Statistical Methods II

Derek L. Sonderegger

2016-09-22

Contents

Chapter 1

Matrix Theory

Almost all of the calculations done in classical statistics require formulas with large number of subscripts and many different sums. In this chapter we will develop the mathematical machinery to write these formulas in a simple compact formula using *matrices*.

1.1 Types of Matrices

We will first introduce the idea behind a matrix and give several special types of matrices that we will encounter.

1.1.1 Scalars

To begin, we first define a *scalar*. A scalar is just a single number, either real or complex. The key is that a scalar is just a single number. For example, 6 is a scalar, as is -3. By convention, variable names for scalars will be lower case and not in bold typeface.

Examples could be $a = 5$, $b = \sqrt{3}$, or $\sigma = 2$.

1.1.2 Vectors

A vector is collection of scalars, arranged as a row or column. Our convention will be that a vector will be a lower cased letter but written in a bold type. In other branches of mathematics is common to put a bar over the variable name to denote that it is a vector, but in statistics, we have already used a bar to denote a mean.

Examples of column vectors could be

$$\boldsymbol{a} \quad = \quad \begin{bmatrix} 2 \\ -3 \\ 4 \end{bmatrix} \qquad \boldsymbol{b} = \begin{bmatrix} 2 \\ 8 \\ 3 \\ 4 \\ 1 \end{bmatrix}$$

and examples of row vectors are

$$\boldsymbol{c} = \begin{bmatrix} 8 & 10 & 43 & -22 \end{bmatrix}$$

$$\boldsymbol{d}=\left[\begin{array}{ccc}-1 & 5 & 2\end{array}\right]$$

To denote a specific entry in the vector, we will use a subscript. For example, the second element of $\boldsymbol{d}$ is $d_2=5$. Notice, that we do not bold this symbol because the second element of the vector is the scalar value 5.

1.1.3 Matrix

Just as a vector is a collection of scalars, a matrix can be viewed as a collection of vectors (all of the same length). We will denote matrices with bold capitalized letters. In general, I try to use letters at the end of the alphabet for matrices. Likewise, I try to use symmetric letters to denote symmetric matrices.

For example, the following is a matrix with two rows and three columns

$$\boldsymbol{W}=\left[\begin{array}{ccc}1 & 2 & 3\\ 4 & 5 & 6\end{array}\right]$$

and there is no requirement that the number of rows be equal, less than, or greater than the number of columns. In denoting the size of the matrix, we first refer to the number of rows and then the number of columns. Thus $\boldsymbol{W}$ is a 2×3 matrix and it sometimes is helpful to remind ourselves of this by writing $\boldsymbol{W}_{2\times 3}$.

To pick out a particular element of a matrix, I will again use a subscripting notation, always with the row number first and then column. Notice the notational shift to lowercase, non-bold font.

$$w_{1,2}=2 \qquad \text{and} \qquad w_{2,3}=6$$

There are times I will wish to refer to a particular row or column of a matrix and we will use the following notation

$$\boldsymbol{w}_{1,\cdot}=\left[\begin{array}{ccc}1 & 2 & 3\end{array}\right]$$

is the first row of the matrix $\boldsymbol{W}$. The second column of matrix $\boldsymbol{W}$ is

$$\boldsymbol{w}_{\cdot,2}=\left[\begin{array}{c}2\\ 5\end{array}\right]$$

1.1.4 Square Matrices

A square matrix is a matrix with the same number of rows as columns. The following are square

$$\boldsymbol{Z}=\left[\begin{array}{cc}3 & 6\\ 8 & 10\end{array}\right] \qquad \boldsymbol{X}=\left[\begin{array}{ccc}1 & 2 & 3\\ 2 & 1 & 2\\ 3 & 2 & 1\end{array}\right]$$

1.1.5 Symmetric Matrices

In statistics we are often interested in square matrices where the i,j element is the same as the j,i element. For example, $x_{1,2}=x_{2,1}$ in the above matrix $\boldsymbol{X}$.

Consider a matrix $\boldsymbol{D}$ that contains the distance from four towns to each of the other four towns. Let $d_{i,j}$ be the distance from town i to town j. It only makes sense that the distance doesn't matter which direction you are traveling, and we should therefore require that $d_{i,j}=d_{j,i}$.

In this example, it is the values $d_{i,i}$ represent the distance from a town to itself, which should be zero. It turns out that we are often interested in the terms $d_{i,i}$ and I will refer to those terms as the *main diagonal* of matrix $\boldsymbol{D}$.

Symmetric matrices play a large role in statistics because matrices that represent the covariances between random variables must be symmetric because $Cov\left(Y,Z\right)=Cov\left(Z,Y\right)$.

1.1.6 Diagonal Matrices

A square matrix that has zero entries in every location except the main diagonal is called a diagonal matrix. Here are two examples:

$$\boldsymbol{Q} = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 6 \end{bmatrix} \qquad \boldsymbol{R} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 3 \end{bmatrix}$$

Sometimes to make matrix more clear, I will replace the 0 with a dot to emphasize the non-zero components.

$$\boldsymbol{R} = \begin{bmatrix} 1 & \cdot & \cdot & \cdot \\ \cdot & 2 & \cdot & \cdot \\ \cdot & \cdot & 2 & \cdot \\ \cdot & \cdot & \cdot & 3 \end{bmatrix}$$

1.1.7 Identity Matrices

A diagonal matrix with main diagonal values exactly 1 is called the identity matrix. The 3×3 identity matrix is denoted I_3.

$$\boldsymbol{I}_3 = \begin{bmatrix} 1 & \cdot & \cdot \\ \cdot & 1 & \cdot \\ \cdot & \cdot & 1 \end{bmatrix}$$

1.2 Operations on Matrices

1.2.1 Transpose

The simplest operation on a square matrix matrix is called *transpose*. It is defined as $\boldsymbol{M} = \boldsymbol{W}^T$ if and only if $m_{i,j} = w_{j,i}$.

$$\boldsymbol{Z} = \begin{bmatrix} 1 & 6 \\ 8 & 3 \end{bmatrix} \qquad \boldsymbol{Z}^T = \begin{bmatrix} 1 & 8 \\ 6 & 3 \end{bmatrix}$$

$$\boldsymbol{M} = \begin{bmatrix} 3 & 1 & 2 \\ 9 & 4 & 5 \\ 8 & 7 & 6 \end{bmatrix} \qquad \boldsymbol{M}^T = \begin{bmatrix} 3 & 9 & 8 \\ 1 & 4 & 7 \\ 2 & 5 & 6 \end{bmatrix}$$

We can think of this as swapping all elements about the main diagonal. Alternatively we could think about the transpose as making the first row become the first column, the second row become the second column, etc. In this fashion we could define the transpose of a non-square matrix.

$$\boldsymbol{W} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

$$\boldsymbol{W}^T = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$$

1.2.2 Addition and Subtraction

Addition and subtraction are performed *element-wise*. This means that two matrices or vectors can only be added or subtracted if their dimensions match.

$$\begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} + \begin{bmatrix} 5 \\ 6 \\ 7 \\ 8 \end{bmatrix} = \begin{bmatrix} 6 \\ 8 \\ 10 \\ 12 \end{bmatrix}$$

$$\begin{bmatrix} 5 & 8 \\ 2 & 4 \\ 11 & 15 \end{bmatrix} - \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & -6 \end{bmatrix} = \begin{bmatrix} 4 & 6 \\ -1 & 0 \\ 6 & 21 \end{bmatrix}$$

1.2.3 Multiplication

Multiplication is the operation that is vastly different for matrices and vectors than it is for scalars. There is a great deal of mathematical theory that suggests a useful way to define multiplication. What is presented below is referred to as the *dot-product* of vectors in calculus, and is referred to as the standard *inner-product* in linear algebra.

1.2.4 Vector Multiplication

We first define multiplication for a row and column vector. For this multiplication to be defined, both vectors must be the same length. The product is the sum of the element-wise multiplications.

$$\begin{bmatrix} 1 & 2 & 3 & 4 \end{bmatrix} \begin{bmatrix} 5 \\ 6 \\ 7 \\ 8 \end{bmatrix} = (1 \cdot 5) + (2 \cdot 6) + (3 \cdot 7) + (4 \cdot 8) = 5 + 12 + 21 + 32 = 70$$

1.2.5 Matrix Multiplication

Matrix multiplication is just a sequence of vector multiplications. If $\boldsymbol{X}$ is a $m \times n$ matrix and $\boldsymbol{W}$ is $n \times p$ matrix then $\boldsymbol{Z} = \boldsymbol{X}\boldsymbol{W}$ is a $m \times p$ matrix where $z_{i,j} = \boldsymbol{x}_{i,\cdot}\boldsymbol{w}_{\cdot,j}$ where $\boldsymbol{x}_{i,\cdot}$ is the ith column of $\boldsymbol{X}$ and $\boldsymbol{w}_{\cdot,j}$ is the jth column of $\boldsymbol{W}$. For example, let

$$\boldsymbol{X} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{bmatrix} \qquad \boldsymbol{W} = \begin{bmatrix} 13 & 14 \\ 15 & 16 \\ 17 & 18 \\ 19 & 20 \end{bmatrix}$$

so $\boldsymbol{X}$ is 3×4 (which we remind ourselves by adding a 3×4 subscript to $\boldsymbol{X}$ as $\boldsymbol{X}_{3\times4}$) and $\boldsymbol{W}$ is $\boldsymbol{W}_{4\times2}$. Because the *inner* dimensions match for this multiplication, then $\boldsymbol{Z}_{3\times2} = \boldsymbol{X}_{3\times4}\boldsymbol{W}_{4\times2}$ is defined where

$$\begin{aligned} z_{1,1} &= \boldsymbol{x}_{1,\cdot}\boldsymbol{w}_{\cdot,1} \\ &= (1 \cdot 13) + (2 \cdot 15) + (3 \cdot 17) + (4 \cdot 19) = 170 \end{aligned}$$

and similarly

$$\begin{aligned} z_{2,1} &= \boldsymbol{x}_{2,\cdot}\boldsymbol{w}_{\cdot,1} \\ &= (5 \cdot 13) + (6 \cdot 15) + (7 \cdot 17) + (8 \cdot 19) = 426 \end{aligned}$$

so that

$$\boldsymbol{Z} = \begin{bmatrix} 170 & 180 \\ 426 & 452 \\ 682 & 724 \end{bmatrix}$$

For another example, we note that

$$\begin{aligned} \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 2 & 2 \\ 1 & 2 \end{bmatrix} &= \begin{bmatrix} 1+4+3 & 2+4+6 \\ 2+6+4 & 4+6+8 \end{bmatrix} \\ &= \begin{bmatrix} 8 & 12 \\ 12 & 18 \end{bmatrix} \end{aligned}$$

Notice that this definition of multiplication means that the order matters. Above, we calculated $\boldsymbol{X}_{3\times4}\boldsymbol{W}_{4\times2}$ but we cannot reverse the order because the inner dimensions do not match up.

1.2.6 Scalar times a Matrix

Strictly speaking, we are not allowed to multiply a matrix by a scalar because the dimensions do not match. However, it is often notationally convenient. So we define $a\boldsymbol{X}$ to be the *element-wise* multiplication of each element of $\boldsymbol{X}$ by the scalar a. Because this is just a notational convenience, the mathematical theory about inner-products does not apply to this operation.

$$5 \begin{bmatrix} 4 & 5 \\ 7 & 6 \\ 9 & 10 \end{bmatrix} = \begin{bmatrix} 20 & 25 \\ 35 & 30 \\ 45 & 50 \end{bmatrix}$$

Because of this definition, it is clear that $a\boldsymbol{X} = \boldsymbol{X}a$ and the order does not matter. Thus when mixing scalar multiplication with matrices, it is acceptable to reorder scalars, but not matrices.

1.2.7 Determinant

The determinant is defined only for square matrices and can be thought of as the matrix equivalent of the absolute value or magnitude (i.e. $|-6| = 6$). The determinant gives a measure of the multi-dimensional size of a matrix (say the matrix $\boldsymbol{A}$) and as such is denoted $\det(\boldsymbol{A})$ or $|\boldsymbol{A}|$. Generally this is a very tedious thing to calculate by hand and for completeness sake, we will give a definition and small examples.

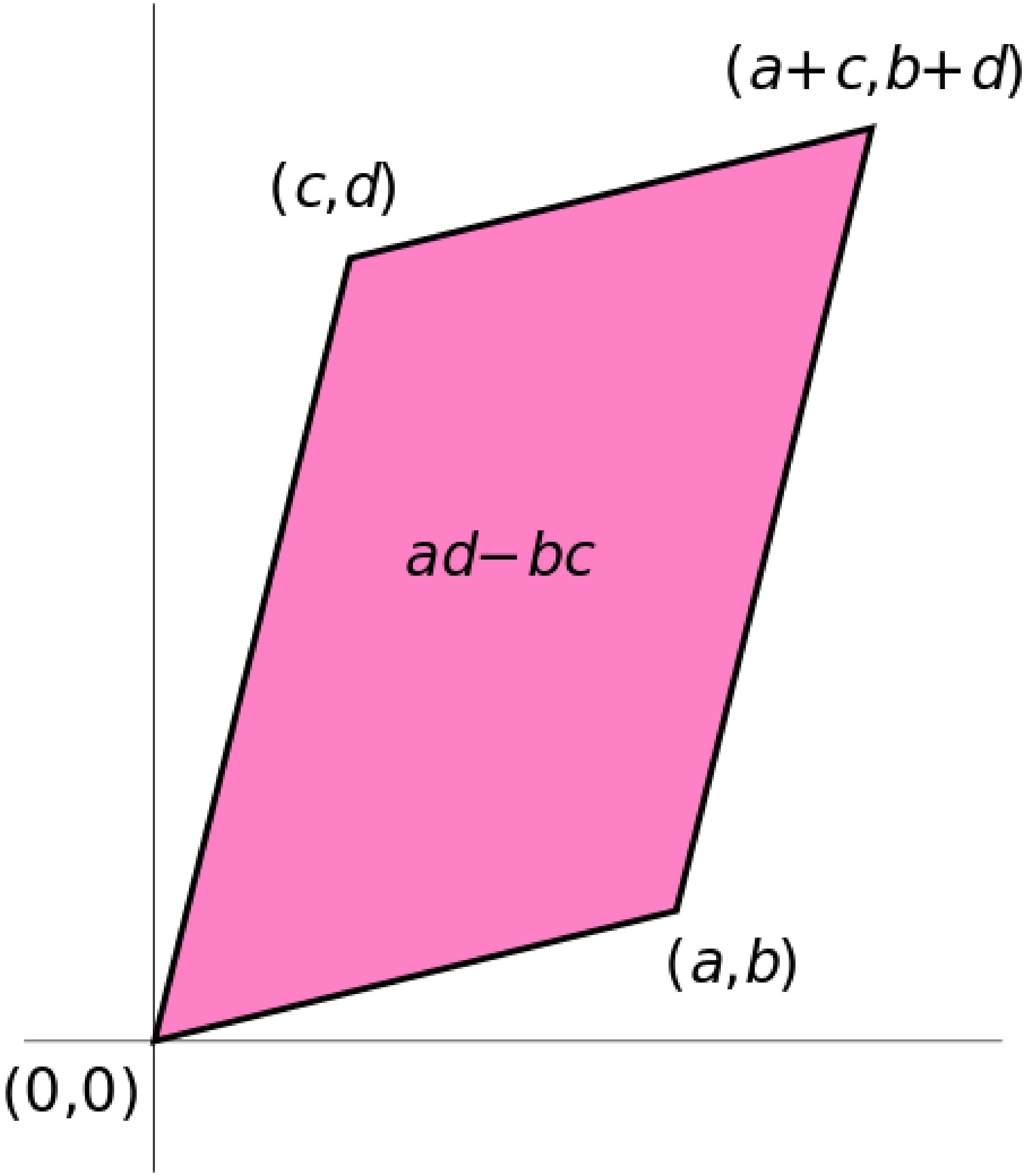
For a 2×2 matrix

$$\begin{vmatrix} a & c \\ b & d \end{vmatrix} = ad - cb$$

So a simple example of a determinant is

$$\begin{vmatrix} 5 & 2 \\ 3 & 10 \end{vmatrix} = 50 - 6 = 44$$

The determinant can be thought of as the area of the parallelogram created by the row or column vectors of the matrix.


$(a+c,b+d)$
(c,d)
$ad-bc$
(a,b)
$(0,0)$

1.2.8 Inverse

In regular algebra, we are often interested in solving equations such as

$$5x = 15$$

for x. To do so, we multiply each side of the equation by the inverse of 5, which is $1/5$.

$$\begin{aligned} 5x &= 15 \\ \frac{1}{5} \cdot 5 \cdot x &= \frac{1}{5} \cdot 15 \\ 1 \cdot x &= 3 \\ x &= 3 \end{aligned}$$

For scalars, we know that the inverse of scalar a is the value that when multiplied by a is 1. That is we see to find a^{-1} such that $aa^{-1} = 1$.

In the matrix case, I am interested in finding $\boldsymbol{A}^{-1}$ such that $\boldsymbol{A}^{-1}\boldsymbol{A} = \boldsymbol{I}$ and $\boldsymbol{A}\boldsymbol{A}^{-1} = \boldsymbol{I}$. For both of these multiplications to be defined, $\boldsymbol{A}$ must be a square matrix and so the inverse is only defined for square matrices.

For a 2×2 matrix

$$\boldsymbol{W} = \left[\begin{array}{cc} a & b \\ c & d \end{array} \right]$$

the inverse is given by:

$$\boldsymbol{W}^{-1} = \frac{1}{\det \boldsymbol{W}} \left[\begin{array}{cc} d & -b \\ -c & a \end{array} \right]$$

For example, suppose

$$\boldsymbol{W} = \left[\begin{array}{cc} 1 & 2 \\ 5 & 3 \end{array} \right]$$

then $\det W = 3 - 10 = -7$ and

$$\begin{aligned} \boldsymbol{W}^{-1} &= \frac{1}{-7} \left[\begin{array}{cc} 3 & -2 \\ -5 & 1 \end{array} \right] \\ &= \left[\begin{array}{cc} -\frac{3}{7} & \frac{2}{7} \\ \frac{5}{7} & -\frac{1}{7} \end{array} \right] \end{aligned}$$

and thus

$$\begin{aligned} \boldsymbol{W}\boldsymbol{W}^{-1} &= \left[\begin{array}{cc} 1 & 2 \\ 5 & 3 \end{array} \right] \left[\begin{array}{cc} -\frac{3}{7} & \frac{2}{7} \\ \frac{5}{7} & -\frac{1}{7} \end{array} \right] \\ &= \left[\begin{array}{cc} -\frac{3}{7} + \frac{10}{7} & \frac{2}{7} - \frac{2}{7} \\ & \\ -\frac{15}{7} + \frac{15}{7} & \frac{10}{7} - \frac{3}{7} \end{array} \right] \\ &= \left[\begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array} \right] = \boldsymbol{I}_2 \end{aligned}$$

Not every square matrix has an inverse. If the determinant of the matrix (which we think of as some measure of the magnitude or *size* of the matrix) is zero, then the formula would require us to divide by zero. Just as we cannot find the inverse of zero (i.e. solve $0x = 1$ for x), a matrix with zero determinate is said to have no inverse.

1.3 Exercises

Consider the following matrices:

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 6 & 5 & 4 \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} 6 & 4 & 3 \\ 8 & 7 & 6 \end{bmatrix} \qquad \mathbf{c} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \qquad \mathbf{d} = \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix} \qquad \mathbf{E} = \begin{bmatrix} 1 & 2 \\ 2 & 6 \end{bmatrix}$$

1. Find $\mathbf{Bc}$
2. Find $\mathbf{AB}^T$
3. Find $\mathbf{c}^T\mathbf{d}$
4. Find $\mathbf{cd}^T$
5. Confirm that $\mathbf{E}^{-1} = \begin{bmatrix} 3 & -1 \\ -1 & 1/2 \end{bmatrix}$ is the inverse of $\mathbf{E}$ by calculating $\mathbf{EE}^{-1} = \mathbf{I}$.

Chapter 2

Parameter Estimation

We have previously looked at ANOVA and regression models and, in many ways, they felt very similar. In this chapter we will introduce the theory that allows us to understand both models as a particular flavor of a larger class of models known as *linear models.*

First we clarify what a linear model is. A linear model is a model where the data (which we will denote using roman letters as $\boldsymbol{x}$ and $\boldsymbol{y}$) and parameters of interest (which we denote using greek letters such as $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$) interact only via addition and multiplication. The following are linear models:

Model	Formula
ANOVA	$y_{ij} = \mu + \tau_i + \epsilon_{ij}$
Simple Regression	$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$
Quadratic Term	$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \epsilon_i$
General Regression	$y_i = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \cdots + \beta_p x_{i,p} + \epsilon_i$

Notice in the Quadratic model, the square is not a parameter and we can consider x_i^2 as just another column of data. This leads to the second example of multiple regression where we just add more slopes for other covariates where the p^{th} covariate is denoted $\boldsymbol{x}_{\cdot,p}$ and might be some transformation (such as x^2 or $\log x$) of another column of data. The critical point is that the transformation to the data $\boldsymbol{x}$ does not depend on a parameter. Thus the following is *not* a linear model

$$y_i = \beta_0 + \beta_1 x_i^{\alpha} + \epsilon_i$$

2.1 Simple Regression

We would like to represent all linear models in a similar compact matrix representation. This will allow us to make the transition between simple and multiple regression (and ANCOVA) painlessly.

To begin, we think about how to write the simple regression model using matrices and vectors that correspond the the data and the parameters. Notice we have

$$\begin{aligned}
y_1 &= \beta_0 + \beta_1 x_1 + \epsilon_1 \\
y_2 &= \beta_0 + \beta_1 x_2 + \epsilon_2 \\
y_3 &= \beta_0 + \beta_1 x_3 + \epsilon_3 \\
&\vdots \\
y_{n-1} &= \beta_0 + \beta_1 x_{n-1} + \epsilon_{n-1} \\
y_n &= \beta_0 + \beta_1 x_n + \epsilon_n
\end{aligned}$$

where, as usual, $\epsilon_i \overset{iid}{\sim} N\left(0, \sigma^2\right)$. These equations can be written using matrices as

$$\underbrace{\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{n-1} \\ y_n \end{bmatrix}}_{\boldsymbol{y}} = \underbrace{\begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & x_3 \\ \vdots & \vdots \\ 1 & x_{n-1} \\ 1 & x_n \end{bmatrix}}_{\boldsymbol{X}} \underbrace{\begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}}_{\boldsymbol{\beta}} + \underbrace{\begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \vdots \\ \epsilon_{n-1} \\ \epsilon_n \end{bmatrix}}_{\boldsymbol{\epsilon}}$$

and we compactly write the model as

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

where $\boldsymbol{X}$ is referred to as the *design matrix* and $\boldsymbol{\beta}$ is the vector of *location parameters* we are interested in estimating.

2.1.1 Estimation of Location Paramters

Our next goal is to find the best estimate of $\boldsymbol{\beta}$ given the data. To justify the formula, consider the case where there is no error terms (i.e. $\epsilon_i = 0$ for all i). Thus we have

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta}$$

and our goal is to solve for $\boldsymbol{\beta}$. To do this, we must use a matrix inverse, but since inverses only exist for square matrices, we pre-multiple by $\boldsymbol{X}^T$ (notice that $\boldsymbol{X}^T\boldsymbol{X}$ is a symmetric 2×2 matrix).

$$\boldsymbol{X}^T\boldsymbol{y} = \boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\beta}$$

and then pre-multiply by $\left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}$.

$$\begin{aligned}
\left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{y} &= \left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\beta} \\
\left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{y} &= \boldsymbol{\beta}
\end{aligned}$$

This exercise suggests that $\left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{y}$ is a good place to start when looking for the maximum-likelihood estimator for $\boldsymbol{\beta}$. It turns out that this quantity is in fact the maximum-likelihood estimator (and equivalently minimizes the sum-of-squared error). Therefore we will use it as our estimate of $\boldsymbol{\beta}$.

$$\hat{\boldsymbol{\beta}} = \left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

2.1.2 Estimation of Variance Parameter

Recall our model is

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

where $\epsilon_i \overset{iid}{\sim} N\left(0, \sigma^2\right)$.

Using our estimates $\hat{\boldsymbol{\beta}}$ we can obtain predicted values for the regression line at any x-value. In particular we can find the predicted value for each x_i value in our dataset.

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$$

Using matrix notation, I would write $\hat{\boldsymbol{y}} = \boldsymbol{X}\hat{\boldsymbol{\beta}}$.

As usual we will find estimates of the noise terms (which we will call residuals or errors) via

$$\begin{aligned} \hat{\epsilon}_i &= y_i - \hat{y}_i \\ &= y_i - \left(\hat{\beta}_0 + \hat{\beta}_1 x_i\right) \end{aligned}$$

Writing $\hat{\boldsymbol{y}}$ in matrix terms we have

$$\begin{aligned} \hat{\boldsymbol{y}} &= \boldsymbol{X}\hat{\boldsymbol{\beta}} \\ &= \boldsymbol{X}\left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{y} \\ &= \boldsymbol{H}\boldsymbol{y} \end{aligned}$$

where $\boldsymbol{H} = \boldsymbol{X}\left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T$ is often called the *hat-matrix* because it takes y to $\hat{y}$ and has many interesting theoretical properties.[1]

We can now estimate the error terms via

$$\begin{aligned} \hat{\boldsymbol{\epsilon}} &= \boldsymbol{y} - \hat{\boldsymbol{y}} \\ &= \boldsymbol{y} - \boldsymbol{H}\boldsymbol{y} \\ &= \left(\boldsymbol{I}_n - \boldsymbol{H}\right)\boldsymbol{y} \end{aligned}$$

As usual we estimate σ^2 using the mean-squared error

$$\begin{aligned} \hat{\sigma}^2 &= \frac{1}{n-2}\sum_{i=1}^{n}\hat{\epsilon}_i^2 \\ &= \frac{1}{n-2}\hat{\boldsymbol{\epsilon}}^T\hat{\boldsymbol{\epsilon}} \end{aligned}$$

In the general linear model case where $\boldsymbol{\beta}$ has p elements (and thus we have $n-p$ degrees of freedom), the formula is

$$\hat{\sigma}^2 = \frac{1}{n-p}\hat{\boldsymbol{\epsilon}}^T\hat{\boldsymbol{\epsilon}}$$

[1] Mathematically, $\boldsymbol{H}$ is the projection matrix that takes a vector in n-dimensional space and projects it onto a p-dimension subspace spanned by the vectors in $\boldsymbol{X}$. Projection matrices have many useful properties and much of the theory of linear models utilizes $\boldsymbol{H}$.

2.1.3 Expectation and variance of a random vector

Just as we needed to derive the expected value and variance of $\bar{x}$ in the previous semester, we must now do the same for $\hat{\boldsymbol{\beta}}$. But to do this, we need some properties of expectations and variances.

In the following, let $\boldsymbol{A}_{n\times p}$ and $\boldsymbol{b}_{n\times 1}$ be constants and $\boldsymbol{\epsilon}_{n\times 1}$ be a random vector.

Expectations are very similar to the scalar case where

$$E\left[\boldsymbol{\epsilon}\right]=\left[\begin{array}{c} E\left[\epsilon_{1}\right]\\ E\left[\epsilon_{2}\right]\\ \vdots\\ E\left[\epsilon_{n}\right] \end{array}\right]$$

and any constants are pulled through the expectation

$$E\left[\boldsymbol{A}^{T}\boldsymbol{\epsilon}+\boldsymbol{b}\right]=\boldsymbol{A}^{T}\,E\left[\boldsymbol{\epsilon}\right]+\boldsymbol{b}$$

Variances are a little different. The variance of the vector $\boldsymbol{\epsilon}$ is

$$Var\left(\boldsymbol{\epsilon}\right)=\left[\begin{array}{cccc} Var\left(\epsilon_{1}\right) & Cov\left(\epsilon_{1},\epsilon_{2}\right) & \dots & Cov\left(\epsilon_{1},\epsilon_{n}\right)\\ Cov\left(\epsilon_{2},\epsilon_{1}\right) & Var\left(\epsilon_{2}\right) & \dots & Cov\left(\epsilon_{2},\epsilon_{n}\right)\\ \vdots & \vdots & \ddots & \vdots\\ Cov\left(\epsilon_{n},\epsilon_{1}\right) & Cov\left(\epsilon_{n},\epsilon_{2}\right) & \dots & Var\left(\epsilon_{1}\right) \end{array}\right]$$

and additive constants are ignored, but multiplicative constants are pulled out as follows:

$$Var\left(\boldsymbol{A}^{T}\boldsymbol{\epsilon}+\boldsymbol{b}\right)=Var\left(\boldsymbol{A}^{T}\boldsymbol{\epsilon}\right)=\boldsymbol{A}^{T}\,Var\left(\boldsymbol{\epsilon}\right)\,\boldsymbol{A}$$

2.1.4 Variance of Location Parameters

We next derive the sampling variance of our estimator $\hat{\boldsymbol{\beta}}$ by first noting that $\boldsymbol{X}$ and $\boldsymbol{\beta}$ are constants and therefore

$$\begin{aligned} Var\left(\boldsymbol{y}\right) &= Var\left(\boldsymbol{X\beta}+\boldsymbol{\epsilon}\right)\\ &= Var\left(\boldsymbol{\epsilon}\right)\\ &= \sigma^{2}\boldsymbol{I}_{n} \end{aligned}$$

because the error terms are independent and therefore $Cov\left(\epsilon_{i},\epsilon_{j}\right)=0$ when $i\neq j$ and $Var\left(\epsilon_{i}\right)=\sigma^{2}$. Recalling that constants come out of the variance operator as the constant *squared,*

$$\begin{aligned} Var\left(\hat{\boldsymbol{\beta}}\right) &= Var\left(\left(\boldsymbol{X}^{T}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^{T}\boldsymbol{y}\right)\\ &= \left(\boldsymbol{X}^{T}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^{T}\,Var\left(\boldsymbol{y}\right)\,\boldsymbol{X}\left(\boldsymbol{X}^{T}\boldsymbol{X}\right)^{-1}\\ &= \left(\boldsymbol{X}^{T}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^{T}\,\sigma^{2}\boldsymbol{I}_{n}\,\boldsymbol{X}\left(\boldsymbol{X}^{T}\boldsymbol{X}\right)^{-1}\\ &= \sigma^{2}\left(\boldsymbol{X}^{T}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^{T}\boldsymbol{X}\left(\boldsymbol{X}^{T}\boldsymbol{X}\right)^{-1}\\ &= \sigma^{2}\left(\boldsymbol{X}^{T}\boldsymbol{X}\right)^{-1} \end{aligned}$$

Using this, the standard error (i.e. the estimated standard deviation) of $\hat{\beta}_j$ (for any j in $1,\dots,p$) is

$$StdErr\left(\hat{\beta}_j\right)=\sqrt{\hat{\sigma}^2\left[\left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\right]_{jj}}$$

2.1.5 Confidence intervals and hypothesis tests

We can now state the general method of creating confidence intervals and perform hypothesis tests for any element of $\boldsymbol{\beta}$.

