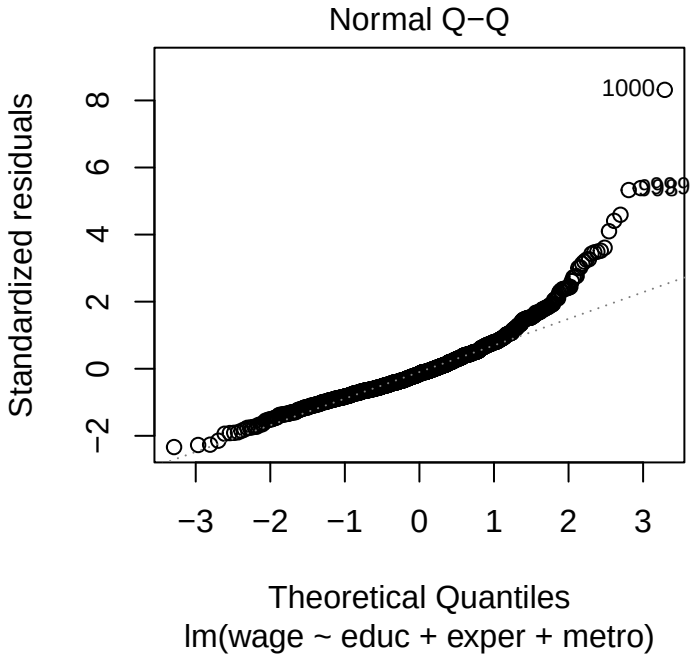
3.5.1.1 Diagnostics

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

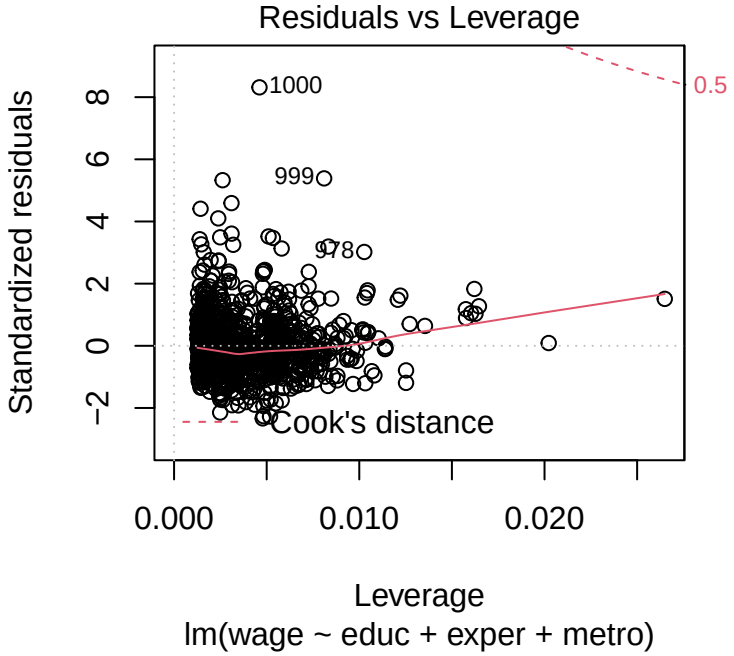
```
# residuals versus fitted
plot(lm_fit, which = 1)
```




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




```r
# 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.

3.5.1.2 Sandwich standard errors

To get standard errors robust to this heteroskedasticity, we can use one of the robust estimators discussed in Section 3.2. 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 obtain 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),
  boostrap_variance = diag(bootstrap_cov)
)

## # A tibble: 4 x 4
##   variable    usual_variance HW_variance boostrap_variance
##   <chr>                <dbl>       <dbl>             <dbl>
## 1 (Intercept)       1.16        1.48              1.34
## 2 educ              0.00489     0.00705           0.00596
## 3 exper             0.000232    0.000252          0.000245
## 4 metro             0.186       0.120             0.124
```

Note that the bootstrap standard errors are closer to the HW ones than the standard ones.

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

# 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

# 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

# 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
```

3.5.1.3 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.

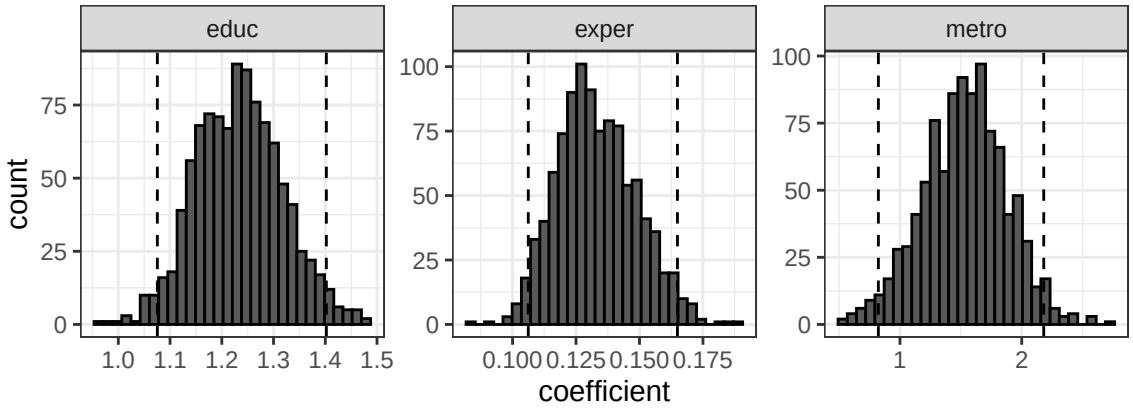
```
library(simpleboot)
boot_out <- lm.boot(
  lm.object = lm_fit, # input the fit object from lm()
  R = 1000
) # R is the number of bootstrap replicates
perc(boot_out) # get the percentile 95% confidence intervals

##       (Intercept)     educ     exper     metro
## 2.5%   -12.402014 1.075425 0.1061925 0.8229787
## 97.5%   -7.623951 1.402054 0.1649488 2.1808994

# Note: lm.boot implements the residual bootstrap as well. For this option, set rows = FALSE.
```

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) -7.8607649 -10.8830664 -9.983998 -9.2859532 -9.839665
## educ         1.0730422   1.2926975  1.208323  1.1587949  1.233057
## exper        0.1227519   0.1412297  0.147260  0.1244842  0.129010
## metro        1.3091946   1.5626078  1.653429  1.8491848  1.576849
```

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.

3.5.2 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 <- filter(nlswork_orig, idcode <= 100) %>%
  select(idcode, year, ln_wage, age, tenure, union) %>%
  filter(complete.cases(.)) %>%
  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
## # ... with 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 an 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`:

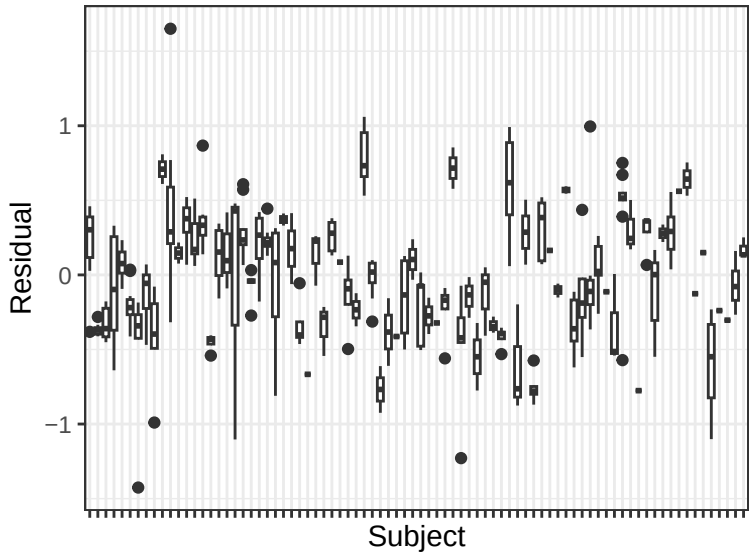
```
lm_fit <- lm(ln_wage ~ age + tenure + union + tenure:union,
  data = nlswork
)
summary(lm_fit)
```

```
##
## 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
```

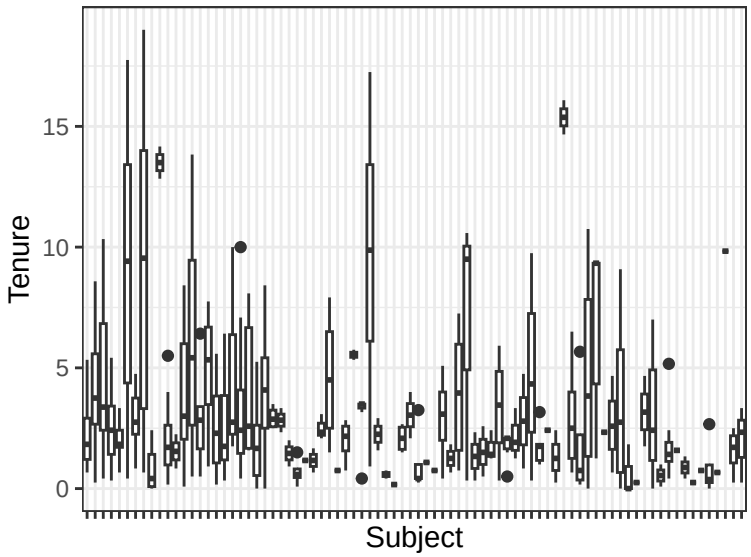
Let's plot the residuals against the individuals:

```
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.

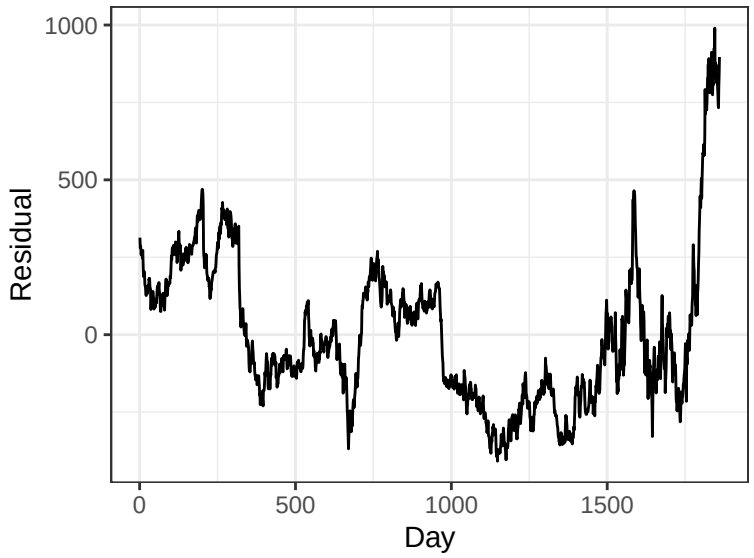
3.5.3 Autocorrelated errors

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
```

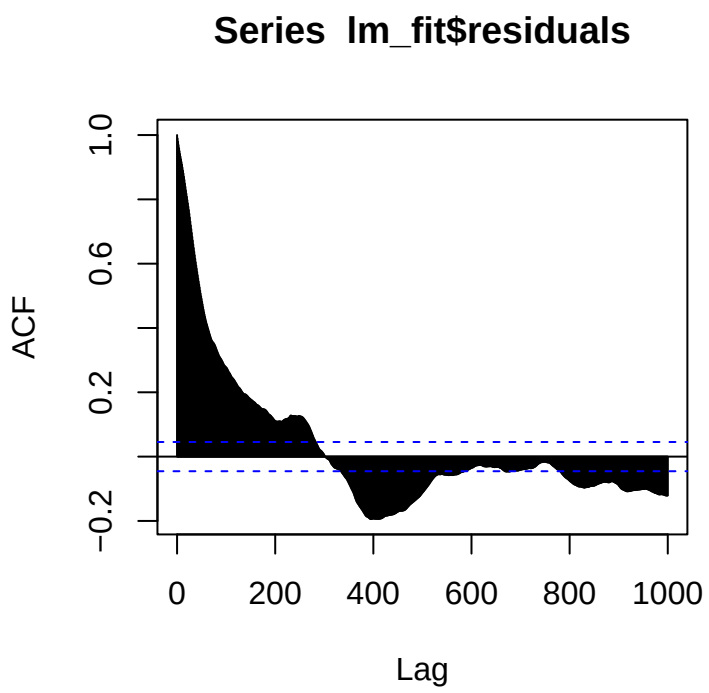
We find an extremely significant association between the two stock indices. But let's examine the residuals for autocorrelation:

```
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.

3.5.4 Outliers

Let's take a look at the crime data from HW2:

```
# 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) %>%
```

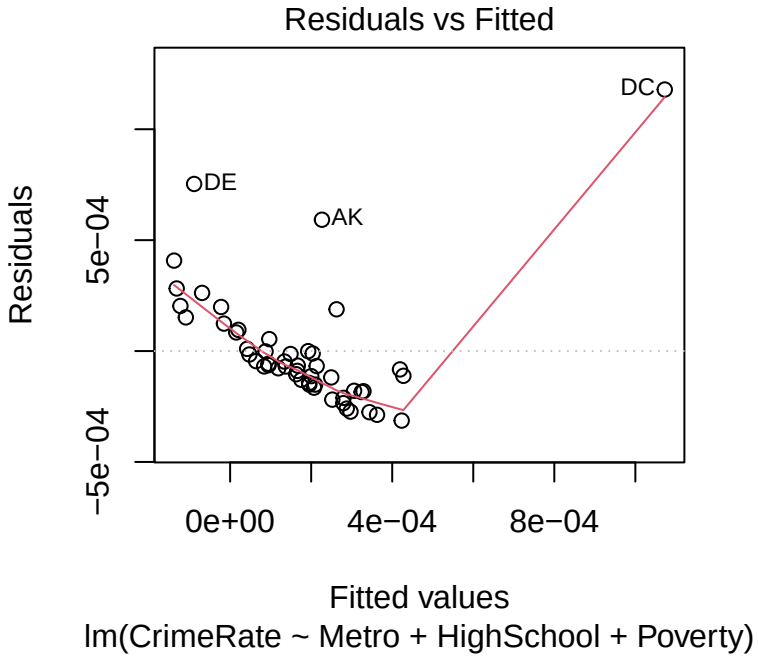
```
  rename(state_name = State, state_pop = Pop)

# 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) %>%
  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
## # ... with 41 more rows
```

Let's fit the linear regression:

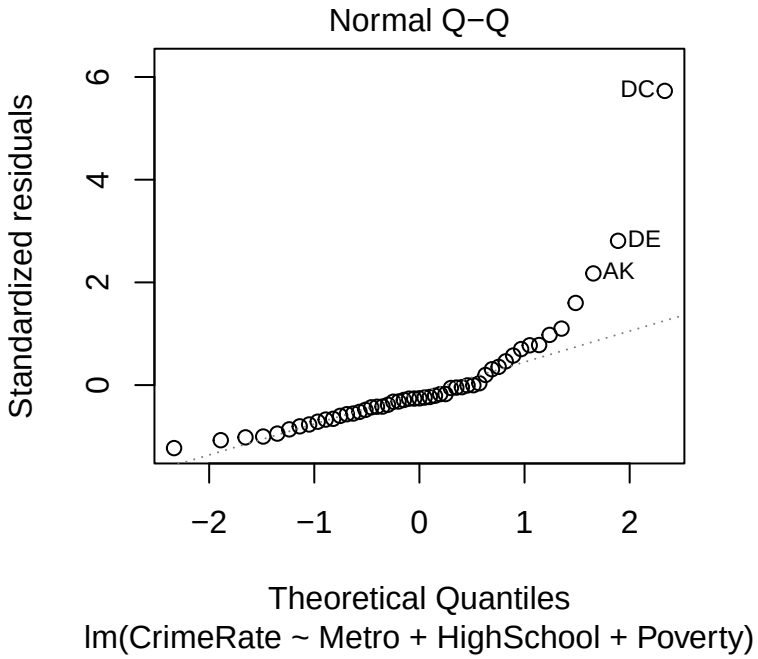
```
# note: we make the state abbreviations row names for better diagnostic plots
lm_fit <- lm(CrimeRate ~ Metro + HighSchool + Poverty,
  data = crime_data %>% column_to_rownames(var = "state_abbrev")
)
```

We can get the standard linear regression diagnostic plots as follows:

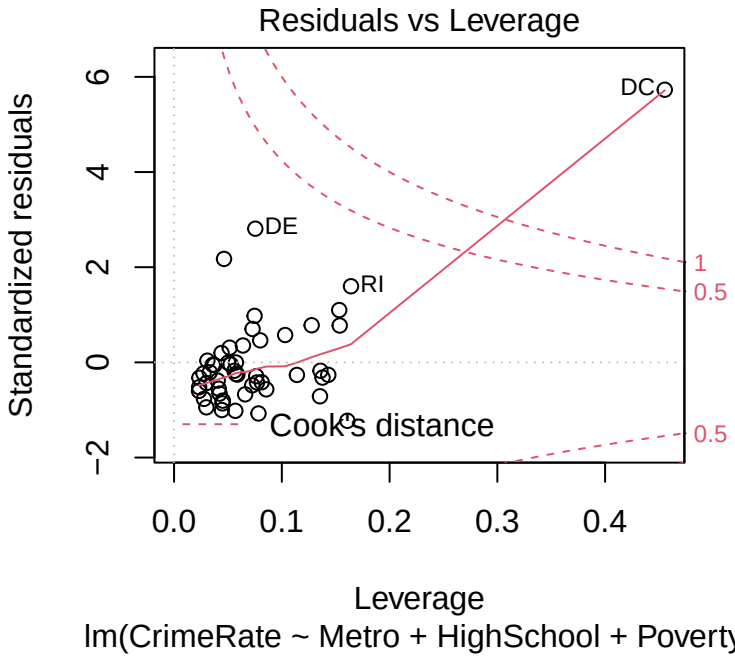
```
# residuals versus fitted
plot(lm_fit, which = 1)
```




```r
# residual QQ plotxw
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
## # ... with 41 more rows
```

Clearly DC is an outlier. We can either run a robust estimation procedure or we can 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:

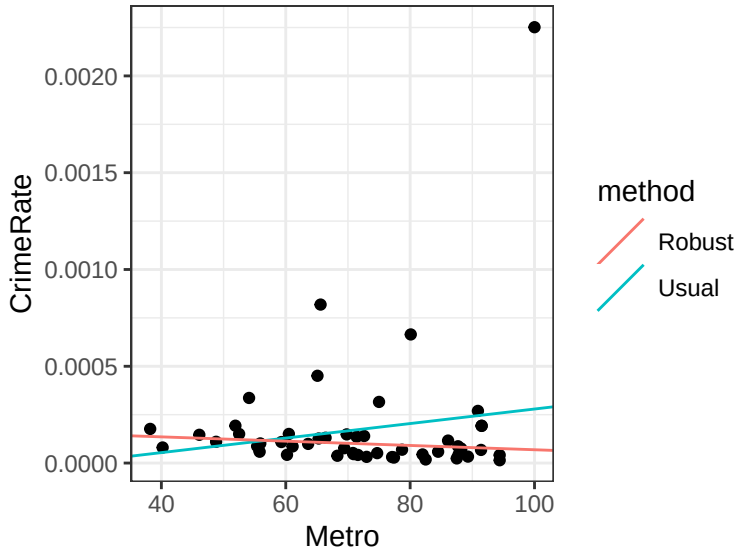
```
summary(rlm_fit)$coef %>%
  as.data.frame() %>%
  rename(Estimate = Value) %>%
  mutate(`p value` = 2 * dnorm(-abs(`t value`)))

##                  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"],
```

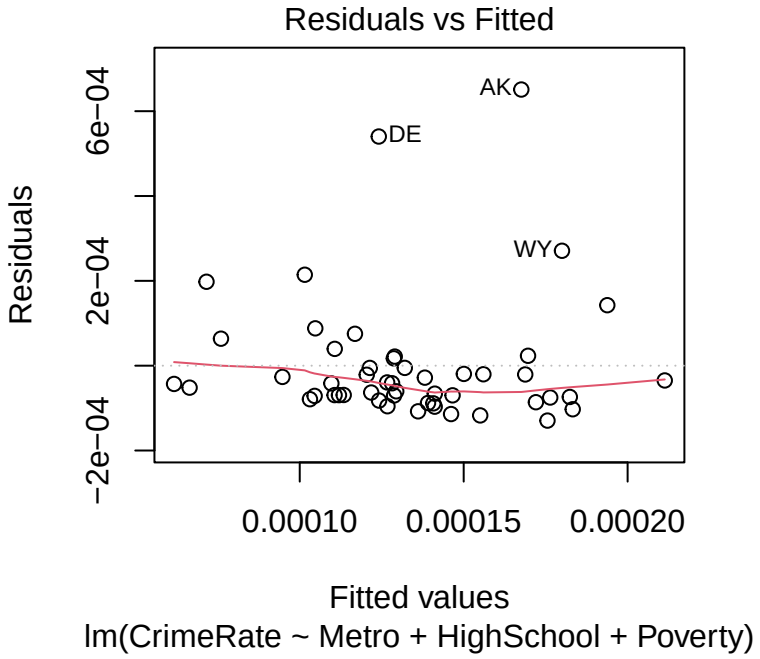
```
    coef(rlm_fit)["Metro"]
  )
)
```

```
# 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
  )
```

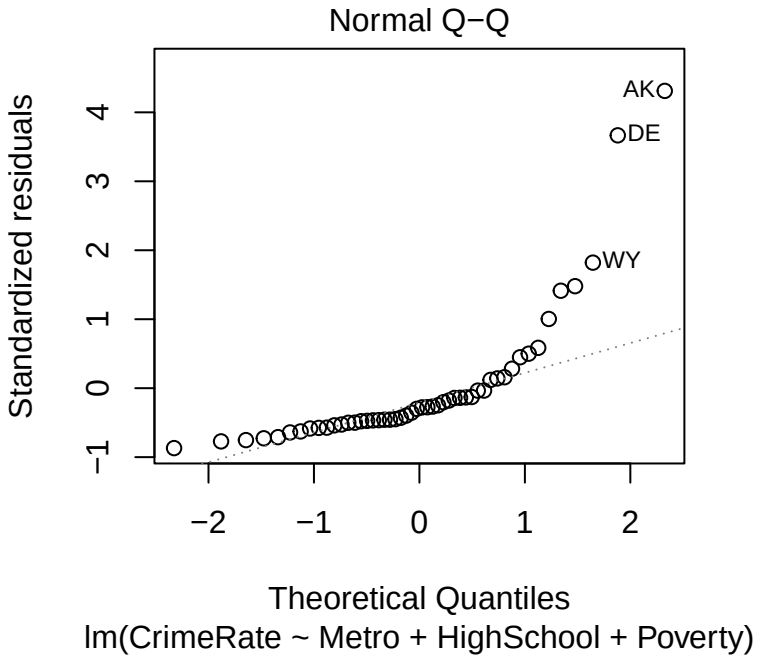


Next, let's try removing DC and running a usual linear regression.

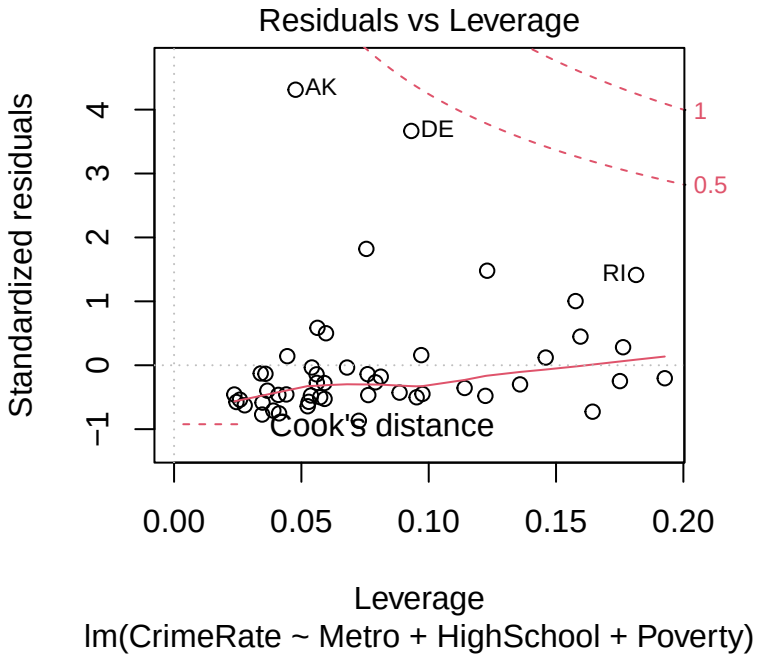
```
lm_fit_no_dc <- lm(CrimeRate ~ Metro + HighSchool + Poverty,
  data = crime_data %>%
    filter(state_abbrev != "DC") %>%
    column_to_rownames(var = "state_abbrev")
)