The confidence interval formula is (as usual)

$$\hat{\beta}_j \pm t_{n-p}^{1-\alpha/2}\, StdErr\left(\hat{\beta}_j\right)$$

and a test statistic for testing $H_0 : \beta_j = 0$ versus $H_a : \beta_j \neq 0$ is

$$t_{n-p}=\frac{\hat{\beta}_j-0}{StdErr\left(\hat{\beta}_j\right)}$$

2.1.6 Summary of pertinent results

- $\hat{\boldsymbol{\beta}}=\left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{y}$ is the unbiased maximum-likelihood estimator of $\boldsymbol{\beta}$.
- The Central Limit Theorem applies to each element of $\boldsymbol{\beta}$. That is, as $n\to\infty$, the distribution of $\hat{\beta}_j \to N\left(\beta_j, \left[\sigma^2\left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\right]_{jj}\right)$.
- The error terms can be calculated via

$$\begin{aligned}\hat{\boldsymbol{y}} &= \boldsymbol{X}\hat{\boldsymbol{\beta}}\\ \hat{\boldsymbol{\epsilon}} &= \boldsymbol{y}-\hat{\boldsymbol{y}}\end{aligned}$$

- The estimate of σ^2 is

$$\hat{\sigma}^2=\frac{1}{n-p}\,\hat{\boldsymbol{\epsilon}}^T\hat{\boldsymbol{\epsilon}}$$

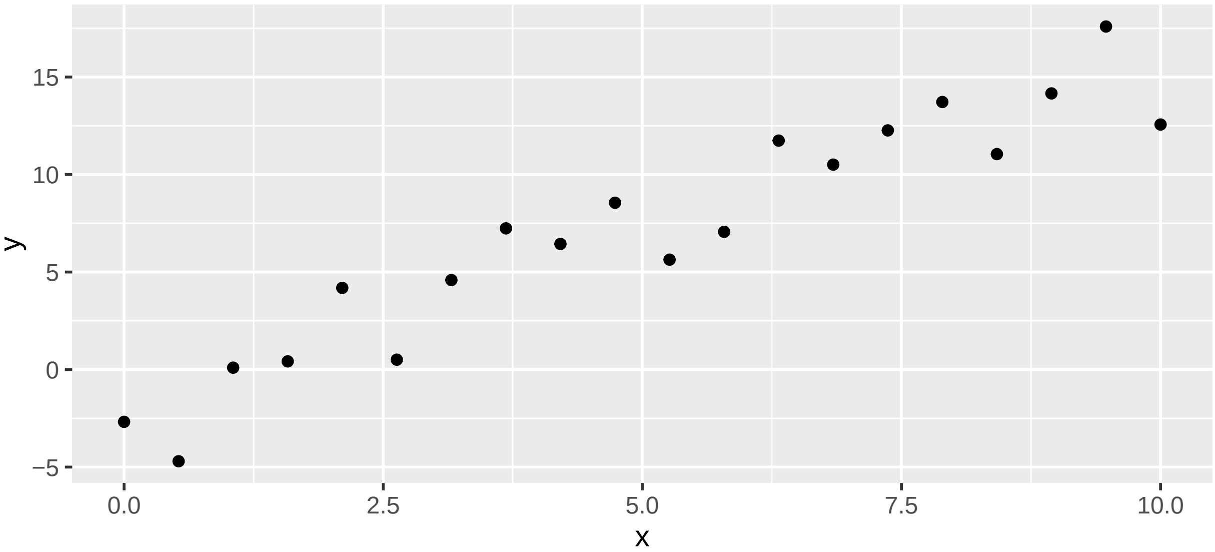
- The standard error (i.e. the estimated standard deviation) of $\hat{\beta}_j$ (for any j in $1,\dots,p$) is

$$StdErr\left(\hat{\beta}_j\right)=\sqrt{\hat{\sigma}^2\left[\left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\right]_{jj}}$$

2.1.7 An example in R

Here we will work an example in R and see the calculations. Consider the following data:

```r
library(ggplot2)
n <- 20
x <- seq(0,10, length=n)
y <- -3 + 2*x + rnorm(n, sd=2)
my.data <- data.frame(x=x, y=y)
ggplot(my.data) + geom_point(aes(x=x,y=y))
```



First we must create the design matrix $\boldsymbol{X}$. Recall

$$\boldsymbol{X}=\left[\begin{array}{cc}1 & x_1\\ 1 & x_2\\ 1 & x_3\\ \vdots & \vdots\\ 1 & x_{n-1}\\ 1 & x_n\end{array}\right]$$

and can be created in R via the following:

```
X <- cbind( rep(1,n), x)
```

Given $\boldsymbol{X}$ and $\boldsymbol{y}$ we can calculate

$$\hat{\boldsymbol{\beta}}=\left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

in R using the following code:

```
XtXinv <- solve( t(X) %*% X )
beta.hat <- XtXinv %*% t(X) %*% y
beta.hat
```

```
##         [,1]
##   -2.175664
## x  1.844463
```

Our next step is to calculate the predicted values $\hat{\boldsymbol{y}}$ and the residuals $\hat{\boldsymbol{\epsilon}}$

$$\begin{aligned}\hat{\boldsymbol{y}} &= \boldsymbol{X}\hat{\boldsymbol{\beta}}\\ \hat{\boldsymbol{\epsilon}} &= \boldsymbol{y}-\hat{\boldsymbol{y}}\end{aligned}$$

```
y.hat <- X %*% beta.hat
residuals <- y - y.hat
```

Now that we have the residuals, we can calculate $\hat{\sigma}^2$ and the standard errors of $\hat{\beta}_j$

$$\hat{\sigma}^2=\frac{1}{n-p}\hat{\boldsymbol{\epsilon}}^T\hat{\boldsymbol{\epsilon}}$$

$$StdErr\left(\hat{\beta}_j\right) = \sqrt{\hat{\sigma}^2 \left[\left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\right]_{jj}}$$

```
sigma2.hat <- ( t(residuals) %*% residuals) / (n-2)
sigma.hat <- sqrt( sigma2.hat )
std.errs <- sqrt( sigma2.hat * diag(XtXinv) )
```

We now print out the important values and compare them to the summary output given by the `lm()` function in R.

```
beta.hat
```

```
##        [,1]
##   -2.175664
## x  1.844463
```

```
sigma.hat
```

```
##         [,1]
## [1,] 2.02257
```

```
std.errs
```

```
## [1] 0.8716179 0.1490206
```

```
model <- lm(y~x)
summary(model)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7057 -1.5547  0.1971  1.4965  2.6206
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -2.1757     0.8716  -2.496   0.0225 *
## x             1.8445     0.1490  12.377 3.06e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.023 on 18 degrees of freedom
## Multiple R-squared:  0.8949, Adjusted R-squared:  0.889
## F-statistic: 153.2 on 1 and 18 DF,  p-value: 3.061e-10
```

We calculate 95% confidence intervals via:

```
lwr <- beta.hat - qt(.975, n-2) * std.errs
upr <- beta.hat + qt(.975, n-2) * std.errs
CI <- cbind(lwr,upr)
colnames(CI) <- c('lower','upper')
rownames(CI) <- c('Intercept', 'x')
CI
```

```
##               lower      upper
## Intercept -4.006865 -0.3444628
## x          1.531382  2.1575435
```

These intervals are the same as what we get when we use the `confint()` function.

```
confint(model)
```

```
##                 2.5 %     97.5 %
## (Intercept) -4.006865 -0.3444628
## x            1.531382  2.1575435
```

2.2 ANOVA model

The anova model is also a linear model and all we must do is create a appropriate design matrix. Given the design matrix $\boldsymbol{X}$, all the calculations are identical as in the simple regression case.

2.2.1 Cell means representation

Recall the cell means representation is

$$y_{i,j} = \mu_i + \epsilon_{i,j}$$

where $y_{i,j}$ is the jth observation within the ith group. To clearly show the creation of the $\boldsymbol{X}$ matrix, let the number of groups be $p = 3$ and the number of observations per group be $n_i = 4$. We now expand the formula to show all the data.

$$\begin{aligned}
y_{1,1} &= \mu_1 + \epsilon_{1,1} \\
y_{1,2} &= \mu_1 + \epsilon_{1,2} \\
y_{1,3} &= \mu_1 + \epsilon_{1,3} \\
y_{1,4} &= \mu_1 + \epsilon_{1,4} \\
y_{2,1} &= \mu_2 + \epsilon_{2,1} \\
y_{2,2} &= \mu_2 + \epsilon_{2,2} \\
y_{2,3} &= \mu_2 + \epsilon_{2,3} \\
y_{2,4} &= \mu_2 + \epsilon_{2,4} \\
y_{3,1} &= \mu_3 + \epsilon_{3,1} \\
y_{3,2} &= \mu_3 + \epsilon_{3,2} \\
y_{3,3} &= \mu_3 + \epsilon_{3,3} \\
y_{3,4} &= \mu_3 + \epsilon_{3,4}
\end{aligned}$$

In an effort to write the model as $\boldsymbol{y} = \boldsymbol{X\beta} + \boldsymbol{\epsilon}$ we will write the above as

$$\begin{aligned}
y_{1,1} &= 1\mu_1 + 0\mu_2 + 0\mu_3 + \epsilon_{1,1}\\
y_{1,2} &= 1\mu_1 + 0\mu_2 + 0\mu_3 + \epsilon_{1,2}\\
y_{1,3} &= 1\mu_1 + 0\mu_2 + 0\mu_3 + \epsilon_{1,3}\\
y_{1,4} &= 1\mu_1 + 0\mu_2 + 0\mu_3 + \epsilon_{1,4}\\
y_{2,1} &= 0\mu + 1\mu_2 + 0\mu_3 + \epsilon_{2,1}\\
y_{2,2} &= 0\mu + 1\mu_2 + 0\mu_3 + \epsilon_{2,2}\\
y_{2,3} &= 0\mu + 1\mu_2 + 0\mu_3 + \epsilon_{2,3}\\
y_{2,4} &= 0\mu + 1\mu_2 + 0\mu_3 + \epsilon_{2,4}\\
y_{3,1} &= 0\mu + 0\mu_2 + 1\mu_3 + \epsilon_{3,1}\\
y_{3,2} &= 0\mu + 0\mu_2 + 1\mu_3 + \epsilon_{3,2}\\
y_{3,3} &= 0\mu + 0\mu_2 + 1\mu_3 + \epsilon_{3,3}\\
y_{3,4} &= 0\mu + 0\mu_2 + 1\mu_3 + \epsilon_{3,4}
\end{aligned}$$

and we will finally be able to write the matrix version

$$\underbrace{\begin{bmatrix} y_{1,1}\\ y_{1,2}\\ y_{1,3}\\ y_{1,4}\\ y_{2,1}\\ y_{2,2}\\ y_{2,3}\\ y_{2,4}\\ y_{3,1}\\ y_{3,2}\\ y_{3,3}\\ y_{3,4} \end{bmatrix}}_{\boldsymbol{y}} = \underbrace{\begin{bmatrix} 1 & 0 & 0\\ 1 & 0 & 0\\ 1 & 0 & 0\\ 1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 1 & 0\\ 0 & 1 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\\ 0 & 0 & 1\\ 0 & 0 & 1\\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{X}} \underbrace{\begin{bmatrix} \mu_1\\ \mu_2\\ \mu_3 \end{bmatrix}}_{\boldsymbol{\beta}} + \underbrace{\begin{bmatrix} \epsilon_{1,1}\\ \epsilon_{1,2}\\ \epsilon_{1,3}\\ \epsilon_{1,4}\\ \epsilon_{2,1}\\ \epsilon_{2,2}\\ \epsilon_{2,3}\\ \epsilon_{2,4}\\ \epsilon_{3,1}\\ \epsilon_{3,2}\\ \epsilon_{3,3}\\ \epsilon_{3,4} \end{bmatrix}}_{\boldsymbol{\epsilon}}$$

Notice that each column of the $\boldsymbol{X}$ matrix is acting as an indicator if the observation is an element of the appropriate group. As such, these are often called *indicator variables*. Another term for these, which I find less helpful, is *dummy variables*.

2.2.2 Offset from reference group

In this model representation of ANOVA, we have an overall mean and then offsets from the control group (which will be group one). The model is thus

$$y_{i,j} = \mu + \tau_i + \epsilon_{i,j}$$

where $\tau_1 = 0$. We can write this in matrix form as

$$\underbrace{\begin{bmatrix} y_{1,1} \\ y_{1,2} \\ y_{1,3} \\ y_{1,4} \\ y_{2,1} \\ y_{2,2} \\ y_{2,3} \\ y_{2,4} \\ y_{3,1} \\ y_{3,2} \\ y_{3,3} \\ y_{3,4} \end{bmatrix}}_{\boldsymbol{y}} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix}}_{\mathbf{X}} \underbrace{\begin{bmatrix} \mu \\ \tau_2 \\ \tau_3 \end{bmatrix}}_{\boldsymbol{\beta}} + \underbrace{\begin{bmatrix} \epsilon_{1,1} \\ \epsilon_{1,2} \\ \epsilon_{1,3} \\ \epsilon_{1,4} \\ \epsilon_{2,1} \\ \epsilon_{2,2} \\ \epsilon_{2,3} \\ \epsilon_{2,4} \\ \epsilon_{3,1} \\ \epsilon_{3,2} \\ \epsilon_{3,3} \\ \epsilon_{3,4} \end{bmatrix}}_{\boldsymbol{\epsilon}}$$

2.3 Exercises

1. We will do a simple ANOVA analysis on example 8.2 from Ott & Longnecker using the matrix representation of the model. A clinical psychologist wished to compare three methods for reducing hostility levels in university students, and used a certain test (HLT) to measure the degree of hostility. A high score on the test indicated great hostility. The psychologist used 24 students who obtained high and nearly equal scores in the experiment. eight were selected at random from among the 24 problem cases and were treated with method 1. Seven of the remaining 16 students were selected at random and treated with method 2. The remaining nine students were treated with method 3. All treatments were continued for a one-semester period. Each student was given the HLT test at the end of the semester, with the results show in the following table. (This analysis was done in section 8.3 of my STA 570 notes)

Method	Values
1	96, 79, 91, 85, 83, 91, 82, 87
2	77, 76, 74, 73, 78, 71, 80
3	66, 73, 69, 66, 77, 73, 71, 70, 74

We will be using the cell means model of ANOVA

$$y_{ij} = \beta_i + \epsilon_{ij}$$

where β_i is the mean of group i and $\epsilon_{ij} \overset{iid}{\sim} N\left(0, \sigma^2\right)$.

a. Create one vector of all 24 hostility test scores `y`. (Use the `c()` function.)

b. Create a design matrix `X` with dummy variables for columns that code for what group an observation belongs to. Notice that `X` will be a 24 rows by 3 column matrix. An R function that might be handy is `cbind(a,b)` which will bind two vectors or matrices together along the columns. (There is also a corresponding `rbind()` function that binds vectors/matrices along rows.)

c) Find $\hat{\boldsymbol{\beta}}$ using the matrix formula given in class. The R function `t(A)` computes the matrix transpose $\mathbf{A}^T$, `solve(A)` computes $\mathbf{A}^{-1}$, and the operator `%*%` does matrix multiplication (used as `A %*% B`).

d) Examine the matrix $\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T$. What do you notice about it? In particular, think about the result when you right multiply by $\mathbf{y}$. How does this matrix calculate the appropriate group means and using the appropriate group sizes n_i?

2. We will calculate the y-intercept and slope estimates in a simple linear model using matrix notation. We will use a data set that gives the diameter at breast height (DBH) versus tree height for a randomly selected set of trees. In addition, for each tree, a ground measurement of crown closure (CC) was taken. Larger values of crown closure indicate more shading and is often associated with taller tree morphology (possibly). We will be interested in creating a regression model that predicts height based on DBH and CC. In the interest of reduced copying, we will only use 10 observations. *(Note: I made this data up and the DBH values might be unrealistic. Don't make fun of me.)*

DBH	30.5	31.5	31.7	32.3	33.3	35	35.4	35.6	36.3	37.8
CC	0.74	0.69	0.65	0.72	0.58	0.5	0.6	0.7	0.52	0.6
Height	58	64	65	70	68	63	78	80	74	76

We are interested in fitting the regression model

$$y_i = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \epsilon_i$$

where β_0 is the y-intercept and β_1 is the slope parameter associated with DBH and β_2 is the slope parameter associated with Crown Closure.

a) Create a vector of all 10 heights $\mathbf{y}$.

b) Create the design matrix $\mathbf{X}$.

c) Find $\hat{\boldsymbol{\beta}}$ using the matrix formula given in class.

d) Compare your results to the estimated coefficients you get using the `lm()` function. To add the second predictor to the model, your call to `lm()` should look something like `lm(Height ~ DBH + CrownClosure)`.

Chapter 3

Inference

3.1 F-tests

We wish to develop a rigorous way to compare nested models and decide if a complicated model explains enough more variability than a simple model to justify the additional intellectual effort of thinking about the data in the complicated fashion.

It is important to specify that we are developing a way of testing nested models. By nested, we mean that the simple model can be created from the full model just by setting one or more model parameters to zero.

3.1.1 Theory

Recall that in the simple regression and ANOVA cases we were interested in comparing a simple model versus a more complex model. For each model we computed the residual sum of squares (RSS) and said that if the complicated model performed much better than the simple then $RSS_{simple} \gg RSS_{complex}$. To do this we needed to standardize by the number of parameters added to the model and the degrees of freedom remaining in the full model. We first defined $RSS_{diff} = RSS_{simple} - RSS_{complex}$ and let df_{diff} be the number of parameters difference between the simple and complex models. Then we had

$$F = \frac{RSS_{difference}/df_{diff}}{RSS_{complex}/df_{complex}}$$

and we claimed that if the null hypothesis was true (i.e. the complex model is an unnecessary obfuscation of the simple), then this ratio follows an F -distribution with degrees of freedom df_{diff} and $df_{complex}$.

The critical assumption for the F-test to be appropriate is that the error terms are independent and normally distributed with constant variance.

We will consider a data set from Johnson and Raven (1973) which also appears in Weisberg (1985). This data set is concerned with the number of tortoise species on $n = 30$ different islands in the Galapagos. The variables of interest in the data set are:

Variable	Description
`Species`	Number of tortoise species found on the island
`Endimics`	Number of tortoise species endemic to the island
`Elevation`	Elevation of the highest point on the island
`Area`	Area of the island (km^2)
`Nearest`	Distance to the nearest neighboring island (km)
`Scruz`	Distance to the Santa Cruz islands (km)
`Adjacent`	Area of the nearest adjacent island (km^2)

We will first read in the data set from the package `faraway`.

```
library(faraway)     # load the library
data(gala)           # import the data set
head(gala)           # show the first couple of rows
```

```
##              Species Endemics  Area Elevation Nearest Scruz Adjacent
## Baltra            58       23 25.09       346     0.6   0.6     1.84
## Bartolome         31       21  1.24       109     0.6  26.3   572.33
## Caldwell           3        3  0.21       114     2.8  58.7     0.78
## Champion          25        9  0.10        46     1.9  47.4     0.18
## Coamano            2        1  0.05        77     1.9   1.9   903.82
## Daphne.Major      18       11  0.34       119     8.0   8.0     1.84
```

First we will create the full model that predicts the number of species as a function of elevation, area, nearest, scruz and adjacent. Notice that this model has $p = 6$ β_i values (one for each coefficient plus the intercept).

```
M.c <- lm(Species ~ Area + Elevation + Nearest + Scruz + Adjacent, data=gala)
summary(M.c)
```

```
##
## Call:
## lm(formula = Species ~ Area + Elevation + Nearest + Scruz + Adjacent,
##     data = gala)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -111.679  -34.898   -7.862   33.460  182.584
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.068221  19.154198   0.369 0.715351
## Area        -0.023938   0.022422  -1.068 0.296318
## Elevation    0.319465   0.053663   5.953 3.82e-06 ***
## Nearest      0.009144   1.054136   0.009 0.993151
## Scruz       -0.240524   0.215402  -1.117 0.275208
## Adjacent    -0.074805   0.017700  -4.226 0.000297 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 60.98 on 24 degrees of freedom
## Multiple R-squared:  0.7658, Adjusted R-squared:  0.7171
## F-statistic:  15.7 on 5 and 24 DF,  p-value: 6.838e-07
```

3.1.2 Testing All Covariates

The first test we might want to do is to test if any of the covariates are significant. That is to say that we want to test the full model versus the simple null hypothesis model

$$y_i = \beta_0 + \epsilon_i$$

that has no covariates and only a y-intercept. So we will create a simple model

```
M.s <- lm(Species ~ 1, data=gala)
```

and calculate the appropriate Residual Sums of Squares (RSS) for each model, along with the difference in degrees of freedom between the two models.

```
RSS.c <- sum(resid(M.c)^2)
RSS.s <- sum(resid(M.s)^2)
df.diff <- 5               # complex model has 5 additional parameters
df.c <- 30 - 6             # complex model has 24 degrees of freedom left
```

The F-statistic for this test is therefore

```
F.stat <-  ( (RSS.s - RSS.c) / df.diff ) / ( RSS.c / df.c )
F.stat
```

```
## [1] 15.69941
```

and should be compared against the F-distribution with 5 and 24 degrees of freedom. Because a large difference between RSS.s and RSS.c would be evidence for the alternative, larger model, the p-value for this test is

$$p - value = P\left(F_{5,24} \geq \texttt{F.stat}\right)$$

```
p.value <-  1 - pf(15.699, 5, 24)
p.value
```

```
## [1] 6.839486e-07
```

Both the F.stat and its p-value are given at the bottom of the summary table. However, I might be interested in creating an ANOVA table for this situation.

Source	df	Sum Sq	Mean Sq	F	p-value
Difference	$p-1$	RSS_d	$MSE_d = RSS_d/(p-1)$	MSE_d/MSE_c	$P(F > F_{p-1,n-p})$
Complex	$n-p$	RSS_c	$MSE_c = RSS_c/(n-p)$		
Simple	$n-1$	RSS_s			

This table can be obtained from R by using the `anova()` function on the two models of interest. As usual with R, it does not show the simple row, but rather concentrates on the difference row.

```
anova(M.s, M.c)
```

```
## Analysis of Variance Table
##
## Model 1: Species ~ 1
## Model 2: Species ~ Area + Elevation + Nearest + Scruz + Adjacent
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1     29 381081
## 2     24  89231  5    291850 15.699 6.838e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

3.1.3 Testing a Single Covariate

For a particular covariate, β_j, we might wish to perform a test to see if it can be removed from the model. It can be shown that the F-statistic can be re-written as

$$\begin{aligned}
F &= \frac{\left[RSS_s - RSS_c\right]/1}{RSS_c/\left(n-p\right)} \\
&= \vdots \\
&= \left[\frac{\hat{\beta}_j}{SE\left(\hat{\beta}_j\right)}\right]^2 \\
&= t^2
\end{aligned}$$

where t has a t-distribution with $n-p$ degrees of freedom under the null hypothesis that the simple model is sufficient.

We consider the case of removing the covariate **Area** from the model and will calculate our test statistic using both methods.

```
M.c <- lm(Species ~ Area + Elevation + Nearest + Scruz + Adjacent, data=gala)
M.s <- lm(Species ~        Elevation + Nearest + Scruz + Adjacent, data=gala)
RSS.c <- sum( resid(M.c)^2 )
RSS.s <- sum( resid(M.s)^2 )
df.d <- 1
df.c <- 30-6
F.stat <- ((RSS.s - RSS.c)/1) / (RSS.c / df.c)
F.stat
```

```
## [1] 1.139792
```

```
1 - pf(F.stat, 1, 24)
```

```
## [1] 0.296318
```

```
sqrt(F.stat)
```

```
## [1] 1.067611
```

To calculate it using the estimated coefficient and its standard error, we must grab those values from the summary table

```
temp <- summary(M.c)
temp$coefficients
```

```
##                 Estimate  Std. Error      t value     Pr(>|t|)
## (Intercept)  7.068220709 19.15419782  0.369016796 7.153508e-01
## Area        -0.023938338  0.02242235 -1.067610554 2.963180e-01
## Elevation    0.319464761  0.05366280  5.953187968 3.823409e-06
## Nearest      0.009143961  1.05413595  0.008674366 9.931506e-01
## Scruz       -0.240524230  0.21540225 -1.116628222 2.752082e-01
## Adjacent    -0.074804832  0.01770019 -4.226216850 2.970655e-04
```

```
beta.area <- temp$coefficients[2,1]
SE.beta.area <- temp$coefficients[2,2]
t <- beta.area / SE.beta.area
t
```

```
## [1] -1.067611
```

```
2 * pt(t, 24)
```

```
## [1] 0.296318
```

All that hand calculation is tedious, so we can again use the anova() command to compare the two models.

```r
anova(M.s, M.c)
```

```
## Analysis of Variance Table
##
## Model 1: Species ~ Elevation + Nearest + Scruz + Adjacent
## Model 2: Species ~ Area + Elevation + Nearest + Scruz + Adjacent
##   Res.Df   RSS Df Sum of Sq      F Pr(>F)
## 1     25 93469
## 2     24 89231  1    4237.7 1.1398 0.2963
```

3.1.4 Testing a Subset of Covariates

Often a researcher will want to remove a subset of covariates from the model. In the Galapagos example, Area, Nearest, and Scruz all have non-significant p-values and would be removed when comparing the full model to the model without that one covariate. While each of them might be non-significant, is the sum of all three significant?

Because the individual $\hat{\beta}_j$ values are not independent, then we cannot claim that the subset is not statistically significant just because each variable in turn was insignificant. Instead we again create simple and complex models in the same fashion as we have previously done.

```r
M.c <- lm(Species ~ Area + Elevation + Nearest + Scruz + Adjacent, data=gala)
M.s <- lm(Species ~        Elevation +                   Adjacent, data=gala)
anova(M.s, M.c)
```

```
## Analysis of Variance Table
##
## Model 1: Species ~ Elevation + Adjacent
## Model 2: Species ~ Area + Elevation + Nearest + Scruz + Adjacent
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     27 100003
## 2     24  89231  3     10772 0.9657  0.425
```

We find a large p-value associated with this test and can safely stay with the null hypothesis, that the simple model is sufficient to explain the observed variability in the number of species of tortoise.

3.2 Confidence Intervals for location parameters

Recall that

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

and it is easy to calculate the estimate of σ^2. This estimate will be the “average” squared residual

$$\hat{\sigma}^2 = \frac{RSS}{df}$$

where RSS is the residual sum of squares and df is the degrees of freedom $n - p$ where p is the number of β_j parameters. Therefore the standard error of the $\hat{\beta}_j$ values is

$$SE\left(\hat{\beta}_j\right) = \sqrt{\hat{\sigma}^2\left(\mathbf{X}^T\mathbf{X}\right)_{jj}^{-1}}$$

We can see this calculation in the summary regression table. We again consider the Galapagos Island data set. First we must create the design matrix

```r
y <- gala$Species
X <- cbind( rep(1,30), gala$Elevation, gala$Adjacent )
```

And then create $\left(\mathbf{X}^{T}\mathbf{X}\right)^{-1}$

```r
XtXinv <- solve(  t(X) %*% X )
XtXinv
```

```
##                [,1]          [,2]          [,3]
## [1,]  6.094829e-02 -8.164025e-05  9.312123e-06
## [2,] -8.164025e-05  2.723835e-07 -7.126027e-08
## [3,]  9.312123e-06 -7.126027e-08  6.478031e-08
```

```r
diag(XtXinv)
```

```
## [1] 6.094829e-02 2.723835e-07 6.478031e-08
```

Eventually we will need $\hat{\boldsymbol{\beta}}$

```r
beta.hat <- XtXinv %*% t(X) %*% y
beta.hat
```

```
##            [,1]
## [1,]  1.4328722
## [2,]  0.2765683
## [3,] -0.0688855
```

And now find the estimate $\hat{\sigma}$

```r
H <- X %*% XtXinv %*% t(X)
y.hat <- H %*% y
RSS <- sum( (y-y.hat)^2 )
sigma.hat <- sqrt(  RSS/(30-3) )
sigma.hat
```

```
## [1] 60.85898
```

The standard errors of $\hat{\beta}$ is thus

```r
sqrt( sigma.hat^2 * diag(XtXinv) )
```

```
## [1] 15.02468680  0.03176253  0.01548981
```

We can double check that this is what R calculates in the summary table

```r
model <- lm(Species ~ Elevation + Adjacent, data=gala)
summary(model)
```

```
##
## Call:
## lm(formula = Species ~ Elevation + Adjacent, data = gala)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -103.41  -34.33  -11.43   22.57  203.65
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.43287   15.02469   0.095 0.924727
## Elevation    0.27657    0.03176   8.707 2.53e-09 ***
## Adjacent    -0.06889    0.01549  -4.447 0.000134 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 60.86 on 27 degrees of freedom
## Multiple R-squared:  0.7376, Adjusted R-squared:  0.7181
## F-statistic: 37.94 on 2 and 27 DF,  p-value: 1.434e-08
```

It is highly desirable to calculate confidence intervals for the regression parameters. Recall that the general form of a confidence interval is

$$Estimate\ \pm Critical\,Value\ \cdot\ StandardError\left(Estimate\right)$$

For any specific β_j we will have

$$\hat{\beta}_j \pm t_{n-p}^{1-\alpha/2}\,\hat{\sigma}\sqrt{\left(\mathbf{X}^T\mathbf{X}\right)_{jj}^{-1}}$$

where $\hat{\sigma}^2\left(\mathbf{X}^T\mathbf{X}\right)_{jj}^{-1}$ is the $[j,j]$ element of the variance/covariance of $\hat{\boldsymbol{\beta}}$.

To demonstrate this, we return to the Galapagos Island data set.

Finally we can calculate confidence intervals for our three β_j values

```
lower <- beta.hat - qt(.975, 27) * sigma.hat * sqrt( diag(XtXinv) )
upper <- beta.hat + qt(.975, 27) * sigma.hat * sqrt( diag(XtXinv) )
cbind(lower, upper)
```

```
##             [,1]        [,2]
## [1,] -29.395239 32.26098305
## [2,]   0.211397  0.34173962
## [3,]  -0.100668 -0.03710303
```

That is certainly a lot of work to do by hand (even with R doing all the matrix multiplication) but we can get these from R by using the confint() command.

```
confint(model)
```

```
##                   2.5 %      97.5 %
## (Intercept) -29.395239 32.26098305
## Elevation     0.211397  0.34173962
## Adjacent     -0.100668 -0.03710303
```

3.3 Prediction and Confidence Intervals for a response

Given a vector of predictor covariates $\boldsymbol{x}_0$ (think of $\boldsymbol{x}_0^T$ as potentially one row in $\boldsymbol{X}$. Because we might want to predict some other values than what we observe, we do not restrict ourselves to *only* rows in $\boldsymbol{X}$), we want to make inference on the expected value $\hat{y}_0$. We can calculate the value by

$$\hat{y}_0 = \boldsymbol{x}_0^T\hat{\boldsymbol{\beta}}$$

and we are interested in two different types of predictions.

1. We might be interested in the uncertainty of a new data point. This uncertainty has two components: the uncertainty of the regression model and uncertainty of a new data point from its expected value.

2. Second, we might be interested in only the uncertainty about the regression model.

We note that because $\boldsymbol{x}_0^T$ is just a constant, we can calculate the variance of this value as

$$\begin{aligned}Var\left(\boldsymbol{x}_0^T\hat{\boldsymbol{\beta}}\right) &= \boldsymbol{x}_0^T\,Var\left(\hat{\boldsymbol{\beta}}\right)\,\boldsymbol{x}_0 \\ &= \boldsymbol{x}_0^T\,\left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\sigma^2\,\boldsymbol{x}_0 \\ &= \boldsymbol{x}_0^T\,\left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\,\boldsymbol{x}_0\,\sigma^2\end{aligned}$$

and use this to calculate two types of intervals. First, a prediction interval for a new observation is

$$\hat{y}_0 \pm t_{n-p}^{1-\alpha/2}\,\hat{\sigma}\sqrt{1+\boldsymbol{x}_0^T\,\left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\,\boldsymbol{x}_0}$$

and a confidence interval for the mean response for the given $\boldsymbol{x}_0$ is

$$\hat{y}_0 \pm t_{n-p}^{1-\alpha/2}\,\hat{\sigma}\sqrt{\boldsymbol{x}_0^T\,\left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\,\boldsymbol{x}_0}$$

Again using the Galapagos Island data set as an example, we might be interested in predicting the number of tortoise species of an island with highest point 400 meters and nearest adjacent island with area $200km^2$. We then have

$$\boldsymbol{x}_0^T = \left[\begin{array}{ccc}1 & 400 & 200\end{array}\right]$$

and we can calculate

```
x0 <- c(1, 400, 200)
y0 <- t(x0) %*% beta.hat
y0
```

```
##          [,1]
## [1,] 98.28309
```

and then calculate $\boldsymbol{x}_0^T\,\left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\,\boldsymbol{x}_0$

```
xt.XtXinv.x <- t(x0) %*% solve( t(X) %*% X ) %*% x0
```

Thus the prediction interval will be

```
c(y0 - qt(.975, 27) * sigma.hat * sqrt(1 + xt.XtXinv.x),
  y0 + qt(.975, 27) * sigma.hat * sqrt(1 + xt.XtXinv.x))
```

```
## [1] -28.70241 225.26858
```

while a confidence interval for the expectation is

```
c(y0 - qt(.975, 27) * sigma.hat * sqrt(xt.XtXinv.x),
  y0 + qt(.975, 27) * sigma.hat * sqrt(xt.XtXinv.x))
```

```
## [1]  75.21317 121.35301
```

These prediction and confidence intervals can be calculated in R using the predict() function

```
x0 <- data.frame(Elevation=400, Adjacent=200)
predict(model, newdata=x0, interval='prediction')
```

```
##        fit       lwr      upr
## 1 98.28309 -28.70241 225.2686
```

```
predict(model, newdata=x0, interval='confidence')
```

```
##        fit      lwr     upr
## 1 98.28309 75.21317 121.353
```

3.4 Interpretation with Correlated Covariates

The standard interpretation of the slope parameter is that β_j is the amount of increase in y for a one unit increase in the jth covariate, provided that all other covariates stayed the same.

The difficulty with this interpretation is that covariates are often related, and the phrase “all other covariates stayed the same” is often not reasonable. For example, if we have a dataset that models the mean annual temperature of a location as a function of latitude, longitude, and elevation, then it is not physically possible to hold latitude, and longitude constant while changing elevation.

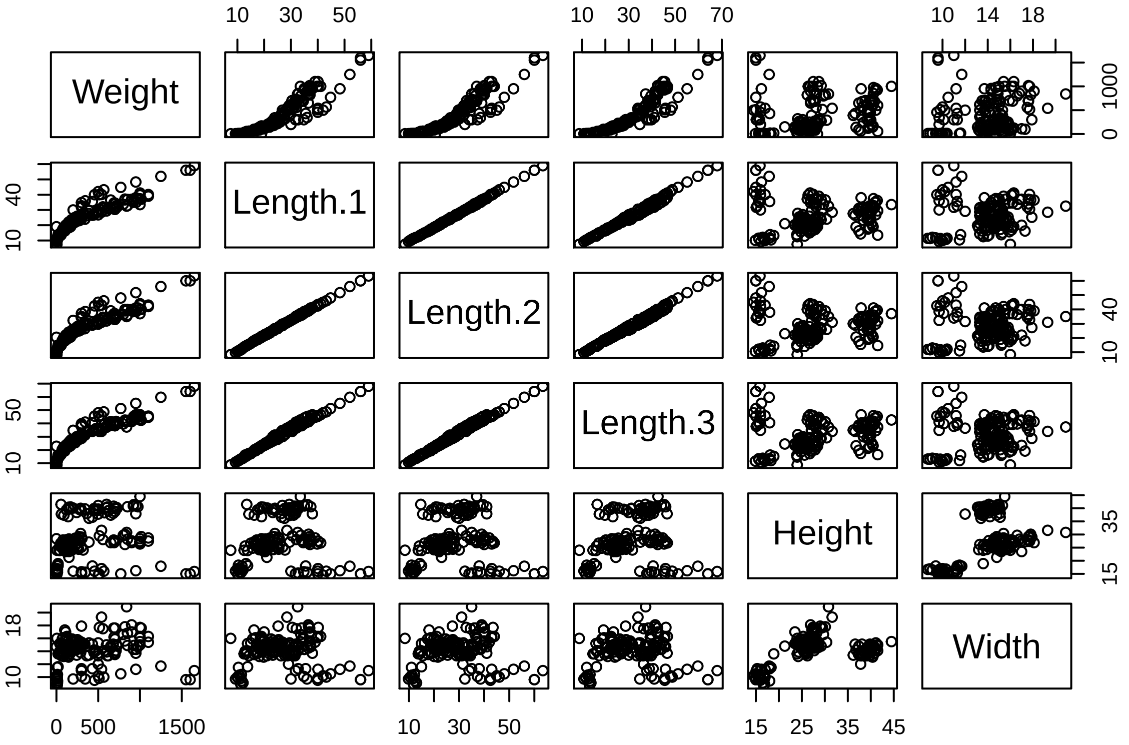
One common issue that make interpretation difficult is that covariates can be highly correlated.

Perch Example: We might be interested in estimating the weight of a fish based off of its length and width. The dataset we will consider is from fishes are caught from the same lake (Laengelmavesi) near Tampere in Finland. The following variables were observed:

Variable	Interpretation
`Weight`	Weight (g)
`Length.1`	Length from nose to beginning of Tail (cm)
`Length.2`	Length from nose to notch of Tail (cm)
`Length.3`	Length from nose to tip of tail (cm)
`Height`	Maximal height as a percentage of `Length.3`
`Width`	Maximal width as a percentage of `Length.3`
`Sex`	0=Female, 1=Male
`Species`	Which species of perch (1-7)

We first look at the data and observe the expected relationship between length and weight.

```r
file <- 'https://raw.githubusercontent.com/dereksonderegger/STA_571_Book/master/data-raw/Fish.csv'
fish <- read.table(file, header=TRUE, skip=111, sep=',')
pairs(fish[,c('Weight','Length.1','Length.2','Length.3','Height','Width')])
```



Naively, we might consider the linear model with all the length effects present.

```r
model <- lm(Weight ~ Length.1 + Length.2 + Length.3 + Height + Width, data=fish)
summary(model)
```

```
##
## Call:
## lm(formula = Weight ~ Length.1 + Length.2 + Length.3 + Height +
##     Width, data = fish)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -302.22  -79.72  -39.88   92.63  344.85
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -724.539     77.133  -9.393   <2e-16 ***
## Length.1      32.389     45.134   0.718   0.4741
## Length.2      -9.184     48.367  -0.190   0.8497
## Length.3       8.747     16.283   0.537   0.5919
## Height         4.947      2.768   1.787   0.0759 .
## Width          8.636      6.972   1.239   0.2174
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 132.9 on 152 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.8675, Adjusted R-squared:  0.8631
## F-statistic:   199 on 5 and 152 DF,  p-value: < 2.2e-16
```

This is crazy. There is a negative relationship between `Length.2` and `Weight`. That does not make any sense unless you realize that this is the effect of `Length.2` assuming the other covariates are in the model and can be held constant while changing the value of `Length.2`, which is obviously ridiculous.

If we remove the highly correlated covariates then we see a much better behaved model

```r
model <- lm(Weight ~ Length.2 + Height + Width, data=fish)
summary(model)
```

```
##
## Call:
## lm(formula = Weight ~ Length.2 + Height + Width, data = fish)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -306.14  -75.11  -36.45   89.54  337.95
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -701.0750    71.0438  -9.868  < 2e-16 ***
## Length.2      30.4360     0.9841  30.926  < 2e-16 ***
## Height         5.5141     1.4311   3.853 0.000171 ***
## Width          5.6513     5.2016   1.086 0.278974
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 132.3 on 154 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.8669, Adjusted R-squared:  0.8643
## F-statistic: 334.2 on 3 and 154 DF,  p-value: < 2.2e-16
```

When you have two variables in a model that are highly positively correlated, you often find that one will have a positive coefficient and the other will be negative. Likewise, if two variables are highly negatively correlated, the two regression coefficients will often be the same sign.

In this case the sum of the three length covariate estimates was approximately 31 in both cases, but with three length variables, the second could be negative the third be positive with approximately the same magnitude and we get approximately the same model as with both the second and third length variables missing from the model.

In general, you should be very careful with the interpretation of the regression coefficients when the covariates are highly correlated. We will talk about how to recognize these situations and what to do about them later in the course.

3.5 Exercises

1. The dataset prostate in package `faraway` has information about a study of 97 men with prostate cancer. We import the data and examine the first four observations using the following commands.

   ```
   library(faraway)
   data(prostate)
   head(prostate)
   ```

   It is possible to get information about the data set using the command `help(prostate)`. Fit a model with `lpsa` as the response and all the other variables as predictors.

   a) Compute 90% and 95% confidence intervals for the parameter associated with `age`. Using just these intervals, what could we deduced about the p-value for age in the regression summary. *Hint: look at the help for the function `confint()`. You'll find the `level` option to be helpful.*

   b) Remove all the predictors that are not significant at the 5% level. Test this model against the original model. Which is preferred?

2. Thirty samples of cheddar cheese were analyzed for their content of acetic acid, hydrogen sulfide and lactic acid. Each sample was tasted and scored by a panel of judges and the average taste score produces. Used the `cheddar` dataset from the `faraway` package (import it the same way you did in problem one, but now use `cheddar`) to answer the following:

   a) Fit a regression model with taste as the response and the three chemical contents as predictors. Identify the predictors that are statistically significant at the 5% level.

   b) `Acetic` and `H2S` are measured on a $\log_{10}$ scale. Create two new columns in the `cheddar` data frame that contain the values on their original scale. Fit a linear model that uses the three covariates on their non-log scale. Identify the predictors that are statistically significant at the 5% level for this model.

   c) Can we use an F-test to compare these two models? Explain why or why not. Which model provides a better fit to the data? Explain your reasoning.

   d) If H2S is increased by 0.01 for the model in (a), what change in taste would be expected? What caveates must be made in this interpretation.

3. The `sat` data set in the `faraway` package gives data collected to study the relationship between expenditures on public education and test results.

   a) Fit a model that with `total` SAT score as the response and only the intercept as a covariate.

b) Fit a model with `total` SAT score as the response and `expend`, `ratio`, and `salary` as predictors (along with the intercept).

c) Compare the models in parts (a) and (b) using an F-test. Is the larger model superior?

d) Examine the summary table of the larger model? Does this contradict your results in part (c)? What might be causing this issue? Create a graph or summary diagnostics to support your guess.

e) Fit the model with `salary` and `ratio` (along with the intercept) as predictor variables and examine the summary table. Which covariates are significant?

f) Now add `takers` to the model (so the model now includes three predictor variables along with the intercept). Test the hypothesis that $\beta_{takers} = 0$ using the summary table.

g) Discuss why `ratio` was not significant in the model in part (e) but was significant in part (f). *Hint: Look at the Residual Standard Error $\hat{\sigma}$ in each model and argue that each t-statistic is some variant of a "signal-to-noise" ratio and that the "noise" part is reduced in the second model.*

Chapter 4

Analysis of Covariance (ANCOVA)

```r
library(faraway)    # for the data
library(ggplot2)    # my favorite graphing package
library(dplyr)      # for the %>% operator
```

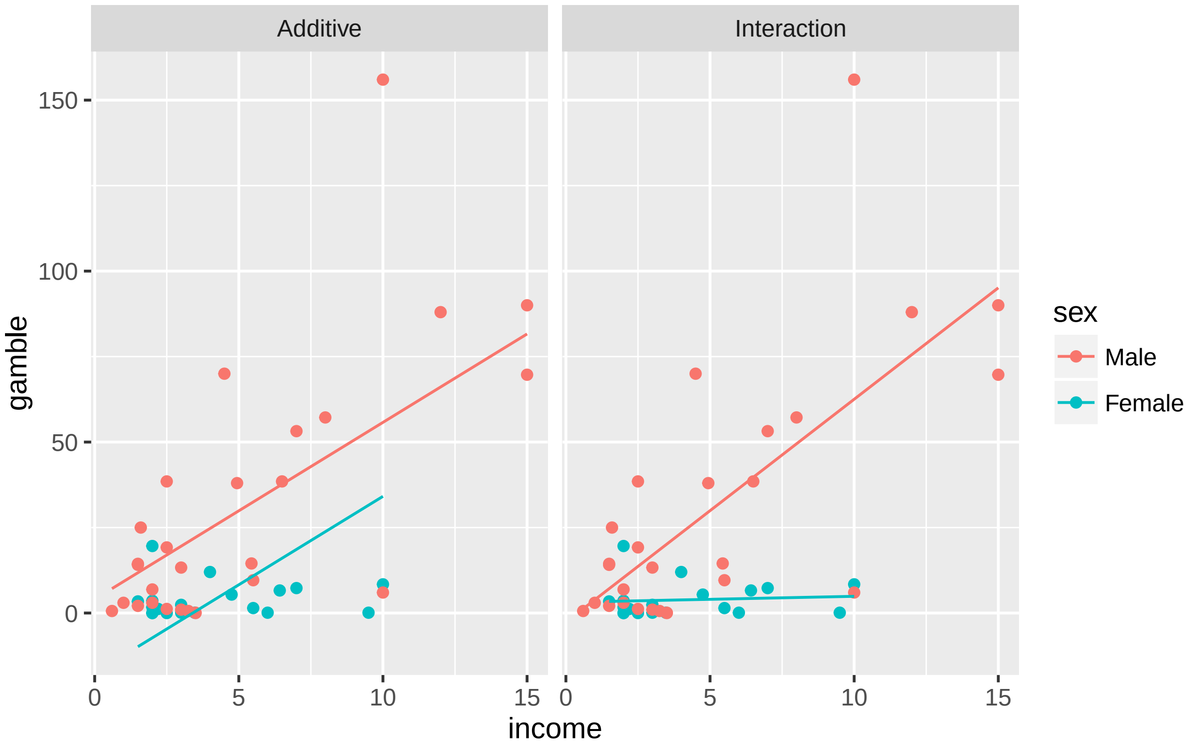
One way that we could extend the ANOVA and regression models is to have both categorical and continuous predictor variables. This model is commonly called ANCOVA which stands for *Analysis of Covariance*.

The dataset `teengamb` in the package `faraway` has data regarding the rates of gambling among teenagers in Britain and their gender and socioeconomic status. One question we might be interested in is how gender and income relate to how much a person gambles. But what should be the effect of gender be?

There are two possible ways that gender could enter the model. Either:

1. We could fit two lines to the data one for males and one for females but require that the lines be parallel (i.e. having the same slopes for income). This is accomplished by having a separate y-intercept for each gender. In effect, the line for the females would be offset by a constant amount from the male line.

2. We could fit two lines but but allow the slopes to differ as well as the y-intercept. This is referred to as an “interaction” between income and gender.



We will now see how to go about fitting these two models. As might be imagined, these can be fit in the same fashion we have been solving the linear models, but require a little finesse in defining the appropriate design matrix $\boldsymbol{X}$.

4.1 Offset parallel Lines (aka additive models)

In order to get offset parallel lines, we want to write a model

$$y_i = \begin{cases} \beta_0 + \beta_1 + \beta_2 x_i + \epsilon_i & \text{if female} \\ \beta_0 + \beta_2 x_i + \epsilon_i & \text{if male} \end{cases}$$

where β_1 is the vertical offset of the female group regression line to the reference group, which is the males regression line. Because the first 19 observations are female, we can this in in matrix form as

$$\begin{bmatrix} y_1 \\ \vdots \\ y_{19} \\ y_{20} \\ \vdots \\ y_{47} \end{bmatrix} = \begin{bmatrix} 1 & 1 & x_1 \\ \vdots & \vdots & \vdots \\ 1 & 1 & x_{19} \\ 1 & 0 & x_{20} \\ \vdots & \vdots & \vdots \\ 1 & 0 & x_{47} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \vdots \\ \epsilon_{19} \\ \epsilon_{20} \\ \vdots \\ \epsilon_{47} \end{bmatrix}$$

I like this representation where β_1 is the offset from the male regression line because it makes it very convenient to test if the offset is equal to zero. The second column of the design matrix referred to as a "dummy variable" or "indicator variable" that codes for the female gender. Notice that even though I have two genders, I only had to add one additional variable to my model because we already had a y-intercept β_0 and we only added one indicator variable for females.

What if we had a third group? Then we would fit another column of indicator variable for the third group. The new beta coefficient in the model would be the offset of the new group to the reference group. For example we consider $n = 9$ observations with $n_i = 3$ observations per group where $y_{i,j}$is the j th replication of the ith group.

$$\begin{bmatrix} y_{1,1} \\ y_{1,2} \\ y_{1,3} \\ y_{2,1} \\ y_{2,2} \\ y_{2,3} \\ y_{3,1} \\ y_{3,2} \\ y_{3,3} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & x_{1,1} \\ 1 & 0 & 0 & x_{1,2} \\ 1 & 0 & 0 & x_{1,3} \\ 1 & 1 & 0 & x_{2,1} \\ 1 & 1 & 0 & x_{2,2} \\ 1 & 1 & 0 & x_{2,3} \\ 1 & 0 & 1 & x_{3,1} \\ 1 & 0 & 1 & x_{3,2} \\ 1 & 0 & 1 & x_{3,3} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} + \begin{bmatrix} \epsilon_{1,1} \\ \epsilon_{1,2} \\ \epsilon_{1,3} \\ \epsilon_{2,1} \\ \epsilon_{2,2} \\ \epsilon_{2,3} \\ \epsilon_{3,3} \\ \epsilon_{3,2} \\ \epsilon_{3,3} \end{bmatrix}$$

In this model, β_0 is the y-intercept for group 1. The paramter β_1 is the vertical offset from the reference group (group 1) for the second group. Similarly β_2 is the vertical offset for group 3. All groups will share the same slope, β_4.

4.2 Lines with different slopes (aka Interaction model)

We can now include a discrete random variable and create regression lines that are parallel, but often that is inappropriate, such as in the teenage gambling dataset. We want to be able to fit a model that has different slopes.

$$y_i = \begin{cases} (\beta_0 + \beta_1) + (\beta_2 + \beta_3)\, x_i + \epsilon_i & \text{if female} \\ \beta_0 + \beta_2 x_i + \epsilon_i & \text{if male} \end{cases}$$

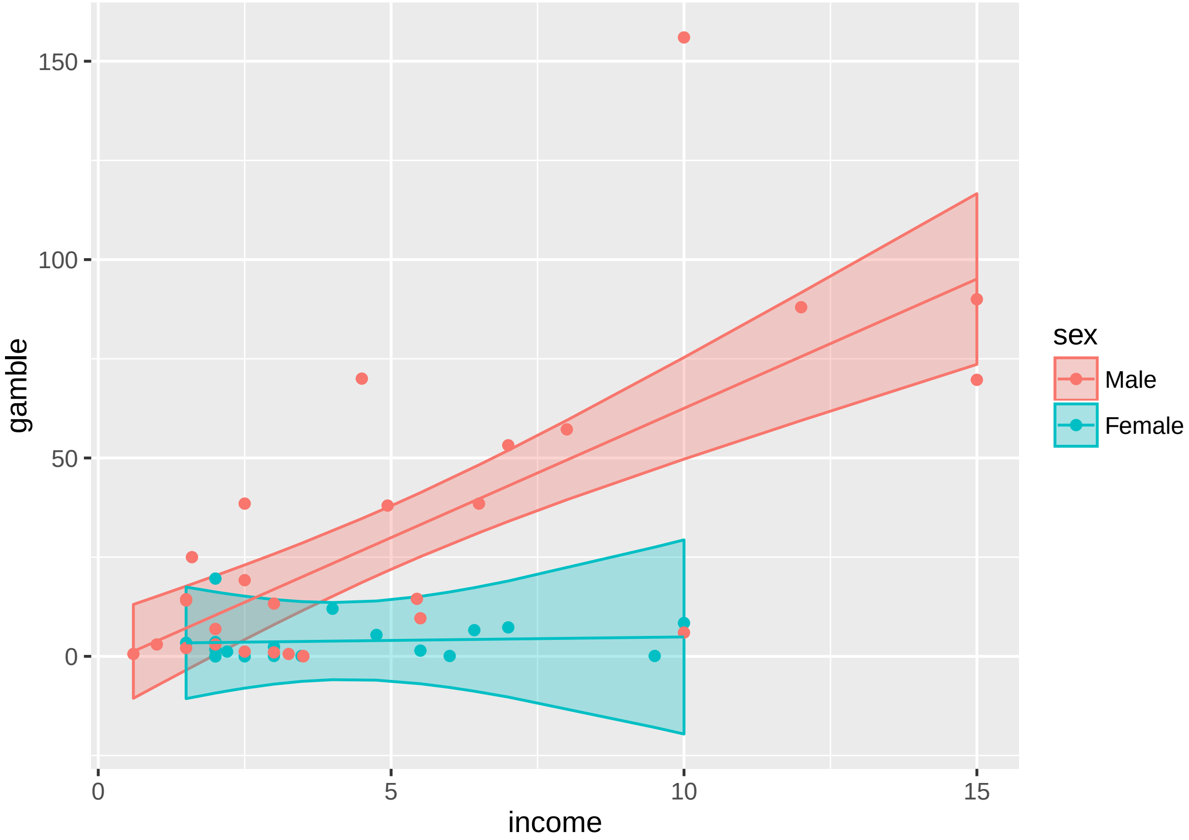
Where β_1 is the offset in y-intercept of the female group from the male group, and β_3 is the offset in slope. Now our matrix formula looks like

$$\begin{bmatrix} y_1 \\ \vdots \\ y_{19} \\ y_{20} \\ \vdots \\ y_{47} \end{bmatrix} = \begin{bmatrix} 1 & 1 & x_1 & x_1 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & x_{19} & x_{19} \\ 1 & 0 & x_{20} & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & x_{47} & 0 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \vdots \\ \epsilon_{19} \\ \epsilon_{20} \\ \vdots \\ \epsilon_{47} \end{bmatrix}$$

where the new fourth column is the what I would get if I multiplied the $\boldsymbol{x}$ column element-wise with the dummy-variable column. To fit this model in R we have

```r
library(ggplot2)
library(faraway)

# Forces R to recognize that 0, 1 are categorical, also
# relabels the levels to something I understand.
teengamb$sex <- factor(teengamb$sex, labels=c('Male','Female'))

# Fit a linear model with the interaction of sex and income
# Interactions can be specified useing a colon :
m1 <- lm( gamble ~ sex + income + sex:income, data=teengamb )

# R allows a shortcut for the prior definition
m1 <- lm( gamble ~ sex * income, data=teengamb )

# save the fit, lwr, upr values for each observation
# these are the yhat and CI
```

```
teengamb <- cbind(teengamb, predict(m1, interval='conf'))

# Make a nice plot that includes the regreesion line.
ggplot(teengamb, aes(x=income, col=sex, fill=sex)) +
      geom_ribbon(aes(ymin=lwr, ymax=upr),
                  alpha=.3) +   # how solid the layer is
      geom_point(aes(y=gamble)) +
      geom_line(aes(y=fit))
```



```
# print the model summary
summary(m1)
```

```
##
## Call:
## lm(formula = gamble ~ sex * income, data = teengamb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -56.522  -4.860  -1.790   6.273  93.478
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -2.6596     6.3164  -0.421  0.67580
## sexFemale           5.7996    11.2003   0.518  0.60724
## income              6.5181     0.9881   6.597 4.95e-08 ***
## sexFemale:income   -6.3432     2.1446  -2.958  0.00502 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.98 on 43 degrees of freedom
## Multiple R-squared:  0.5857, Adjusted R-squared:  0.5568
## F-statistic: 20.26 on 3 and 43 DF,  p-value: 2.451e-08
```

4.3 Iris Example

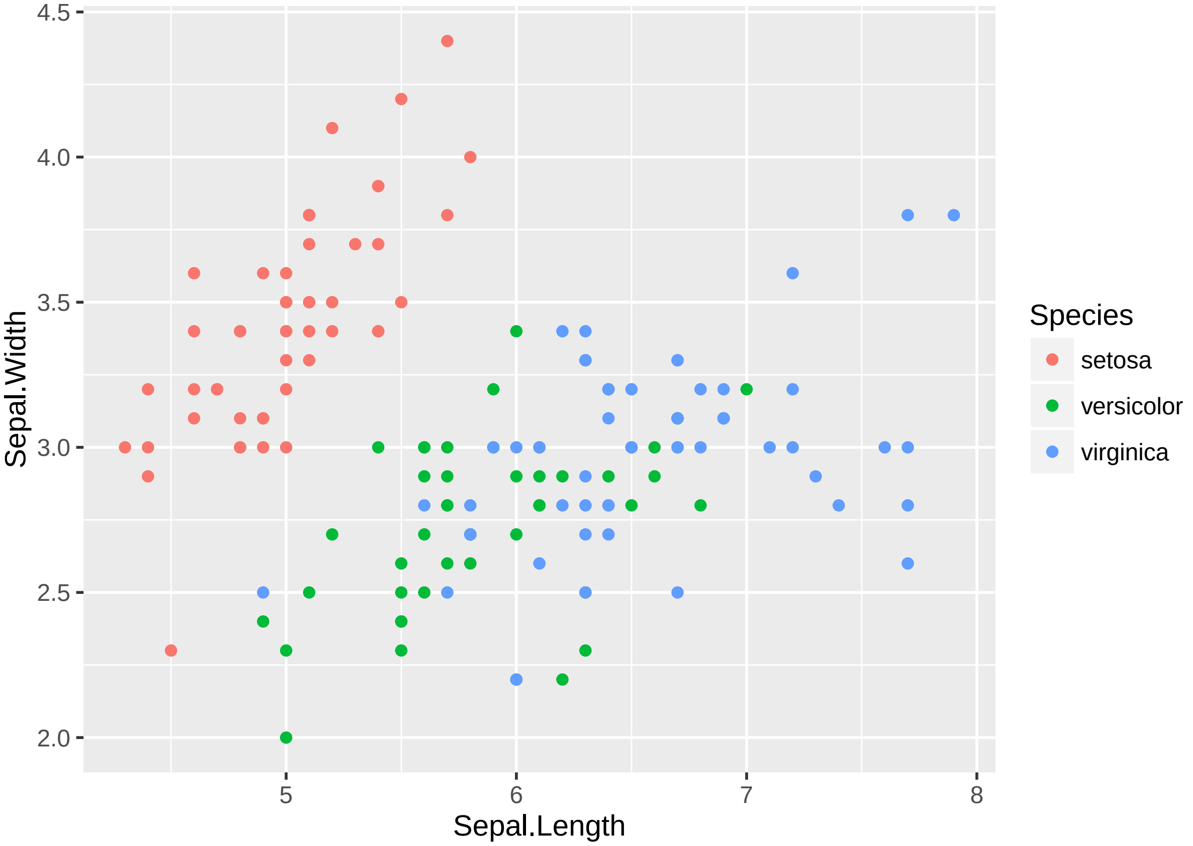
For a second example, we will explore the relationship between sepal length and sepal width for three species of irises. This data set is available in R as `iris`.

```
data(iris)            # read in the iris dataset
levels(iris$Species)  # notice the order of levels of Species
```

```
## [1] "setosa"     "versicolor" "virginica"
```

The very first thing we should do when encountering a dataset is to do some sort of graphical summary to get an idea of what model seems appropriate.

```
ggplot(iris, aes(x=Sepal.Length, y=Sepal.Width, color=Species)) +
  geom_point()
```



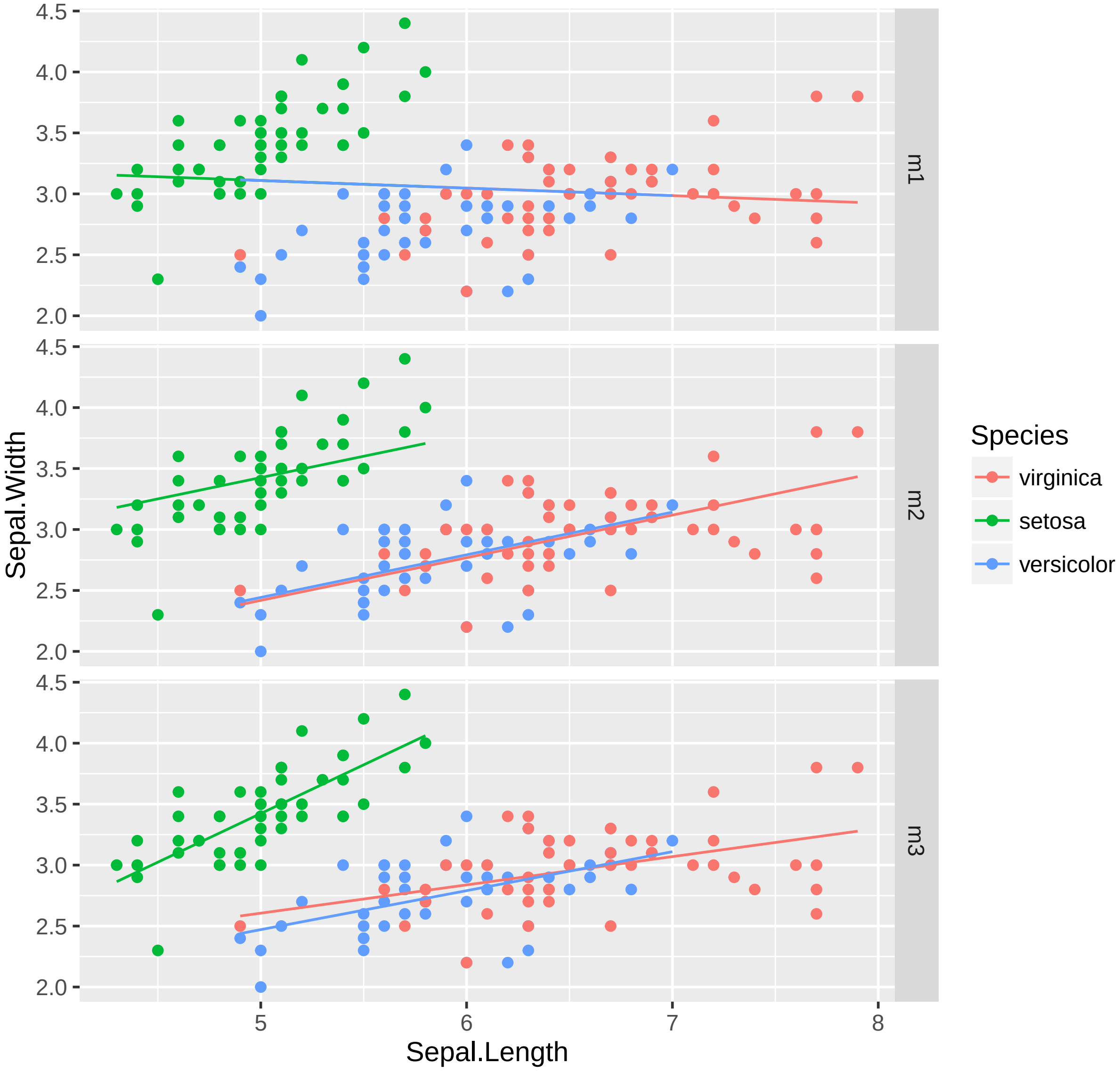
Looking at this graph, it seems that I will likely have a model with different y-intercepts for each species, but it isn't clear to me if we need different slopes.

We consider the sequence of building successively more complex models:

```
# make virginica the reference group
iris$Species <- relevel(iris$Species, ref='virginica')
```

```
m1 <- lm( Sepal.Width ~ Sepal.Length, data=iris )           # One line
m2 <- lm( Sepal.Width ~ Sepal.Length + Species, data=iris )  # Parallel Lines
m3 <- lm( Sepal.Width ~ Sepal.Length * Species, data=iris )  # Non-parallel Lines
```

The three models we consider are the following:



Looking at these, it seems obvious that the simplest model where we ignore Species is horrible. The other two models seem decent, and I am not sure about the parallel lines model vs the differing slopes model.

```
summary(m1)$coefficients %>% round(digits=3)
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.419      0.254  13.484    0.000
## Sepal.Length   -0.062      0.043  -1.440    0.152
```

For the simplest model, there is so much unexplained noise that the slope variable isn't significant.

Moving onto the next most complicated model, where each species has their own y-intercept, but they share a slope, we have

```
summary(m2)$coefficients %>% round(digits=3)
```

```
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)          0.669      0.308   2.174    0.031
## Sepal.Length         0.350      0.046   7.557    0.000
## Speciessetosa        1.008      0.093  10.798    0.000
## Speciesversicolor    0.024      0.065   0.370    0.712
```

The first two lines are the y-intercept and slope associated with the reference group and the last two lines are the y-intercept offsets from the reference group to *Setosa* and *Versicolor*, respectively. We have that the slope associated with increasing Sepal Length is significant and that *Setosa* has a statistically different y-intercept than the reference group *Virginica* and that *Versicolor* does not have a statstically different y-intercept than the reference group.

Finally we consider the most complicated model that includes two more slope parameters

```
summary(m3)$coefficients %>% round(digits=3)
```

```
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       1.446      0.405   3.572    0.000
## Sepal.Length                      0.232      0.061   3.790    0.000
## Speciessetosa                    -2.016      0.686  -2.938    0.004
## Speciesversicolor                -0.574      0.605  -0.950    0.344
## Sepal.Length:Speciessetosa        0.567      0.126   4.490    0.000
## Sepal.Length:Speciesversicolor    0.088      0.097   0.905    0.367
```

These parameters are:

Meaning	R-label
Reference group y-intercept	(Intercept)
Reference group slope	Sepal.Length
offset to y-intercept for Setosa	Speciessetosa
offset to y-intercept for Versicolor	Speciesversicolor
offset to slope for Setosa	Sepal.Length:Speciessetosa
offset to slope for Versicolor	Sepal.Length:Speciesversicolor

It appears that slope for *Setosa* is different from the reference group *Virginica*. However because we've added 2 parameters to the model, testing Model2 vs Model3 is not equivalent to just looking at the p-value for that one slope. Instead we need to look at the F-test comparing the two models which will evaluate if the decrease in SSE is sufficient to justify the addition of two paramters.

```
anova(m2, m3)
```

```
## Analysis of Variance Table
##
## Model 1: Sepal.Width ~ Sepal.Length + Species
## Model 2: Sepal.Width ~ Sepal.Length * Species
##   Res.Df    RSS Df Sum of Sq      F   Pr(>F)
## 1    146 12.193
## 2    144 10.680  2    1.5132 10.201 7.19e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The F-test concludes that there is sufficent decrease in the SSE to justify adding two additional parameters to the model.

4.4 Exercises

1. The in the `faraway` package, there is a dataset named `phbirths` that gives babies birth weights along with their gestational time in utero along with the mother's smoking status.
   a. Load and inspect the dataset using

   ```
   library(faraway)     # load the package
   data(phbirths)       # load the data within the package
   ?phbirths
   ```

   b. Create a plot of the birthweight vs the gestational age. Color code the points based on the mother's smoking status. Does it appear that smoking matters?
   c. Fit the simple model (one regression line) along with both the main effects (parallel lines) and interaction (non-parallel lines) ANCOVA model to these data. Which model is preferred?
   d. Using whichever model you selected in the previous section, create a graph of the data along with the confidence region for the regression line(s).
   e. Now consider only the "full term babies" which are babies with gestational age at birth ≥ 36 weeks. With this reduced dataset, repeat parts c,d.
   f. Interpret the relationship between gestational length and mother's smoking status on birthweight.
2. The in the `faraway` package, there is a dataset named `clot` that gives information about the time for blood to clot verses the blood dilution concentration when the blood was diluted with prothrombin-free plasma. Unfortunately the researchers had to order the plasma in two different lots (could think of this as two different sources) and need to assertain if the lot number makes any difference in clotting time.
   a. Log transform the `time` and `conc` variable and plot the log-transformed data with color of the data point indicating the lot number.
   b. Ignoring the slight remaining curvature in the data, perform the appropriate analysis using transformed variables. Does `lot` matter?
3. In the `faraway` package, there is a data set `ToothGrowth` which is data from an experiment giving Vitamin C to guinea pigs. Guinea pigs were give vitamin C doses either via orange juice or ascorbic acid and the response of interest was a measure of tooth growth.
   a. Log transform the `dose` and use that throughout this problem. Use e as the base, which R does by default when you use the `log()` function.
   b. Graph the data, fit appropriate ANCOVA models, and describe the relationship between the delivery method, log(dose) level, and tooth growth. Produce a graph with the data and the regression line(s) along with the confidence region for the line(s).
   c. Just using your graphs and visual inspection, at low dose levels, say $\log(dose) = -0.7$, is there a difference in delivery method? What about at high dose levels, say $\log(dose) = 0.7$? At this point we don't know how to answer this question using appropriate statistical inference, but we will address this in the chapter on contrasts.

Chapter 5

Contrasts

We often are interested in estimating a function of the parameters $\boldsymbol{\beta}$. For example in the offset representation of the ANOVA model with 3 groups we have

$$y_{ij} = \mu + \tau_i + \epsilon_{ij}$$

where

$$\boldsymbol{\beta} = [\mu \; \tau_2 \; \tau_3]^T$$

and μ is the mean of the control group, group one is the control group and thus $\tau_1 = 0$, and τ_2 and τ_3 are the offsets of group two and three from the control group. In this representation, the mean of group two is $\mu + \tau_2$ and is estimated with $\hat{\mu} + \hat{\tau}_2$.

5.1 Estimate and variance

A contrast is a linear combinations of elements of $\hat{\boldsymbol{\beta}}$, which is a fancy way of saying that it is a function of the elements of $\hat{\boldsymbol{\beta}}$ where the elements can be added, subtracted, or multiplied by constants. In particular, the contrast can be represented by the vector $\boldsymbol{c}$ such that the function we are interested in is $\boldsymbol{c}^T\hat{\boldsymbol{\beta}}$.

In the ANOVA case with $k = 3$ where we have the offset representation, I might be interested in the mean of group 2, which could be written as

$$\mu + \tau_2 = \underbrace{\left[\begin{array}{ccc} 1 & 1 & 0 \end{array}\right]}_{\boldsymbol{c}^T} \cdot \underbrace{\left[\begin{array}{c} \hat{\mu} \\ \hat{\tau}_2 \\ \hat{\tau}_3 \end{array}\right]}_{\hat{\boldsymbol{\beta}}}$$

Similarly in the simple regression case, I will be interested in the height of the regression line at x_0. This height can be written as

$$\hat{\beta}_0 + \hat{\beta}_1 x_0 = \underbrace{\left[\begin{array}{cc} 1 & x_0 \end{array}\right]}_{\boldsymbol{c}^T} \cdot \underbrace{\left[\begin{array}{c} \hat{\beta}_0 \\ \hat{\beta}_1 \end{array}\right]}_{\hat{\boldsymbol{\beta}}}$$

In this manner, we could think of the predicted values $\hat{y}_i$ as just the result of the contrasts $\boldsymbol{X}^T\hat{\boldsymbol{\beta}}$ where our design matrix takes the role of the contrasts.