# 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)
```



Chapter 4

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 4.1), 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 4.2). Next we discuss maximum likelihood inference in GLMs (Section 4.3). Finally, we discuss how to carry out statistical inference in GLMs (Section 4.4).

4.1 Exponential dispersion model (EDM) distributions

4.1.1 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 $\mathbb{R}$:

$$f_{\mu,\sigma^2}(y) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(y-\mu)^2\right) = \exp\left(\frac{\mu y - \frac{1}{2}\mu^2}{\sigma^2}\right) \cdot \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}y^2\right). \tag{4.1}$$

We can consider a more general class of densities with respect to any measure ν:

$$f_{\theta,\phi}(y) \equiv \exp\left(\frac{\theta y - \psi(\theta)}{\phi}\right) h(y,\phi), \quad \theta \in \Theta \subseteq \mathbb{R},\ \phi > 0. \tag{4.2}$$

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, $\phi > 0$ 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). Sometimes, we parameterize this distribution using its mean and dispersion, writing

$$y \sim \text{EDM}(\mu, \phi). \tag{4.3}$$

When $\phi = 1$, the distribution becomes a *one-parameter natural exponential family*. The support of an EDM distribution remains fixed as (θ, ϕ) vary.

4.1.2 Examples

Normal distribution. As derived above, $y \sim N(\mu, \sigma^2)$ is an EDM with

$$\theta = \mu, \quad \psi(\theta) = -\frac{1}{2}\theta^2, \quad \phi = \sigma^2, \quad h(y, \phi) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}y^2\right). \tag{4.4}$$

Bernoulli distribution. Suppose $y \sim \text{Ber}(\mu)$. Then, we have

$$f(y) = \mu^y(1-\mu)^{1-y} = \exp\left(y \log\frac{\mu}{1-\mu} + \log(1-\mu)\right). \tag{4.5}$$

Therefore, we have $\theta = \log\frac{\mu}{1-\mu}$, so that $\log(1-\mu) = -\log(1+e^\theta)$. It follows that

$$\theta = \log\frac{\mu}{1-\mu}, \quad \psi(\theta) = \log(1+e^\theta), \quad \phi = 1, \quad h(y) = 1. \tag{4.6}$$

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 $\{0, 1\}$.

Binomial distribution. Consider the binomial proportion y: $my \sim \text{Bin}(m, \mu)$. We have

$$f(y) = \binom{m}{my}\mu^{my}(1-\mu)^{m(1-y)} = \exp\left(m\left(y \log\frac{\mu}{1-\mu} + \log(1-\mu)\right)\right)\binom{m}{my}, \tag{4.7}$$

so

$$\theta = \log\frac{\mu}{1-\mu}, \quad \psi(\theta) = \frac{e^\theta}{1+e^\theta}, \quad \phi = 1/m, \quad h(y, \phi) = \binom{m}{my}. \tag{4.8}$$

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 $\{0, \frac{1}{m}, \frac{2}{m}, \ldots, 1\}$.

Poisson distribution. Suppose $y \sim \text{Poi}(\mu)$. We have

$$f(y) = e^{-\mu}\frac{\mu^y}{y!} = \exp(y \log\mu - \mu)\frac{1}{y!}. \tag{4.9}$$

Therefore, we have $\theta = \log\mu$, so that $\mu = e^\theta$. It follows that

$$\theta = \log\mu, \quad \psi(\theta) = e^\theta, \quad \phi = 1, \quad h(y) = \frac{1}{y!}. \tag{4.10}$$

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.

4.1.3 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

$$\int f_{\theta,\phi}(y)d\nu(y) = \int \exp\left(\frac{\theta y - \psi(\theta)}{\phi}\right) h(y,\phi)d\nu(y) = 1. \tag{4.11}$$

Differentiating in θ and interchanging the derivative and the integral, we obtain

$$0 = \frac{d}{d\theta}\int f_{\theta,\phi}(y)dy = \int \frac{y - \dot{\psi}(\theta)}{\phi} f_{\theta,\phi}(y)dy, \tag{4.12}$$

from which it follows that

$$\dot{\psi}(\theta) = \int \dot{\psi}(\theta) f_{\theta,\phi}(y)dy = \int y f_{\theta,\phi}(y)dy = \mathbb{E}[y] \equiv \mu. \tag{4.13}$$

Thus, the first derivative of the log partition function is the mean of y. Differentiating again, we get

$$\phi\cdot\ddot{\psi}(\theta) = \phi\int \ddot{\psi}(\theta) f_{\theta,\phi}(y)d\nu(y) = \int (y-\dot{\psi}(\theta))^2 f_{\theta,\phi}(y)dy = \int (y-\mu)^2 f_{\theta,\phi}(y)d\nu(y) = \mathrm{Var}[y]. \tag{4.14}$$

Thus, the second derivative of the log-partition function multiplied by the dispersion parameter is the variance of y.

4.1.4 Relationships among the mean, variance, and natural parameter

Relationship between the mean and the natural parameter. The log-partition function ψ induces a connection (4.13) between the natural parameter θ and the mean μ. Because

$$\frac{d\mu}{d\theta} = \frac{d}{d\theta}\dot{\psi}(\theta) = \ddot{\psi}(\theta) = \frac{1}{\phi}\mathrm{Var}[y] > 0, \tag{4.15}$$

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.

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

$$V(\mu) \equiv \frac{d\mu}{d\theta}, \tag{4.16}$$

so that $\mathrm{Var}[y] = \phi V(\mu)$. 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).

4.1.5 The unit deviance and the saddlepoint approximation

The unit deviance. It's possible to rewrite the EDM distribution in terms of μ rather than in terms of θ. Take the quantity in the numerator of the exponential in the EDM density (4.2) and call it $t(y, \mu)$:

$$t(y, \mu) \equiv \theta y - \psi(\theta). \tag{4.17}$$

Let's consider the shape of this function by taking the first two derivatives with respect to θ:

$$\frac{\partial}{\partial \theta} t(y, \mu) = y - \mu \tag{4.18}$$

and

$$\frac{\partial^2}{\partial \theta^2} t(y, \mu) = -V(\mu) < 0. \tag{4.19}$$

Hence, $t(y, \mu)$ has a unique global maximum at $\mu = y$. For instance, in the normal case, we have $t(y, \mu) = -\frac{1}{2}(y - \mu)^2$. We can then define the *unit deviance* $d(y, \mu)$ as twice the distance between the value $t(y, \mu)$ and the optimal value $t(y, y)$:

$$d(y, \mu) \equiv 2(t(y, y) - t(y, \mu)). \tag{4.20}$$

The unit deviance is nonnegative, and minimized by $\mu = y$. For the normal distribution, the unit deviance is $d(y, \mu) = (y - \mu)^2$. The unit deviance can therefore be viewed as a "distance" between the mean μ and the observation y.

Example. For the Poisson distribution, we have $t(y, \mu) = y \log \mu - \mu$, so

$$d(y, \mu) \equiv 2(t(y, y) - t(y, \mu)) = 2(y \log y - y - y \log \mu + \mu) = 2\left(y \log \frac{y}{\mu} - (y - \mu)\right). \tag{4.21}$$

See Figure 4.1 for an example of the shape of this function.

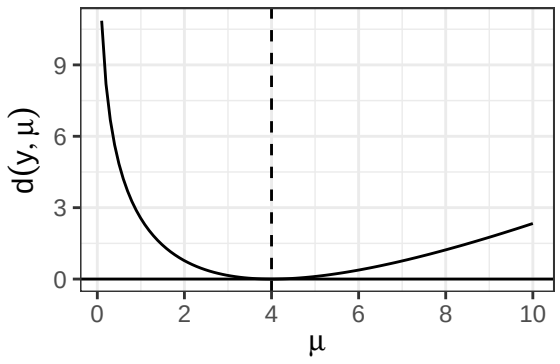


Figure 4.1: The Poisson unit deviance for $y = 4$.

The saddlepoint approximation. We have

$$\begin{aligned} f_{\theta,\phi}(y) &\equiv \exp\left(\frac{\theta y - \psi(\theta)}{\phi}\right) h(y,\phi) \\ &= \exp\left(\frac{t(y,\mu)}{\phi}\right) h(y,\phi) \\ &= \exp\left(\frac{-2(t(y,y) - t(y,\mu)) + 2t(y,y)}{2\phi}\right) h(y,\phi) \\ &= \exp\left(-\frac{d(y,\mu)}{2\phi}\right) \tilde{h}(y,\phi). \end{aligned} \tag{4.22}$$

As it turns out, in certain cases we have the approximation

$$\tilde{h}(y,\phi) \approx \frac{1}{\sqrt{2\pi\phi V(y)}}, \quad \text{so that} \quad f_{\theta,\phi}(y) \approx \frac{1}{\sqrt{2\pi\phi V(y)}} \exp\left(-\frac{d(y,\mu)}{2\phi}\right). \tag{4.23}$$

This is called the *saddlepoint approximation*. For example, if $y \sim \text{Poi}(\mu)$, then the saddlepoint approximation is $f_\theta(y) \approx \frac{1}{\sqrt{2\pi y}} \exp(-y \log \frac{y}{\mu} + (y - \mu))$. The significance of this approximation is that it implies that

$$\frac{d(y,\mu)}{\phi} \dot{\sim} \chi^2_1. \tag{4.24}$$

This approximation (which follows from the saddle approximation but is not immediately obvious) underlies *small dispersion asymptotics*, which will be useful for inferential purposes. As this name suggests, the saddlepoint approximation is accurate in cases when the dispersion ϕ is small, e.g. for binomial EDMs with large m. The saddlepoint approximation is also accurate for Poisson EDMs with large μ. As a rule of thumb, the approximation is good when $\min(my, m(1-y)) \geq 3$ for binomial proportions y or when $y \geq 3$ for Poisson-distributed y.

4.2 Generalized linear models and examples

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):

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, and observation weights w_i if

$$y_i \overset{\text{ind}}{\sim} \text{EDM}(\mu_i, \phi_0/w_i), \quad \eta_i \equiv g(\mu_i) = o_i + \boldsymbol{x}_{i*}^T\boldsymbol{\beta}. \tag{4.25}$$

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*

$$g(\mu) = \dot{\psi}^{-1}(\mu), \tag{4.26}$$

which, given the relationship (4.13), gives $\eta = \dot{\psi}^{-1}(\mu) = \theta$, i.e. the linear predictor coincides with the natural parameter. As discussed in the context of equation (4.15), $\dot{\psi}^{-1}$ is a valid link function because it is monotonic and differentiable. Canonical link functions are very commonly used with EDMs because they lead to various nice properties that general EDMs do not enjoy (e.g. concave log-likelihood).

Example: Linear regression model. The linear regression models is a special case of a GLM, with $\phi_0 = \sigma^2$ (unknown), $w_i = 1$, $o_i = 0$, and identity (canonical) link function:

$$y_i \overset{\text{ind}}{\sim} N(\mu_i, \sigma^2); \quad \eta_i = \mu_i = \boldsymbol{x}_{i*}^T \boldsymbol{\beta}. \tag{4.27}$$

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:

$$y_i \overset{\text{ind}}{\sim} N(\mu_i, \tfrac{\sigma^2}{m_i}); \quad \eta_i = \mu_i = \boldsymbol{x}_{i*}^T \boldsymbol{\beta}. \tag{4.28}$$

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:

$$y_i \overset{\text{ind}}{\sim} \text{Ber}(\mu_i); \quad \eta_i = \theta_i = \log \frac{\mu_i}{1 - \mu_i} = \boldsymbol{x}_{i*}^T \boldsymbol{\beta}. \tag{4.29}$$

Thus the canonical link function for logistic regression is the *logistic link function* $g(\mu) = \log \frac{\mu}{1-\mu}$.

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:

$$m_i y_i \sim \text{Bin}(m_i, \mu_i); \quad \eta_i = \log \frac{\mu_i}{1 - \mu_i} = o_i + \boldsymbol{x}_{i*}^T \boldsymbol{\beta}. \tag{4.30}$$

Note that a binomial proportion y_i based on 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.

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:

$$y_i \overset{\text{ind}}{\sim} \text{Poi}(\mu_i); \quad \eta_i = \theta_i = \log \mu_i = \boldsymbol{x}_{i*}^T \boldsymbol{\beta}. \tag{4.31}$$

Thus the canonical link function for Poisson regression is the *log link function* $g(\mu) = \log \mu$.

4.3 Parameter estimation in GLMs

4.3.1 The GLM likelihood, score, and Fisher information

The log-likelihood of a GLM is

$$\log \mathcal{L}(\boldsymbol{\beta}) = \sum_{i=1}^{n} \frac{\theta_i y_i - \psi(\theta_i)}{\phi_0 / w_i} + \sum_{i=1}^{n} \log h(y_i, \phi_0 / w_i). \tag{4.32}$$

Let's differentiate this with respect to $\boldsymbol{\beta}$, using the chain rule:

$$\begin{aligned}
\frac{\partial \log \mathcal{L}(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} &= \frac{\partial \log \mathcal{L}(\boldsymbol{\beta})}{\partial \boldsymbol{\theta}} \frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\mu}} \frac{\partial \boldsymbol{\mu}}{\partial \boldsymbol{\eta}} \frac{\partial \boldsymbol{\eta}}{\partial \boldsymbol{\beta}} \\
&= (\boldsymbol{y} - \boldsymbol{\mu})^T \operatorname{diag}(\phi_0 / w_i)^{-1} \cdot \operatorname{diag}(\ddot{\psi}(\theta_i))^{-1} \cdot \operatorname{diag}\left(\frac{\partial \mu_i}{\partial \eta_i}\right) \cdot \boldsymbol{X} \\
&= \frac{1}{\phi_0} (\boldsymbol{y} - \boldsymbol{\mu})^T \operatorname{diag}\left(\frac{w_i}{V(\mu_i)(d\eta_i/d\mu_i)^2}\right) \cdot \operatorname{diag}\left(\frac{\partial \eta_i}{\partial \mu_i}\right) \cdot \boldsymbol{X} \\
&\equiv \frac{1}{\phi_0} (\boldsymbol{y} - \boldsymbol{\mu})^T \boldsymbol{W} \boldsymbol{M} \boldsymbol{X}.
\end{aligned} \tag{4.33}$$

Here, $\boldsymbol{W} \equiv \operatorname{diag}(W_i)$ is a diagonal matrix of *working weights* and $\boldsymbol{M} \equiv \operatorname{diag}\left(\frac{\partial \eta_i}{\partial \mu_i}\right) = \operatorname{diag}(g'(\mu_i))$ is a diagonal matrix of link derivatives. Transposing, we get the score vector

$$\boldsymbol{U}(\boldsymbol{\beta}) = \frac{1}{\phi_0} \boldsymbol{X}^T \boldsymbol{M} \boldsymbol{W} (\boldsymbol{y} - \boldsymbol{\mu}). \tag{4.34}$$

To get the Fisher information matrix, note first that

$$\operatorname{Var}[\boldsymbol{y}] = \operatorname{diag}\left(\phi_0 \frac{V(\mu_i)}{w_i}\right) = \phi_0 \boldsymbol{W}^{-1} \boldsymbol{M}^{-2} \tag{4.35}$$

we can compute the covariance matrix of the score vector:

$$\begin{aligned}
\boldsymbol{I}(\boldsymbol{\beta}) = \operatorname{Var}[\boldsymbol{U}(\boldsymbol{\beta})] &= \frac{1}{\phi_0^2} \boldsymbol{X}^T \boldsymbol{M} \boldsymbol{W} \operatorname{Var}[\boldsymbol{y}] \boldsymbol{M} \boldsymbol{W} \boldsymbol{X} \\
&= \frac{1}{\phi_0^2} \boldsymbol{X}^T \boldsymbol{M} \boldsymbol{W} \phi_0 \boldsymbol{W}^{-1} \boldsymbol{M}^{-2} \boldsymbol{M} \boldsymbol{W} \boldsymbol{X} \\
&= \frac{1}{\phi_0} \boldsymbol{X}^T \boldsymbol{W} \boldsymbol{X}.
\end{aligned} \tag{4.36}$$

4.3.2 Maximum likelihood estimation of $\boldsymbol{\beta}$

To estimate $\boldsymbol{\beta}$, we can set the score vector to zero:

$$\frac{1}{\phi_0} \boldsymbol{X}^T \widehat{\boldsymbol{M}} \widehat{\boldsymbol{W}} (\boldsymbol{y} - \widehat{\boldsymbol{\mu}}) = 0 \quad \Longleftrightarrow \quad \boldsymbol{X}^T \operatorname{diag}\left(\frac{w_i}{V(\widehat{\mu}_i) g'(\widehat{\mu}_i)}\right) (\boldsymbol{y} - \widehat{\boldsymbol{\mu}}) = 0. \tag{4.37}$$

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 4.3.3. 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

$$\widehat{\boldsymbol{M}}\widehat{\boldsymbol{W}} = \operatorname{diag}\left(\frac{\widehat{\partial\mu_i/\partial\theta_i}}{V(\widehat{\mu}_i)}\right) = \frac{\ddot{\psi}(\widehat{\theta}_i)}{\ddot{\psi}(\widehat{\theta}_i)} = 1, \tag{4.38}$$

so the normal equations reduce to

$$\boldsymbol{X}^T(\boldsymbol{y} - \widehat{\boldsymbol{\mu}}) = 0. \tag{4.39}$$

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 (4.39) apply to the latter regressions as well.

In the linear regression case, we interpreted the normal equations (4.39) as an orthogonality statement: $\boldsymbol{y} - \widehat{\boldsymbol{\mu}} \perp C(\boldsymbol{X})$. In the case of GLMs, the $C(\boldsymbol{X}) \equiv \{\boldsymbol{\mu} = \mathbb{E}[\boldsymbol{y}] : \boldsymbol{\beta} \in \mathbb{R}^p\}$ 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 $\mathbb{R}^n$):

$$C(\boldsymbol{X}) \equiv \{\boldsymbol{\mu} = \mathbb{E}[\boldsymbol{y}] : \boldsymbol{\beta} \in \mathbb{R}^p\} = \{g^{-1}(\boldsymbol{X}\boldsymbol{\beta}) : \boldsymbol{\beta} \in \mathbb{R}^p\}. \tag{4.40}$$

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 (4.22). Taking a logarithm and summing over $i = 1, \dots, n$, we find the following expression for the negative log likelihood:

$$-\log \mathcal{L}(\boldsymbol{\beta}) = \frac{1}{2\phi}\sum_{i=1}^{n} d(y_i, \mu_i) = \frac{\sum_{i=1}^{n} w_i d(y_i, \mu_i)}{2\phi_0} \equiv \frac{D(\boldsymbol{y}, \boldsymbol{\mu})}{2\phi_0} \equiv \frac{1}{2}D^*(\boldsymbol{y}, \boldsymbol{\mu}). \tag{4.41}$$

$D(\boldsymbol{y}, \boldsymbol{\mu})$ 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, $D(\boldsymbol{y}, \boldsymbol{\mu}) = \|\boldsymbol{y} - \boldsymbol{\mu}\|^2$. Therefore, maximizing the GLM log likelihood is equivalent to minimizing the deviance:

$$\widehat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} D(\boldsymbol{y}, \boldsymbol{\mu}(\boldsymbol{\beta})), \quad \text{so that} \quad \widehat{\boldsymbol{\mu}} = \arg\min_{\boldsymbol{\mu} \in C(\boldsymbol{X})} D(\boldsymbol{y}, \boldsymbol{\mu}). \tag{4.42}$$