One of the properties of maximum likelihood estimator (MLEs), is that they are invariant under transformations. Meaning that since $\hat{\boldsymbol{\beta}}$ is the MLE of $\boldsymbol{\beta}$, then $\boldsymbol{c}^T\hat{\boldsymbol{\beta}}$ is the MLE of $\boldsymbol{c}^T\boldsymbol{\beta}$. The only thing we need to perform hypotheses tests and create confidence intervals is an estimate of the variance of $\boldsymbol{c}^T\hat{\boldsymbol{\beta}}$.

Because we know the variance of $\hat{\boldsymbol{\beta}}$ is

$$Var\left(\hat{\boldsymbol{\beta}}\right)=\sigma^{2}\left(\boldsymbol{X}^{T}\boldsymbol{X}\right)^{-1}$$

and because $\boldsymbol{c}$ is a constant, then

$$Var\left(\boldsymbol{c}^{T}\hat{\boldsymbol{\beta}}\right)=\sigma^{2}\boldsymbol{c}^{T}\left(\boldsymbol{X}^{T}\boldsymbol{X}\right)^{-1}\boldsymbol{c}$$

and the standard error is found by plugging in our estimate of σ^2 and taking the square root.

$$StdErr\left(\boldsymbol{c}^{T}\hat{\boldsymbol{\beta}}\right)=\sqrt{\hat{\sigma}^{2}\boldsymbol{c}^{T}\left(\boldsymbol{X}^{T}\boldsymbol{X}\right)^{-1}}\boldsymbol{c}=\hat{\sigma}\sqrt{\boldsymbol{c}^{T}\left(\boldsymbol{X}^{T}\boldsymbol{X}\right)^{-1}\boldsymbol{c}}$$

As usual, we can now calculate confidence intervals for $\boldsymbol{c}^{T}\hat{\boldsymbol{\beta}}$ using the usual formula

$$Est\pm t_{n-p}^{1-\alpha/2}\;StdErr\left(\,Est\,\right)$$

$$\boldsymbol{c}^{T}\hat{\boldsymbol{\beta}}\pm t_{n-p}^{1-\alpha/2}\;\hat{\sigma}\sqrt{\boldsymbol{c}^{T}\left(\boldsymbol{X}^{T}\boldsymbol{X}\right)^{-1}\boldsymbol{c}}$$

Recall the hostility example which was an ANOVA with three groups with the data

Method	Test Scores
1	96 79 91 85 83 91 82 87
2	77 76 74 73 78 71 80
3	66 73 69 66 77 73 71 70 74

We have analyzed this data using both the cell means model and the offset and we will demonstrate how to calculate the group means from the offset representation. Thus we are interested in estimating $\mu+\tau_2$ and $\mu+\tau_3$. I am also interested in estimating the difference between treatment 2 and 3 and will therefore be interested in estimating $\tau_2-\tau_3$.

```
y <- c(96,79,91,85,83,91,82,87,
       77,76,74,73,78,71,80,
       66,73,69,66,77,73,71,70,74)
groups <- factor(c( rep('Group1',8), rep('Group2',7),rep('Group3',9) ))
```

We can fit the offset model and obtain the design matrix and estimate of $\hat{\sigma}$ via the following code.

```
m <- lm(y ~ groups)            # Fit the ANOVA model (offset representation)
coef(m)                        # Show me beta.hat
```

```
##  (Intercept) groupsGroup2 groupsGroup3
##     86.75000    -11.17857    -15.75000
```

```
X <- model.matrix(m)           # obtains the design matrix
sigma.hat <- summary(m)$sigma  # create the summary table and grab sigma.hat
beta.hat <- coef(m)
XtX.inv <- solve( t(X) %*% X )
```

Now we calculate

```
contr <- c(1,1,0)  # define my contrast
ctb <- t(contr) %*% beta.hat
std.err <- sigma.hat * sqrt( t(contr) %*% XtX.inv %*% contr )
```

```
ctb
```

```
##          [,1]
## [1,] 75.57143
```

```
std.err
```

```
##          [,1]
## [1,] 1.622994
```

and notice this is the exact same estimate and standard error we got for group two when we fit the cell means model.

```
CellMeansModel <- lm(y ~ groups - 1)
summary(CellMeansModel)$coefficients
```

```
##              Estimate Std. Error  t value     Pr(>|t|)
## groupsGroup1 86.75000   1.518172 57.14108 1.567052e-24
## groupsGroup2 75.57143   1.622994 46.56296 1.117034e-22
## groupsGroup3 71.00000   1.431347 49.60364 2.993023e-23
```

5.2 Estimating contrasts in R

Instead of us doing all the matrix calculations ourselves, all we really need is to is specify the row vector $\boldsymbol{c}^T$. The function that will do the rest of the calculations is the generalized linear hypothesis test function `glht()` that can be found in the multiple comparisons package `multcomp`. The p-values will be adjusted to correct for testing multiple hypothesis, so there may be slight differences compared to the p-value seen in just the regular summary table.

5.2.1 1-way ANOVA

We will again use the Hostility data set and demonstrate how to calculate the point estimates, standard errors and confidence intervals for the group means given a model fit using the offset representation.

```
y <- c(96,79,91,85,83,91,82,87,
       77,76,74,73,78,71,80,
       66,73,69,66,77,73,71,70,74)
groups <- factor(c( rep('Group1',8), rep('Group2',7),rep('Group3',9) ))
m <- lm(y ~ groups)
summary(m)$coefficients
```

```
##               Estimate Std. Error   t value     Pr(>|t|)
## (Intercept)   86.75000   1.518172 57.141079 1.567052e-24
## groupsGroup2 -11.17857   2.222377 -5.030008 5.583942e-05
## groupsGroup3 -15.75000   2.086528 -7.548424 2.063600e-07
```

We will now define a row vector (and it needs to be a matrix or else `glht()` will throw an error. First we note that the simple contrast $\boldsymbol{c}^T = [1\ 0\ 0]$ just grabs the first coefficient and gives us the same estimate and standard error as the summary did.

```
library(multcomp)
contr <- rbind("Intercept"=c(1,0,0)) # 1x3 matrix with row named "Intercept"
test <- glht(m, linfct=contr)        # the linear function to be tested is contr
summary(test)
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
```

```
##
## Fit: lm(formula = y ~ groups)
##
## Linear Hypotheses:
##                Estimate Std. Error t value Pr(>|t|)
## Intercept == 0   86.750      1.518   57.14   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

Next we calculate the estimate of all the group means μ, $\mu+\tau_2$ and $\mu+\tau_3$ and the difference between group 2 and 3. Notice I can specify more than one contrast at a time.

```
contr <- rbind("Mean of Group 1"=c(1,0,0),
               "Mean of Group 2"=c(1,1,0),
               "Mean of Group 3"=c(1,0,1),
               "Diff G2-G3"     =c(0,1,-1))
test <- glht(m, linfct=contr)
summary(test)
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Fit: lm(formula = y ~ groups)
##
## Linear Hypotheses:
##                      Estimate Std. Error t value Pr(>|t|)
## Mean of Group 1 == 0   86.750      1.518  57.141   <0.001 ***
## Mean of Group 2 == 0   75.571      1.623  46.563   <0.001 ***
## Mean of Group 3 == 0   71.000      1.431  49.604   <0.001 ***
## Diff G2-G3 == 0         4.571      2.164   2.112    0.144
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

Finally we calculate confidence intervals in the usual manner using the `confint()` function.

```
confint(test, level=0.95)
```

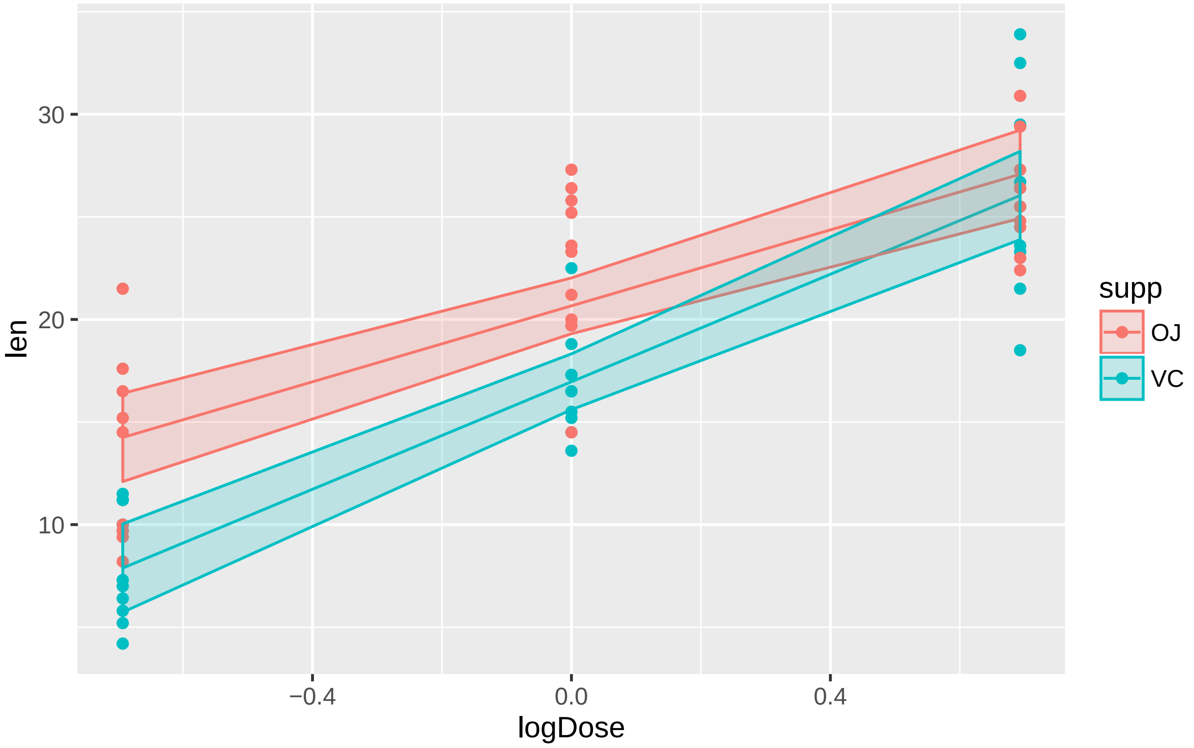
```
##
##   Simultaneous Confidence Intervals
##
## Fit: lm(formula = y ~ groups)
##
## Quantile = 2.6454
## 95% family-wise confidence level
##
##
## Linear Hypotheses:
##                      Estimate lwr     upr
## Mean of Group 1 == 0 86.7500  82.7338 90.7662
## Mean of Group 2 == 0 75.5714  71.2779 79.8649
## Mean of Group 3 == 0 71.0000  67.2135 74.7865
## Diff G2-G3 == 0       4.5714  -1.1532 10.2961
```

5.2.2 ANCOVA example

In the ANCOVA case, there are more interesting contrasts to be made. For this example we will use the `ToothGrowth` dataset from the faraway package. This data measuring the effects of different dose levels of vitamin C on tooth growth of guinea pigs. The two different delivery methods are encoded by the variable supp which has levels of orange juice (OJ) and ascorbic acid (VC).

We first fit a ANCOVA model with an interaction between log(dose) level and delivery method and graph the result.

```
library(ggplot2)
library(faraway)
data(ToothGrowth)
ToothGrowth$logDose <- log( ToothGrowth$dose )
m <- lm(len ~ logDose * supp, data=ToothGrowth)

# predict() gives me the yhat values and optional CI
# these are just the "contrasts" defined by X matrix!
ToothGrowth$len.hat <- predict(m, interval='confidence')[,1]
ToothGrowth$len.lwr <- predict(m, interval='confidence')[,2]
ToothGrowth$len.upr <- predict(m, interval='confidence')[,3]

# Plot the results using ggplot2
ggplot( ToothGrowth, aes(x=logDose, col=supp, fill=supp)) +
    geom_point(aes(y=len)) +
    geom_line(aes(y=len.hat)) +
    geom_ribbon(aes(ymin=len.lwr, ymax=len.upr), alpha=.2)
```



R has fit this model using the offset representation. First we present the summary coefficients so we know which parameters are which.

```
summary(m)$coefficient
```

```
##                  Estimate Std. Error   t value     Pr(>|t|)
## (Intercept)     20.663333  0.6791481 30.425371 1.629404e-36
## logDose          9.254889  1.2000095  7.712346 2.302639e-10
## suppVC          -3.700000  0.9604605 -3.852319 3.033467e-04
## logDose:suppVC   3.844782  1.6970696  2.265542 2.736578e-02
```

For a warm-up, we will calculate the y-intercepts of both groups and the slopes of both. For the OJ group, this is just the 1st and 2nd coefficients, while for the VC group it is $\beta_0 + \beta_2$ and $\beta_1 + \beta_3$.

```
# Add a heading so that I know which parameter is which!
#                                Int   logDose   suppVC    logDose:suppVC
contr <- rbind("Intercept OJ" = c(1,     0,      0,           0          ),
               "Slope     OJ" = c(0,     1,      0,           0          ),
               "Intercept VC" = c(1,     0,      1,           0          ),
               "Slope     VC" = c(0,     1,      0,           1          ) )
test <- glht(m, linfct=contr)
summary(test)
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Fit: lm(formula = len ~ logDose * supp, data = ToothGrowth)
##
## Linear Hypotheses:
##                   Estimate Std. Error t value Pr(>|t|)
## Intercept OJ == 0  20.6633     0.6791  30.425   <1e-09 ***
## Slope     OJ == 0   9.2549     1.2000   7.712   <1e-09 ***
## Intercept VC == 0  16.9633     0.6791  24.977   <1e-09 ***
## Slope     VC == 0  13.0997     1.2000  10.916   <1e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

In our original table of summary coefficients, the (intercept) term corresponds to the tooth growth of a guinea pig when the logDose level is 0 (and therefore dose=1). If I wanted to estimate the tooth growth of a guinea pig fed OJ but with only 1/2 a dose (and therefore $logDose = \log(1/2) = -0.69$), then we want

$$\begin{aligned}
\hat{y}_{oj,dose=0.5} &= \hat{\beta}_0 + \hat{\beta}_1 \log(0.5) \\
&= 20.6 + 9.25 \log(0.5) \\
&= 20.6 + 9.25\,(-0.69) \\
&= 14.21
\end{aligned}$$

which I can write as

$$y_{oj,dose=0.5} = \left[\begin{array}{cccc} 1 & -0.69 & 0 & 0 \end{array}\right] \left[\begin{array}{c} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \hat{\beta}_2 \\ \hat{\beta}_3 \end{array}\right] = \boldsymbol{c}_1^T \hat{\boldsymbol{\beta}}$$

To calculate the same value for the VC group, we need the following contrast:

$$
\begin{aligned}
\hat{y}_{vc,dose=0.5} &= 16.9633 + 13.0997 \log(0.5) \\
&= (20.66 - 3.7) + (9.25 + 3.8) \log(0.5) \\
&= \left(\hat{\beta}_0 + \hat{\beta}_2\right) + \left(\hat{\beta}_1 + \hat{\beta}_4\right)(-0.69) \\
&= \left[\begin{array}{cccc} 1 & -0.69 & 1 & -0.69 \end{array}\right] \left[\begin{array}{c} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \hat{\beta}_2 \\ \hat{\beta}_3 \end{array}\right] \\
&= \boldsymbol{c}_2^T \hat{\boldsymbol{\beta}}
\end{aligned}
$$

```
#                                Int  logDose  suppVC   logDose:suppVC
contr <- rbind("OJ; 1/2 dose" = c(1,   -0.69,    0,          0         ),
               "VC; 1/2 dose" = c(1,   -0.69,    1,        -0.69       ))
test <- glht(m, linfct=contr)
summary(test)
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Fit: lm(formula = len ~ logDose * supp, data = ToothGrowth)
##
## Linear Hypotheses:
##                     Estimate Std. Error t value Pr(>|t|)
## OJ; 1/2 dose == 0   14.277      1.071   13.33  < 1e-10 ***
## VC; 1/2 dose == 0    7.925      1.071    7.40 1.51e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

Finally we might be interested in testing if there is a treatment difference between OJ and VC at a 1/2 dose level. So to do this, we want to calculate the difference between these two previous contrasts, i.e.

$$
\begin{aligned}
\boldsymbol{c}_1^T \hat{\boldsymbol{\beta}} - \boldsymbol{c}_2^T \hat{\boldsymbol{\beta}} &= \left(\boldsymbol{c}_1^T - \boldsymbol{c}_2^T\right) \hat{\boldsymbol{\beta}} \\
&= \left(\left[\begin{array}{cccc} 1 & -0.69 & 0 & 0 \end{array}\right] - \left[\begin{array}{cccc} 1 & -0.69 & 1 & -0.69 \end{array}\right]\right) \hat{\boldsymbol{\beta}} \\
&= \left[\begin{array}{cccc} 0 & 0 & -1 & 0.69 \end{array}\right] \hat{\boldsymbol{\beta}}
\end{aligned}
$$

and we can calculate

```
contr <- rbind("OJ - VC; 1/2 dose" = c(0, 0, -1, 0.69))
test <- glht(m, linfct=contr)
summary(test)
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Fit: lm(formula = len ~ logDose * supp, data = ToothGrowth)
##
## Linear Hypotheses:
##                          Estimate Std. Error t value Pr(>|t|)
## OJ - VC; 1/2 dose == 0    6.353      1.514   4.195 9.83e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

When we do the same test, but at dose level 2 (logDose $=\log 2 = 0.69$) we see that the difference is not statistically significant.

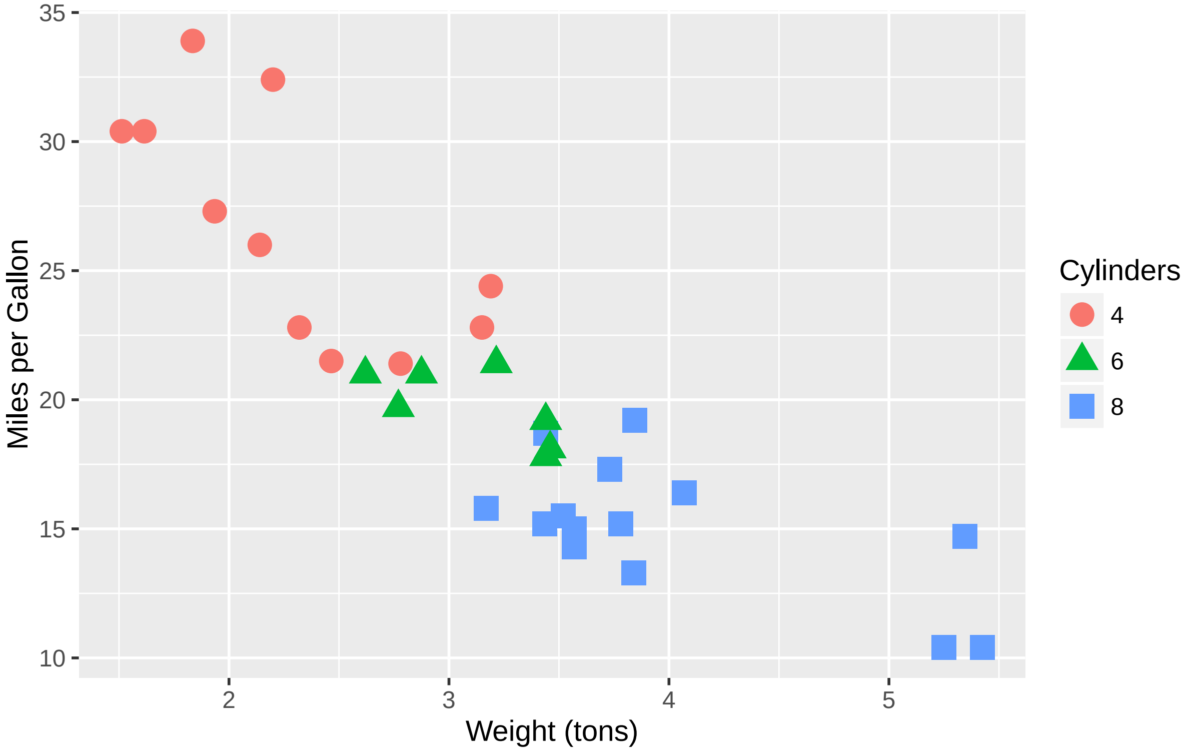
```r
contr <- rbind("OJ - VC; dose 2" = c(0, 0, -1, -0.69))
test <- glht(m, linfct=contr)
summary(test)
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Fit: lm(formula = len ~ logDose * supp, data = ToothGrowth)
##
## Linear Hypotheses:
##                       Estimate Std. Error t value Pr(>|t|)
## OJ - VC; dose 2 == 0    1.047      1.514   0.691    0.492
## (Adjusted p values reported -- single-step method)
```

5.3 Exercises

1. We will examine a dataset that summarizes cars featured in the magazine Motor Trend during the year 1974. In particular we will examine the relationship between the vehicles gas milage (`mpg`) versus the weight (`wt`) and number of cylinders (`cyl`) of the vehicle.

```r
library(ggplot2)
# treat cylinders as categorical
data(mtcars)
mtcars$cyl <- factor(mtcars$cyl)
ggplot(mtcars, aes(x=wt, col=cyl)) +
  geom_point(aes(y=mpg, shape=cyl), size=4) +
  xlab('Weight (tons)') +
  ylab('Miles per Gallon') +
  labs(color='Cylinders', shape='Cylinders')
```



a. Fit the model that predicts mpg using both weight and number of cylinders and their interaction using the command

```
model <- lm(mpg ~ wt*cyl, data=mtcars)
```

b. Create a vector of fitted values, add it to the data frame mtcars and then create a plot that includes the regression lines.

c. Denote the coefficients obtained from the summary of your `lm()` call as $\hat{\beta}_1$ to $\hat{\beta}_6$ (in the order given by R). Write your interaction model out using subscript notation and should be in the form

$$y_i = \begin{cases} ??????? & \text{if 4 cylinder} \\ ??????? & \text{if 6 cylinder} \\ ??????? & \text{if 8 cylinder} \end{cases}$$

and give an interpretation for each β_j value.

i. $\hat{\beta}_1 = ???$
ii. $\hat{\beta}_2 = ???$
iii. $\hat{\beta}_3 = ???$
iv. $\hat{\beta}_4 = ???$
v. $\hat{\beta}_5 = ???$
vi. $\hat{\beta}_6 = ???$

d. What is the estimated mpg of a 6-cylinder vehicle weighing 3.5 tons? Give an associated 95% confidence interval. Calculate this using the `predict()` function. *Warning: When making the new data frame, make sure that R interprets cyl as a factor by either coercing it to a factor after creation or inputing the cylinder as a string (i.e. as "6" instead of 6).*

e. Answer the previous question but using the `glht()` function.

f. What is the estimated mpg (and 95% confidence interval) of an 8-cylinder vehicle weighing 3.5 tons? Compute this using the `glht()` function.

g. What is the estimated difference (and 95% confidence interval) in mpg between the 8 and 6 cylinder vehicles that weigh 3.5 tons? Is there a statistically significant difference in mpg at 3.5 tons?

h. Are the slopes of the 8-cylinder and 6-cylinder vehicles statistically different?

Chapter 6

Diagnostics and Transformations

We will be interested in analyzing whether or not our linear model is a good model and whether or not the data violate any of the assumptions that are required. In general we will be interested in three classes of assumption violations and our diagnostic measures might be able detect one or more of the following issues:

1. Unusual observations that contribute too much influence to the analysis. These few observations might drastically change the outcome of the model.
2. Model misspecification. Our assumption that $E\left[\boldsymbol{y}\right]=\boldsymbol{X}\boldsymbol{\beta}$ might be wrong and we might need to include different covariates in the model to get a satisfactory result.
3. Error distribution. We have assumed that $\boldsymbol{\epsilon}\sim MVN\left(\boldsymbol{0},\sigma^{2}\boldsymbol{I}\right)$ but autocorrelation, heteroscedasticity, and non-normality might be present.

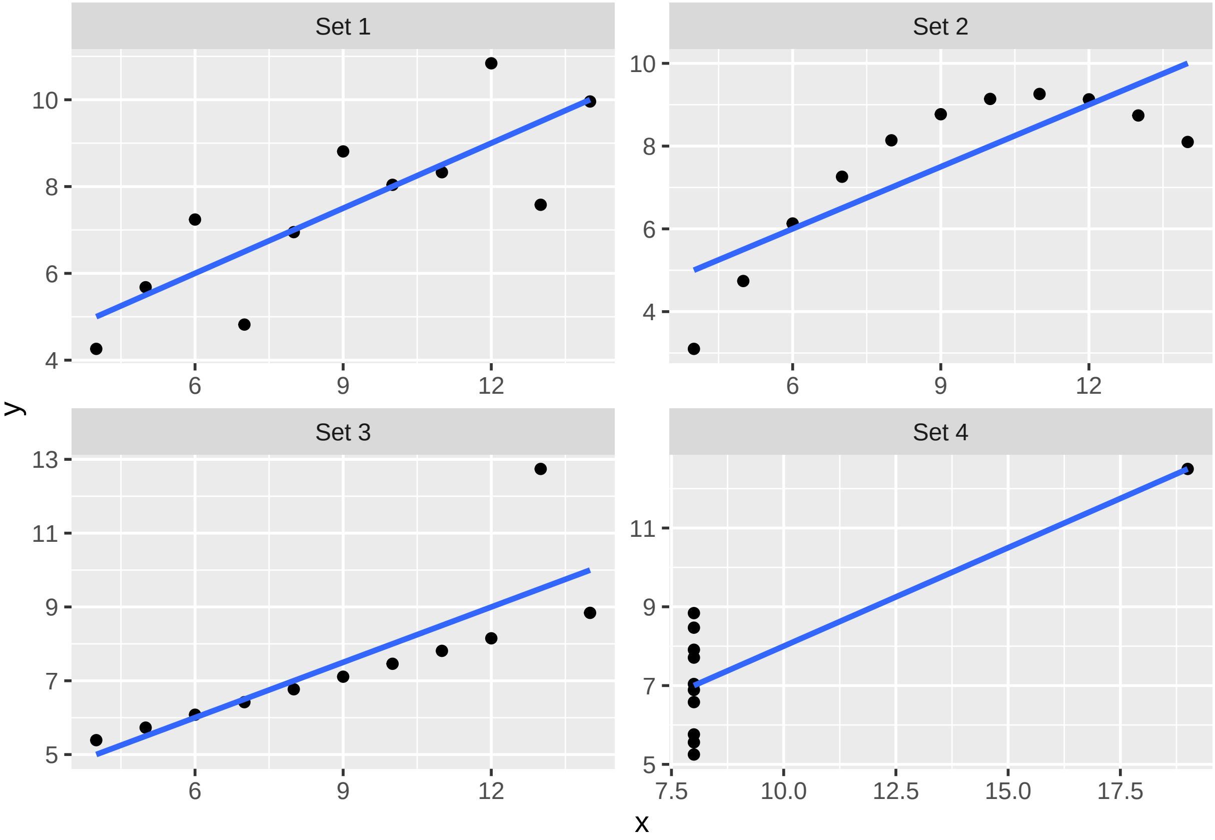
Often problems with one of these can be corrected by tranforming either the explanatory or response variables.

6.1 Detecting Assumption Violations

Throughout this chapter I will use data created by Francis Anscombe that show how simple linear regression can be misused. In particular, these data sets will show how our diagnostic measures will detect various departures from the model assumptions.

The data are available in R as a data frame `anscombe` and is loaded by default. The data consists of four datasets, each having the same linear regression $\hat{y}=3+0.5\,x$ but the data are drastically different.

```
library(ggplot2)
library(dplyr)

# The anscombe dataset has 8 columns - x1,x2,x3,x4,y1,y2,y3,y4
# and I want it to have 3 columns - Set, X, Y
Anscombe <- rbind(
  data.frame(x=anscombe$x1, y=anscombe$y1, set='Set 1'),
  data.frame(x=anscombe$x2, y=anscombe$y2, set='Set 2'),
  data.frame(x=anscombe$x3, y=anscombe$y3, set='Set 3'),
  data.frame(x=anscombe$x4, y=anscombe$y4, set='Set 4'))

# order them by their x values, and add an index column
Anscombe <- Anscombe %>%
  group_by(set) %>%       # Every subsequent action happens by dataset
  arrange(x,y) %>%        # sort them on the x-values and if tied, by y-value
  mutate( index = 1:n() ) # give each observation within a set, an ID number

# Make a nice graph
ggplot(Anscombe, aes(x=x, y=y)) +
 geom_point() +
 facet_wrap(~set, scales='free') +
 stat_smooth(method="lm", formula=y~x, se=FALSE)
```



6.1.1 Measures of Influence

6.1.1.1 Standardized Residuals (aka Studentized)

Recall that we have

$$\begin{aligned}\hat{\boldsymbol{y}} &= \boldsymbol{X}\hat{\boldsymbol{\beta}} \\ &= \boldsymbol{X}\left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{y} \\ &= \boldsymbol{H}\boldsymbol{y}\end{aligned}$$

where the "Hat Matrix" is $\boldsymbol{H} = \boldsymbol{X}\left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T$ because we have $\hat{\boldsymbol{y}} = \boldsymbol{H}\boldsymbol{y}$. The elements of $\boldsymbol{H}$ can be quite useful in diagnostics. It can be shown that the variance of the $i$the residual is

$$Var\left(\hat{\epsilon}_i\right) = \sigma^2\left(1 - \boldsymbol{H}_{ii}\right)$$

where $\boldsymbol{H}_{ii}$ is the ith element of the main diagonal of $\boldsymbol{H}$. This suggests that I could rescale my residuals to

$$\hat{\epsilon}_i^* = \frac{\hat{\epsilon}_i}{\hat{\sigma}\sqrt{1-\boldsymbol{H}_{ii}}}$$

which, if the normality and homoscedasticity assumptions hold, should behave as a $N(0,1)$ sample.

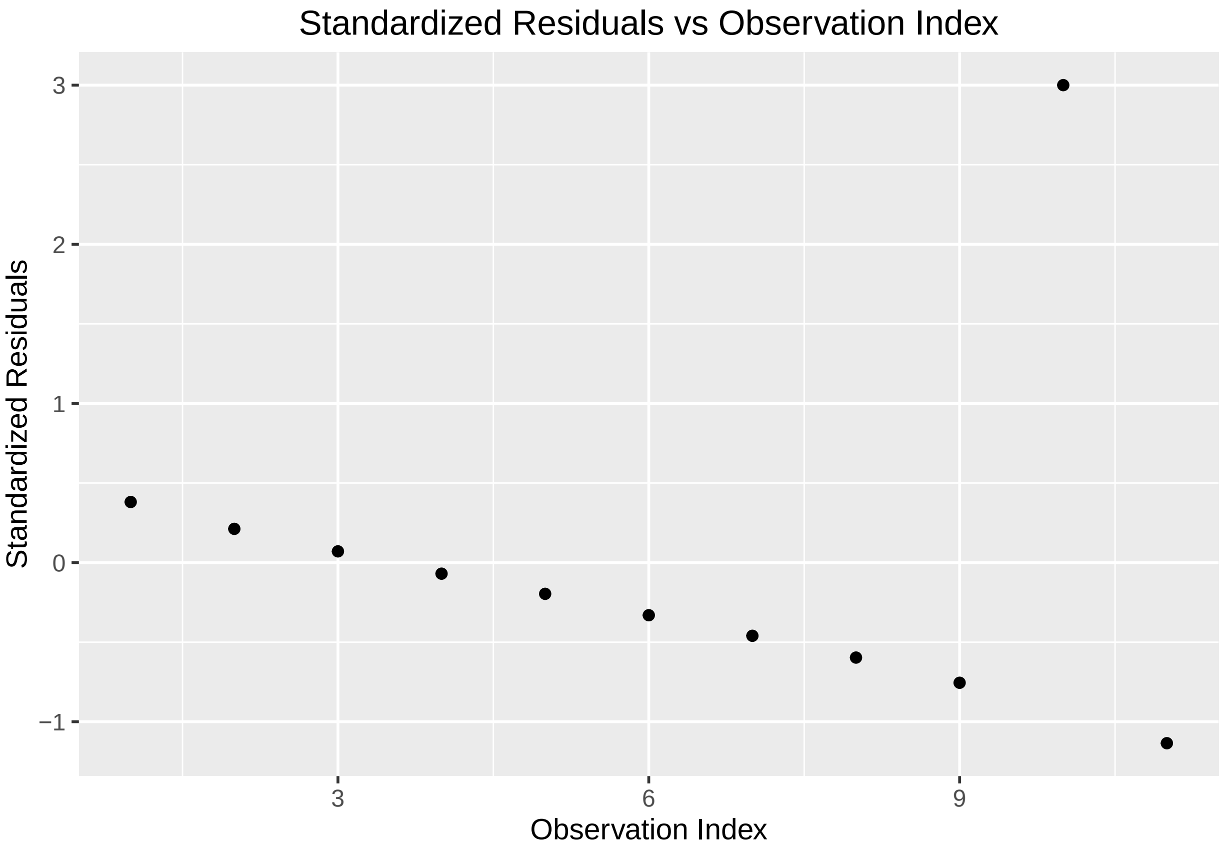
These rescaled residuals are called "studentized residuals", though R typically refers to them as "standardized". Since we have a good intuition about the scale of a standard normal distribution, the scale of standardized residuals will give a good indicator if normality is violated.

There are actually two types of studentized residuals, typically called *internal* and *external* among statisticians. The version presented above is the *internal* version which can be obtained using the R function `rstandard()` while the *external* version is available using `rstudent()`. Whenever you see R present standardized residuals, they are talking about internally studentized residuals. For sake of clarity, I will use the term *standardized* as well.

6.1.1.1.1 Example - Anscombe's set 3

For the third dataset, the outlier is the ninth observation with $x_9 = 13$ and $y_9 = 12.74$. We calculate the standardized residuals using the function `rstandard()` and plot them

```
Set3 <- Anscombe %>% filter(set == 'Set 3') # Just set 3
model <- lm(y ~ x, data=Set3)               # Fit the regression line
Set3$stdresid <- rstandard(model)           # rstandard() returns the standardized residuals

ggplot(Set3, aes(x=index, y=stdresid)) +    # make a plot
  geom_point() +
  labs(x='Observation Index',
       y='Standardized Residuals',
       title='Standardized Residuals vs Observation Index')
```



and we notice that the outlier residual is really big. If the model assumptions were true, then the standardized residuals should follow a standard normal distribution, and I would need to have hundreds of observations before I wouldn't be surprised to see a residual more than 3 standard deviations from 0.

6.1.1.2 Leverage

The extremely large standardized residual suggests that this data point is important, but we would like to quantify how important this observation actually is.

One way to quantify this is to look at the elements of $\boldsymbol{H}$. Because

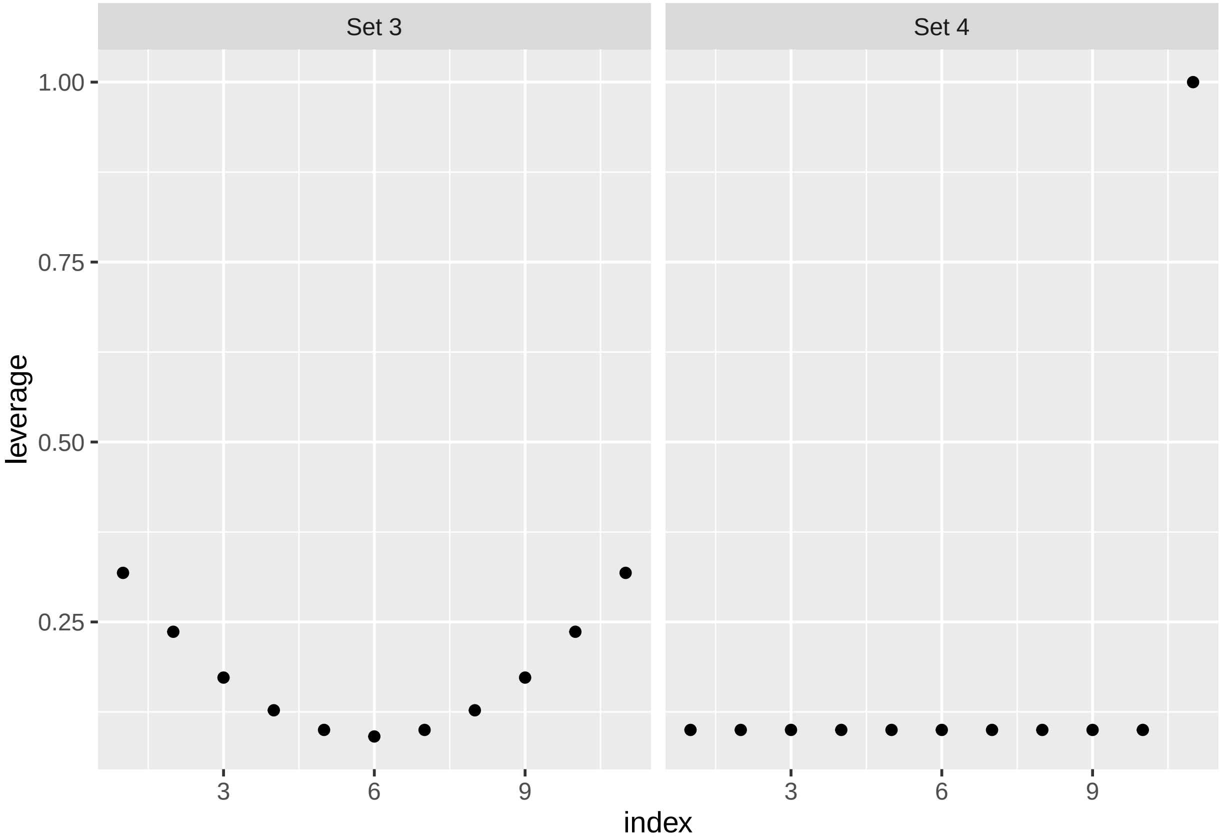
$$\hat{y}_i = \sum_{j=1}^{n} \boldsymbol{H}_{ij} y_j$$

then the ith row of $\boldsymbol{H}$ is a vector of weights that tell us how influential a point y_j is for calculating the predicted value $\hat{y}_i$. If I look at just the main diagonal of $\boldsymbol{H}$, these are how much weight a point has on its predicted value. As such, I can think of the $\boldsymbol{H}_{ii}$ as the amount of leverage a particular data point has on the regression line.

Fortunately there is already a function `hatvalues()` to compute these $\boldsymbol{H}_{ii}$ values for me. We will compare the leverages from Anscombe's set 3 versus set 4.

```
Set3 <- Anscombe %>% filter( set == 'Set 3')
Set4 <- Anscombe %>% filter( set == 'Set 4')

model3 <- lm(y ~ x, data = Set3 )
model4 <- lm(y ~ x, data = Set4 )
```

```
Set3 <- Set3 %>% mutate(leverage = hatvalues(model3))  # add leverage columns
Set4 <- Set4 %>% mutate(leverage = hatvalues(model4))

ggplot( rbind(Set3,Set4), aes(x=index, y=leverage) ) +
  geom_point() +
  facet_grid( . ~ set )
```



This leverage idea only picks out the *potential* for a specific value of x to be influential, but does not actually measure influence. It has picked out the issue with the fourth data set, but does not adequately address the outlier in set 3.

6.1.1.3 Cook's Distance

To attempt to measure the actual influence of an observation $\left\{y_i, \boldsymbol{x}_i^T\right\}$ on the linear model, we consider the effect on the regression if we removed the observation and fit the same model. Let

$$\hat{\boldsymbol{y}} = \boldsymbol{X}\hat{\boldsymbol{\beta}}$$

be the vector of predicted values, where $\hat{\boldsymbol{\beta}}$ is created using all of the data, and $\hat{\boldsymbol{y}}_{(i)} = \boldsymbol{X}\hat{\boldsymbol{\beta}}_{(i)}$ be the vector of predicted values where $\hat{\boldsymbol{\beta}}_{(i)}$ was estimated using all of the data except the ith observation. Letting p be the number of β_j parameters as usual we define Cook's distance of the ith observation as

$$D_i = \frac{\left(\hat{\boldsymbol{y}} - \hat{\boldsymbol{y}}_{(i)}\right)^T \left(\hat{\boldsymbol{y}} - \hat{\boldsymbol{y}}_{(i)}\right)}{p\hat{\sigma}^2}$$

which boils down to saying if the predicted values have large changes when the ith element is removed, then the distance is big. It can be shown that this formula can be simplified to

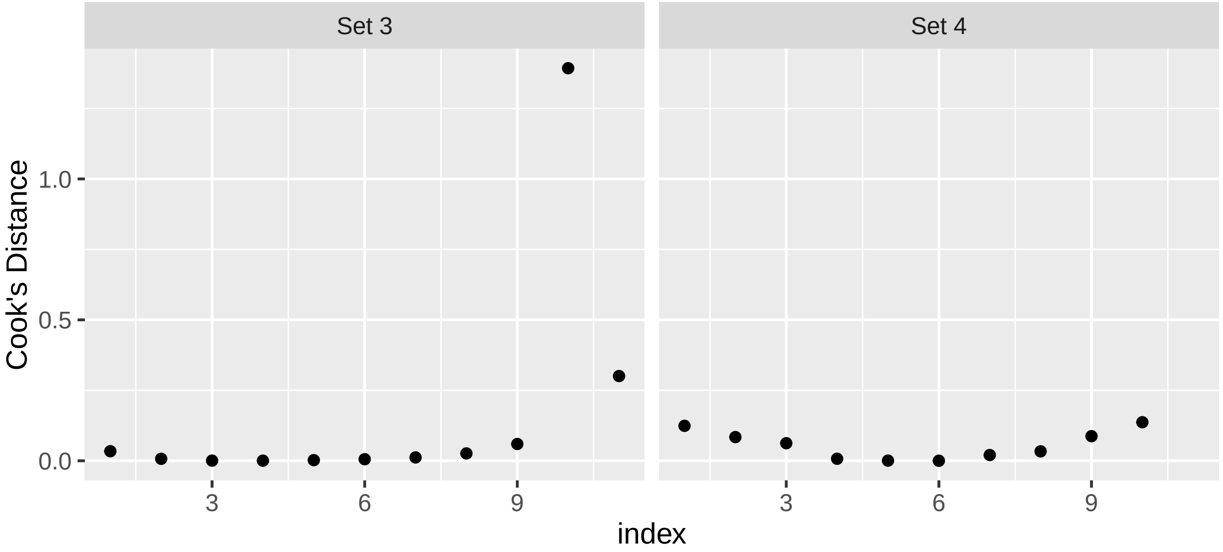
$$D_i = \frac{\hat{\epsilon}_i^* \boldsymbol{H}_{ii}}{p\,(1 - H_{ii})}$$

which expresses Cook's distance in terms of the ith studentized residual and the ith leverage.

Nicely, the R function `cooks.distance()` will calculate Cook's distance.

```
Set3 <- Set3 %>% mutate(cooksd = cooks.distance(model3))
Set4 <- Set4 %>% mutate(cooksd = cooks.distance(model4))

# Note: The high leverage point in set 4 has a Cook's distance of Infinity.
ggplot(rbind(Set3,Set4), aes(x=index, y=cooksd)) +
  geom_point() +
  facet_grid(. ~ set) +
  labs(y="Cook's Distance")
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```



Some texts will give a rule of thumb that points with Cook's distances greater than 1 should be considered influential, while other books claim a reasonable rule of thumb is $4/(n-p-1)$ where n is the sample size, and p is the number of parameters in $\boldsymbol{\beta}$. My take on this, is that you should look for values that are highly different from the rest of your data.

6.1.2 Diagnostic Plots

After fitting a linear model in R, you have the option of looking at diagnostic plots that help to decide if any assumptions are being violated. We will step through each of the plots that are generated by the function `plot(model)` or using `ggplot2` using the package `ggfortify`.

In the package `ggfortify` there is a function that will calculate the diagnostics measures and add them to your dataset. This will simplify our graphing process.

```
Set1 <- Anscombe %>% filter(set == 'Set 1')
model <- lm( y ~ x, data=Set1)
Set1 <- fortify(model)  # add dignostic measures to the dataset
Set1 %>% round(digits=3) # show the dataset nicely
```

```
##        y  x  .hat .sigma .cooksd .fitted .resid .stdresid
## 1   4.26  4 0.318  1.273   0.123   5.000 -0.740    -0.725
## 2   5.68  5 0.236  1.310   0.004   5.501  0.179     0.166
## 3   7.24  6 0.173  1.220   0.127   6.001  1.239     1.102
## 4   4.82  7 0.127  1.147   0.154   6.501 -1.681    -1.455
## 5   6.95  8 0.100  1.311   0.000   7.001 -0.051    -0.043
## 6   8.81  9 0.091  1.218   0.062   7.501  1.309     1.110
## 7   8.04 10 0.100  1.312   0.000   8.001  0.039     0.033
## 8   8.33 11 0.127  1.310   0.002   8.501 -0.171    -0.148
## 9  10.84 12 0.173  1.100   0.279   9.001  1.839     1.635
## 10  7.58 13 0.236  1.056   0.489   9.501 -1.921    -1.778
## 11  9.96 14 0.318  1.311   0.000  10.001 -0.041    -0.041
```

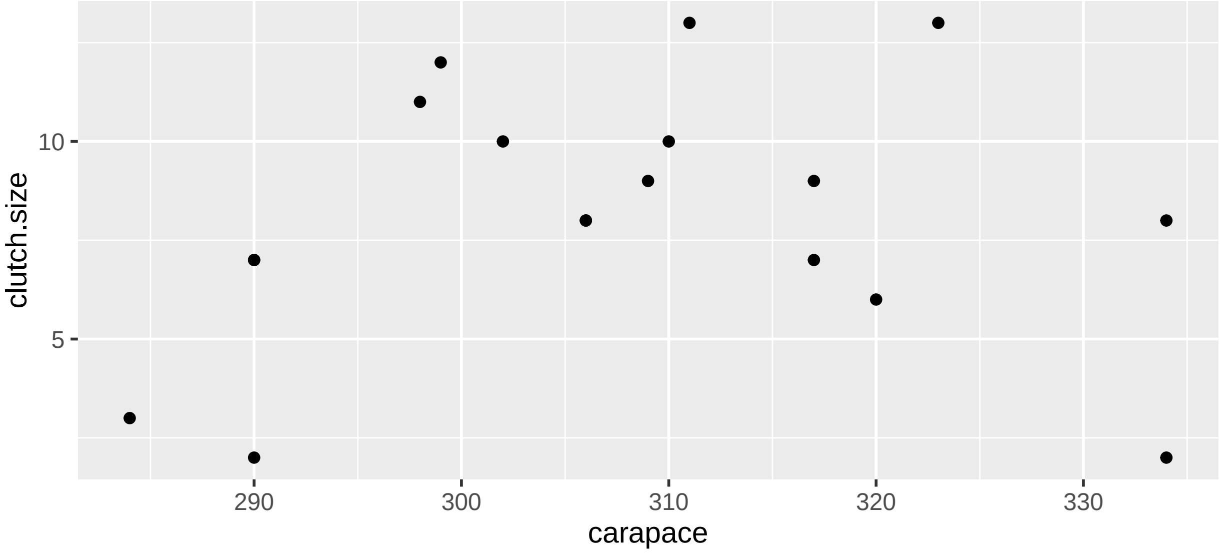
6.1.2.1 Residuals vs Fitted

In the simple linear regression the most useful plot to look at was the residuals versus the x-covariate, but we also saw that this was similar to looking at the residuals versus the fitted values. In the general linear model, we will look at the residuals versus the fitted values or possibly the studentized residuals versus the fitted values.

6.1.2.1.1 Polynomial relationships

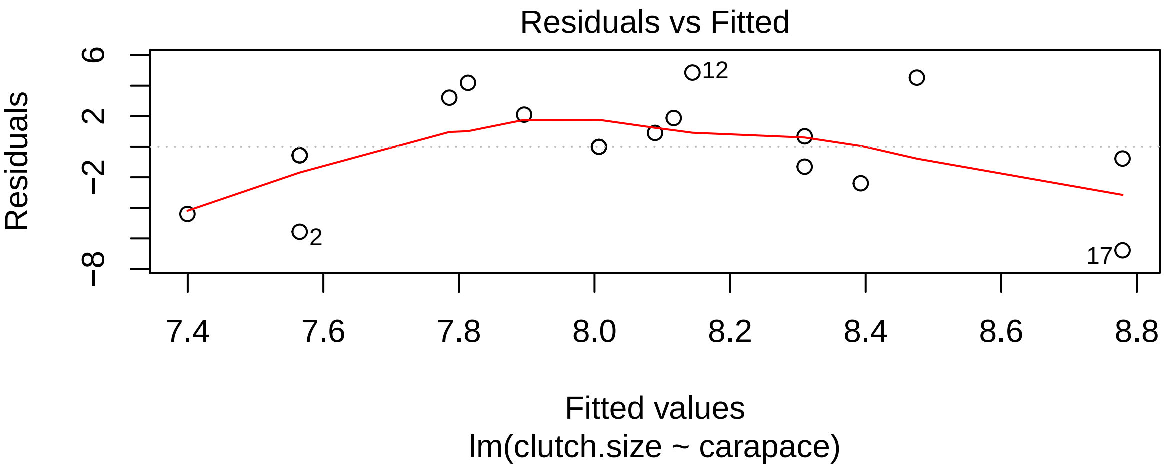
To explore how this plot can detect non-linear relationships between y and x, we will examine a data set from Ashton et al. (2007) that relates the length of a tortoise's carapace to the number of eggs laid in a clutch. The data are

```
Eggs <- data.frame(
  carapace    = c(284,290,290,290,298,299,302,306,306,
                  309,310,311,317,317,320,323,334,334),
  clutch.size = c(3,2,7,7,11,12,10,8,8,
                  9,10,13,7,9,6,13,2,8))
ggplot(Eggs, aes(x=carapace, y=clutch.size)) +
  geom_point()
```

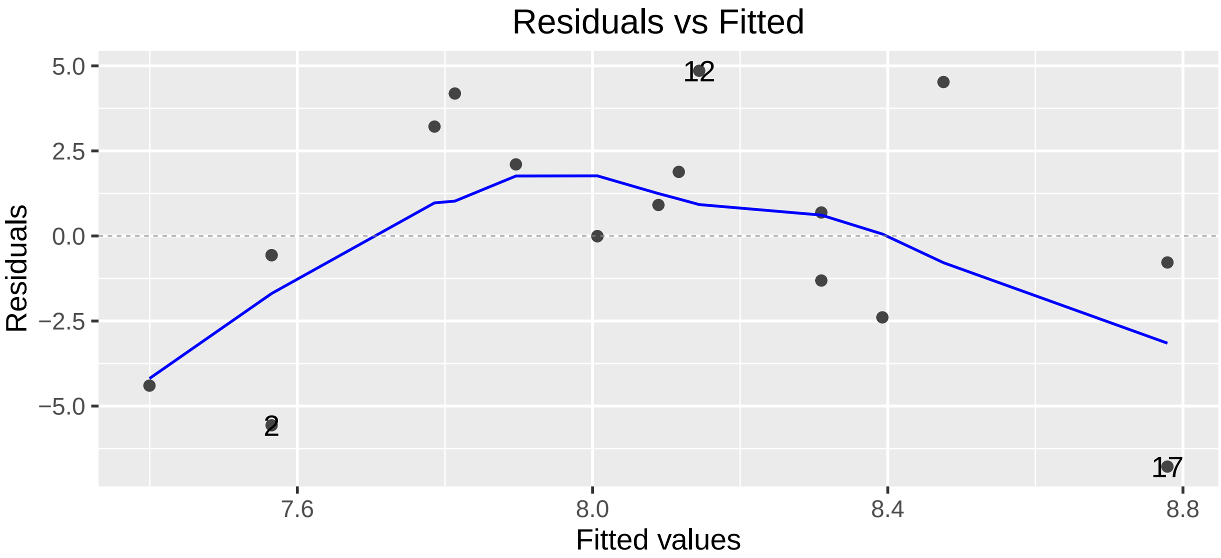


Looking at the data, it seems that we are violating the assumption that a linear model is appropriate, but we will fit the model anyway and look at the residual graph.

```
model <- lm( clutch.size ~ carapace, data=Eggs )
plot(model, which=1)      # which=1 tells R to only make the first plot
```



```
library(ggfortify)         # need the ggfortify library for autoplot.lm() to work
autoplot(model, which=1)   # same plot using ggplot2
```



The red/blue lines going through the plot is a smoother of the residuals. Ideally this should be a flat line and I should see no trend in this plot. Clearly there is a quadratic trend as larger tortoises have larger clutch sizes until some point where the extremely large tortoises start laying fewer (perhaps the extremely large tortoises are extremely old as well). To correct for this, we should fit a model that is quadratic in `carapace length`. We will create a new covariate, `carapace.2`, which is the square of the carapace length and add it to the model.

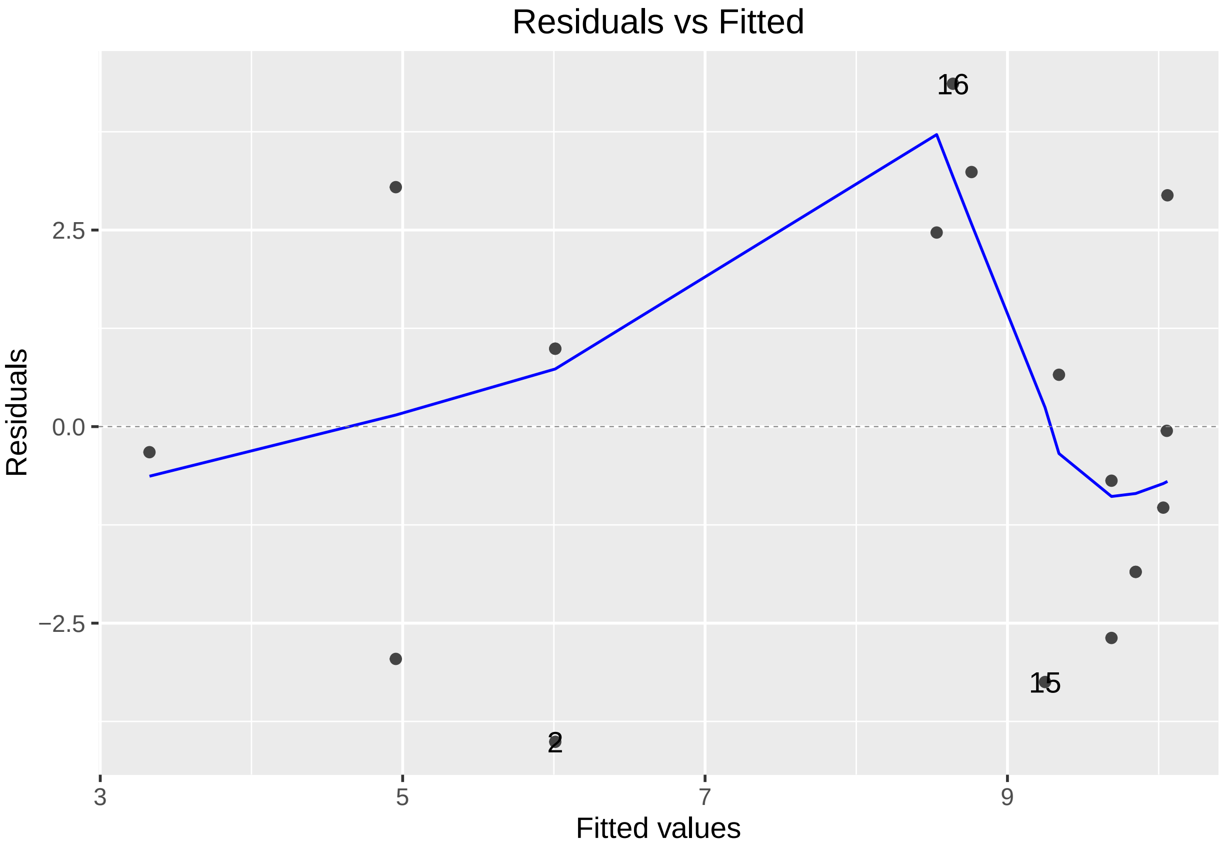
In general I could write the quadratic model as

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \epsilon_i$$

and note that my model is still a linear model with respect to covariates $\boldsymbol{x}$ and $\boldsymbol{x}^2$ because I can still write the model as

$$\begin{aligned}\boldsymbol{y} &= \boldsymbol{X}\boldsymbol{\beta}+\boldsymbol{\epsilon}\\ &= \begin{bmatrix} 1 & x_1 & x_1^2\\ 1 & x_2 & x_2^2\\ 1 & x_3 & x_3^2\\ \vdots & \vdots & \vdots\\ 1 & x_n & x_n^2 \end{bmatrix}\begin{bmatrix}\beta_0\\ \beta_1\\ \beta_2\end{bmatrix}+\begin{bmatrix}\epsilon_1\\ \epsilon_2\\ \epsilon_3\\ \vdots\\ \epsilon_n\end{bmatrix}\end{aligned}$$

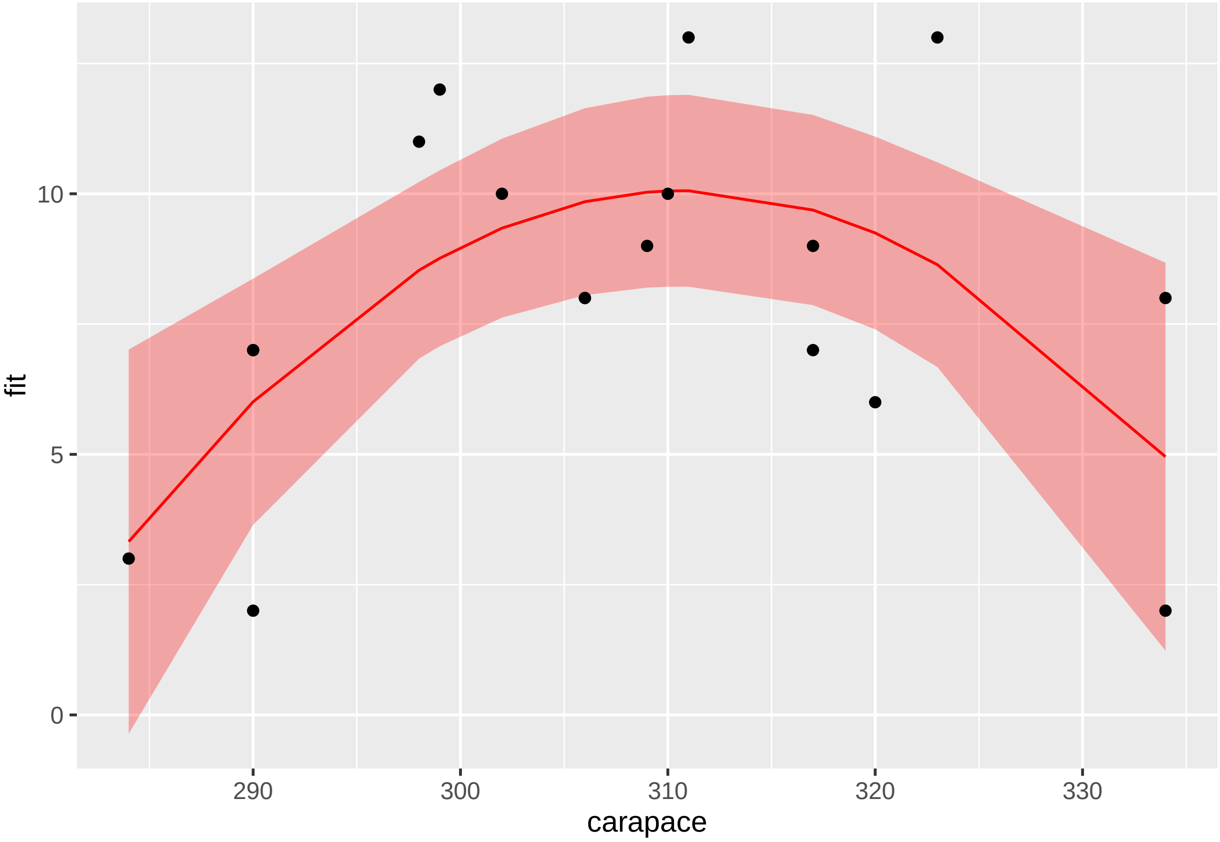
```r
# add a new column that is carapace^2
Eggs2 <- Eggs %>% mutate( carapace.2 = carapace^2 )
model <- lm( clutch.size ~ carapace + carapace.2,    data=Eggs2 )

# make R do it inside the formula... convenient
model <- lm( clutch.size ~ carapace + I(carapace^2), data=Eggs )

# Fit an arbitrary degree polynomial
model <- lm( clutch.size ~ poly(carapace, 2),        data=Eggs )

# If you use poly() in the formula, you must use 'data=' here,
# otherwise you can skip it and R will do the right thing.
autoplot(model, which=1, data=Eggs)
```



Now our residual plot versus fitted values does not show any trend, suggesting that the quadratic model is fitting the data well. Graphing the original data along with the predicted values confirms this.

```r
# add the fitted and CI lwr/upr columns to my dataset
Eggs <- Eggs %>% cbind( predict(model, interval='confidence') )
```

```
ggplot(Eggs, aes(x=carapace)) +
  geom_ribbon( aes(ymin=lwr, ymax=upr), fill='red', alpha=.3) +
  geom_line(aes(y=fit), color='red') +
  geom_point(aes(y=clutch.size))
```



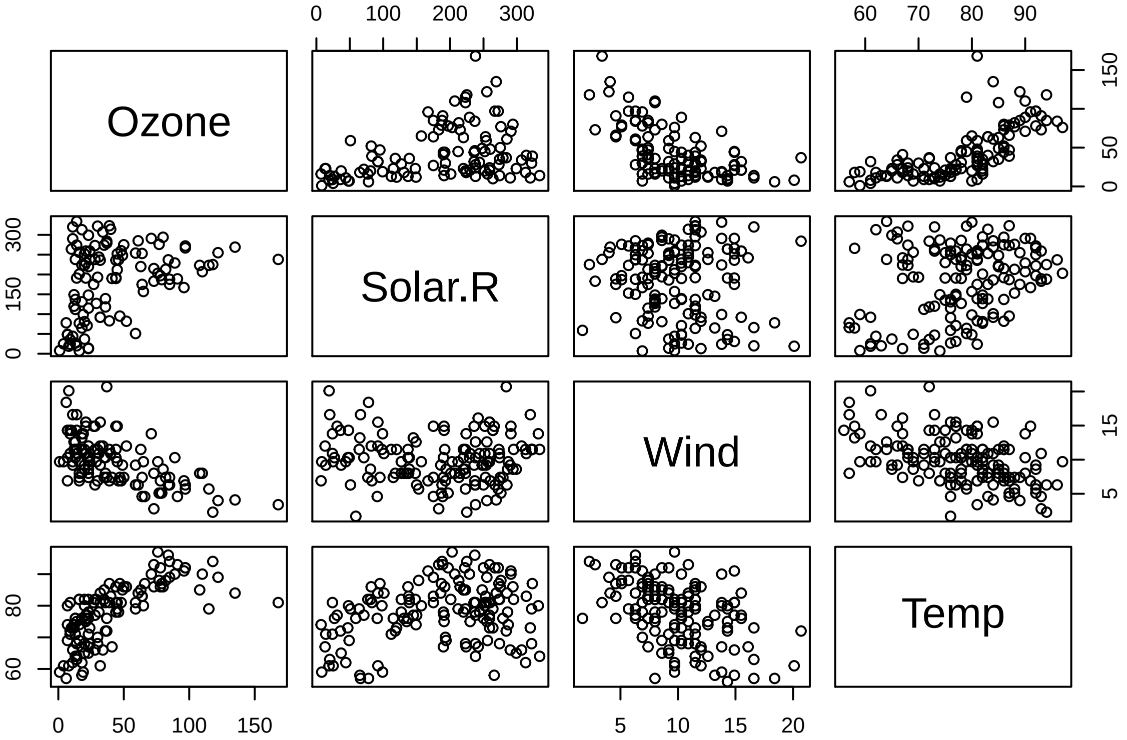
6.1.2.1.2 Heteroskedasticity

The plot of residuals versus fitted values can detect heteroskedasticity (non-constant variance) in the error terms.

To illustrate this, we turn to another dataset in the Faraway book. The dataset airquality uses data taken from an environmental study that measured four variables, ozone, solar radiation, temperature and windspeed for 153 consecutive days in New York. The goal is to predict the level of ozone using the weather variables.

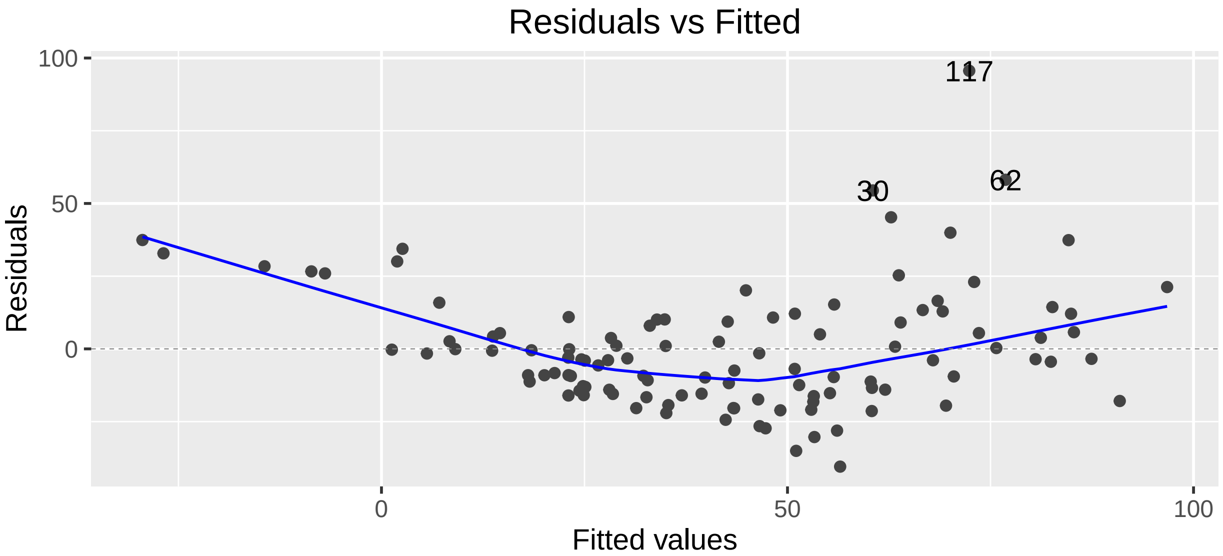
We first graph all pairs of variables in the dataset.

```
data(airquality)
pairs(~ Ozone + Solar.R + Wind + Temp, data=airquality)
```



and notice that ozone levels are positively correlated with solar radiation and temperature, and negatively correlated with wind speed. A linear relationship with wind might be suspect as is the increasing variability in the response to high temperature. However, we don't know if those trends will remain after fitting the model, because there is some covariance amongst the predictors.

```
model <- lm(Ozone ~ Solar.R + Wind + Temp, data=airquality)
autoplot(model, which=1)
```

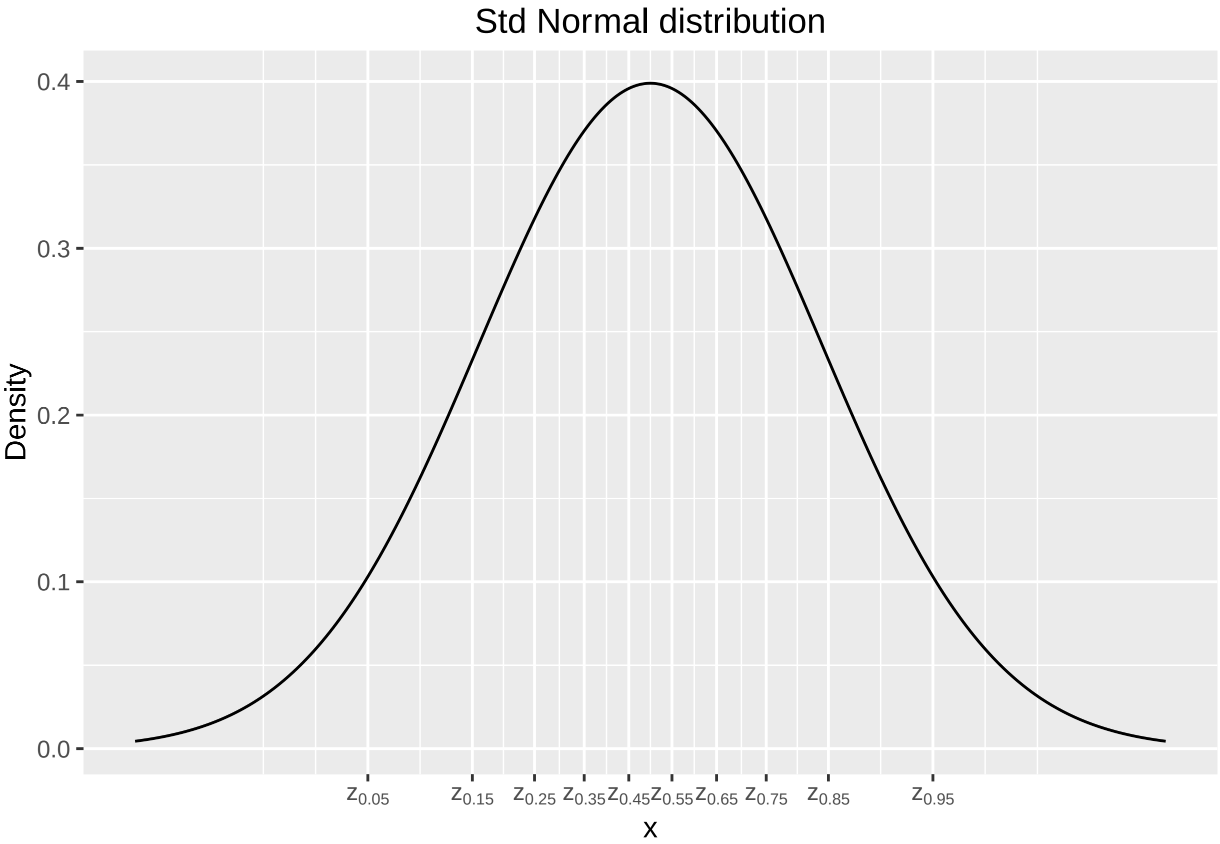


As we feared, we have both a non-constant variance and a non-linear relationship. A transformation of the y variable might be able to fix our problem.