4.3.3 Iteratively reweighted least squares

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 4.3.1. *If g is the canonical link function, then the function $\log \mathcal{L}(\boldsymbol{\beta})$ defined in (4.32) 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 (4.3.1) gives us confidence that an iterative algorithm will converge to the global maximum of the likelihood. We present such an iterative algorithm next.

Newton-Raphson. We can maximize the log-likelihood (4.32) via the Newton Raphson algorithm, which involves the gradient and Hessian of the function we'd like to maximize. The gradient is the score vector (4.34), while the Hessian is the Fisher information (4.36). The Newton-Raphson iteration is therefore

$$\begin{aligned}\widehat{\boldsymbol{\beta}}^{(t+1)} &= \widehat{\boldsymbol{\beta}}^{(t)} - (\nabla^2_{\boldsymbol{\beta}} \log \mathcal{L}(\widehat{\boldsymbol{\beta}}^{(t)}))^{-1} \nabla_{\boldsymbol{\beta}} \log \mathcal{L}(\widehat{\boldsymbol{\beta}}^{(t)}) \\ &= \widehat{\boldsymbol{\beta}}^{(t)} + (\boldsymbol{X}^T \widehat{\boldsymbol{W}}^{(t)} \boldsymbol{X})^{-1} \boldsymbol{X}^T \widehat{\boldsymbol{W}}^{(t)} \widehat{\boldsymbol{M}}^{(t)} (\boldsymbol{y} - \widehat{\boldsymbol{\mu}}^{(t)}).\end{aligned} \tag{4.43}$$

See Figure 4.2.

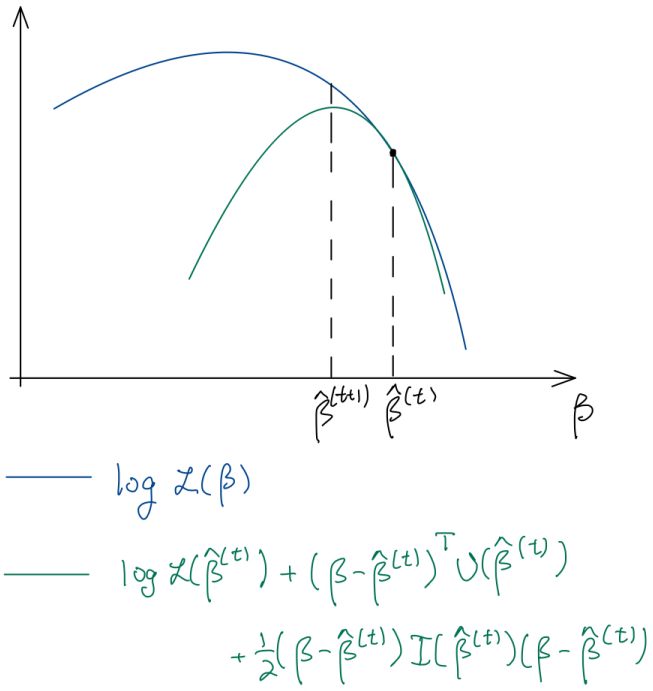


Figure 4.2: Newton-Raphson iteratively approximates the log likelihood via a quadratic function and maximizing that function.

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

$$\mathbb{E}_{\boldsymbol{\beta}}[\boldsymbol{y}] = g^{-1}(\boldsymbol{X}\boldsymbol{\beta}) \approx g^{-1}(\boldsymbol{X}\widehat{\boldsymbol{\beta}}^{(t)}) + \operatorname{diag}(\partial \mu_i / \partial \eta_i)(\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}^{(t)}) = \widehat{\boldsymbol{\mu}}^{(t)} + (\widehat{\boldsymbol{M}}^{(t)})^{-1}(\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}^{(t)})$$

and

$$\operatorname{Var}_{\boldsymbol{\beta}}[\boldsymbol{y}] \approx \phi_0 (\widehat{\boldsymbol{W}}^{(t)})^{-1} (\widehat{\boldsymbol{M}}^{(t)})^{-2},$$

recalling equation (4.35). Thus, up to the first two moments, near $\boldsymbol{\beta} = \widehat{\boldsymbol{\beta}}^{(t)}$ the distribution of $\boldsymbol{y}$ is approximately

$$\boldsymbol{y} = \widehat{\boldsymbol{\mu}}^{(t)} + (\widehat{\boldsymbol{M}}^{(t)})^{-1}(\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}^{(t)}) + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim N\left(\mathbf{0}, \phi_0 (\widehat{\boldsymbol{W}}^{(t)})^{-1} (\widehat{\boldsymbol{M}}^{(t)})^{-2}\right), \tag{4.44}$$

or, equivalently,

$$\boldsymbol{z}^{(t)} \equiv \widehat{\boldsymbol{M}}^{(t)}(\boldsymbol{y} - \widehat{\boldsymbol{\mu}}^{(t)}) + \boldsymbol{X}\widehat{\boldsymbol{\beta}}^{(t)} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}', \quad \boldsymbol{\epsilon}' \sim N(\boldsymbol{0}, \phi_0(\widehat{\boldsymbol{W}}^{(t)})^{-1}). \tag{4.45}$$

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

$$\widehat{\boldsymbol{\beta}}^{(t+1)} = (\boldsymbol{X}^T\widehat{\boldsymbol{W}}^{(t)}\boldsymbol{X})^{-1}\boldsymbol{X}^T\widehat{\boldsymbol{W}}^{(t)}\boldsymbol{z}^{(t)}, \tag{4.46}$$

which we define as our next iterate. It's easy to verify that the IRLS iteration (4.46) is equivalent to the Newton-Raphson iteration (4.43).

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:

$$r_i^D \equiv \text{sign}(y_i - \widehat{\mu}_i)\sqrt{w_i d(y_i, \widehat{\mu}_i)}. \tag{4.47}$$

On the other hand, Pearson residuals are defined as variance-normalized residuals:

$$r_i^P \equiv \frac{y_i - \widehat{\mu}_i}{\sqrt{V(\widehat{\mu}_i)/w_i}}. \tag{4.48}$$

These residuals can be viewed as residuals from the (converged) weighted linear regression model (4.45). 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*

$$\widetilde{\phi}_0^D \equiv \frac{1}{n-p}\|r^D\|^2 \equiv \frac{1}{n-p}\sum_{i=1}^n w_i d(y_i, \widehat{\mu}_i) \equiv \frac{1}{n-p}D(\boldsymbol{y}; \widehat{\boldsymbol{\mu}}); \tag{4.49}$$

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*:

$$\widetilde{\phi}_0^P \equiv \frac{1}{n-p}X^2 \equiv \frac{1}{n-p}\|r^P\|^2 \equiv \frac{1}{n-p}\sum_{i=1}^n w_i \frac{(y_i - \widehat{\mu}_i)^2}{V(\mu_i)}. \tag{4.50}$$

X^2 is known as the Pearson X^2 statistic. The deviance estimator can be more accurate when the EDM model is well-specified; the Pearson estimator can be more robust when only the first two moments of the EDM model are correct.

4.4 Inference in GLMs

Inferential goals. There are two types of inferential goals: hypothesis testing and confidence interval construction. Within hypothesis testing, we can test $H_0 : \beta_j = 0$ (importance of a single coefficient), $H_0 : \boldsymbol{\beta}_S = \boldsymbol{0}$ for some $S \subset \{0, \ldots, p-1\}$ (importance of a group of coefficients), or $H_0 : \boldsymbol{\eta} = \boldsymbol{X}\boldsymbol{\beta}$ (goodness of fit). To elaborate on the latter, we would like to test

$$H_0 : y_i \overset{\text{ind}}{\sim} \text{EDM}(g^{-1}(o_i + \boldsymbol{x}_{i*}^T\boldsymbol{\beta}), \phi_0/w_i), \ \boldsymbol{\beta} \in \mathbb{R}^p \quad \text{versus} \quad H_1 : y_i \overset{\text{ind}}{\sim} \text{EDM}(\mu_i, \phi_0/w_i), \ \boldsymbol{\mu} \in \mathbb{R}^n.$$

Within confidence intervals, we may want to construct intervals for the coefficients β_j or for fitted values η_i or μ_i.

Inferential tools. Inference in GLMs is based on asymptotic likelihood theory. Hypothesis tests (and, by inversion, confidence intervals) can be constructed in three asymptotically equivalent ways: Wald tests, likelihood ratio tests (LRT), and score tests. 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 4.3.

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 4.3: A comparison of the three asymptotic methods for GLM inference.

4.4.1 Wald inference

Wald inference is based on the following asymptotic normality statement:

$$\widehat{\boldsymbol{\beta}} \dot{\sim} N(\boldsymbol{\beta}, \boldsymbol{I}^{-1}(\boldsymbol{\beta})) = N(\boldsymbol{\beta}, \phi_0(\boldsymbol{X}^T\boldsymbol{W}(\boldsymbol{\beta})\boldsymbol{X})^{-1}), \tag{4.51}$$

recalling our derivation of the Fisher information from equation (4.36). This is a *large-sample approximation*. Wald inference is easy to carry out, and for this reason is considered the default type of inference. However, as we'll 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.

Wald test for $\beta_j = \beta_j^0$ (known ϕ_0). Based on the Wald approximation (4.51), under the null hypothesis, we have

$$\widehat{\beta}_j \dot{\sim} N(\beta_j^0, \phi_0[(\boldsymbol{X}^T\boldsymbol{W}(\boldsymbol{\beta})\boldsymbol{X})^{-1}]_{jj}) \approx N(\beta_j^0, \phi_0[(\boldsymbol{X}^T\boldsymbol{W}(\widehat{\boldsymbol{\beta}})\boldsymbol{X})^{-1}]_{jj}) \equiv N(0, \mathrm{SE}(\beta_j)^2), \tag{4.52}$$

where we have used a plug-in estimator of the variance. This leaves us with the Wald z-test

$$\phi(\boldsymbol{X}, \boldsymbol{y}) \equiv \mathbb{1}\left(\left|\frac{\widehat{\beta}_j - \beta_j^0}{\mathrm{SE}(\beta_j)}\right| > z_{1-\alpha/2}\right). \tag{4.53}$$

Wald test for $\boldsymbol{\beta}_S = \boldsymbol{\beta}_S^0$ (known ϕ_0). Extending the reasoning above, we have under the null hypothesis that

$$\widehat{\boldsymbol{\beta}}_S \dot{\sim} N(\boldsymbol{\beta}_S^0, \phi_0[(\boldsymbol{X}^T\boldsymbol{W}(\boldsymbol{\beta})\boldsymbol{X})^{-1}]_{S,S}) \approx N(\beta_j^0, \phi_0[(\boldsymbol{X}^T\boldsymbol{W}(\widehat{\boldsymbol{\beta}})\boldsymbol{X})^{-1}]_{S,S}), \tag{4.54}$$

and therefore

$$\frac{1}{\phi_0}(\widehat{\boldsymbol{\beta}}_S - \boldsymbol{\beta}_S^0)^T \left([(\boldsymbol{X}^T\boldsymbol{W}(\widehat{\boldsymbol{\beta}})\boldsymbol{X})^{-1}]_{S,S}\right)^{-1} (\widehat{\boldsymbol{\beta}}_S - \boldsymbol{\beta}_S^0) \mathrel{\dot\sim} \chi^2_{|S|}. \tag{4.55}$$

Hence we have the Wald χ^2 test

$$\phi(\boldsymbol{X}, \boldsymbol{y}) \equiv \mathbb{1}\left(\frac{1}{\phi_0}(\widehat{\boldsymbol{\beta}}_S - \boldsymbol{\beta}_S^0)^T \left([(\boldsymbol{X}^T\boldsymbol{W}(\widehat{\boldsymbol{\beta}})\boldsymbol{X})^{-1}]_{S,S}\right)^{-1} (\widehat{\boldsymbol{\beta}}_S - \boldsymbol{\beta}_S^0) > \chi^2_{|S|}(1-\alpha)\right). \tag{4.56}$$

Wald interval for β_j (known ϕ_0). Inverting the Wald test for β_j, we get a Wald confidence interval:

$$\mathrm{CI}(\widehat{\beta}_j) \equiv \widehat{\beta}_j \pm z_{1-\alpha/2} \cdot \mathrm{SE}(\beta_j), \quad \text{where} \quad \mathrm{SE}(\beta_j) \equiv \sqrt{\phi_0[(\boldsymbol{X}^T\boldsymbol{W}(\widehat{\boldsymbol{\beta}})\boldsymbol{X})^{-1}]_{jj}}. \tag{4.57}$$

Wald intervals for η_i and μ_i (known ϕ_0). Given the Wald approximation (4.51), we have

$$\widehat{\eta}_i \equiv o_i + \boldsymbol{x}_{i*}^T\widehat{\boldsymbol{\beta}} \mathrel{\dot\sim} N(\eta_i, \phi_0 \cdot \boldsymbol{x}_{i*}^T(\boldsymbol{X}^T\boldsymbol{W}(\widehat{\boldsymbol{\beta}})\boldsymbol{X})^{-1}\boldsymbol{x}_{i*}) \equiv N(\eta_i, \mathrm{SE}(\eta_i)^2).$$

Hence, the Wald interval for η_i is

$$\mathrm{CI}(\eta_i) \equiv o_i + \boldsymbol{x}_{i*}^T\widehat{\boldsymbol{\beta}} \pm z_{1-\alpha/2} \cdot \mathrm{SE}(\eta_i), \quad \text{where} \quad \mathrm{SE}(\eta_i) \equiv \sqrt{\phi_0 \boldsymbol{x}_{i*}^T(\boldsymbol{X}^T\boldsymbol{W}(\widehat{\boldsymbol{\beta}})\boldsymbol{X})^{-1}\boldsymbol{x}_{i*}}. \tag{4.58}$$

A confidence interval for $\mu_i \equiv \mathbb{E}_{\boldsymbol{\beta}}[y_i] = g^{-1}(\eta_i)$ can be obtained by applying the strictly increasing 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.

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 $\mathrm{SE}(\beta_j) \equiv \sqrt{\widetilde{\phi}_0 \cdot [(\boldsymbol{X}^T\boldsymbol{W}(\widehat{\boldsymbol{\beta}})\boldsymbol{X})^{-1}]_{jj}}$, and our test statistic for $H_0 : \beta_j = \beta_j^0$ is

$$\frac{\widehat{\beta}_j - \beta_j^0}{\sqrt{\widetilde{\phi}_0}\sqrt{[(\boldsymbol{X}^T\boldsymbol{W}(\widehat{\boldsymbol{\beta}})\boldsymbol{X})^{-1}]_{jj}}}. \tag{4.59}$$

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

$$\phi(\boldsymbol{X}, \boldsymbol{y}) \equiv \mathbb{1}\left(\left|\frac{\widehat{\beta}_j - \beta_j^0}{\mathrm{SE}(\beta_j)}\right| > t_{n-p}(1-\alpha/2)\right). \tag{4.60}$$

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,

$$\frac{\frac{1}{|S|}(\widehat{\boldsymbol{\beta}}_S - \boldsymbol{\beta}_S^0)^T \left([(\boldsymbol{X}^T\boldsymbol{W}(\widehat{\boldsymbol{\beta}})\boldsymbol{X})^{-1}]_{S,S}\right)^{-1} (\widehat{\boldsymbol{\beta}}_S - \boldsymbol{\beta}_S^0)}{\widetilde{\phi}_0} \mathrel{\dot\sim} F_{|S|,n-p}.$$

4.4.2 Likelihood ratio inference

Let $\ell(\boldsymbol{y}, \boldsymbol{\mu}) = -\frac{D(\boldsymbol{y},\boldsymbol{\mu})}{2\phi_0} + C$ be the GLM log-likelihood. 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 $\widehat{\boldsymbol{\mu}}_{\text{-}S}$ be the maximum likelihood estimate under the null hypothesis. Likelihood ratio inference is based on the following asymptotic chi square distribution:

$$2(\ell(\boldsymbol{y}, \widehat{\boldsymbol{\mu}}) - \ell(\boldsymbol{y}, \widehat{\boldsymbol{\mu}}_{\text{-}S})) = \frac{D(\boldsymbol{y}, \widehat{\boldsymbol{\mu}}_{\text{-}S}) - D(\boldsymbol{y}, \widehat{\boldsymbol{\mu}})}{\phi_0} \mathrel{\dot\sim} \chi^2_{|S|}. \tag{4.61}$$

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,

$$\frac{d(y_i, \mu_i)}{\phi_0/w_i} \mathrel{\dot\sim} \chi^2_1, \tag{4.62}$$

so

$$\frac{D(\boldsymbol{y}, \boldsymbol{\mu})}{\phi_0} = \sum_{i=1}^n \frac{d(y_i, \mu_i)}{\phi_0/w_i} \mathrel{\dot\sim} \chi^2_n. \tag{4.63}$$

Testing one or more coefficients (ϕ_0 known). Suppose we wish to test the null hypothesis $H_0 : \boldsymbol{\beta}_S = \boldsymbol{\beta}_S^0$. Then, based on the approximation (4.61) we can define the likelihood ratio test

$$\phi(\boldsymbol{X}, \boldsymbol{y}) \equiv \mathbb{1}\left(\frac{D(\boldsymbol{y}, \widehat{\boldsymbol{\mu}}_{\text{-}S}) - D(\boldsymbol{y}, \widehat{\boldsymbol{\mu}})}{\phi_0} > \chi^2_{|S|}(1-\alpha)\right). \tag{4.64}$$

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

$$\text{CI}(\beta_j) \equiv \left\{\beta_j : \frac{D(\boldsymbol{y}, \widehat{\boldsymbol{\mu}}_{\text{-}j}(\beta_j)) - D(\boldsymbol{y}, \widehat{\boldsymbol{\mu}})}{\phi_0} \le \chi^2_{|S|}(1-\alpha)\right\}. \tag{4.65}$$

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 get computationally out of hand due to the curse of dimensionality.

Goodness of fit testing (ϕ_0 known). For ϕ_0 known, we can also construct a goodness of fit test: This includes comparing the GLM to a saturated model, to get a goodness of fit test via

$$\frac{D(\boldsymbol{y}, \widehat{\boldsymbol{\mu}}) - D(\boldsymbol{y}, \boldsymbol{y})}{\phi_0} = \frac{D(\boldsymbol{y}; \widehat{\boldsymbol{\mu}})}{\phi_0} \mathrel{\dot\sim} \chi^2_{n-p}, \tag{4.66}$$

assuming the saturated model can be estimated relatively well (small dispersion asymptotics).

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:

$$F \equiv \frac{(D(\boldsymbol{y}; \widehat{\boldsymbol{\mu}}_{\text{-}S}) - D(\boldsymbol{y}; \widehat{\boldsymbol{\mu}}))/|S|}{\widetilde{\phi}_0}. \tag{4.67}$$

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 construction 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.

4.4.3 Score-based inference

Score-based inference can be used for the same set of inferential tasks as likelihood ratio inference, but in practice it is primarily applied in the context of goodness of fit testing. Consider the model

$$y_i \overset{\text{ind}}{\sim} \text{EDM}(\mu_i, \phi_0/w_i).$$

Now, consider the hypothesis testing problem

$$H_0 : \boldsymbol{\theta} = \dot{\psi}^{-1}(g(\boldsymbol{o} + \boldsymbol{X}\boldsymbol{\beta})),\ \boldsymbol{\beta} \in \mathbb{R}^p \quad \text{versus} \quad H_1 : \boldsymbol{\theta} \in \mathbb{R}^n.$$

Score-based-inference is based on the approximation

$$\boldsymbol{U}(\boldsymbol{\theta}) \dot{\sim} N(0, \boldsymbol{I}(\boldsymbol{\theta})). \tag{4.68}$$

Plugging in a consistent estimate of $\boldsymbol{\theta}$ under the null hypothesis, $\widehat{\boldsymbol{\theta}}$ from fitting the GLM, we get

$$\boldsymbol{U}(\widehat{\boldsymbol{\theta}})^T \boldsymbol{I}(\widehat{\boldsymbol{\theta}})^{-1} \boldsymbol{U}(\widehat{\boldsymbol{\theta}}) \dot{\sim} \chi^2_{n-p}. \tag{4.69}$$

Based on the model parameterized by $\boldsymbol{\theta}$, we compute that

$$\boldsymbol{U}(\boldsymbol{\theta}) = \frac{1}{\phi_0}\text{diag}(w_i)(\boldsymbol{y} - \boldsymbol{\mu}) \quad \text{and} \quad \boldsymbol{I}(\boldsymbol{\theta}) = \frac{1}{\phi_0}\text{diag}(w_i V(\mu_i)). \tag{4.70}$$

Hence,

$$\boldsymbol{U}(\widehat{\boldsymbol{\theta}})^T \boldsymbol{I}(\widehat{\boldsymbol{\theta}})^{-1} \boldsymbol{U}(\widehat{\boldsymbol{\theta}}) = \frac{1}{\phi_0}(\boldsymbol{y} - \widehat{\boldsymbol{\mu}})^T \text{diag}(w_i/V(\widehat{\mu}_i))(\boldsymbol{y} - \widehat{\boldsymbol{\mu}}) = \frac{1}{\phi_0}\sum_{i=1}^{n} \frac{w_i(y_i - \widehat{\mu}_i)^2}{V(\widehat{\mu}_i)} \equiv \frac{1}{\phi_0} X^2, \tag{4.71}$$

where X^2 is the Pearson chi-square statistic, which he proposed in 1900. It was only pointed out many decades later that this is a score test.

4.5 R demo

4.5.1 Crime data

Let's revisit the crime data from Homework 2, this time fitting a logistic regression to it.

```
# 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
## # ... with 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)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -21.043   -9.176    0.418    9.053   47.174
##
## 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 mean 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 data set.

4.5.2 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
## # ... with 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)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2826  -1.1220  -0.8011   0.4159   3.3511
##
## 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
```

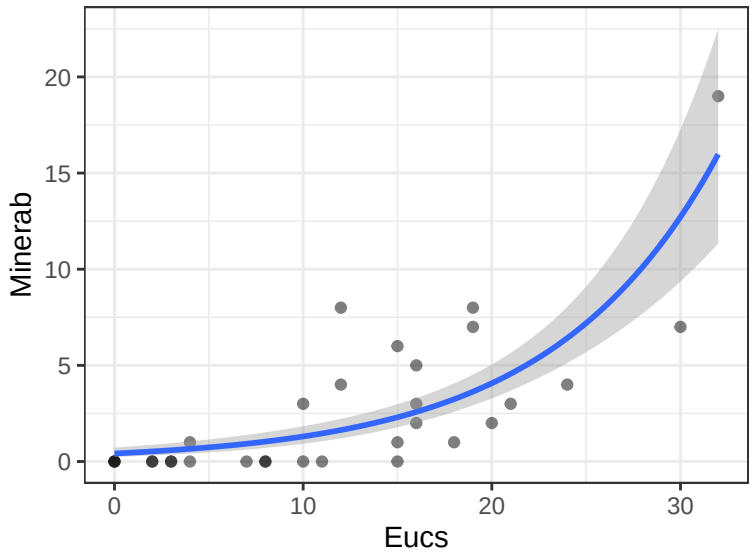
We see that the prediction on the linear predictor scale is exactly the logarithm of the prediction on the mean scale. However, the standard error given on the mean scale uses the delta method. We prefer to directly transform the confidence interval from the linear scale using the inverse link function (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.

Chapter 5

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.

5.1 Logistic regression

5.1.1 Model definition and interpretation

Model definition. Recall from Chapter 4 that the logistic regression model is

$$m_i y_i \overset{\text{ind}}{\sim} \text{Bin}(m_i, \pi_i); \quad \text{logit}(\pi_i) = \log \frac{\pi_i}{1-\pi_i} = \boldsymbol{x}_{i*}^T \boldsymbol{\beta}. \tag{5.1}$$

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.

Generative model equivalent. Consider the following generative model for $(\boldsymbol{x}, y) \in \mathbb{R}^{p-1} \times \{0, 1\}$:

$$y \sim \text{Ber}(\pi); \quad \boldsymbol{x}|y \sim \begin{cases} N(\boldsymbol{\mu}_0, \boldsymbol{V}) & \text{if } y = 0 \\ N(\boldsymbol{\mu}_1, \boldsymbol{V}) & \text{if } y = 1 \end{cases}. \tag{5.2}$$

Then, we can derive that $y|\boldsymbol{x}$ follows a logistic regression model (called a *discriminative* model because it conditions on $\boldsymbol{x}$). Indeed,

$$\begin{aligned}\text{logit}(p(y=1|\boldsymbol{x})) &= \log \frac{p(y=1)p(\boldsymbol{x}|y=1)}{p(y=0)p(\boldsymbol{x}|y=0)} \\ &= \log \frac{\pi \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_1)^T\boldsymbol{V}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_1)\right)}{(1-\pi)\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_0)^T\boldsymbol{V}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_0)\right)} \\ &= \beta_0 + \boldsymbol{x}^T\boldsymbol{V}^{-1}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_0) \\ &\equiv \beta_0 + \boldsymbol{x}^T\boldsymbol{\beta}_{\text{-}0}.\end{aligned} \tag{5.3}$$

This is another natural route to motivating the logistic regression model.

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*:

$$e^{\beta_1} = \frac{\mathbb{P}[y=1|x=1]/\mathbb{P}[y=0|x=1]}{\mathbb{P}[y=1|x=0]/\mathbb{P}[y=0|x=0]}. \tag{5.4}$$

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 5.1). 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.

	Success	Failure	Total
Treatment	9	12	21
Control	7	17	24
Total	16	29	45

Table 5.1: An example of a 2×2 contingency table.

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 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,

$$\text{logit}(\mathbb{P}[y=1|z=1,\boldsymbol{x}]) = \log\frac{\rho_1\mathbb{P}[y=1|\boldsymbol{x}]}{\rho_0\mathbb{P}[y=0|\boldsymbol{x}]} = \log\frac{\rho_1}{\rho_0} + \text{logit}(\mathbb{P}[y=1|\boldsymbol{x}]) = \left(\log\frac{\rho_1}{\rho_0} + \beta_0\right) + \boldsymbol{x}^T\boldsymbol{\beta}_{\text{-}0}.$$

Thus, conditioning on retrospective sampling changes only the intercept term, but preserves the coefficients of $\boldsymbol{x}$. Therefore, we can carry out inference for $\boldsymbol{\beta}_{\text{-}0}$ in the same way regardless of whether the study design is prospective or retrospective.

5.1.2 Estimation and inference

Score and Fisher information. Recall from Chapter 4 that

$$\boldsymbol{U}(\boldsymbol{\beta}) = \frac{1}{\phi_0}\boldsymbol{X}^T\boldsymbol{M}\boldsymbol{W}(\boldsymbol{y}-\boldsymbol{\mu}) \quad \text{and} \quad \boldsymbol{I}(\boldsymbol{\beta}) = \frac{1}{\phi_0}\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X}, \tag{5.5}$$

where

$$\boldsymbol{W} \equiv \text{diag}\left(\frac{w_i}{V(\mu_i)(d\eta_i/d\mu_i)^2}\right) \quad \text{and} \quad \boldsymbol{M} \equiv \text{diag}\left(\frac{\partial\eta_i}{\partial\mu_i}\right). \tag{5.6}$$

Since logistic regression uses a canonical link function, we get the following simplifications:

$$\boldsymbol{W} \equiv \text{diag}\left(w_i V(\mu_i)\right) \quad \text{and} \quad \boldsymbol{M} \equiv \text{diag}\left(\frac{1}{V(\mu_i)}\right). \tag{5.7}$$

We substitute 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

$$0 = \boldsymbol{X}^T\text{diag}\left(m_i\right)\left(\boldsymbol{y}-\widehat{\boldsymbol{\mu}}\right) \quad \Longleftrightarrow \quad \sum_{i=1}^{n} m_i x_{ij}(y_i - \widehat{\pi}_i) = 0, \quad j = 0,\ldots,p-1. \tag{5.8}$$

We can solve these equations using IRLS. The Fisher information is

$$\boldsymbol{I}(\boldsymbol{\beta}) = \boldsymbol{X}^T\text{diag}\left(m_i\pi_i(1-\pi_i)\right)\boldsymbol{X}. \tag{5.9}$$

Wald inference. Using the results in the previous paragraph, we can carry out Wald inference based on the normal approximation

$$\widehat{\boldsymbol{\beta}} \dot{\sim} N\left(\boldsymbol{\beta}, \left(\boldsymbol{X}^T\text{diag}(m_i\widehat{\pi}_i(1-\widehat{\pi}_i))\boldsymbol{X}\right)^{-1}\right). \tag{5.10}$$

This approximation holds for $\sum_{i=1}^{n} m_i \to \infty$. Unfortunately, Wald inference in finite samples does not always perform very well. The Wald test above is known to be conservative due to the *Hauck-Donner effect*. As an example, consider testing $H_0 : \beta_0 = 0.5$ in the intercept-only model

$$my \sim \text{Bin}(m,\pi); \quad \text{logit}(\pi) = \beta_0. \tag{5.11}$$

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 $y \to 1$ (Figure 5.1), 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!

Perfect separability. If we have a situation where a hyperplane in covariate space separates observations with $y_i = 0$ from those with $y_i = 1$, we have *perfect separability*. It turns out that some of the maximum likelihood estimates are infinite in this case. The Wald test completely fails in this case, since it uses the parameter estimates as test statistics.

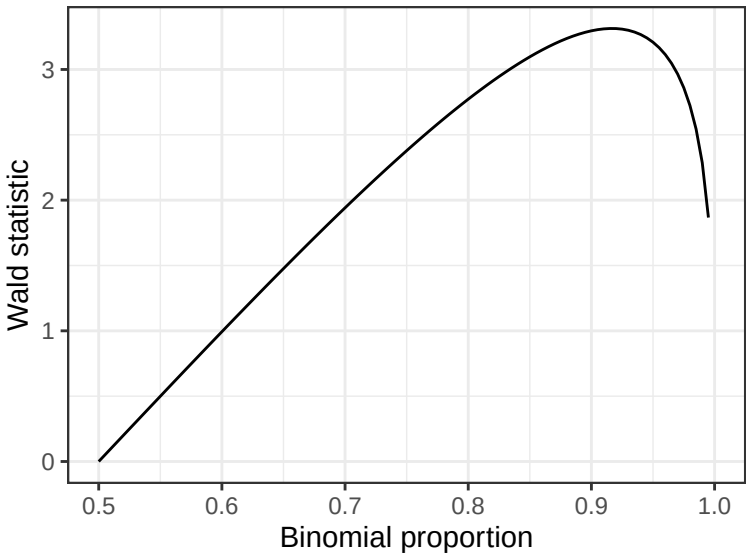


Figure 5.1: The Hauck-Donner effect: The Wald statistic for testing $H_0 : \pi = 0.5$ within the model $my \sim \text{Bin}(m, \pi)$ decreases as the proportion y approaches 1. Here, $m = 25$.

Likelihood ratio inference. Let's first compute the deviance of a logistic regression model. The unit deviance is

$$t(y, \pi) = y \log \pi + (1 - y) \log(1 - \pi). \tag{5.12}$$

Hence, we have

$$t(y, y) = y \log y + (1 - y) \log(1 - y). \tag{5.13}$$

In particular, for $y \in \{0, 1\}$, by taking the limit we find that $t(0, 0) = t(1, 1) = 0$. Hence, the unit deviance is

$$d(y, \mu) \equiv 2(t(y, y) - t(y, \pi)) = 2 \left(y \log \frac{y}{\pi} + (1 - y) \log \frac{1 - y}{1 - \pi} \right). \tag{5.14}$$

The total deviance, therefore, is

$$D(\boldsymbol{y}, \boldsymbol{\pi}) \equiv \sum_{i=1}^{n} w_i d(y_i, \widehat{\pi}_i) = 2 \sum_{i=1}^{n} \left(m_i y_i \log \frac{y_i}{\widehat{\pi}_i} + m_i (1 - y_i) \log \frac{1 - y_i}{1 - \widehat{\pi}_i} \right). \tag{5.15}$$

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

$$D(\boldsymbol{y}, \widehat{\boldsymbol{\pi}}_0) - D(\boldsymbol{y}, \widehat{\boldsymbol{\pi}}_1) = 2 \sum_{i=1}^{n} \left(m_i y_i \log \frac{\widehat{\pi}_{i1}}{\widehat{\pi}_{i0}} + m_i (1 - y_i) \log \frac{1 - \widehat{\pi}_{i1}}{1 - \widehat{\pi}_{i0}} \right). \tag{5.16}$$

We can then construct a likelihood ratio test in the usual way. Likelihood ratio inference can give nontrivial conclusions in cases when Wald inference cannot, e.g. in the case of perfect separability. Indeed, suppose that

$$m_i y_i \sim \text{Bin}(m_i, \pi_i), \quad \text{logit}(\pi_i) = \beta_0 + \beta_1 x_i, \quad i = 1, 2. \tag{5.17}$$

We would like to test $H_0 : \beta_1 = 0$. Suppose that we observe $(x_1, y_1) = (0, 0)$, $(x_2, y_2) = (1, 1)$, giving us complete separability. Can we still get a meaningful test of H_0? We can write out the likelihood

ratio test statistic, which is

$$D(\boldsymbol{y}; \widehat{\boldsymbol{\pi}}) = 2\left(m_1 \log \frac{1}{1 - \frac{m_2}{m_1+m_2}} + m_2 \log \frac{1}{\frac{m_2}{m_1+m_2}}\right) = 2\left(m_1 \log \frac{m_1+m_2}{m_1} + m_2 \log \frac{m_1+m_2}{m_2}\right).$$

This is a number that we can compare to the χ^2_1 distribution to get a p-value, as usual.

Goodness of fit tests. We can test goodness of fit in the grouped logistic regression model by comparing the deviance statistic (5.15) to the asymptotic null distribution χ^2_{n-p}. We can alternatively use the score test, which gives us Pearson's X^2 statistic:

$$X^2 = \sum_{i=1}^n \frac{w_i(y_i - \widehat{\mu}_i)^2}{V(\widehat{\mu}_i)} = \sum_{i=1}^n \frac{m_i(y_i - \widehat{\pi}_i)^2}{\widehat{\pi}_i(1-\widehat{\pi}_i)}. \tag{5.18}$$

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

$$s_1 = m_1 y_1 \sim \text{Bin}(m_1, \pi_1) \quad \textit{and} \quad s_2 = m_2 y_2 \sim \text{Bin}(m_2, \pi_2). \tag{5.19}$$

The trick is to conduct inference *conditional on* $s_1 + s_2$. Note that under $H_0 : \pi_1 = \pi_2$, we have

$$\begin{aligned}
\mathbb{P}[s_1 = t | s_1 + s_2 = v] &= \mathbb{P}[s_1 = t | s_1 + s_2 = v] \\
&= \frac{\mathbb{P}[s_1 = t, s_2 = v - t]}{\mathbb{P}[s_1 + s_2 = v]} \\
&= \frac{\binom{m_1}{t}\pi^t(1-\pi)^{m_1-t}\binom{m_2}{v-t}\pi^{v-t}(1-\pi)^{m_2-(v-t)}}{\binom{m_1+m_2}{v}\pi^v(1-\pi)^{m_1+m_2-v}} \\
&= \frac{\binom{m_1}{t}\binom{m_2}{v-t}}{\binom{m_1+m_2}{v}}.
\end{aligned} \tag{5.20}$$

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.

5.2 Poisson regression

The Poisson regression model (with offsets) is

$$y_i \overset{\text{ind}}{\sim} \text{Poi}(\mu_i); \quad \log \mu_i = o_i + \boldsymbol{x}_{i*}^T \boldsymbol{\beta}. \tag{5.21}$$

Because the log of the mean is linear in the predictors, Poisson regression models are often called *loglinear models*. We have seen in Chapter 4 how to carry out inference for this model based on the Wald, likelihood ratio, and score tests. Recall, for example, that the deviance of this model is

$$D(\boldsymbol{y}; \widehat{\boldsymbol{\mu}}) = \sum_{i=1}^n y_i \log \frac{y_i}{\widehat{\mu}_i}. \tag{5.22}$$

5.2.1 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

$$y_i \overset{\text{ind}}{\sim} \text{Poi}(\pi_i t_i), \quad \log \pi_i = \boldsymbol{x}_{i*}^T \boldsymbol{\beta}, \tag{5.23}$$

where π_i represents the rate of events per day / per square mile / per capita, etc. In other words,

$$y_i \overset{\text{ind}}{\sim} \text{Poi}(\mu_i), \quad \log \mu_i = \log t_i + \boldsymbol{x}_{i*}^T \boldsymbol{\beta}, \tag{5.24}$$

which is exactly equation (5.21) 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.

5.2.2 Relationship between Poisson and multinomial distributions

Suppose that $y_i \overset{\text{ind}}{\sim} \text{Poi}(\mu_i)$ for $i = 1, \ldots, n$. Then,

$$\begin{aligned} \mathbb{P}\left[y_1 = m_1, \ldots, y_n = m_n \,\middle|\, \sum_i y_i = m \right] &= \frac{\mathbb{P}[y_1 = m_1, \ldots, y_n = m_n]}{\mathbb{P}[\sum_i y_i = m]} \\ &= \frac{\prod_{i=1}^n e^{-\mu_i} \frac{\mu_i^{y_i}}{y_i!}}{e^{-\sum_i \mu_i} \frac{(\sum_i \mu_i)^m}{m!}} \\ &= \binom{m}{m_1, \ldots, m_n} \prod_{i=1}^n \pi_i^{y_i}; \quad \pi_i \equiv \frac{\mu_i}{\sum_{i'=1}^n \mu_{i'}}. \end{aligned} \tag{5.25}$$

We recognize the last expression 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.

5.2.3 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

$$(y_{00}, y_{01}, y_{10}, y_{11}) | n \sim \text{Mult}(n, (\pi_{00}, \pi_{01}, \pi_{10}, \pi_{11})). \tag{5.26}$$

Our primary question is whether these two random variables are independent, i.e.