6.1.2.2 QQplots

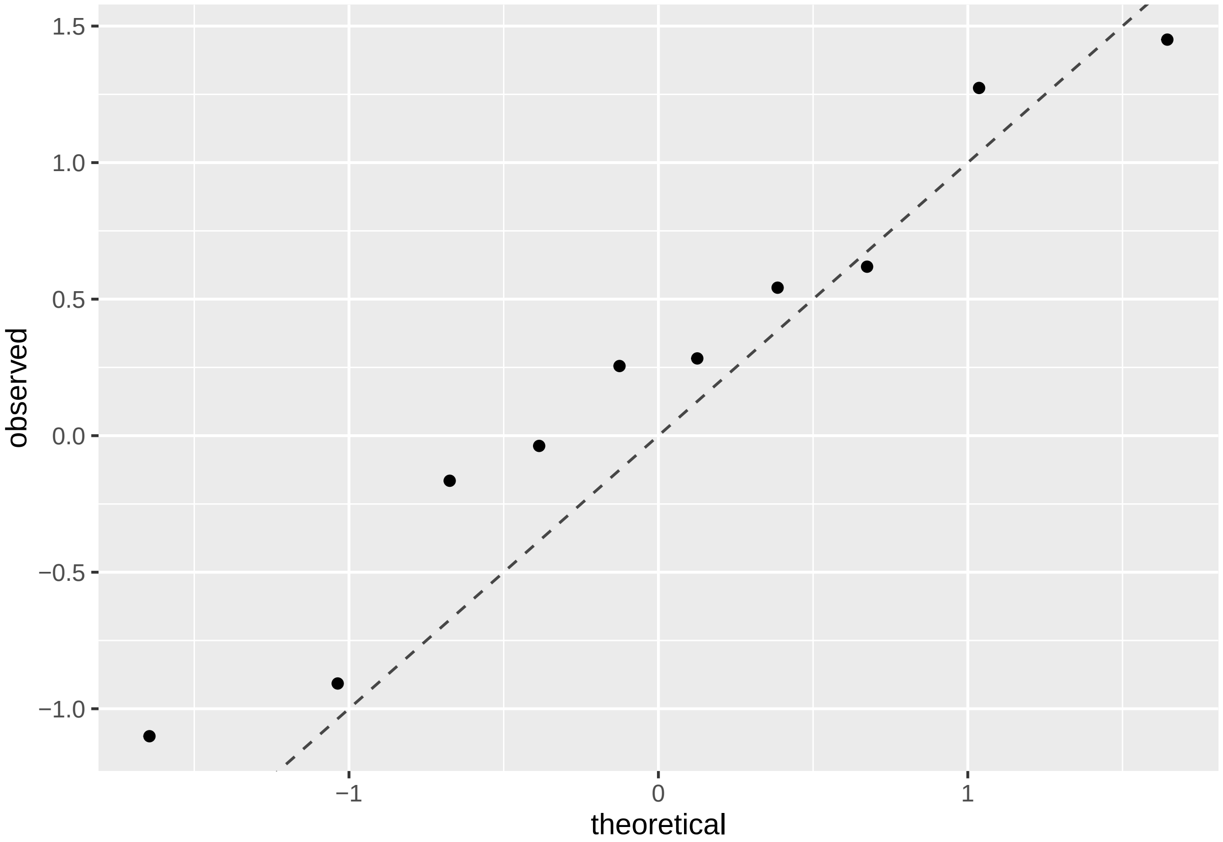
If we are taking a sample of size $n = 10$ from a standard normal distribution, then I should expect that the smallest observation will be negative. Intuitively, you would expect the smallest observation to be near the 10th percentile of the standard normal, and likewise the second smallest should be near the 20th percentile.

This idea needs a little modification because the largest observation cannot be near the 100th percentile (because that is ∞). So we'll adjust the estimates to still be spaced at $(1/n)$ quantile increments, but starting at the $0.5/n$ quantile instead of the $1/n$ quantile. So the smallest observation should be near the 0.05 quantile, the second smallest should be near the 0.15 quantile, and the largest observation should be near the 0.95 quantile. I will refer to these as the *theoretical quantiles*.



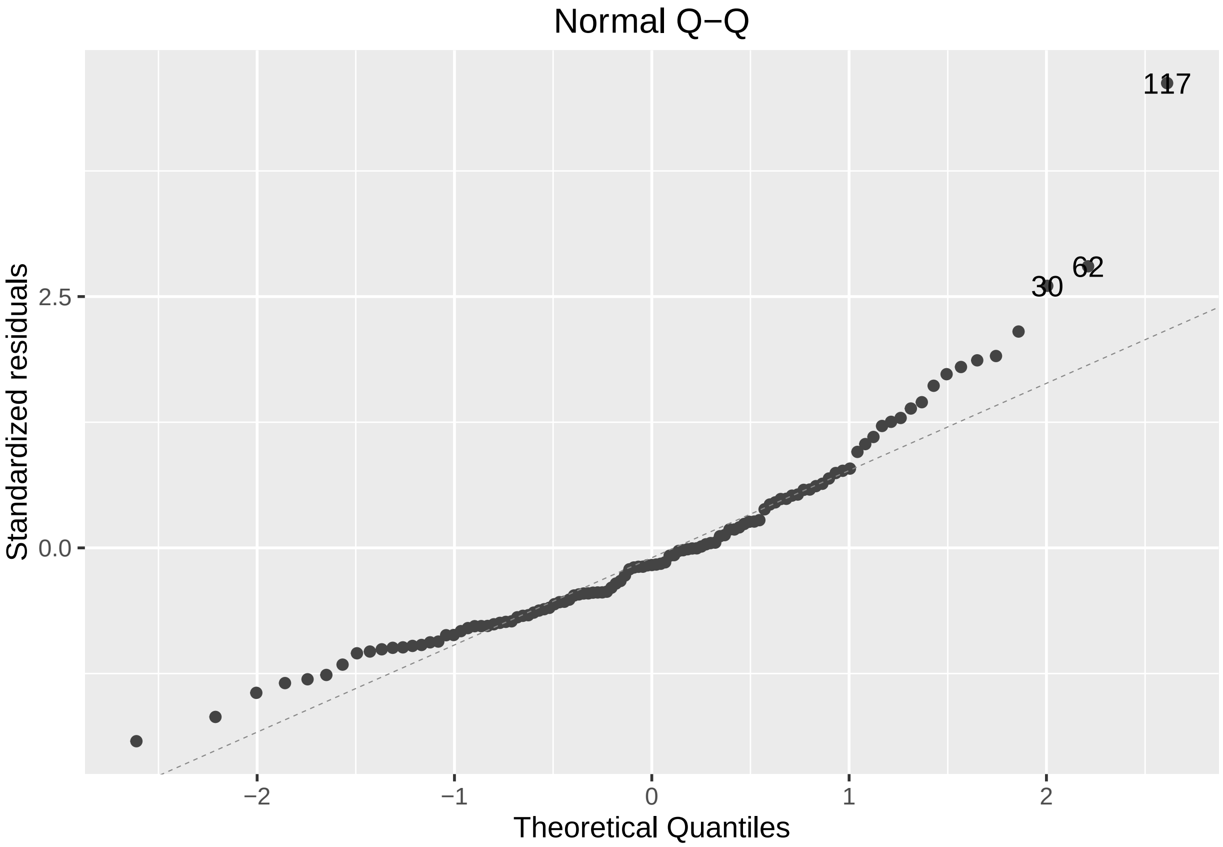
I can then graph the theoretical quantiles vs my observed values and if they lie on the 1-to-1 line, then my data comes from a standard normal distribution.

```r
set.seed(93516)  # make random sample in the next code chunk consistant run-to-run

n <- 10
data <- data.frame( observed = rnorm(n, mean=0, sd=1) ) %>%
  arrange(observed) %>%
  mutate( theoretical = qnorm( (1:n -.5)/n ) )

ggplot(data, aes(x=theoretical, y=observed) ) +
  geom_point() +
  geom_abline( intercept=0, slope=1,  linetype=2, alpha=.7) +
  labs(main='Q-Q Plot: Observed vs Normal Distribution')
```



In the context of a regression model, we wish to look at the residuals and see if there are obvious departures from normality. Returning to the airquality example, R will calculate the qqplot for us.

```
model <- lm(Ozone ~ Solar.R + Wind + Temp, data=airquality)
autoplot(model, which=2)
```



In this case, we have a large number of residuals that are bigger than I would expect them to be based on them being from a normal distribution. We could further test this using the Shapiro-Wilks test and compare the standardized residuals against a $N(0,1)$ distribution.

```
shapiro.test( rstandard(model) )
```

```
##
##  Shapiro-Wilk normality test
##
## data:  rstandard(model)
## W = 0.9151, p-value = 2.819e-06
```

The tail of the distribution of observed residuals is *far* from what we expect to see.

6.1.2.3 Scale-Location Plot

This plot is a variation on the fitted vs residuals plot, but the y-axis uses the square root of the absolute value of the standardized residuals. Supposedly this makes detecting increasing variance easier to detect, but I'm not convinced.

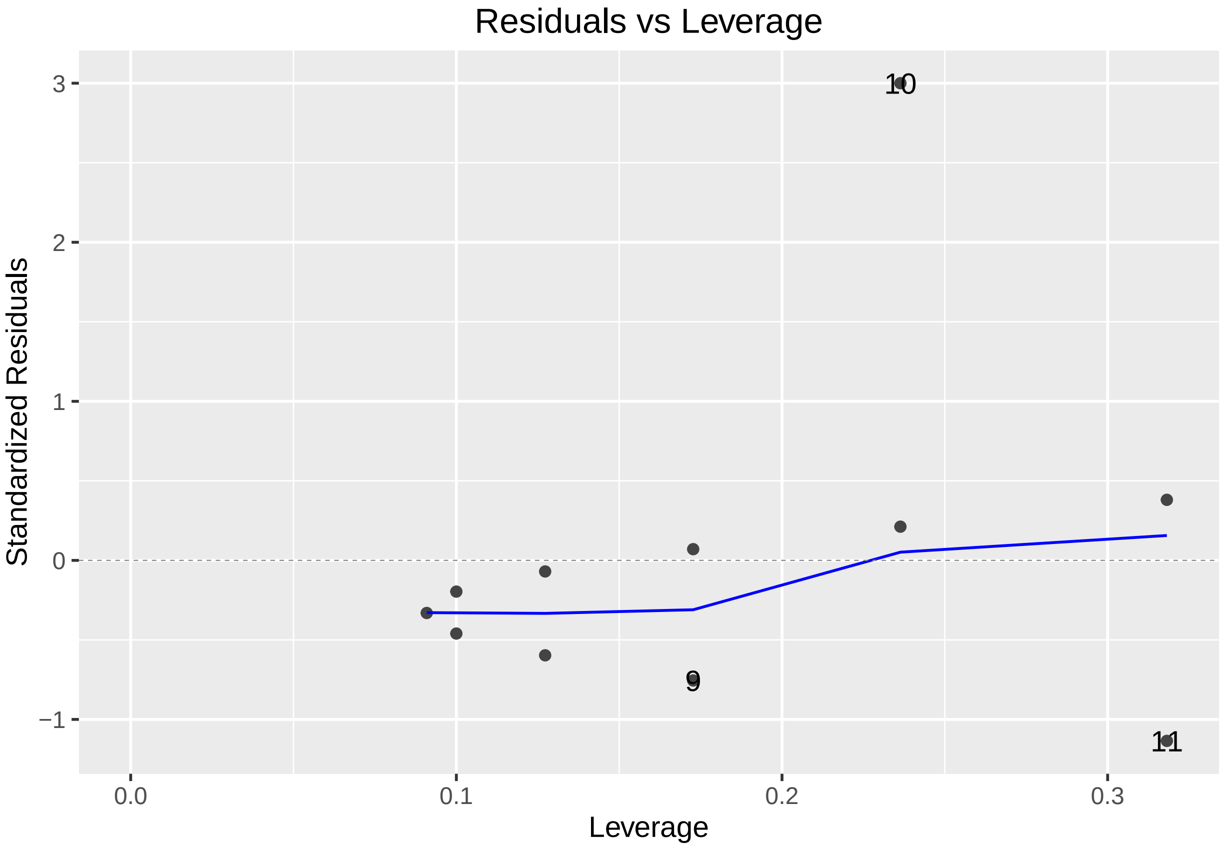
6.1.2.4 Residuals vs Leverage (plus Cook's Distance)

This plot lets the user examine the which observations have a high potential for being influential (i.e. high leverage) versus how large the residual is. Because Cook's distance is a function of those two traits, we can also divide the graph up into regions by the value of Cook's Distance.

Returning to Anscombe's third set of data, we see

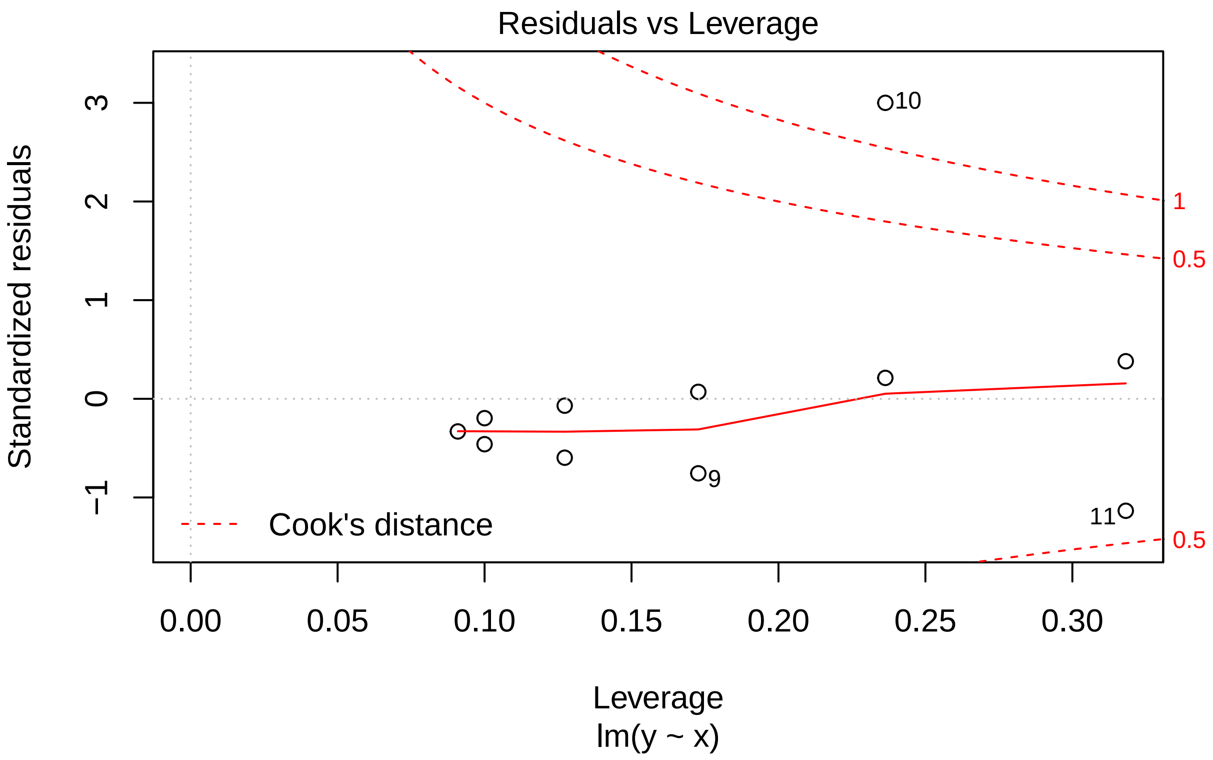
<<>>=

```r
model3 <- lm(y ~ x, data=Set3)
autoplot(model3, which=5)
```



that one data point (observation 10) has an extremely large standardized residual. This is one plot where I prefer what the base graphics in R does compared to the ggfortify version. The base version of R adds some contour lines that mark where the contours of where Cook's distance is 1/2 and 1.

```r
plot(model3, which=5)
```



6.2 Transformations

Transformations of the response variable and/or the predictor variables can drastically improve the model fit and can correct violations of the model assumptions. We might also create new predictor variables that are functions of existing variables. These include quadratic and higher order polynomial terms and interaction terms.

Often we are presented with data and we would like to fit a linear model to the data. Unfortunately the data might not satisfy all of the assumptions of a linear model. For the simple linear model

$$y = \beta_0 + \beta_1 x + \epsilon$$

where $\epsilon \sim N\left(0, \sigma^2\right)$, the necessary assumptions are:

1. Independent errors
2. Errors have constant variance, no matter what the x-value (or equivalently the fitted value)
3. Errors are normally distributed
4. The model contains all the appropriate covariates and no more.

In general, a transformation of the response variable can be used to address the 2nd and 3rd assumptions, and adding new covariates to the model will be how to address deficiencies of assumption 4.

6.2.1 Transforming the Response

When the normality or constant variance assumption is violated, sometimes it is possible to transform the response to satisfy the assumption. Often times count data is analyzed as `log(count)` and weights are analyzed after taking a square root or cube root transform. Statistics involving income or other monetary values are usually analyzed on the log scale so as to reduce the leverage of high income observations.

While we may want to transform the response in order to satisfy the statistical assumptions, it is often necessary to back-transform to the original scale. For example if we fit a linear model for income (y) based on the amount of schooling the individual has received (x)

$$\log y = \beta_0 + \beta_1 x + \epsilon$$

then we might want to give a prediction interval for an x_0 value. The predicted $log(income)$ value is

$$\log\left(\hat{y}_0\right) = \hat{\beta}_0 + \hat{\beta}_x x_0$$

and we could calculate the appropriate predicted income as $\hat{y}_0 = e^{log(\hat{y}_0)}$. Likewise if we had a confidence interval or prediction interval for $\log\left(\hat{y}_0\right)$ of the form (l, u) then the appropriate interval for $\hat{y}_0$ is $\left(e^l, e^u\right)$. Notice that while (l, u) might be symmetric about $\log\left(\hat{y}_0\right)$, the back-transformed interval is not symmetric about $\hat{y}_0$.

Unfortunately the interpretation of the regression coefficients $\hat{\beta}_0$ and $\hat{\beta}_1$ on the untransformed scale becomes more complicated. This is a very serious difficulty and might sway a researcher from transforming their data.

6.2.1.1 Box-Cox Family of Transformations

The Box-Cox method is a popular way of determining what transformation to make. It is intended for responses that are strictly positive (because $\log 0 = -\infty$ and the square root of a number gives complex numbers, which we don't know how to address in regression). The transformation is defined as

$$g\left(y\right) = \begin{cases} \frac{y^{\lambda}-1}{\lambda} & \lambda \neq 0 \\ \log y & \lambda = 0 \end{cases}$$

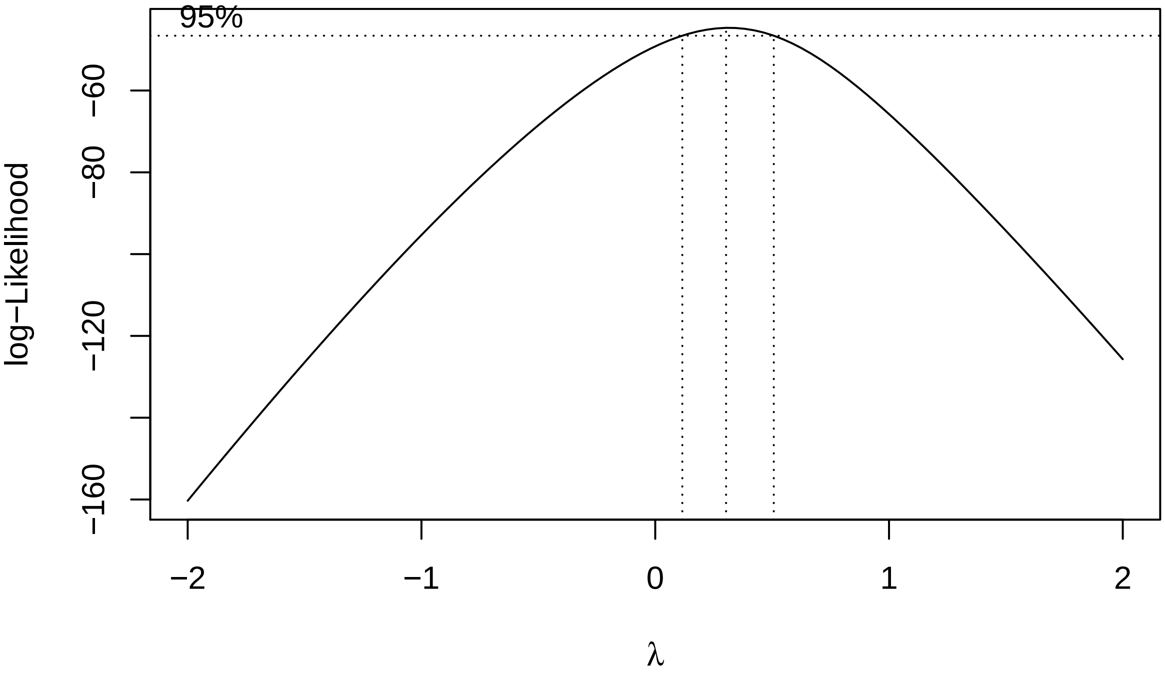
This transformation is a smooth family of transformations because

$$\lim_{\lambda \to 0} \frac{y^{\lambda} - 1}{\lambda} = \log y$$

In the case that $\lambda \neq 0$, then a researcher will usually use the simpler transformation y^{λ} because the subtraction and division does not change anything in a non-linear fashion. Thus for purposes of addressing the assumption violations, all we care about is the y^{λ} and prefer the simpler (i.e. more interpretable) transformation.

Finding the best transformation can be done by adding the λ parameter to the model and finding the value that maximizes the log-likelihood function. Fortunately, we don't have to do this by hand, as the function `boxcox()` in the `MASS` library will do all the heavy calculation for us.

```r
library(faraway)
library(MASS)
data(gala)
g <- lm(Species ~ Area + Elevation + Nearest + Scruz + Adjacent, data=gala)
boxcox(g)
```



The optimal transformation for these data would be $y^{1/4} = \sqrt[4]{y}$ but that is an extremely uncommon transformation. Instead we should pick the nearest "standard" transformation which would suggest that we should use either the $\log y$ or $\sqrt{y}$ transformation.

Thoughts on the Box-Cox transformation:

1. In general, I prefer to using a larger-than-optimal model when picking a transformation and then go about the model building process. After a suitable model has been chosen, I'll double check the my transformation was appropriate given the model that I ended up with.
2. Outliers can have a profound effect on this method. If the "optimal" transformation is extreme ($\lambda = 5$ or something silly) then you might have to remove the outliers and refit the transformation.
3. If the range of the response y is small, then the method is not as sensitive.
4. These are not the only possible transformations. For example, for binary data, the `logit` and `probit` transformations are common.

6.2.2 Transforming the predictors

6.2.2.1 Polynomials of a predictor

Perhaps the most common transformation to make is to make a quadratic function in x. Often the relationship between x and y follows a curve and we want to fit a quadratic model

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x + \hat{\beta}_2 x^2$$

and we should note that this is still a linear model because $\hat{y}$ is a linear function of x and x^2. As we have already seen, it is easy to fit the model. Adding the column of x^2 values to the design matrix does the trick.

The difficult part comes in the interpretation of the parameter values. No longer is $\hat{\beta}_1$ the increase in y for every one unit increase in x. Instead the three parameters in my model interact in a complicated fashion. For example, the peak of the parabola is at $-\hat{\beta}_1/2\hat{\beta}_2$ and whether the parabola is cup shaped vs dome shaped and its steepness is controlled by $\hat{\beta}_2$. Because my geometric understanding of degree q polynomials relies on have all factors of degree q or lower, whenever I include a covariate raised to a power, I should include all the lower powers as well.

6.2.2.2 Log and Square Root of a predictor

Often the effect of a covariate is not linearly related to response, but rather some function of the covariate. For example the area of a circle is not linearly related to its radius, but it is linearly related to the radius squared.

$$Area = \pi r^2$$

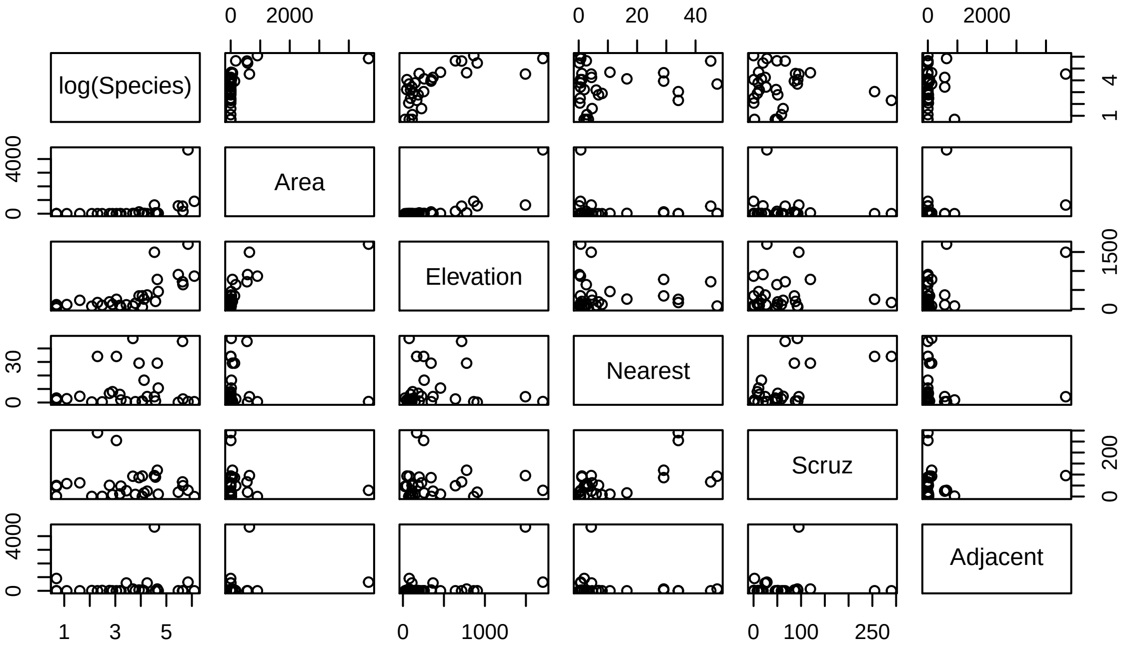
Similar situations might arise in biological settings, such as the volume of conducting tissue being related to the square of the diameter. Or perhaps an animals metabolic requirements are related to some power of body length. In sociology, it is often seen that the utility of, say, $1000 drops off in a logarithmic fashion according to the person's income. To a graduate student, $1K is a big deal, but to a corporate CEO, $1K is just another weekend at the track. Making a log transformation on any monetary covariate, might account for the non-linear nature of "utility".

Picking a good transformation for a covariate is quite difficult, but most fields of study have spent plenty of time thinking about these issues. When in doubt, look at scatter plots of the covariate vs the response and ask what transformation would make the data fall onto a line?

6.2.2.3 Examples of transformation of predictors

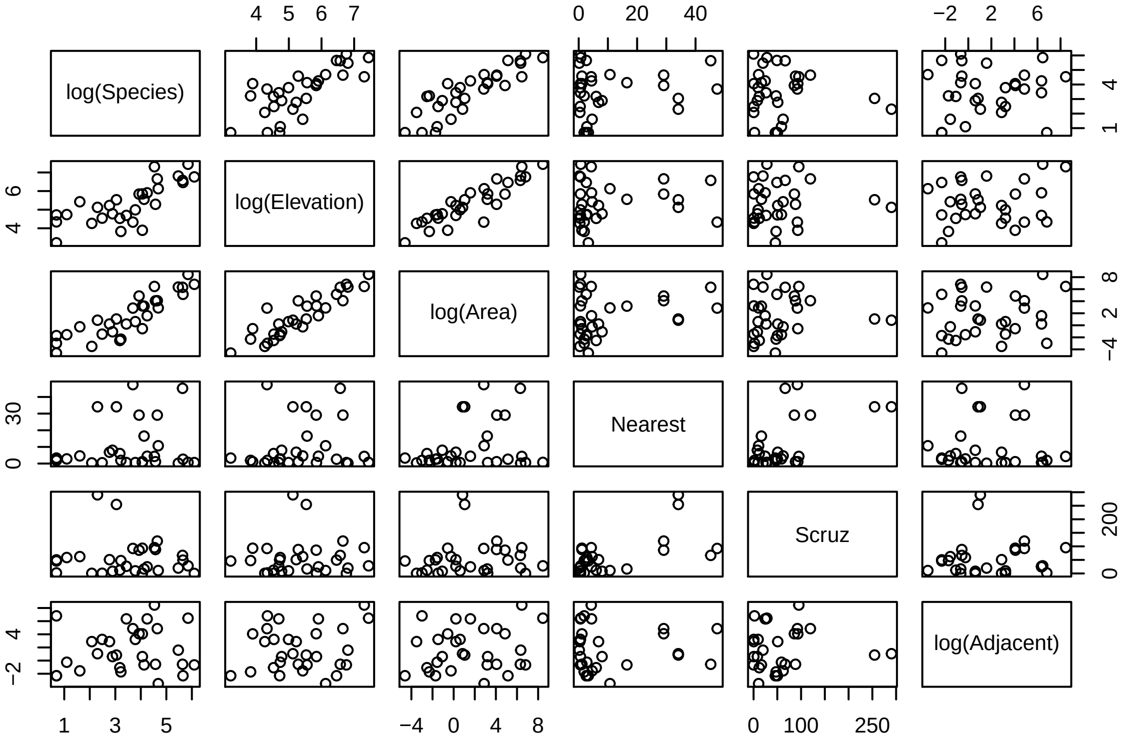
To illustrate how to add a transformation of a predictor to a linear model in R, we will consider the Galapagos data in `faraway`.

```r
library(faraway)
data(gala)
# look at all the scatterplots
pairs(log(Species) ~ Area + Elevation + Nearest + Scruz + Adjacent, data=gala)
```



Looking at these graphs, I think I should transform `elevation`, `Adjacent`, and `Area` and perhaps a log transformation is a good idea.

```r
pairs( log(Species) ~ log(Elevation) + log(Area) +
                         Nearest + Scruz + log(Adjacent), data=gala)
```



Looking at these graphs, it is clear that `log(Elevation)` and `log(Area)` are highly correlated and we should probably have one or the other, but not both in the model.

```
m.c <- lm(log(Species) ~  log(Area) + Nearest + Scruz + log(Adjacent), data=gala)
summary(m.c)$coefficients %>% round(digits=3) # more readable...
```

```
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)        3.122      0.208  14.979    0.000
## log(Area)          0.398      0.044   9.027    0.000
## Nearest            0.000      0.013  -0.022    0.983
## Scruz             -0.003      0.003  -1.238    0.227
## log(Adjacent)     -0.021      0.045  -0.468    0.644
```

We will remove all the parameters that appear to be superfluous, and perorm an F-test to confirm that the simple model is sufficient.

```
m.s <- lm(log(Species) ~ log(Area), data=gala)
anova(m.s, m.c)
```

```
## Analysis of Variance Table
##
## Model 1: log(Species) ~ log(Area)
## Model 2: log(Species) ~ log(Area) + Nearest + Scruz + log(Adjacent)
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     28 17.218
## 2     25 15.501  3    1.7176 0.9234 0.4439
```

Next we will look at the coefficients.

```
summary(m.s)
```

```
##
## Call:
## lm(formula = log(Species) ~ log(Area), data = gala)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.5442 -0.4001  0.0941  0.5449  1.3752
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.9037     0.1571  18.484  < 2e-16 ***
## log(Area)     0.3886     0.0416   9.342 4.23e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7842 on 28 degrees of freedom
## Multiple R-squared:  0.7571, Adjusted R-squared:  0.7484
## F-statistic: 87.27 on 1 and 28 DF,  p-value: 4.23e-10
```

The slope coefficient (0.3886) is the increase in log(Species) for every 1 unit increase in log(Area). Unfortunately that is not particularly convenient to interpretation and we will address this in the next section of this chapter.