$$\pi_{jk} = \pi_{j+}\pi_{+k}, \quad \text{where} \quad \pi_{j+} \equiv \mathbb{P}[x_1 = j] = \pi_{j1} + \pi_{j2}; \quad \pi_{+k} \equiv \mathbb{P}[x_2 = k] = \pi_{1k} + \pi_{2k}. \tag{5.27}$$

We can express this equivalently as

$$\pi_{00}(\pi_{00}+\pi_{01}+\pi_{10}+\pi_{11}) = \pi_{00} = \pi_{0+}\pi_{+0} = (\pi_{00}+\pi_{01})(\pi_{00}+\pi_{10}) \quad \Longleftrightarrow \quad \pi_{00}\pi_{11} = \pi_{01}\pi_{10}. \tag{5.28}$$

In other words, we can express the independence hypothesis concisely as

$$H_0 : \frac{\pi_{11}\pi_{00}}{\pi_{10}\pi_{01}} = 1. \tag{5.29}$$

Let's arbitrarily assume that, additionally, $n \sim \text{Poi}(\mu_{++})$. Then,

$$(y_{00}, y_{01}, y_{10}, y_{11}) \sim \text{Poi}(\mu_{++}\pi_{00}) \times \text{Poi}(\mu_{++}\pi_{01}) \times \text{Poi}(\mu_{++}\pi_{10}) \times \text{Poi}(\mu_{++}\pi_{11}). \tag{5.30}$$

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

$$y_i \overset{\text{ind}}{\sim} \text{Poi}(\mu_i); \quad \log \mu_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_{12} x_{i1} x_{i2}, \quad i = 1, \ldots, 4, \tag{5.31}$$

where

$$\beta_0 = \log \mu_{++} + \log \pi_{00}; \quad \beta_1 = \log \frac{\pi_{10}}{\pi_{00}}; \quad \beta_2 = \log \frac{\pi_{01}}{\pi_{00}}; \quad \beta_{12} = \log \frac{\pi_{11}\pi_{00}}{\pi_{10}\pi_{01}}. \tag{5.32}$$

Note that the independence hypothesis (5.29) reduces to the hypothesis $H_0 : \beta_{12} = 0$:

$$H_0 : \frac{\pi_{11}\pi_{00}}{\pi_{10}\pi_{01}} = 1 \quad \iff \quad H_0 : \beta_{12} = 0. \tag{5.33}$$

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. The normal equations state that the observed cell counts equal those that would have been expected under the null hypothesis. Using the formulation (5.27), we obtain

$$y_{jk} = \mathbb{E}[y_{jk}] = \widehat{\mu}_{++}\widehat{\pi}_{j+}\widehat{\pi}_{+k}, \tag{5.34}$$

so that

$$\widehat{\mu} = y_{++}; \quad \widehat{\mu}_{++}\widehat{\pi}_{j+} = y_{j+}; \quad \widehat{\mu}_{++}\widehat{\pi}_{+k} = y_{+k}, \tag{5.35}$$

from which it follows that

$$\widehat{\mu}_{jk} = \widehat{\mu}_{++}\widehat{\pi}_{j+}\widehat{\pi}_{+k} = y_{++}\frac{y_{j+}}{y_{++}}\frac{y_{+k}}{y_{++}} = \frac{y_{j+}y_{+k}}{y_{++}}. \tag{5.36}$$

Hence, we have

$$X^2 = \sum_{j,k=0}^{1} \frac{(y_{jk} - \widehat{\mu}_{jk})^2}{\widehat{\mu}_{jk}}. \tag{5.37}$$

Alternatively, we can use the likelihood ratio test, which gives

$$G^2 = \sum_{j,k=0}^{1} y_{jk} \log \frac{y_{jk}}{\widehat{\mu}_{jk}}. \tag{5.38}$$

5.2.4 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, we claim that the likelihood ratio statistic is the same for the Poisson and multinomial models. Indeed, let's write the Poisson likelihood as follows:

$$p_{\boldsymbol{\mu}}(\boldsymbol{y}) = p_{\mu_{++}}(y_{++} = n) p_{\boldsymbol{\pi}}(\boldsymbol{y} | y_{++} = n). \tag{5.39}$$

Note that the fitted parameter $\widehat{\mu}_{++}$ is the same under the null and alternative hypotheses: $\widehat{\mu}^0_{++} = \widehat{\mu}^1_{++}$, so we have

$$\frac{p_{\widehat{\boldsymbol{\mu}}^1}(\boldsymbol{y})}{p_{\widehat{\boldsymbol{\mu}}^0}(\boldsymbol{y})} = \frac{p_{\widehat{\mu}^1_{++}}(y_{++} = n) p_{\widehat{\boldsymbol{\pi}}^1}(\boldsymbol{y} | y_{++} = n)}{p_{\widehat{\mu}^0_{++}}(y_{++} = n) p_{\widehat{\boldsymbol{\pi}}^0}(\boldsymbol{y} | y_{++} = n)} = \frac{p_{\widehat{\boldsymbol{\pi}}^1}(\boldsymbol{y} | y_{++} = n)}{p_{\widehat{\boldsymbol{\pi}}^0}(\boldsymbol{y} | y_{++} = n)}. \tag{5.40}$$

The latter expression is the likelihood ratio statistic for the multinomial model. 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.

5.2.5 Equivalence among Poisson and logistic regressions

We've talked above 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. Note that under the Poisson model, we have

$$\text{logit } \mathbb{P}[x_2 = 1|x_1 = 0] = \log \frac{\pi_{01}}{\pi_{00}} = \beta_2 \tag{5.41}$$

and

$$\text{logit } \mathbb{P}[x_2 = 1|x_1 = 1] = \log \frac{\pi_{11}}{\pi_{10}} = \log \frac{\pi_{01}}{\pi_{00}} + \log \frac{\pi_{11}\pi_{00}}{\pi_{10}\pi_{01}} = \beta_2 + \beta_{12}. \tag{5.42}$$

In other words,

$$\text{logit } \mathbb{P}[x_2 = 1|x_1] = \beta_2 + \beta_{12}x_1. \tag{5.43}$$

Therefore, the β_{12} parameter for the Poisson regression (5.31) is the same as it is for the logistic regression (5.43).

5.2.6 Poisson models for $J \times K$ contingency tables

Suppose now that $x_1 \in \{1, \dots, J\}$ and $x_2 \in \{1, \dots, K\}$. 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

$$y_{jk} \overset{\text{ind}}{\sim} \text{Poi}(\mu_{jk}); \quad \log \mu_{jk} = \beta_0 + \beta_j^1 + \beta_k^2. \tag{5.44}$$

The Pearson statistic for this test is

$$\sum_{j=1}^{J} \sum_{k=1}^{K} \frac{(y_{ij} - \widehat{\mu}_{ij})^2}{\widehat{\mu}_{ij}}; \quad \widehat{\mu}_{ij} = \widehat{y}_{++} \frac{y_{i+}}{y_{++}} \frac{y_{+j}}{y_{++}}. \tag{5.45}$$

Like with the 2×2 case, the test is the same regardless if 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 created a lot of animosity between these two statisticians.

5.2.7 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 $x_1 \in \{1, \dots, J\}, x_2 \in \{1, \dots, K\}, x_3 \in \{1, \dots, L\}$, 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

$$y_{jkl} \overset{\text{ind}}{\sim} \text{Poi}(\mu_{jkl}), \tag{5.46}$$

where

$$\log \mu_{ijk} = \beta_0 + \beta_j^1 + \beta_k^2 + \beta_l^3 \quad \text{(mutual independence)}; \tag{5.47}$$

$$\log \mu_{ijk} = \beta_0 + \beta_j^1 + \beta_k^2 + \beta_l^3 + \beta_{kl}^{2,3} \quad \text{(joint independence)}; \tag{5.48}$$

$$\log \mu_{ijk} = \beta_0 + \beta_j^1 + \beta_k^2 + \beta_l^3 + \beta_{jk}^{1,2} + \beta_{jl}^{1,3} \quad \text{(mutual independence)}. \tag{5.49}$$

5.3 Negative binomial regression

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

$$\mathbb{E}[y|x_1] = \mathbb{E}[\mathbb{E}[y|x_1, x_2]|x_1] = \mathbb{E}[e^{\beta_0 + \beta_1 x_1 + \beta_2 x_2}|x_1] = e^{\beta_0 + \beta_1 x_1}\mathbb{E}[e^{\beta_2 x_2}] = e^{\beta_0' + \beta_1 x_1}. \tag{5.50}$$

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:

$$\text{Var}[y|x_1] = \mathbb{E}[\text{Var}[y|x_1, x_2]|x_1] + \text{Var}[\mathbb{E}[y|x_1, x_2]|x_1] = e^{\beta_0' + \beta_1 x_1} + e^{2(\beta_0' + \beta_1 x_1)}\text{Var}[e^{\beta_2 x_2}] > e^{\beta_0' + \beta_1 x_1} = \mathbb{E}[y|x_1]. \tag{5.51}$$

So indeed, the variance is larger than what we would have expected under the Poisson model.

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

$$f(\lambda; k, \mu) = \frac{(k/\mu)^k}{\Gamma(k)} e^{-k\lambda/\mu} \lambda^{k-1}, \tag{5.52}$$

with mean and variance given by

$$\mathbb{E}[\lambda] = \mu; \quad \text{Var}[\lambda] = \mu^2/k. \tag{5.53}$$

Therefore, it makes sense to augment the Poisson regression model as follows:

$$\lambda|\boldsymbol{x} \sim \Gamma(\mu, k), \quad \log \mu = \boldsymbol{x}^T \boldsymbol{\beta}, \quad y|\lambda \sim \text{Poi}(\lambda). \tag{5.54}$$

Negative binomial distribution. A simpler way to write the hierarchical model (5.54) would be to marginalize out λ. Doing so leaves us with a count distribution called the *negative binomial distribution*:

$$\lambda \sim \Gamma(\mu, k),\ y|\lambda \sim \text{Poi}(\lambda) \quad \Longrightarrow \quad y \sim \text{NegBin}(\mu, k). \tag{5.55}$$

The negative binomial probability mass function is

$$p(y;\mu,k) = \int_0^\infty \frac{(k/\mu)^k}{\Gamma(k)} e^{-k\lambda/\mu}\lambda^{k-1}e^{-\lambda}\frac{\lambda^y}{y!}d\lambda = \frac{\Gamma(y+k)}{\Gamma(k)\Gamma(y+1)}\left(\frac{\mu}{\mu+k}\right)^y\left(\frac{k}{\mu+k}\right)^k. \tag{5.56}$$

This random variable has mean and variance given by

$$\mathbb{E}[y] = \mathbb{E}[\lambda] = \mu \quad \text{and} \quad \text{Var}[y] = \mathbb{E}[\lambda] + \text{Var}[\lambda] = \mu + \frac{\mu^2}{k}. \tag{5.57}$$

Negative binomial as exponential dispersion model. If we treat k as known, then the negative binomial distribution is in the exponential family:

$$p(y;\mu,k) = \exp\left(y\log\frac{\mu}{\mu+k} - k\log\frac{\mu+k}{k}\right)\frac{\Gamma(y+k)}{\Gamma(k)\Gamma(y+1)}. \tag{5.58}$$

We can read off that

$$\theta = \log\frac{\mu}{\mu+k}, \quad \psi(\theta) = k\log\frac{\mu+k}{k} = -k\log(1-e^\theta). \tag{5.59}$$

This is a regular exponential family model, and not an exponential dispersion model. Given the extra parameter k controlling the variance, we may have been expecting to see an EDM. We can arrive at the EDM form by putting $1/k$ in the denominator:

$$p(y;\mu,k) = \exp\left(\frac{\frac{y}{k}\log\frac{\mu}{\mu+k} - \log\frac{\mu+k}{k}}{1/k}\right)\frac{\Gamma(y+k)}{\Gamma(k)\Gamma(y+1)}. \tag{5.60}$$

Note that the "normalized" variable y/k has the EDM distribution rather than the count variable y; this parallels our modeling of the binomial *proportion* (rather than the binomial count) as an EDM. We then see that y/k has the dispersion parameter $\phi = 1/k$. An alternate parameterization of the negative binomial model is via $\gamma = \phi = 1/k$. Here, γ is called the negative binomial *dispersion*.

Negative binomial regression. Let's revisit the hierarchical model (5.54), writing it more succinctly in terms of the negative binomial distribution:

$$y_i \overset{\text{ind}}{\sim} \text{NegBin}(\mu_i, \gamma), \quad \log\mu_i = \boldsymbol{x}^T\boldsymbol{\beta}. \tag{5.61}$$

Notice that we typically assume that all observations share the same dispersion parameter γ. Reading off from equation (5.59), 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 our first example of a non-canonical link!

Score and Fisher information. Recall from Chapter 4 that

$$\boldsymbol{U}(\boldsymbol{\beta}) = \frac{1}{\phi_0}\boldsymbol{X}^T\boldsymbol{M}\boldsymbol{W}(\boldsymbol{y}-\boldsymbol{\mu}) \quad \text{and} \quad \boldsymbol{I}(\boldsymbol{\beta}) = \frac{1}{\phi_0}\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X}, \tag{5.62}$$

where

$$\boldsymbol{W} \equiv \text{diag}\left(\frac{w_i}{V(\mu_i)(d\eta_i/d\mu_i)^2}\right) \quad \text{and} \quad \boldsymbol{M} \equiv \text{diag}\left(\frac{\partial\eta_i}{\partial\mu_i}\right). \tag{5.63}$$

In our case, we have

$$w_i = 1; \quad V(\mu_i) = \mu_i + \gamma\mu_i^2; \quad \frac{\partial\eta_i}{\partial\mu_i} = \frac{1}{\mu_i}. \tag{5.64}$$

Putting this together, we find that

$$\boldsymbol{W} = \text{diag}\left(\frac{\mu_i}{1+\gamma\mu_i}\right); \quad \boldsymbol{M} = \text{diag}\left(\frac{1}{1+\gamma\mu_i}\right). \tag{5.65}$$

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

$$\boldsymbol{X}^T\text{diag}\left(\frac{1}{1+\widehat{\gamma}\widehat{\mu}_i}\right)(\boldsymbol{y}-\widehat{\boldsymbol{\mu}}) = 0. \tag{5.66}$$

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.

Wald inference. Wald inference is based on

$$\widehat{\text{Var}}[\widehat{\boldsymbol{\beta}}] = (\boldsymbol{X}^T\widehat{\boldsymbol{W}}\boldsymbol{X})^{-1}, \quad \text{where} \quad \widehat{\boldsymbol{W}} = \text{diag}\left(\frac{\widehat{\mu}_i}{1+\widehat{\gamma}\widehat{\mu}_i}\right). \tag{5.67}$$

Likelihood ratio test inference. The negative binomial deviance is

$$D(\boldsymbol{y};\widehat{\boldsymbol{\mu}}) = 2\sum_{i=1}^{n}\left(y_i\log\frac{y_i}{\widehat{\mu}_i} - \left(y_i+\frac{1}{\widehat{\gamma}}\right)\log\frac{1+\widehat{\gamma}y_i}{1+\widehat{\gamma}\widehat{\mu}_i}\right). \tag{5.68}$$

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.

[1] Having said that, the dependency between $\widehat{\boldsymbol{\beta}}$ and $\widehat{\gamma}$ is weak, as the two are asymptotically independent parameters.

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

$$T^{\text{LRT}} \equiv 2(\ell^{\text{NB}} - \ell^{\text{Poi}}) \dot{\sim} \frac{1}{2}\delta_0 + \frac{1}{2}\chi^2_1.$$

Here, δ_0 is the delta mass at zero. Therefore, the likelihood ratio test for $H_0 : \gamma = 0$ rejects when

$$T^{\text{LRT}} > \chi^2_1(1 - 2\alpha). \tag{5.69}$$

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.

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.

5.4 R demo

5.4.1 Contingency table analysis

Let's take a look at the UC Berkeley admissions data:

```
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
## # ... with 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
```

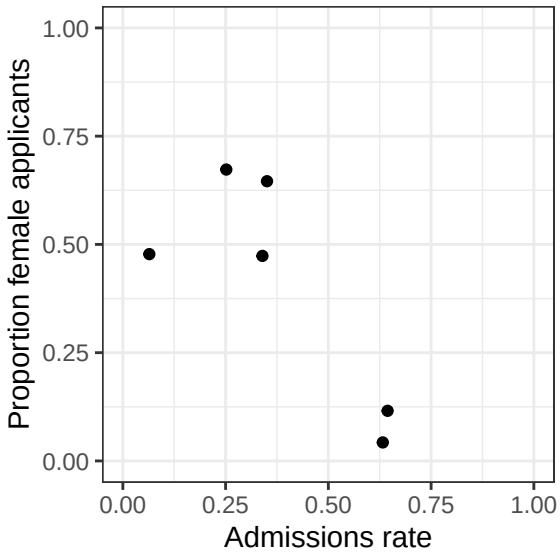
As a sanity check, let's run the Poisson regression underlying the chi-square test above.

```
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)
##
## Deviance Residuals:
## [1]  0  0  0  0
##
## 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: 4.4853e-14  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")
```



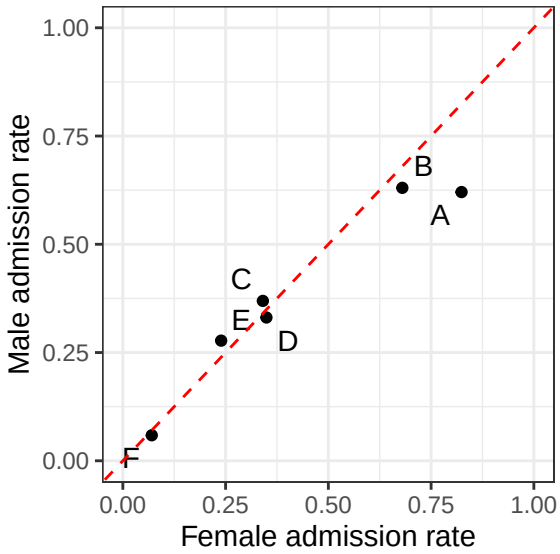
Indeed, it does seem that female applicants typically applied to more selective departments! This suggests that it is very important to control for department when evaluating the association between admissions and gender. To do this, we can run a test for conditional independence in the $J \times K \times L$ 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!

5.4.2 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
## # ... with 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)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -21.042   -9.176    0.418    9.051   47.170
##
## 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)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.62929  -1.02800  -0.54853   0.07234   2.71356
```

```
##
## 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
```

Chapter 6

Further Topics

Chapters 1-5 focused on estimation and inference in linear models and generalized linear models. In Chapter 6, we explore further topics: multiple testing (Section 6.1) and high-dimensional inference under the model-X assumption (Section 6.2).

6.1 Multiple testing

In this class, we have talked a lot about hypothesis testing, e.g. testing the significance of a coefficient in a (generalized) linear model. But frequently, there are multiple hypotheses we care about testing; let us denote these null hypotheses by $H_1, \dots, H_m$. After obtaining p-values for each null hypothesis—denote these by $p_1, \dots, p_m$—we may want to answer questions about this entire collection of hypotheses. In particular:

- Global testing: Test the *global null hypothesis* $H_0 : H_1 \cap \cdots \cap H_m$.
- Multiple testing: Find a subset $S \subseteq \{1, \dots, m\}$ of null hypotheses to reject so that the set S satisfies some notion of Type-I error.

We discuss global testing in Section 6.1.1 and multiple testing in Section 6.1.2.

6.1.1 Global testing

Global testing problem setup. Here we want to test whether *any* of the null hypotheses $H_1, \dots, H_m$ is false. For example, suppose that $H_j : \beta_j = 0$, where β_j are the coefficients in a GLM. Then, $H_0 : \beta_1 = \cdots = \beta_m = 0$. We recognize this hypothesis as something we would test using an F-test or, more generally, a likelihood ratio test. Here we are concerned with the more general problem of aggregating m p-values for individual hypotheses (whatever these hypotheses may be) into one p-value (i.e. one test) for the global null. A level-α test $\phi(p_1, \dots, p_m)$ of the global null must satisfy

$$\mathbb{E}_{H_0}[\phi(p_1, \dots, p_m)] \leq \alpha. \tag{6.1}$$

The multiplicity problem. A naive test would separately test the m hypotheses, and then reject if any are significant:

$$\phi_{\text{naive}}(p_1, \dots, p_m) = \mathbb{1}\left(p_j \leq \alpha \text{ for some } j = 1, \dots, m\right). \tag{6.2}$$

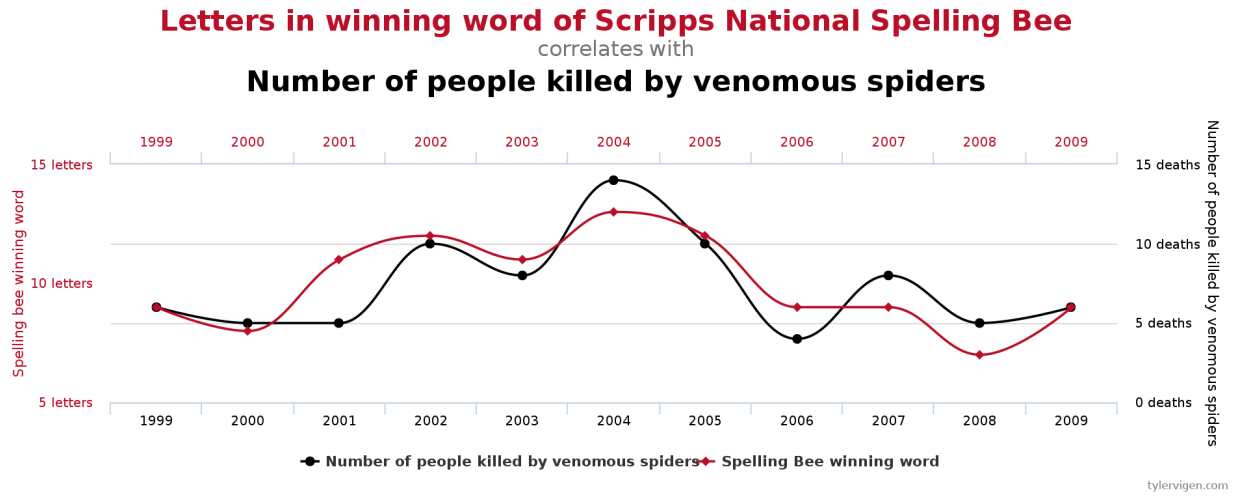


Figure 6.1: A spurious correlation resulting from data snooping.

This test does not control the Type-I error. In fact, assuming the input p-values are independent, we have

$$\mathbb{E}_{H_0}[\phi_{\text{naive}}(p_1, \dots, p_m)] = 1 - (1-\alpha)^m \to 1 \quad \text{as} \quad m \to \infty. \tag{6.3}$$

This is an illustration of *the multiplicity problem*: The more hypotheses we test, the more likely one of them is going to appear significant just by chance. This is related to data-snooping and the issue of selection bias. If we had chosen just one hypothesis a priori, then we can compare its p-value to the nominal level of α. If we chose the hypothesis by looking ("snooping") at the p-values of m hypotheses and choosing the most significant, we have incurred selection bias that must be corrected for. See Figure 6.1. There are several ways of properly correcting for this selection bias, i.e. several valid global tests in the sense of definition (6.1). Here we highlight two:

- Fisher combination test: Powerful against many weak signals.
- Bonferroni test: Powerful against few strong signals.

6.1.1.1 Fisher combination test

Suppose that $p_1, \dots, p_m$ are independent (though this is a strong assumption that is often violated). Then, the Fisher combination test is

$$\phi(p_1, \dots, p_m) \equiv \mathbb{1}\left(-2\sum_{j=1}^{m} \log p_j \geq Q_{1-\alpha}[\chi^2_{2m}]\right). \tag{6.4}$$

Type-I error control (6.1) is based on the fact that

$$\text{if } p_1, \dots, p_m \overset{\text{i.i.d.}}{\sim} U[0,1], \text{ then } -2\sum_{j=1}^{m} \log p_j \sim \chi^2_{2m}. \tag{6.5}$$

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

$$\text{if } X_1, \dots, X_m \overset{\text{i.i.d.}}{\sim} N(0,1), \text{ then } \sum_{j=1}^{m} X_j^2 \sim \chi^2_m. \tag{6.6}$$

It turns out that the tests based on equation (6.5) and (6.6) 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.

6.1.1.2 Bonferroni test

Instead of averaging the signal across p-values, we might want to find the *strongest* signal among the p-values. It makes sense that such a strategy would be powerful against sparse alternatives. We define the Bonferroni test via

$$\phi(p_1, \dots, p_m) \equiv \mathbb{1}\left(\min_{1 \le j \le m} p_j \le \alpha/m\right). \tag{6.7}$$

The Bonferroni global test rejects if any of the p-values crosses the *multiplicity-adjusted* or *Bonferroni-adjusted* significance threshold of α/m. The more hypotheses we test, the more stringent the significance threshold must be. We can verify the Type-I error control of the Bonferroni test via a union bound:

$$\mathbb{P}_{H_0}\left[\min_{1 \le j \le m} p_j \le \alpha/m\right] \le \sum_{j=1}^{m} \mathbb{P}_{H_0}\left[p_j \le \alpha/m\right] = m \cdot \alpha/m = \alpha. \tag{6.8}$$

Importantly, while the Fisher combination test is valid only for independent p-values, *the Bonferroni test is valid for arbitrary p-value dependency structures.* However, the Bonferroni bound derived above is tightest for independent p-values. For example, if the p-values are perfectly dependent, then no multiplicity correction is required at all.