Finally, we might be interested in creating a confidence interval for the expected number of tortoise species for an island with `Area=50`.

```
x0 <- data.frame(Area=50)
log.Species.CI <- predict(m.s, newdata=x0, interval='confidence')
log.Species.CI       # Log(Species) scale
```

```
##        fit      lwr      upr
## 1 4.423903 4.068412 4.779394
```

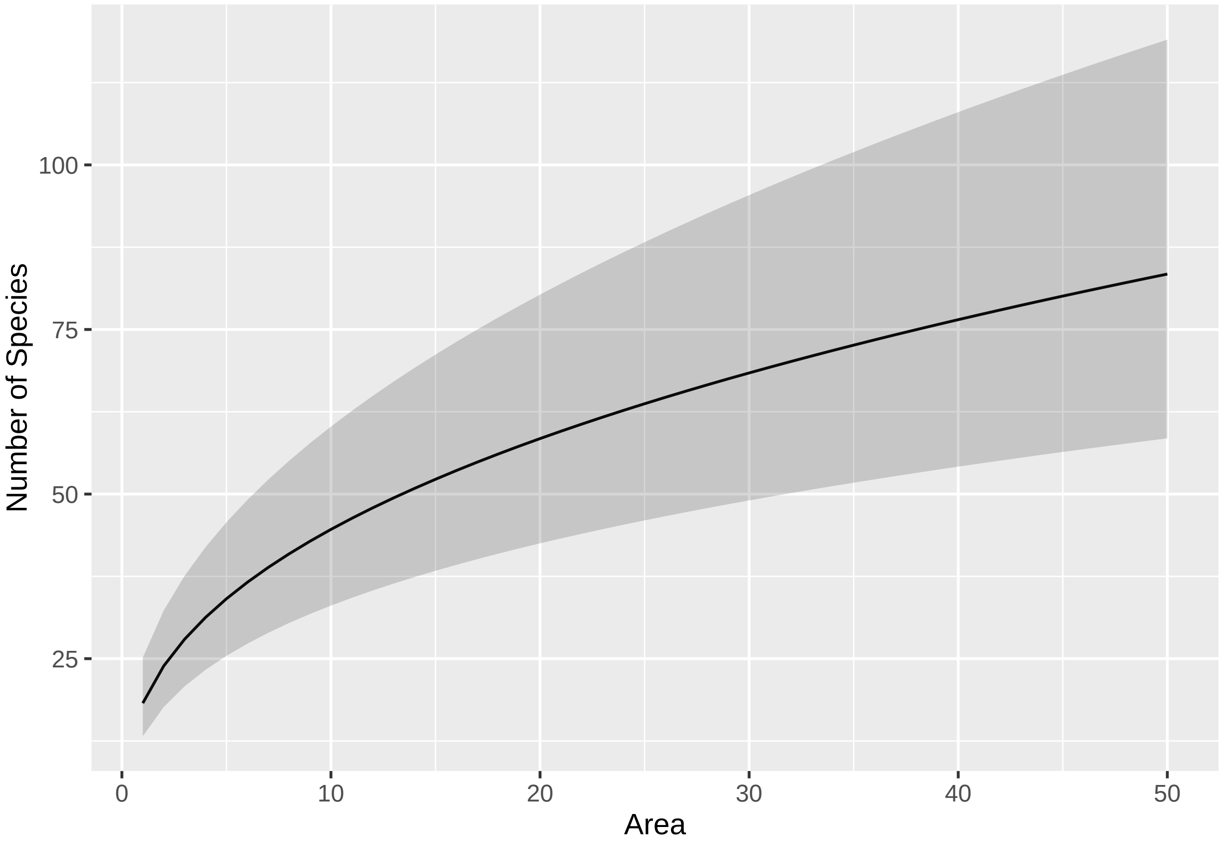
```
exp(log.Species.CI)  # Species scale
```

```
##        fit      lwr      upr
## 1 83.42122 58.46403 119.0322
```

Notice that on the species-scale, we see that the fitted value is not in the center of the confidence interval.

To help us understand what the log transformations are doing, we can produce a plot with the island Area on the x-axis and the expected number of Species on the y-axis and hopefully that will help us understand the relationship between the two.

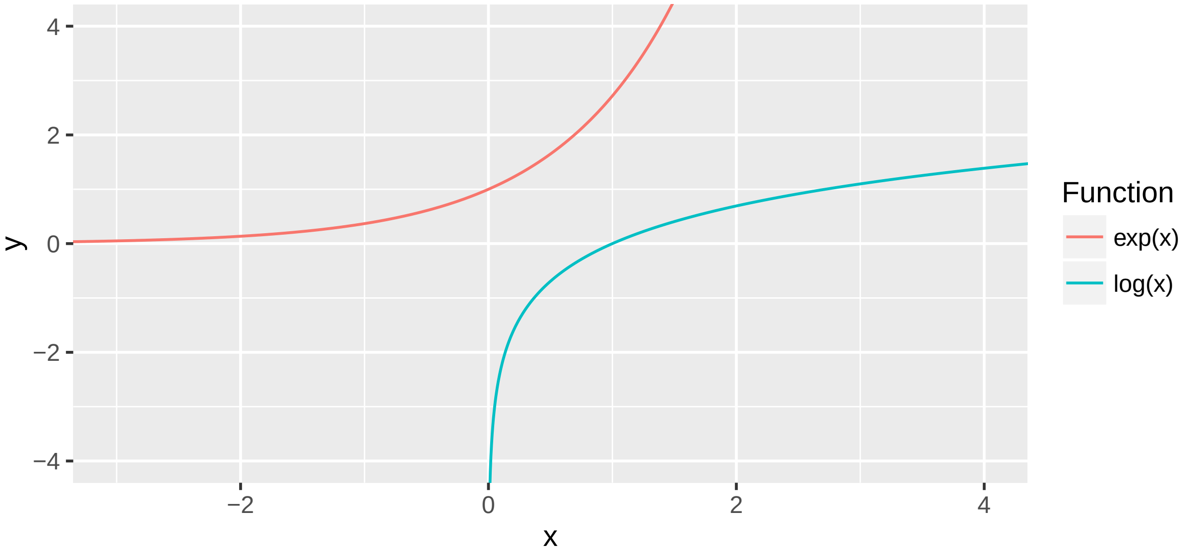
```
library(ggplot2)
pred.data <- data.frame(Area=1:50)
pred.data <- pred.data %>% cbind( predict(m.s, newdata=pred.data, interval='conf'))
ggplot(pred.data, aes(x=Area)) +
  geom_line(aes(y=exp(fit))) +
  geom_ribbon(aes(ymin=exp(lwr), ymax=exp(upr)), alpha=.2) +
  ylab('Number of Species')
```



6.2.3 Interpretation of log transformed variables

One of the most difficult issues surrounding transformed variables is that the interpretation is difficult. Here we look at the interpretation of log transformed variables.

To begin with, we need to remind ourselves of what the functions $\log x$ and e^x look like.



In particular we notice that

$$e^0 = 1$$

and

$$\log(1) = 0$$

and the functions e^x and $\log x$ are inverse functions of each other.

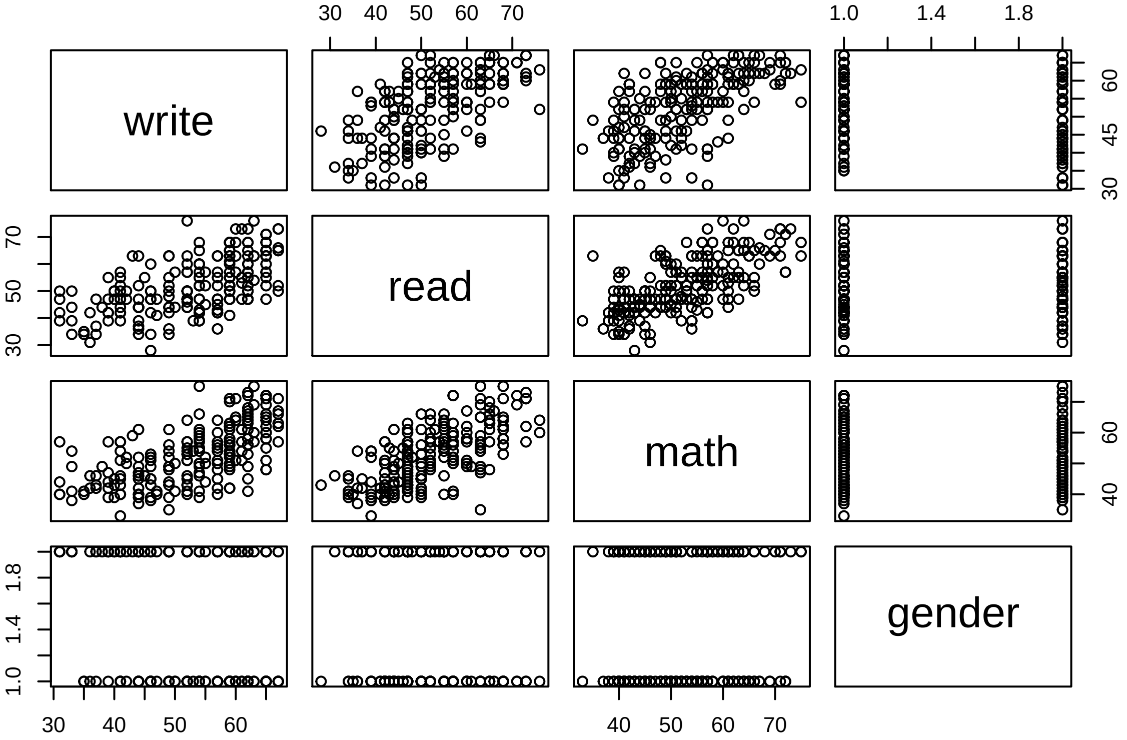
$$e^{\log x} = \log(e^x) = x$$

Also it is important to note that the log function has some interesting properties in that it makes operations "1-operation easier".

$$\log\left(a^b\right) = b\log a \log\left(\frac{a}{b}\right) \quad = \log a - \log b \log(ab) = \log a + \log b$$

To investigate the effects of a log transformation, we'll examine a dataset that predicts the writing scores of $n = 200$ students using the gender, reading and math scores. This example was taken from the UCLA Statistical Consulting Group.

```r
scores <- read.table(
  file='http://www.ats.ucla.edu/stat/mult_pkg/faq/general/lgtrans.csv',
  header=TRUE, sep=',')
scores <- scores %>% mutate(gender = female)
pairs(write~read+math+gender, data=scores)
```



6.2.3.1 Log-transformed response, untransformed covariates

We consider the model where we have transformed the response variable and just an intercept term.

$$\log y = \beta_0 + \epsilon$$

```r
m <- lm(log(write) ~ 1, data=scores)
summary(m)$coef %>% round(digits=3)
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.948      0.014 288.402        0
```

We interpret the intercept as the mean of the log-transformed response values. We could back transform this to the original scale $e^{\hat{\beta}_0} = e^{3.948} = 51.83$ as a typical value of write. To distinguish this from the usually defined mean of the write values, we will call this as the *geometric mean*.

Next we examine how to interpret the model when a categorical variable is added to the model.

$$\log y = \begin{cases} \beta_0 + \epsilon & \text{if female} \\ \beta_0 + \beta_1 + \epsilon & \text{if male} \end{cases}$$

```r
m <- lm(log(write) ~ gender, data=scores)
summary(m)$coef %>% round(digits=3)
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.995      0.018 222.949        0
## gendermale    -0.103      0.027  -3.887        0
```

The intercept is now the mean of the log-transformed `write` responses for the females and the offset for males is the change in `log(write)` from the female group. Notice that for the males, we have

$$\log \hat{y} = \hat{\beta}_0 + \hat{\beta}_1 \hat{y} = e^{\hat{\beta}_0 + \hat{\beta}_1} = \underbrace{e^{\hat{\beta}_0}}_{\text{females}} \qquad * \underbrace{e^{\hat{\beta}_1}}_{\text{percent change fo males}}$$

and therefore we see that males tend to have writing scores $e^{-0.103} = 0.90 = 90\%$ of the females. Typically this sort of result would be reported as the males have a 10% lower writing score than the females.

The model with a continuous covariate has a similar interpretation.

$$\log y = \begin{cases} \beta_0 + \beta_2 x + \epsilon & \text{if female} \\ \beta_0 + \beta_1 + \beta_2 x + \epsilon & \text{if male} \end{cases}$$

We will use the reading score read to predict the writing score. Then $\hat{\beta}_2$ is the predicted increase in `log(write)` for every 1-unit increase in read score. The interpretation of $\hat{\beta}_0$ is now $\log \hat{y}$ when $x = 0$ and therefore $\hat{y} = e^{\hat{\beta}_0}$ when $x = 0$.

```r
m <- lm(log(write) ~ gender + read, data=scores)  # main effects model
summary(m)$coefficients %>% round(digits=3)
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.412      0.055  62.452        0
## gendermale    -0.116      0.021  -5.516        0
## read           0.011      0.001  11.057        0
```

For females, we consider the difference in $\log \hat{y}$ for a 1-unit increase in x and will interpret this on the original write scale.

$$\log \hat{y} = \hat{\beta}_0 + \hat{\beta}_2 x \hat{y} \quad = e^{\hat{\beta}_0 + \hat{\beta}_2 x}$$

therefore we consider $e^{\hat{\beta}_2}$ as the *percent* increase in write score for a 1-unit increase in x because of the following justification:

$$\frac{e^{\hat{\beta}_0 + \hat{\beta}_2 (x+1)}}{e^{\hat{\beta}_0 + \hat{\beta}_2 x}} = \frac{e^{\hat{\beta}_0} e^{\hat{\beta}_2 x} e^{\hat{\beta}_2}}{e^{\hat{\beta}_0} e^{\hat{\beta}_2 x}} = e^{\hat{\beta}_2}$$

For our writing scores example we have that $e^{\hat{\beta}_2} = e^{0.011} = 1.01$ meaning there is an estimated 1% increase in `write` score for every 1-point increase in `read` score.

If we are interested in, say, a 5-unit increase in x, then that would result in an increase of

$$\frac{e^{\hat{\beta}_0+\hat{\beta}_2(x+5)}}{e^{\hat{\beta}_0+\hat{\beta}_2 x}} = \frac{e^{\hat{\beta}_0}e^{\hat{\beta}_2 x}e^{5\hat{\beta}_2}}{e^{\hat{\beta}_0}e^{\hat{\beta}_2 x}} = e^{5\hat{\beta}_2}$$

and for the writing scores we have $e^{5\hat{\beta}_2} = \left(e^{\hat{\beta}_2}\right)^5 = 1.01^5$.

In short, we can interpret $e^{\hat{\beta}_i}$ as the percent increase/decrease in the non-transformed response variable. Some students get confused by what is meant by a % increase or decrease in x.

- A 75% decrease in x has a resulting value of $(1-0.75)\,x = (0.25)\,x$
- A 75% increase in x has a resulting value of $(1+0.75)\,x = (1.75)\,x$
- A 100% increase in x has a resulting value of $(1+1.00)\,x = 2x$ and is a doubling of x.
- A 50% decrease in x has a resulting value of $(1-0.5)\,x = (0.5)\,x$ and is a halving of x.

6.2.3.2 Untransformed response, log-transformed covariate

We consider the model

$$y = \beta_0 + \beta_2 \log x + \epsilon$$

and consider two different values of x (which we'll call x_1 and x_2 and we are considering the effect of moving from x_1 to x_2) and look at the differences between the predicted values $\hat{y}$.

$$\begin{aligned}
\hat{y}_2 - \hat{y}_1 &= \left[\hat{\beta}_0 + \hat{\beta}_2 \log x_2\right] - \left[\hat{\beta}_0 + \hat{\beta}_2 \log x_1\right] \\
&= \hat{\beta}_2 \left[\log x_2 - \log x_1\right] \\
&= \hat{\beta}_2 \log \left[\frac{x_2}{x_1}\right]
\end{aligned}$$

This means that so long as the ratio between the two x-values is constant, then the change in $\hat{y}$ is the same. So doubling the value of x from 1 to 2 changes $\hat{y}$ has the same amount as changing x from 50 to 100.

6.2.3.3 Log-transformed response, log-transformed covariate

This combines the interpretation of the in the previous two section. We consider

$$\log y = \beta_0 + \beta_2 \log x + \epsilon$$

and we again consider two x values (again x_1 and x_2). We then examine the difference in the $\log \hat{y}$ values as

$$\begin{aligned}
\log \hat{y}_2 - \log \hat{y}_1 &= \left[\hat{\beta}_0 + \hat{\beta}_2 \log x_2\right] - \left[\hat{\beta}_0 + \hat{\beta}_2 \log x_1\right] \\
\log \left[\frac{\hat{y}_2}{\hat{y}_1}\right] &= \hat{\beta}_2 \log \left[\frac{x_2}{x_1}\right] \\
\log \left[\frac{\hat{y}_2}{\hat{y}_1}\right] &= \log \left[\left(\frac{x_2}{x_1}\right)^{\hat{\beta}_2}\right] \\
\frac{\hat{y}_2}{\hat{y}_1} &= \left(\frac{x_2}{x_1}\right)^{\hat{\beta}_2}
\end{aligned}$$

This allows us to examine the effect of some arbitrary percentage increase in x value as a percentage increase in y value.

```
m <- lm(log(write) ~ gender + log(read), data=scores)
summary(m)$coefficients %>% round(digits=3)
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.714      0.205   8.358        0
## gendermale    -0.114      0.021  -5.483        0
## log(read)      0.581      0.052  11.148        0
```

which implies for a 10% increase in `read` score, we should see a $1.10^{0.581} = 1.05$ multiplier in `write` score. That is to say, a 10% increase in reading score results in a 5% increase in writing score.

For the Galapagos islands, we had

```
m.s <- lm(log(Species) ~ log(Area), data=gala)
summary(m.s)$coefficients %>% round(digits=3)
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    2.904      0.157  18.484        0
## log(Area)      0.389      0.042   9.342        0
```

and therefore doubling of Area (i.e. the ratio of the $Area_2/Area_1 = 2$) results in a $2^{0.389} = 1.31$ multiplier of the `Species` value. That is to say doubling the island area increases the number of species by 31%.

6.3 Exercises

1. The dataset `infmort` in package faraway has information about infant mortality from countries around the world. Be aware that this is a old data set and does not necessarily reflect current conditions. More information about the dataset can be found using `help(infmort)`. We will be interested in understanding how infant mortality is predicted by per capita income, world region, and oil export status.

    a. Plot the relationship between income and mortality. This can be done using the command

    ```
    library(faraway)
    pairs(mortality ~., data=infmort)
    ```

    What do you notice about the relationship between mortality and income?

    b. Fit a linear model without any interaction terms with all three covariates as predictors of infant mortality. Examine the diagnostic plots. What stands out?

    c. Use the `boxcox()` function in the library MASS to determine a what a good transformation to the mortality response variable.

    d. Make a log transformation to the mortality variable and refit the model without interactions. Use the log transformed mortality for all further questions.

    e. Examine the pairs plot with log(mortality), income, and log(income). Which should be used in our model, `income` or `log(income)`?

    f. Examine models that have a `region:log(income)` interaction and `oil:log(income)` interaction along with the main effect of the third variable. Are either interaction significant vs the model without interactions? What about a model that contains both interactions?

    g. Interpret the effects of income, world region, and oil exports on log infant mortality based on these data. *Hint: graph the data and the predicted values.*

2. Using the `pressure` data available in the `faraway` package, fit a model with pressure as the response and temperature as the predictor using transformations to obtain a good fit. Feel free to experiment with what might be considered a ridiculously complicated model.

   a. Document your process of building your final model. Do not show graphs or computer output that is not relevant to your decision or that you do not wish to comment on.

   b. Comment on the interpretability of your (possibly ridiculously complicated) model.

3. Use transformations to find a good model for `volume` in terms of `girth` and `height` using the `trees` dataset in the `faraway` package.

   a. Document your process of building your final model. Again, only include output or graphs that are relevant to your decisions and include discussion about anything you include.

   b. Create a prediction interval for the volume of a tree with girth=16 and height=70. Notice that if you have transformed your response variable in you model, you'll have to back-transform to the original y-scale.

4. For this problem, we will look at a manufacturing problem. We will investigate the relationship predicting the time taken polishing a newly manufactured dish versus the dish diameter (in inches), type, and price.

   a. The data live in a package I have on GitHub. The following code will download the data package:

   ```
   library(devtools)   # You might need to load this package the usual way...
   install_github('dereksonderegger/dsData') # load an R package that lives on GitHub.
   ```

   b. Load the data and examine it using the commands

   ```
   library(dsData)
   str(Dishes)
   pairs(Time ~ ., data=Dishes)
   ```

   Comment on the relationships and possible transformations to be made.

   c. Fit a linear model predicting Time as a function the main of Diameter, Price, and Type, but with no interaction.

   d. Examine the diagnostic plots. What stands out to you?

   e. While most of the diagnostics look fine, there is weak evidence that there might be some heteroskedasticity. To address this (and provide an interesting model to interpret), refit your linear model but with a log-transformed Time and Price variables.

   f. What is your interpretation of the parameter associated with the Diameter variable on the original scale of the Time variable? Does this make sense to you considering the `pairs()` plot?

   g. How does a 20% increase in Price affect the polishing time?

Chapter 7

Variable Selection

Given a set of data, we are interested in selecting the best subset of predictors for the following reasons:

1. Occam's Razor tells us that from a list of plausible model or explanations, the simplest is usually the best. In the statistics sense, I want the smallest model that adequately explains the observed data patterns.
2. Unnecessary predictors add noise to the estimates of other quantities and will waste degrees of freedom, possibly increasing the estimate of $\hat{\sigma}^2$.
3. We might have variables that are collinear.

The problems that arise in the diagnostics of a model will often lead a researcher to consider other models, for example to include a quadratic term to account for curvature. The model building process is often an iterative procedure where we build a model, examine the diagnostic plots and consider what could be added or modified to correct issues observed.

7.1 Nested Models

Often one model is just a simplification of another and can be obtained by setting some subset of β_i values equal to zero. Those models can be adequately compared by the F-test, which we have already made great use of.

We should be careful to note that we typically do not want to remove the main covariate from the model if the model uses the covariate in a more complicated fashion. For example, if my model is

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \epsilon$$

where $\epsilon \sim N\left(0, \sigma^2\right)$, then considering the simplification $\beta_1 = 0$ and removing the effect of x is not desirable because that forces the parabola to be symmetric about $x = 0$. Similarly, if the model contains an interaction effect, then the removal of the main effect drastically alters the interpretation of the interaction coefficients and should be avoided. Often times removing a lower complexity term while keeping a higher complexity term results in unintended consequences and is typically not recommended.

7.2 Testing-Based Model Selection

Starting with a model that is likely too complex, consider a list of possible terms to remove and remove each in turn while evaluating the resulting model to the starting model using an F-test. Whichever term has the highest p-value is removed and the process is repeated until no more terms have non-significant p-values. This is often referred to as *backward selection*.

It should be noted that the cutoff value for significance here does not have to be $\alpha = 0.05$. If prediction performance is the primary goal, then a more liberal α level is appropriate.

Starting with a model that is likely too small, consider adding terms until there are no more terms that when added to the model are significant. This is called *forward selection*.

This is a hybrid between forward selection and backward elimination. At every stage, a term is either added or removed. This is referred to as *stepwise selection*.

Stepwise, forward, and backward selection are commonly used but there are some issues.

1. Because of the “one-at-a-time” nature of the addition/deletion, the most optimal model might not be found.

2. p-values should not be treated literally. Because the multiple comparisons issue is completely ignored, the p-values are lower than they should be if multiple comparisons were accounted for. As such, it is possible to sort through a huge number of potential covariates and find one with a low p-value simply by random chance. This is “data dredging” and is a serious issue.

3. As a non-thinking algorithm, these methods ignore the science behind that data and might include two variables that are highly collinear or might ignore variables that are scientifically interesting.

7.2.1 Example - U.S. Life Expectancy

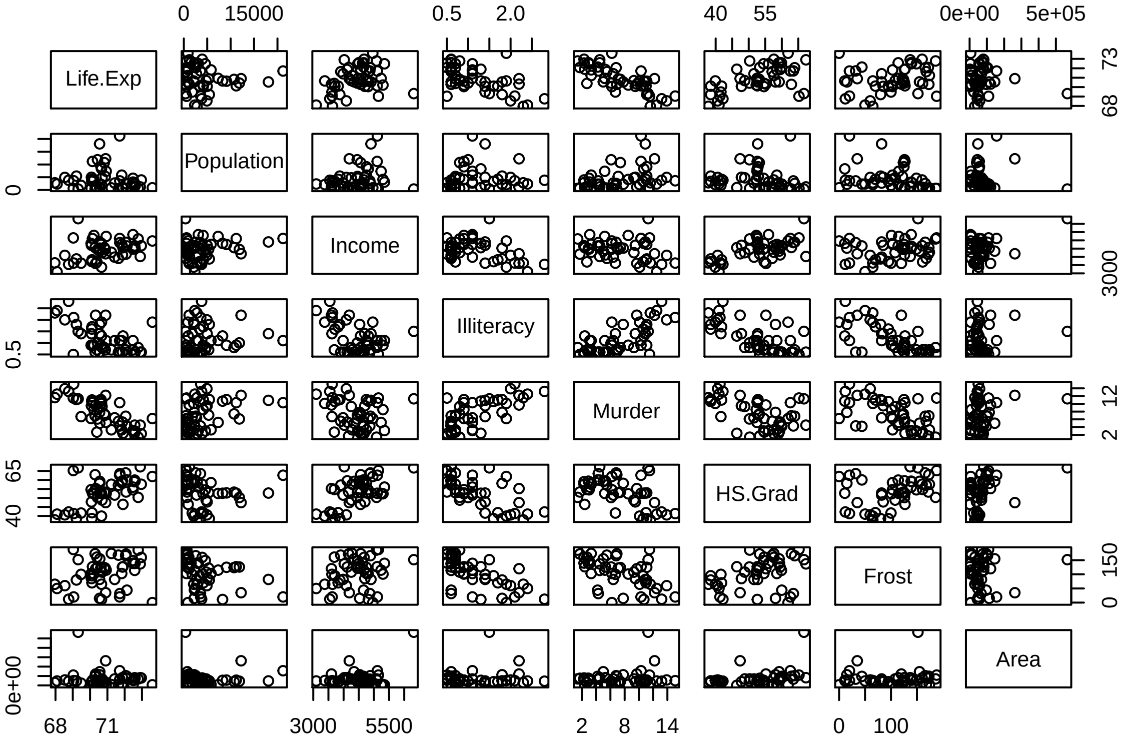
Using data from the Census Bureau we can look at the life expectancy as a response to a number of predictors. One R function that is often convenient to use is the update() function that takes a `lm()` object and adds or removes things from the formula. The notation `. ~ .` means to leave the response and all the predictors alone, while `. ~ . + vnew` will add the main effect of vnew to the model.

```r
library(faraway)
library(dplyr)   # for the %>% operator!

# Convert from a matrix to a data frame with state abbreviations
state.data <- data.frame(state.x77, row.names=state.abb)
str(state.data)
```

```
## 'data.frame':    50 obs. of  8 variables:
##  $ Population: num  3615 365 2212 2110 21198 ...
##  $ Income    : num  3624 6315 4530 3378 5114 ...
##  $ Illiteracy: num  2.1 1.5 1.8 1.9 1.1 0.7 1.1 0.9 1.3 2 ...
##  $ Life.Exp  : num  69 69.3 70.5 70.7 71.7 ...
##  $ Murder    : num  15.1 11.3 7.8 10.1 10.3 6.8 3.1 6.2 10.7 13.9 ...
##  $ HS.Grad   : num  41.3 66.7 58.1 39.9 62.6 63.9 56 54.6 52.6 40.6 ...
##  $ Frost     : num  20 152 15 65 20 166 139 103 11 60 ...
##  $ Area      : num  50708 566432 113417 51945 156361 ...
```

We should first look at the

```
pairs(Life.Exp ~ ., data=state.data)
```



I want to add a quadratic effect for HS.Grad rate and for Income. We'll add it to the data frame and then perform the backward elimination method starting with the model with all predictors as main effects.

```
state.data$HS.Grad.2 <- state.data$HS.Grad ^ 2
state.data$Income.2  <- state.data$Income  ^ 2
# explicitly define my starting model
m1 <- lm(Life.Exp ~ Population + Income + Illiteracy +
                    Murder + HS.Grad + Frost + HS.Grad.2 + Income.2, data=state.data)
#
# Define the same model, but using shorthand
# The '.' means everything else in the data frame
m1 <- lm( Life.Exp ~ ., data=state.data)
summary(m1)$coefficients %>% round(digits=3)
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    63.413      7.449   8.513    0.000
## Population      0.000      0.000   1.310    0.198
## Income          0.004      0.003   1.605    0.116
## Illiteracy      0.255      0.388   0.656    0.515
## Murder         -0.300      0.049  -6.088    0.000
## HS.Grad        -0.053      0.214  -0.248    0.806
## Frost          -0.004      0.003  -1.374    0.177
## Area            0.000      0.000   0.656    0.516
## HS.Grad.2       0.001      0.002   0.460    0.648
## Income.2        0.000      0.000  -1.618    0.114
```

The signs make reasonable sense (higher murder rates decrease life expectancy) but covariates like `Income` are not significant, which is surprising. We will first remove the term with the highest p-value, which is the state's `HS.Grad`. However, I don't want to remove the lower-order graduation term and keep the squared-term. So instead I will remove both of them since they are the highest p-values. Notice that `HS.Grad` is correlated with Income and Illiteracy.

```
# Remove Graduation Rate from the model from the model
m1 <- update(m1, .~. - HS.Grad - HS.Grad.2)
summary(m1)$coefficients %>% round(digits=3)
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   62.497      6.395   9.772    0.000
## Population     0.000      0.000   0.838    0.407
## Income         0.005      0.003   1.804    0.078
## Illiteracy    -0.076      0.358  -0.212    0.833
## Murder        -0.308      0.047  -6.573    0.000
## Frost         -0.006      0.003  -1.948    0.058
## Area           0.000      0.000   1.688    0.099
## Income.2       0.000      0.000  -1.750    0.087
```

```
# Next remove Illiteracy
m1 <- update(m1, .~. - Illiteracy)
summary(m1)$coefficients %>% round(digits=3)
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   61.779      5.367  11.510    0.000
## Population     0.000      0.000   0.860    0.394
## Income         0.005      0.002   2.165    0.036
## Murder        -0.312      0.043  -7.304    0.000
## Frost         -0.006      0.003  -2.261    0.029
## Area           0.000      0.000   1.736    0.090
## Income.2       0.000      0.000  -2.069    0.045
```

```
# And Population...
m1 <- update(m1, .~. - Population)
summary(m1)$coefficients %>% round(digits=3)
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   59.985      4.931  12.166    0.000
## Income         0.006      0.002   2.702    0.010
## Murder        -0.297      0.039  -7.633    0.000
## Frost         -0.006      0.003  -2.471    0.017
## Area           0.000      0.000   1.746    0.088
## Income.2       0.000      0.000  -2.533    0.015
```

```
# Remove Area from the model
m1 <- update(m1, .~. - Area)
summary(m1)$coefficients %>% round(digits=3)
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   63.685      4.552  13.990    0.000
## Income         0.004      0.002   2.078    0.043
## Murder        -0.285      0.039  -7.278    0.000
## Frost         -0.006      0.003  -2.235    0.030
## Income.2       0.000      0.000  -1.864    0.069
```

The removal of Income.2 is a tough decision because the p-value is very close to $\alpha = 0.05$ and might be left in if it makes model interpretation easier or if the researcher feels a quadratic effect in income is appropriate (perhaps rich people are too stressed?).

```
summary(m1)
```

```
##
```

```
## Call:
## lm(formula = Life.Exp ~ Income + Murder + Frost + Income.2, data = state.data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.44396 -0.53231 -0.04994  0.50160  1.51754
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.369e+01  4.552e+00  13.990  < 2e-16 ***
## Income       4.017e-03  1.933e-03   2.078   0.0434 *
## Murder      -2.852e-01  3.918e-02  -7.278 3.95e-09 ***
## Frost       -5.686e-03  2.544e-03  -2.235   0.0304 *
## Income.2    -3.955e-07  2.121e-07  -1.864   0.0688 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7619 on 45 degrees of freedom
## Multiple R-squared:  0.7042, Adjusted R-squared:  0.6779
## F-statistic: 26.78 on 4 and 45 DF,  p-value: 2.112e-11
```

We are left with a model that adequately explains `Life.Exp` but we should be careful to note that just because a covariate was removed from the model does not imply that it isn't related to the response. For example, being a high school graduate is highly correlated with not being illiterate as is `Income` and thus replacing `Illiteracy` shows that illiteracy is associated with lower life expectancy, but it is not as predictive as `Income`.

```
m2 <- lm(Life.Exp ~ Illiteracy+Murder+Frost, data=state.data)
summary(m2)
```

```
##
## Call:
## lm(formula = Life.Exp ~ Illiteracy + Murder + Frost, data = state.data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.59010 -0.46961  0.00394  0.57060  1.92292
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 74.556717   0.584251 127.611  < 2e-16 ***
## Illiteracy  -0.601761   0.298927  -2.013  0.04998 *
## Murder      -0.280047   0.043394  -6.454 6.03e-08 ***
## Frost       -0.008691   0.002959  -2.937  0.00517 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7911 on 46 degrees of freedom
## Multiple R-squared:  0.6739, Adjusted R-squared:  0.6527
## F-statistic: 31.69 on 3 and 46 DF,  p-value: 2.915e-11
```

Notice that the R^2 values for both models are quite similar 0.7042 vs 0.6739 but the first model with the higher R^2 has one more predictor variable? Which model should I prefer? I can't do an F-test because these models are not nested.