6.1.2 Multiple testing

While global testing seeks to detect the presence of *any* signals, multiple testing seeks to *localize* these signals, i.e. find a subset S of the null hypotheses that are false. Let $\{1, \dots, m\} = \mathcal{H}_0 \cup \mathcal{H}_1$, where $\mathcal{H}_0, \mathcal{H}_1$ are the sets of null hypotheses that are true and false, respectively. Ideally, we would like to have $S = \mathcal{H}_1$, but of course we typically cannot do this. We design methods such outputting sets S satisfying satisfying some Type-I error control criterion, and compare their performance based on their power, e.g. as quantified by $\mathbb{E}[|S \cap \mathcal{H}_1|/|\mathcal{H}_1|]$. There are several Type-I error control criteria of interest, but we highlight the two most important ones:

- Family-wise error rate (FWER), defined

$$\text{FWER} \equiv \mathbb{P}[S \cap \mathcal{H}_0 \neq \varnothing]. \tag{6.9}$$

- False discovery rate (FDR), defined

$$\text{FDR} \equiv \mathbb{E}\left[\frac{|S \cap \mathcal{H}_0|}{|S|}\right], \quad \text{where} \quad \frac{0}{0} \equiv 0. \tag{6.10}$$

The random quantity $\frac{|S \cap \mathcal{H}_0|}{|S|}$ is called the *false discovery proportion* (FDP). Note that the FWER is a stricter error rate than the FDR. Controlling the FWER at level α implies that, with probability $1 - \alpha$, the set S contains no false discoveries at all. Controlling the FDR at level q means that, on average, at most a proportion q of the set S can be false discoveries. Many methods have been proposed to control each of these error rates, but we highlight one each.

6.1.2.1 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:

$$S \equiv \{j : p_j \leq \alpha/m\}. \tag{6.11}$$

Note that 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 for it to double as an FWER-controlling procedure. It is easy to verify that the Bonferroni procedure controls the FWER:

$$\mathbb{P}[S \cap \mathcal{H}_0 \neq \varnothing] = \mathbb{P}\left[\min_{j \in \mathcal{H}_0} p_j \leq \alpha/m\right] \leq \sum_{j \in \mathcal{H}_0} \mathbb{P}[p_j \leq \alpha/m] = \frac{|\mathcal{H}_0|}{m}\alpha \leq \alpha. \tag{6.12}$$

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.

6.1.2.2 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 (6.10) couples them together. Both the FDR criterion and the most popular FDR-controlling procedure were proposed by Benjamini and Hochberg in 1995.

Procedure. To define the BH procedure, consider thresholding the p-values at $t \in [0, 1]$. We would expect $\mathbb{E}[|\{j : p_j \leq t\} \cap \mathcal{H}_0|] = |\mathcal{H}_0|t$ false discoveries among $\{j : p_j \leq t\}$. Since $|\mathcal{H}_0|$ is unknown, we can bound it from above by mt. This leads to the FDP estimate

$$\widehat{\mathrm{FDP}}(t) \equiv \frac{mt}{|\{j : p_j \leq t\}|}. \tag{6.13}$$

The BH procedure is then defined via

$$S \equiv \{j : p_j \leq \widehat{t}\}, \quad \text{where} \quad \widehat{t} = \max\{t \in [0, 1] : \widehat{\mathrm{FDP}}(t) \leq q\}. \tag{6.14}$$

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: $\widehat{\mathrm{FDP}}(0) = \frac{0}{0} \equiv 0$.

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).

Proof. We have

$$\begin{aligned} \mathrm{FDR} = \mathbb{E}\left[\mathrm{FDP}(\widehat{t})\right] &= \mathbb{E}\left[\frac{|\{j \in \mathcal{H}_0 : p_j \leq \widehat{t}\}|}{|\{j : p_j \leq \widehat{t}\}|}\right] \\ &= \mathbb{E}\left[\frac{|\{j \in \mathcal{H}_0 : p_j \leq \widehat{t}\}|}{m\widehat{t}}\widehat{\mathrm{FDP}}(\widehat{t})\right] \leq q \cdot \mathbb{E}\left[\frac{|\{j \in \mathcal{H}_0 : p_j \leq \widehat{t}\}|}{m\widehat{t}}\right]. \end{aligned} \tag{6.15}$$

To prove that the last expectation is bounded above by 1, note that

$$M(t) \equiv \frac{|\{j \in \mathcal{H}_0 : p_j \leq t\}|}{mt} \tag{6.16}$$

is a backwards martingale with respect to the filtration

$$\mathcal{F}_t = \sigma(\{p_j : j \in \mathcal{H}_1\}, |\{j \in \mathcal{H}_0 : p_j \leq t'\}| \text{ for } t' \geq t), \tag{6.17}$$

with t running backwards from 1 to 0. Indeed, for $s < t$ we have

$$\mathbb{E}[M(s)|\mathcal{F}_t] = \mathbb{E}\left[\frac{|\{j \in \mathcal{H}_0 : p_j \leq s\}|}{ms}\middle| \mathcal{F}_t\right] = \frac{\frac{s}{t}|\{j \in \mathcal{H}_0 : p_j \leq t\}|}{ms} = \frac{|\{j \in \mathcal{H}_0 : p_j \leq t\}|}{mt} = M(t). \tag{6.18}$$

The threshold $\widehat{t}$ is a stopping time with respect to this filtration, so by the optional stopping theorem, we have

$$\mathbb{E}\left[\frac{|\{j \in \mathcal{H}_0 : p_j \leq \widehat{t}\}|}{m\widehat{t}}\right] = \mathbb{E}[M(\widehat{t})] \leq \mathbb{E}[M(1)] = \frac{|\mathcal{H}_0|}{m} \leq 1. \tag{6.19}$$

This completes the proof. □

FDR control under dependence. The BH procedure has empirically been shown to control the FDR for a wide variety of dependency structures besides independence. However, theoretical FDR control results for the BH procedure are available only for a few dependency structures. A notable example is a type of positive dependency called *positive regression dependence on a subset*, or PRDS. Benjamini and Yekutieli proved FDR control for BH under PRDS in 2001. This theoretical condition is somewhat hard to verify in practice, however. The simplest example of a set of PRDS p-values is when $\boldsymbol{x} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \in \mathbb{R}^m$ where $\boldsymbol{\Sigma}$ has all positive entries and the p-values are derived based on one-sided tests. Outside of this special case, there are few known instances of PRDS p-values.

6.2 High-dimensional inference under Model-X

All of the statistical inference done so far in this class was *low-dimensional*: we assumed that the number of predictors p was fixed and at most equal to the sample size n. However, some modern applications fall outside of this regime and therefore require new statistical methodology. We discuss here a line of work initiated by Candès, E., Fan, Y., Janson, L., & Lv, J. (2018). Panning for gold: 'model-X' knockoffs for high dimensional controlled variable selection. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 80(3), 551–577.

6.2.1 Motivation and problem statement

All statistical inference requires assumptions, and inherently difficult problems like high-dimensional inference require strong assumptions. One such assumption is the

$$\textit{Model-X assumption:} \text{ That the joint distribution of } (x_1, \ldots, x_p) \text{ is known.} \tag{6.20}$$

This assumption is in some sense the opposite of what we have been considering in this class so far: Usually we assume nothing about the joint distribution of covariates (we treat these as fixed anyways), and assume instead that $y|(x_1, \ldots, x_p)$ follows a generalized linear model. Notably, this assumption is stronger than correct specification of a parametric model for $(x_1, \ldots, x_p)$; it states that we know not only a model for this distribution but all of its parameters as well. Below we discuss the motivation for this assumption, and the inference problem that grows out of it.

Motivation: Genome-wide association studies (GWAS). In GWAS, $x_1, \dots, x_p \in \{0, 1, 2\}$ represent *genotypes* of an individual at p genomic locations. Suppose that humans typically have either an A or a T at genomic location j, where A is more common. Since we have two sets of chromosomes (one maternal and one paternal), the *genotype* at this location can either be AA, AT, or TT. The allele T is called the *minor allele* because it is less common, and x_j is defined as the number of minor alleles an individual has at location j: AA implies $x_j = 0$, AT implies $x_j = 1$, TT implies $x_j = 2$. We collect this genotype information at p genomic locations from each individual, as well as a response variable y, like disease status. The goal is to find the genomic locations whose genotypes are associated with the response. The nice thing in this application is that the joint distribution $(x_1, \dots, x_p)$ has been studied extensively in the field of population genetics, and is well-approximated by a *hidden Markov model*. This motivates the model-X assumption.

Problem statement. It turns out that if we have a model for the joint distribution of the predictors, we need not make any assumptions on the distribution of the response given the predictors. But this leaves us with the following awkward question: If we have no parametric model for the response, then what even are the hypotheses we are testing? Well, for each genomic location j, we are trying to test whether the genotype at that location is associated with the response, controlling for the genotypes at other genomic locations. Probabilistically, this may be written as:

$$H_{0j} : x_j \perp\!\!\!\perp y \mid \boldsymbol{x}_{\text{-}j}. \tag{6.21}$$

Under mild assumptions, this hypothesis turns out to coincide with the usual $H_{0j} : \beta_j = 0$ in the case when y does follow a GLM. The problem statement, then, is to test the hypotheses H_{0j} based on data

$$(x_{i1}, \dots, x_{ip}, y_i) \overset{\text{i.i.d.}}{\sim} F_{\boldsymbol{x},y}, \quad i = 1, \dots, n, \tag{6.22}$$

given knowledge of the distribution $F_{\boldsymbol{x}}$. Note that *regularized regression* methods such as the LASSO have been developed to get estimates of regression coefficients in high dimensions. However, the issue with these shrinkage-based estimation methodologies is that they do not come with inferential guarantees and therefore cannot provide valid tests of the conditional independence hypothesis (6.21). Under the model-X assumption, we can get around this roadblock.

6.2.2 Conditional randomization test

One idea is to view x_j as a treatment (though not necessarily binary) and $\boldsymbol{x}_{\text{-}j}$ as a set of covariates. The model-X assumption gives us knowledge of the *propensity function* $p(x_j|\boldsymbol{x}_{\text{-}j})$, i.e. the distribution of treatment assignments given the covariates. In the spirit of Fisher's randomization test (see Homework 5 Problem 1), we can build a null distribution for any test statistic $T(\boldsymbol{X}, \boldsymbol{y})$—e.g. a lasso coefficient—by *randomly reassigning the treatment x_j to each individual based on its covariates $\boldsymbol{x}_{\text{-}j}$*. More explicitly, let

$$\widetilde{x}_{ij} | \boldsymbol{X}, \boldsymbol{y} \overset{\text{ind}}{\sim} F_{x_j | \boldsymbol{x}_{\text{-}j} = \boldsymbol{x}_{i,\text{-}j}}. \tag{6.23}$$

Let $\widetilde{\boldsymbol{X}}$ be the matrix obtained by replacing the jth column in $\boldsymbol{X}$ with $\widetilde{\widetilde{x}}_{*j}$ as defined above. For a test statistic T, we then define the CRT p-value by comparing the test statistic's value on the original data with its distribution under resampling:

$$p_j^{\text{CRT}} \equiv \mathbb{P}[T(\widetilde{\boldsymbol{X}}, \boldsymbol{y}) \geq T(\boldsymbol{X}, \boldsymbol{y}) | \boldsymbol{X}, \boldsymbol{y}]. \tag{6.24}$$

In practice, we approximate this p-value by resampling a finite number B of instances $\widetilde{\boldsymbol{X}}^b$ and setting

$$\widehat{p}_j^{\text{CRT}} \equiv \frac{1}{B+1} \sum_{b=1}^{B} \mathbb{1}(T(\widetilde{\boldsymbol{X}}^b, \boldsymbol{y}) \geq T(\boldsymbol{X}, \boldsymbol{y})). \tag{6.25}$$

The CRT is a simple and elegant inferential framework that gives finite-sample valid p-values for high-dimensional inference. However, its adoption has been slowed by the computational cost of resampling.

6.2.3 Model-X knockoffs

An alternative to the CRT for model-X inference is *model-X knockoffs*. This methodology requires constructing a set of p new *knockoff* variables $(\widetilde{x}_1, \ldots, \widetilde{x}_p)$, whose joint distribution with the original variables satisfies the following exchangeability criterion:

$$\text{for each } j, \quad (x_j, \widetilde{x}_j) \stackrel{d}{=} (\widetilde{x}_j, x_j) \mid \boldsymbol{x}_{\text{-}j}, \widetilde{\boldsymbol{x}}_{\text{-}j}. \tag{6.26}$$

Knockoff variables are meant to serve as valid *negative controls* for the original variables: they have the same dependency structure but they have no association with the response y. Constructing such knockoff variables is a nontrivial endeavor that depends on the joint distribution of the original variables. If this can be done, then we can sample an entire knockoff matrix $\widetilde{\boldsymbol{X}}$, row by row. We then assess the significant of all $2p$ variables using test statistics $Z_1(\boldsymbol{X}, \widetilde{\boldsymbol{X}}, \boldsymbol{y}), \ldots, Z_p(\boldsymbol{X}, \widetilde{\boldsymbol{X}}, \boldsymbol{y}), \widetilde{Z}_1(\boldsymbol{X}, \widetilde{\boldsymbol{X}}, \boldsymbol{y}), \ldots, \widetilde{Z}_p(\boldsymbol{X}, \widetilde{\boldsymbol{X}}, \boldsymbol{y})$, constructed to ensure the following swap-equivariance property: swapping $\boldsymbol{X}_{*j}$ with $\widetilde{\boldsymbol{X}}_{*j}$ results in $Z_j(\boldsymbol{X}, \widetilde{\boldsymbol{X}}, \boldsymbol{y})$ swapping with $\widetilde{Z}_j(\boldsymbol{X}, \widetilde{\boldsymbol{X}}, \boldsymbol{y})$, while all the other test statistics stay the same. For example, consider running the LASSO of $\boldsymbol{y}$ on the *augmented design matrix* $[\boldsymbol{X}, \widetilde{\boldsymbol{X}}]$, and defining the Z_j's as the fitted coefficients for the corresponding variables. With these Z_j's in hand, the idea is to define the significance of the jth original variable by comparing the test statistics for itself and for its knockoff:

$$T_j(\boldsymbol{X}, \widetilde{\boldsymbol{X}}, \boldsymbol{y}) \equiv Z_j(\boldsymbol{X}, \widetilde{\boldsymbol{X}}, \boldsymbol{y}) - \widetilde{Z}_j(\boldsymbol{X}, \widetilde{\boldsymbol{X}}, \boldsymbol{y}). \tag{6.27}$$

Large values of T_j are evidence against H_{0j}. If the knockoffs are constructed correctly, then the test statistics T_j for null j can be shown to have symmetric distributions around zero. In other words, the original variable and its knockoff are equally likely to be more significant. Using this observation, a clever multiple testing algorithm called *Selective SeqStep* can be used to choose a cutoff $\widehat{t}$ for the test statistics in a way that provably controls the FDR at a pre-specified level q. Remarkably, this entire construction bypasses the construction of p-values!

6.2.4 Comparing CRT to MX knockoffs

There are pros and cons to both the CRT and MX knockoffs. Both procedures offer valid, finite-sample inference in high dimensions, which sets them apart from many other inferential methodologies. Both procedures require the model-X assumption, however, which may or may not be reasonable in a given application. MX knockoffs is the more popular methodology at this time, due to its computational speed. It can be used to carry out inference for all p hypotheses in "one shot", by running one big regularized regression on the augmented design matrix. It has been applied successfully to genome-wide association studies, using a hidden Markov model as the model for X. On the other hand, MX knockoffs is a randomized procedure, giving different results for different realizations of $\widetilde{\boldsymbol{X}}$. Furthermore, it does not provide p-values quantifying the significance of individual

predictors, which hinders the interpretability of its results. On the other hand, the CRT requires more computation than knockoffs, so it has been slower to be adopted in practice. But this procedure is not randomized in the same way that knockoffs is; with more computation its results can be arbitrarily "de-randomized." Furthermore, the CRT does have a p-value output, which facilitates easy interpretation and more flexibility for downstream multiple testing.