7.3 Criterion Based Procedures

7.3.1 Information Criterions

It is often necessary to compare models that are not nested. For example, I might want to compare

$$y = \beta_0 + \beta_1 x + \epsilon$$

vs

$$y = \beta_0 + \beta_2 w + \epsilon$$

This comparison comes about naturally when doing forward model selection and we are looking for the "best" covariate to add to the model first.

Akaike introduced his criterion (which he called "An Information Criterion") as

$$AIC = \underbrace{-2\,\log L\left(\hat{\boldsymbol{\beta}}, \hat{\sigma} |\, \text{data}\right)}_{\text{decreases if RSS decreases}} + \underbrace{2p}_{\text{increases as p increases}}$$

where $L\left(\hat{\boldsymbol{\beta}} |\, \text{data}\right)$ is the likelihood function and p is the number of elements in the $\hat{\boldsymbol{\beta}}$ vector and we regard a lower AIC value as better. Notice the $2p$ term is essentially a penalty on adding addition covariates so to lower the AIC value, a new predictor must lower the negative log likelihood more than it increases the penalty.

To convince ourselves that the first summand decreases with decreasing RSS in the standard linear model, we examine the likelihood function

$$\begin{aligned} f\left(\boldsymbol{y}\,|\,\boldsymbol{\beta},\sigma,\boldsymbol{X}\right) &= \frac{1}{\left(2\pi\sigma^2\right)^{n/2}} \exp\left[-\frac{1}{2\sigma^2}\left(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}\right)^T\left(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}\right)\right] \\ &= L\left(\boldsymbol{\beta},\sigma\,|\,\boldsymbol{y},\boldsymbol{X}\right) \end{aligned}$$

and we could re-write this as

$$\begin{aligned} \log L\left(\hat{\boldsymbol{\beta}},\hat{\sigma}\,|\,\text{data}\right) &= -\log\left(\left(2\pi\hat{\sigma}^2\right)^{n/2}\right) - \frac{1}{2\hat{\sigma}^2}\left(\boldsymbol{y}-\boldsymbol{X}\hat{\boldsymbol{\beta}}\right)^T\left(\boldsymbol{y}-\boldsymbol{X}\hat{\boldsymbol{\beta}}\right) \\ &= -\frac{n}{2}\log\left(2\pi\hat{\sigma}^2\right) - \frac{1}{2\hat{\sigma}^2}\left(\boldsymbol{y}-\boldsymbol{X}\hat{\boldsymbol{\beta}}\right)^T\left(\boldsymbol{y}-\boldsymbol{X}\hat{\boldsymbol{\beta}}\right) \\ &= -\frac{1}{2}\left[n\log\left(2\pi\hat{\sigma}^2\right) + \frac{1}{\hat{\sigma}^2}\left(\boldsymbol{y}-\boldsymbol{X}\hat{\boldsymbol{\beta}}\right)^T\left(\boldsymbol{y}-\boldsymbol{X}\hat{\boldsymbol{\beta}}\right)\right] \\ &= -\frac{1}{2}\left[+n\log\left(2\pi\right) + n\log\hat{\sigma}^2 + \frac{1}{\hat{\sigma}^2}RSS\right] \end{aligned}$$

It isn't clear what we should do with the $n\log\left(2\pi\right)$ term in the $\log L()$ function. There are some compelling reasons to ignore it and just use the second, and there are reasons to use both terms. Unfortunately, statisticians have not settled on one convention or the other and different software packages might therefore report different values for AIC.

As a general rule of thumb, if the difference in AIC values is less than two then the models are not significantly different, differences between 2 and 4 AIC units are marginally significant and any difference greater than 4 AIC units is highly significant.

Notice that while this allows us to compare models that are not nested, it does require that the same data are used to fit both models. Because I could start out with my data frame including both x and x^2, (or more generally x and $f\left(x\right)$ for some function $f()$) you can regard a transformation of a covariate as "the same data". However, a transformation of a y-variable is not and therefore we cannot use AIC to compare a models `log(y) ~ x` versus the model `y ~ x`.

Another criterion that might be used is *Bayes Information Criterion* (BIC) which is

$$BIC = -2\,\log L\left(\hat{\boldsymbol{\beta}}, \hat{\sigma} |\,\text{data}\right) + p\log n$$

and this criterion punishes large models more than AIC does (because $\log n > 2$ for $n \geq 8$)

The AIC value of a linear model can be found using the AIC() on a lm() object.

```
AIC(m1)
```

```
## [1] 121.4293
```

```
AIC(m2)
```

```
## [1] 124.2947
```

Because the AIC value for the first model is lower, we would prefer the first model that includes both `Income` and `Income.2` compared to model 2, which was `Life.Exp ~ Illiteracy+Murder+Frost`.

7.3.2 Adjusted `R-sq`

One of the problems with R^2 is that it makes no adjustment for how many parameters in the model. Recall that R^2 was defined as

$$R^2 = \frac{RSS_S - RSS_C}{RSS_S} = 1 - \frac{RSS_C}{RSS_S}$$

where the simple model was the intercept only model. We can create an R^2_{adj} statistic that attempts to add a penalty for having too many parameters by defining

$$R^2_{adj} = 1 - \frac{RSS_C/\left(n-p\right)}{RSS_S/\left(n-1\right)}$$

With this adjusted definition, adding a variable to the model that has no predictive power will decrease R^2_{adj}.

7.3.3 Example

Returning to the life expectancy data, we could start with a simple model add covariates to the model that have the lowest AIC values. R makes this easy with the function add1() which will take a linear model (which includes the data frame that originally defined it) and will sequentially add all of the possible terms that are not currently in the model and report the AIC values for each model.

```
# Define the biggest model I wish to consider
biggest <- Life.Exp ~ Population + Income + Illiteracy + Murder +
                      HS.Grad + Frost + Area + HS.Grad.2 + Income.2

# Define the model I wish to start with
m <- lm(Life.Exp ~ 1, data=state.data)

add1(m, scope=biggest)  # what is the best addition to make?
```

```
## Single term additions
##
## Model:
## Life.Exp ~ 1
##            Df Sum of Sq    RSS    AIC
## <none>                  88.299 30.435
## Population  1     0.409 87.890 32.203
## Income      1    10.223 78.076  26.283
## Illiteracy  1    30.578 57.721  11.179
## Murder      1    53.838 34.461 -14.609
## HS.Grad     1    29.931 58.368  11.737
## Frost       1     6.064 82.235  28.878
## Area        1     1.017 87.282  31.856
## HS.Grad.2   1    27.414 60.885  13.848
## Income.2    1     7.464 80.835  28.020
```

Clearly the additiona of `Murder` to the model results in the lowest AIC value, so we will add `Murder` to the model. Notice the `<none>` row corresponds to the model m which we started with and it has a `RSS=88.299`. For each model considered, R will calculate the `RSS_{C}` for the new model and will calculate the difference between the starting model and the more complicated model and display this in the Sum of Squares column.

```
m <- update(m, . ~ . + Murder)  # add murder to the model
add1(m, scope=biggest)          # what should I add next?
```

```
## Single term additions
##
## Model:
## Life.Exp ~ Murder
##            Df Sum of Sq    RSS     AIC
## <none>                  34.461 -14.609
## Population  1    4.0161 30.445 -18.805
## Income      1    2.4047 32.057 -16.226
## Illiteracy  1    0.2732 34.188 -13.007
## HS.Grad     1    4.6910 29.770 -19.925
## Frost       1    3.1346 31.327 -17.378
## Area        1    0.4697 33.992 -13.295
## HS.Grad.2   1    4.4396 30.022 -19.505
## Income.2    1    1.8972 32.564 -15.441
```

There is a companion function to `add1()` that finds the best term to drop. It is conveniently named `drop1()` but here the `scope` parameter defines the smallest model to be considered.

It would be nice if all of this work was automated. Again, R makes our life easy and the function `step()` does exactly this. The set of models searched is determined by the scope argument which can be a *list* of two formulas with components upper and lower or it can be a single formula, or it can be blank. The right-hand-side of its lower component defines the smallest model to be considered and the right-hand-side of the upper component defines the largest model to be considered. If `scope` is a single formula, it specifies the upper component, and the lower model taken to be the intercept-only model. If scope is missing, the initial model is used as the upper model.

```
smallest <- Life.Exp ~ 1
biggest <- Life.Exp ~ Population + Income + Illiteracy +
                      Murder + HS.Grad + Frost + Area + HS.Grad.2 + Income.2
m <- lm(Life.Exp ~ Income, data=state.data)
step(m, scope=list(lower=smallest, upper=biggest))
```

```
## Start:  AIC=26.28
## Life.Exp ~ Income
##
##                Df Sum of Sq    RSS     AIC
## + Murder        1    46.020 32.057 -16.226
## + Illiteracy    1    21.109 56.968  12.523
## + HS.Grad       1    19.770 58.306  13.684
## + Income.2      1    19.062 59.015  14.288
## + HS.Grad.2   1    17.193 60.884  15.847
## + Area        1     5.426 72.650  24.682
## + Frost       1     3.188 74.889  26.199
## <none>                    78.076  26.283
## + Population  1     1.781 76.295  27.130
## - Income      1    10.223 88.299  30.435
##
## Step:  AIC=-16.23
## Life.Exp ~ Income + Murder
##
##              Df Sum of Sq    RSS     AIC
## + Frost       1     3.918 28.138 -20.745
## + Income.2    1     3.036 29.021 -19.200
## + Population  1     2.552 29.504 -18.374
## + HS.Grad     1     2.388 29.668 -18.097
## + HS.Grad.2   1     2.199 29.857 -17.780
## <none>                    32.057 -16.226
## - Income      1     2.405 34.461 -14.609
## + Illiteracy  1     0.011 32.046 -14.242
## + Area        1     0.000 32.057 -14.226
## - Murder      1    46.020 78.076  26.283
##
## Step:  AIC=-20.74
## Life.Exp ~ Income + Murder + Frost
##
##              Df Sum of Sq    RSS     AIC
## + HS.Grad     1     2.949 25.189 -24.280
## + HS.Grad.2   1     2.764 25.375 -23.914
## + Income.2    1     2.017 26.121 -22.465
## + Population  1     1.341 26.797 -21.187
## <none>                    28.138 -20.745
## + Illiteracy  1     0.950 27.189 -20.461
## + Area        1     0.147 27.991 -19.007
## - Income      1     3.188 31.327 -17.378
## - Frost       1     3.918 32.057 -16.226
## - Murder      1    46.750 74.889  26.199
##
## Step:  AIC=-24.28
## Life.Exp ~ Income + Murder + Frost + HS.Grad
##
##              Df Sum of Sq    RSS     AIC
## + Population  1     1.887 23.302 -26.174
## + Income.2    1     1.864 23.326 -26.124
## - Income      1     0.182 25.372 -25.920
## <none>                    25.189 -24.280
## + HS.Grad.2   1     0.218 24.972 -22.714
## + Illiteracy  1     0.131 25.058 -22.541
## + Area        1     0.058 25.131 -22.395
## - HS.Grad     1     2.949 28.138 -20.745
## - Frost       1     4.479 29.668 -18.097
## - Murder      1    32.877 58.067  15.478
##
## Step:  AIC=-26.17
## Life.Exp ~ Income + Murder + Frost + HS.Grad + Population
```

```
##
##              Df Sum of Sq    RSS     AIC
## - Income      1     0.006 23.308 -28.161
## <none>                    23.302 -26.174
## + Income.2    1     0.790 22.512 -25.899
## - Population  1     1.887 25.189 -24.280
## + HS.Grad.2   1     0.006 23.296 -24.187
## + Illiteracy  1     0.004 23.298 -24.182
## + Area        1     0.000 23.302 -24.174
## - Frost       1     3.037 26.339 -22.048
## - HS.Grad     1     3.495 26.797 -21.187
## - Murder      1    34.739 58.041  17.456
##
## Step:  AIC=-28.16
## Life.Exp ~ Murder + Frost + HS.Grad + Population
##
##              Df Sum of Sq    RSS     AIC
## <none>                    23.308 -28.161
## + Income.2    1     0.031 23.277 -26.229
## + HS.Grad.2   1     0.007 23.301 -26.177
## + Income      1     0.006 23.302 -26.174
## + Illiteracy  1     0.004 23.304 -26.170
## + Area        1     0.001 23.307 -26.163
## - Population  1     2.064 25.372 -25.920
## - Frost       1     3.122 26.430 -23.877
## - HS.Grad     1     5.112 28.420 -20.246
## - Murder      1    34.816 58.124  15.528
```

```
##
## Call:
## lm(formula = Life.Exp ~ Murder + Frost + HS.Grad + Population,
##     data = state.data)
##
## Coefficients:
## (Intercept)       Murder        Frost      HS.Grad   Population
##   7.103e+01   -3.001e-01   -5.943e-03    4.658e-02    5.014e-05
```

Notice that our model selected by `step()` is not the same model we obtained when we started with the biggest model and removed things based on p-values.

The log-likelihood is only defined up to an additive constant, and there are different conventional constants used. This is more annoying than anything because all we care about for model selection is the difference between AIC values of two models and the additive constant cancels. The only time it matters is when you have two different ways of extracting the AIC values. Recall the model we fit using the top-down approach was

```r
# m1 was
m1 <- lm(Life.Exp ~ Income + Murder + Frost + Income.2, data = state.data)
AIC(m1)
```

```
## [1] 121.4293
```

and the model selected by the stepwise algorithm was

```r
m3 <- lm(Life.Exp ~ Murder + Frost + HS.Grad + Population, data = state.data)
AIC(m3)
```

```
## [1] 115.7326
```

Because `step()` and `AIC()` are following different conventions the absolute value of the AICs are different, but the difference between the two is constant no matter which function we use.

First we calculate the difference using the AIC() function:

```
AIC(m1) - AIC(m3)
```

```
## [1] 5.696681
```

and next we use `add1()` on both models to see what the AIC values for each.

```
add1(m1, scope=biggest)
```

```
## Single term additions
##
## Model:
## Life.Exp ~ Income + Murder + Frost + Income.2
##            Df Sum of Sq    RSS     AIC
## <none>                  26.121 -22.465
## Population  1   0.42412 25.697 -21.283
## Illiteracy  1   0.10097 26.020 -20.658
## HS.Grad     1   2.79527 23.326 -26.124
## Area        1   1.69309 24.428 -23.815
## HS.Grad.2   1   2.79698 23.324 -26.127
```

```
add1(m3, scope=biggest)
```

```
## Single term additions
##
## Model:
## Life.Exp ~ Murder + Frost + HS.Grad + Population
##            Df Sum of Sq    RSS     AIC
## <none>                  23.308 -28.161
## Income      1 0.0060582 23.302 -26.174
## Illiteracy  1 0.0039221 23.304 -26.170
## Area        1 0.0007900 23.307 -26.163
## HS.Grad.2   1 0.0073439 23.301 -26.177
## Income.2    1 0.0314248 23.277 -26.229
```

Using these results, we can calculate the difference in AIC values to be the same as we calculated before

$$\begin{aligned} -22.465 - -28.161 &= -22.465 + 28.161 \\ &= 5.696 \end{aligned}$$

7.4 Exercises

1. Consider the `prostate` data from the `faraway` package. The variable `lpsa` is a measurement of a prostate specific antigen which higher levels are indicative of prostate cancer. Use `lpsa` as the response and all the other variables as predictors (no interactions). Determine the "best" model using:

    a. Backward elimination using the analysis of variance F-statistic as the criteria.

    b. Forward selection using AIC as the criteria.

2. Again from the `faraway` package, use the `divusa` which has diviorce rates for each year from 1920-1996 along with other population information for each year. Use `divorce` as the response variable and all other variables as the predictors.

    a. Determine the best model using stepwise selection starting from the intercept only model and the most complex model being all main effects (no interactions). Use the F-statistic to determine significance. Note: add1(), drop1(), and step() allow an option of test=‘F’ to use an F-test instead of AIC.

b. Following the stepwise selection, comment on the relationship between p-values used and the AIC difference observed. Do the AIC rules of thumb match the p-value interpretation?

Chapter 8

One way ANOVA

```r
# Load the libraries I'll use
library(faraway)
library(ggplot2)
library(dplyr)
library(multcompView)
```

Given a categorical covariate (which I will call a factor) with I levels, we are interested in fitting the model

$$y_{ij} = \mu + \tau_i + \epsilon_{ij}$$

where $\epsilon_{ij} \overset{iid}{\sim} N\left(0, \sigma^2\right)$, μ is the overall mean, and τ_i are the offset of factor level i from μ. Unfortunately this model is not identifiable because I could add a constant (say 5) to μ and subtract that same constant from each of the τ_i values and the group mean $\mu + \tau_i$ would not change. There are two easy restrictions we could make to make the model identifiable:

1. Set $\mu = 0$. In this case, τ_i represents the expected value of an observation in group level i. We call this the "cell means" representation.

2. Set $\tau_1 = 0$. Then μ represents the expected value of treatment 1, and the τ_i values will represent the offsets from group 1. The group or level that we set to be zero is then referred to as the reference group. We can call this the "offset from reference" model.

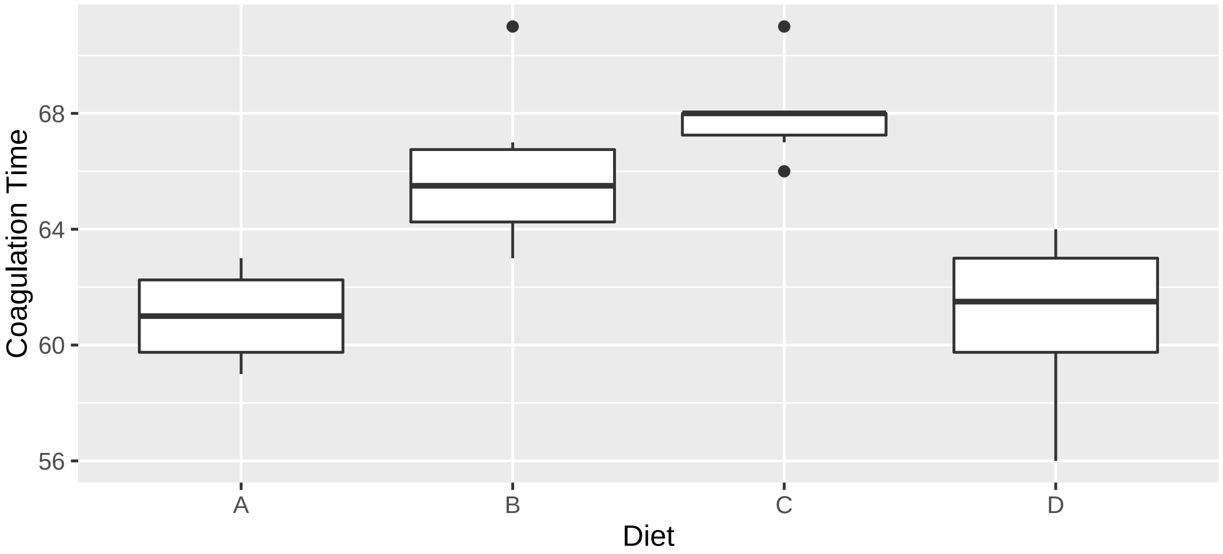
We will be interested in testing the null and alternative hypotheses

$$H_0: \quad y_{ij} = \mu + \epsilon_{ij}$$
$$H_a: \quad y_{ij} = \mu + \alpha_i + \epsilon_{ij}$$

8.1 An Example

We look at a dataset that comes from the study of blood coagulation times: 24 animals were randomly assigned to four different diets and the samples were taken in a random order. The diets are denoted as A,B,C,and D and the response of interest is the amount of time it takes for the blood to coagulate.

```r
data(coagulation)
ggplot(coagulation, aes(x=diet, y=coag)) +
    geom_boxplot() +
    labs( x='Diet', y='Coagulation Time' )
```



Just by looking at the graph, we expect to see that diets A and D are similar while B and C are different from A and D and possibly from each other, too. We first fit the offset model.

```
m <- lm(coag ~ diet, data=coagulation)
summary(m)
```

```
##
## Call:
## lm(formula = coag ~ diet, data = coagulation)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -5.00  -1.25   0.00   1.25   5.00
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 6.100e+01  1.183e+00  51.554  < 2e-16 ***
## dietB       5.000e+00  1.528e+00   3.273 0.003803 **
## dietC       7.000e+00  1.528e+00   4.583 0.000181 ***
## dietD       2.991e-15  1.449e+00   0.000 1.000000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.366 on 20 degrees of freedom
## Multiple R-squared:  0.6706, Adjusted R-squared:  0.6212
## F-statistic: 13.57 on 3 and 20 DF,  p-value: 4.658e-05
```

Notice that diet A is the reference level and it has a mean of 61. Diet B has an offset from A of 5, etc. From the very small F-statistic, we conclude that simple model

$$y_{ij} = \mu + \epsilon_{ij}$$

is not sufficient to describe the data.

8.2 Degrees of Freedom

Throughout the previous example, the degrees of freedom that are reported keeps changed depending on what models we are comparing. The simple model we are considering is

$$y_{ij} \sim \mu + \epsilon_{ij}$$

which has 1 parameter that defines the expected value versus

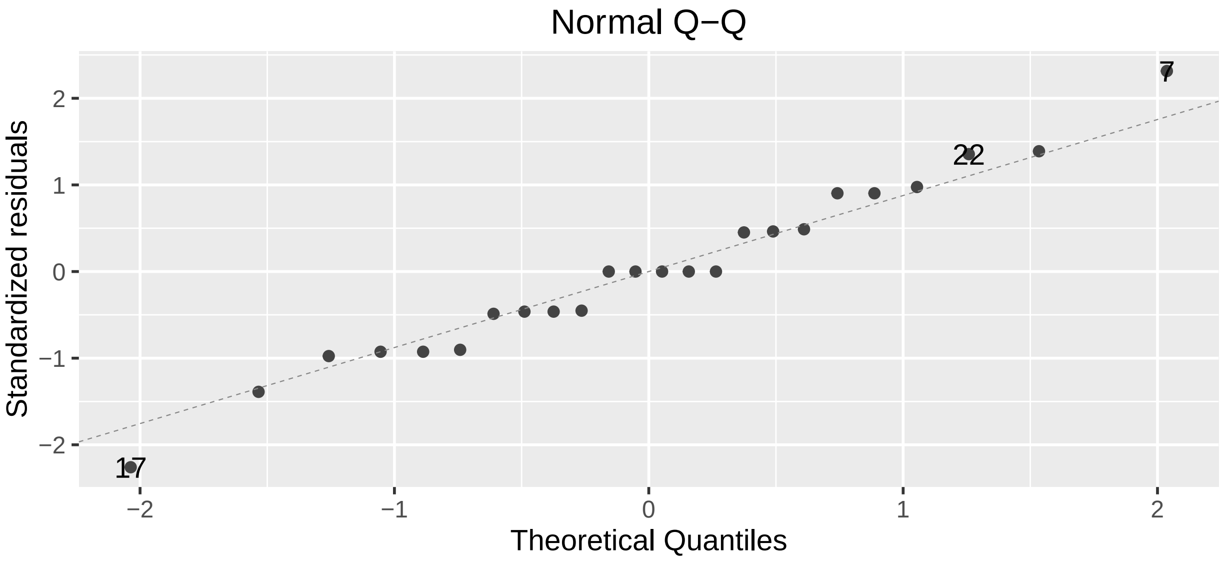
$$y_{ij} \sim \mu + \tau_i + \epsilon_{ij}$$

where there really are only 4 parameters that define the expected value because $\tau_1 = 0$. In general, the larger model is only adding $I - 1$ terms to the model where I is the number of levels of the factor of interest.

8.3 Diagnostics

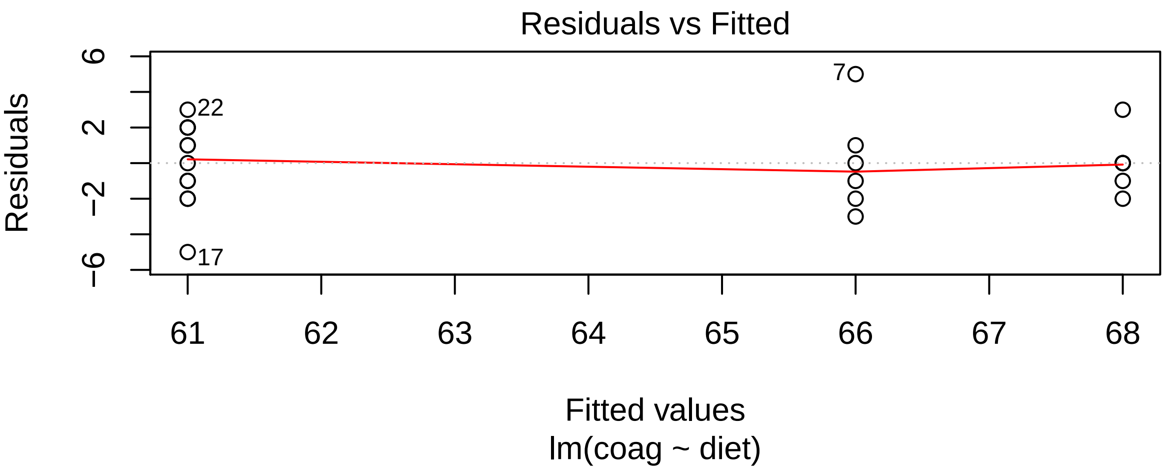
It is still important to check the diagnostics plots, but certain diagnostic plots will be useless. In particular, we need to be concerned about constant variance among the groups and normality of the residuals.

```
library(ggfortify)
m <- lm(coag ~ diet, data=coagulation)
autoplot(m, which=2)  #  QQ plot
```



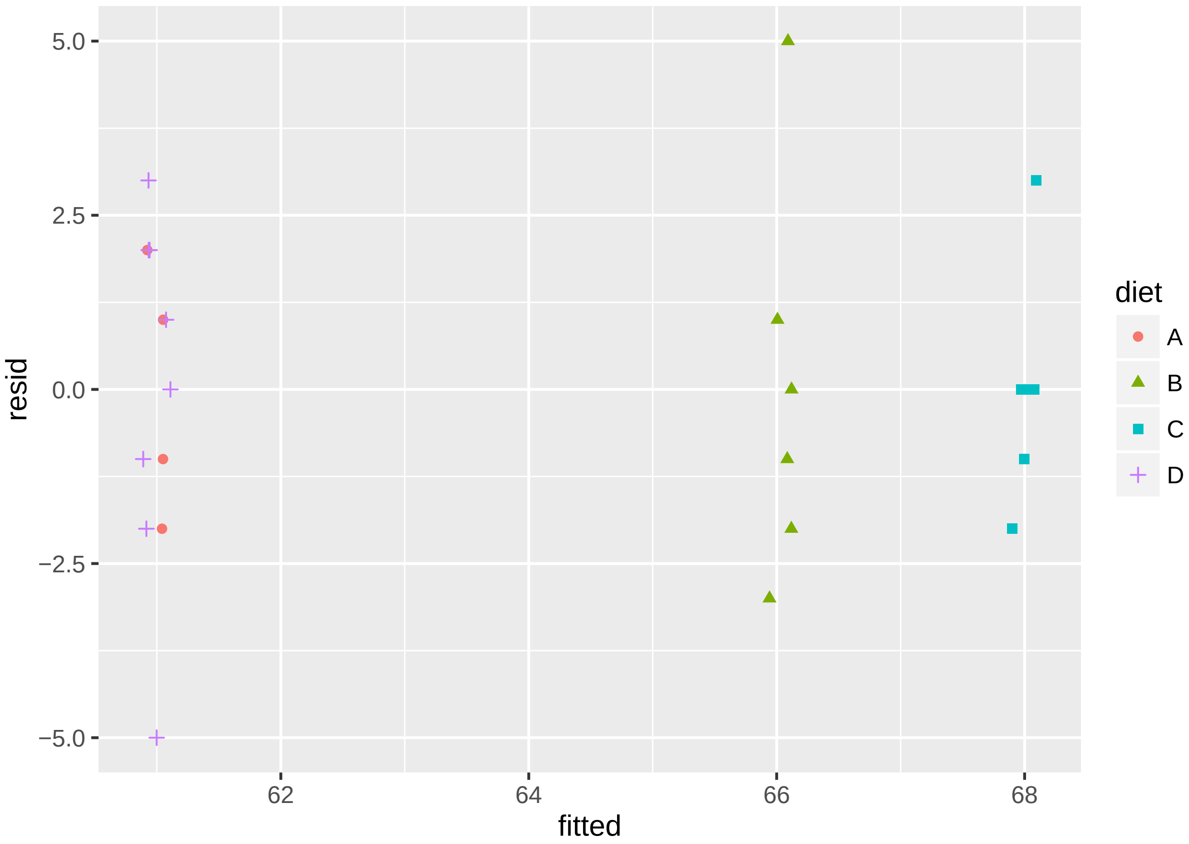
The residual plots however, might need a little bit of extra work, because there are only four possible predicted values (actually 3 because group A and D have the same predicted values). Note that we actually have $n = 24$ observations, but I can only see 16 of them.

```
plot(m, which=1)  #  Residals vs fitted
```



To remedy this, we will plot the residuals vs fitted by hand, and add a little bit of random noise to the fitted value, just so that we don't have points stack up on top of each other. Lets also add a different shape for each diet.

```
coagulation$fitted <- predict(m)
coagulation$resid <- resid(m)
ggplot(coagulation, aes(x=fitted, y=resid, shape=diet, color=diet)) +
  geom_point(position=position_jitter(w=0.3, h=0))
```



8.4 Pairwise Comparisons

After detecting differences in the factor levels, we are often interested in which factor levels are different from which. Often we are interested in comparing the mean of level i with the mean of level j. As usual we let the vector of parameter estimates be $\hat{\boldsymbol{\beta}}$ then the contrast of interested can be written as

$$\boldsymbol{c}^{T}\hat{\boldsymbol{\beta}}\pm t_{n-p}^{1-\alpha/2}\ \hat{\sigma}\sqrt{\boldsymbol{c}^{T}\left(\boldsymbol{X}^{T}\boldsymbol{X}\right)^{-1}\boldsymbol{c}}$$

for some vector $\boldsymbol{c}$.

Unfortunately this interval does not take into account the multiple comparisons issue (i.e. we are making $I(I-1)/2$ contrasts if our factor has I levels). To account for this, we will not use a quantile from a t-distribution, but from Tukey's studentized range distribution $q_{n,n-I}$ divided by $\sqrt{2}$. The intervals we will use are:

$$\boldsymbol{c}^{T}\hat{\boldsymbol{\beta}}\pm\frac{q_{n,n-I}^{1-\alpha/2}}{\sqrt{2}}\ \hat{\sigma}\sqrt{\boldsymbol{c}^{T}\left(\boldsymbol{X}^{T}\boldsymbol{X}\right)^{-1}\boldsymbol{c}}$$

There are several ways to make R calculate this interval (See the contrasts chapter for a more general treatment of this.), but the easiest is to use `TukeyHSD()`. This function will calculate all of these pairwise intervals using the above formula, which is commonly known as Tukey's Honestly Significant Differences. This function expects to receive output from the aov() function. The aov() is similar to lm() but does not accept continuous covariates in the model. Because it is possible to convert a `lm` output object to an `aov` object, I typically will do the following

```
m <- lm(coag ~ diet, data=coagulation)   # use the lm() function as usual
Pvalues <- TukeyHSD( aov(m), level=.90 ) # convert to the format that TukeyHSD likes...
Pvalues
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = m)
##
## $diet
##     diff         lwr       upr     p adj
## B-A    5   0.7245544  9.275446 0.0183283
## C-A    7   2.7245544 11.275446 0.0009577
## D-A    0  -4.0560438  4.056044 1.0000000
## C-B    2  -1.8240748  5.824075 0.4766005
## D-B   -5  -8.5770944 -1.422906 0.0044114
## D-C   -7 -10.5770944 -3.422906 0.0001268
```

Here we see that diets A and D are similar to each other, but different than B and C and that B and C are not statistically different from each other at the 0.10 level.

8.4.1 Presentation of Results

The display of `TukeyHSD` is pretty annoying and often I want to turn the vector of adjusted p-values into a matrix of p-values. We want a matrix that is $p \times p$.

```
# Convert to a matrix of adjusted p-values
Pmatrix <- vec2mat( Pvalues$diet[, 'p adj'] )
Pmatrix
```

```
##             B            C            D            A
## B 1.000000000 0.4766005178 0.0044113688 0.0183282757
## C 0.476600518 1.0000000000 0.0001267866 0.0009576856
## D 0.004411369 0.0001267866 1.0000000000 1.0000000000
## A 0.018328276 0.0009576856 1.0000000000 1.0000000000
```

From this matrix of adjusted p-values, there are many functions that will be helpful. One very common thing is to use a lettering scheme that indicates which groups are different. So if two groups share a letter, they are not significantly different.

```
multcompLetters(Pmatrix)
```

```
##   B   C   D   A
## "a" "a" "b" "b"
```

To make our lives easier while making graphs, we *really* want to be able to add these back into our original data frame. Unfortunately the `multcompView` package that does this doesn't really make it easy to do this. So I wrote us a little function that will suffice:

```
#' Create a data frame with significant groupings
#'
#' This function runs TukeyHSD on the input model and then creates a data frame
#' with a column for the factor and a second for the Significance Group
#'
#' @param model The output of a lm() or aov() call that can be coerced to an aov object.
#' @param variable The variable of interest.
#' @output A data frame with a column for factor and another for the signicance group.
make_TukeyHSD_letters <- function(model, variable){
  Tukey <- TukeyHSD(aov(model))[[variable]]
  temp <- Tukey[,'p adj'] %>%
    vec2mat() %>%
    multcompLetters()
  out <- data.frame(group = names(temp$Letters), SigGroup=temp$Letters)
  colnames(out)[1] <- variable
  out
}
```

Now that the function is defined, we can happily use it.

```
make_TukeyHSD_letters(m, 'diet')
```

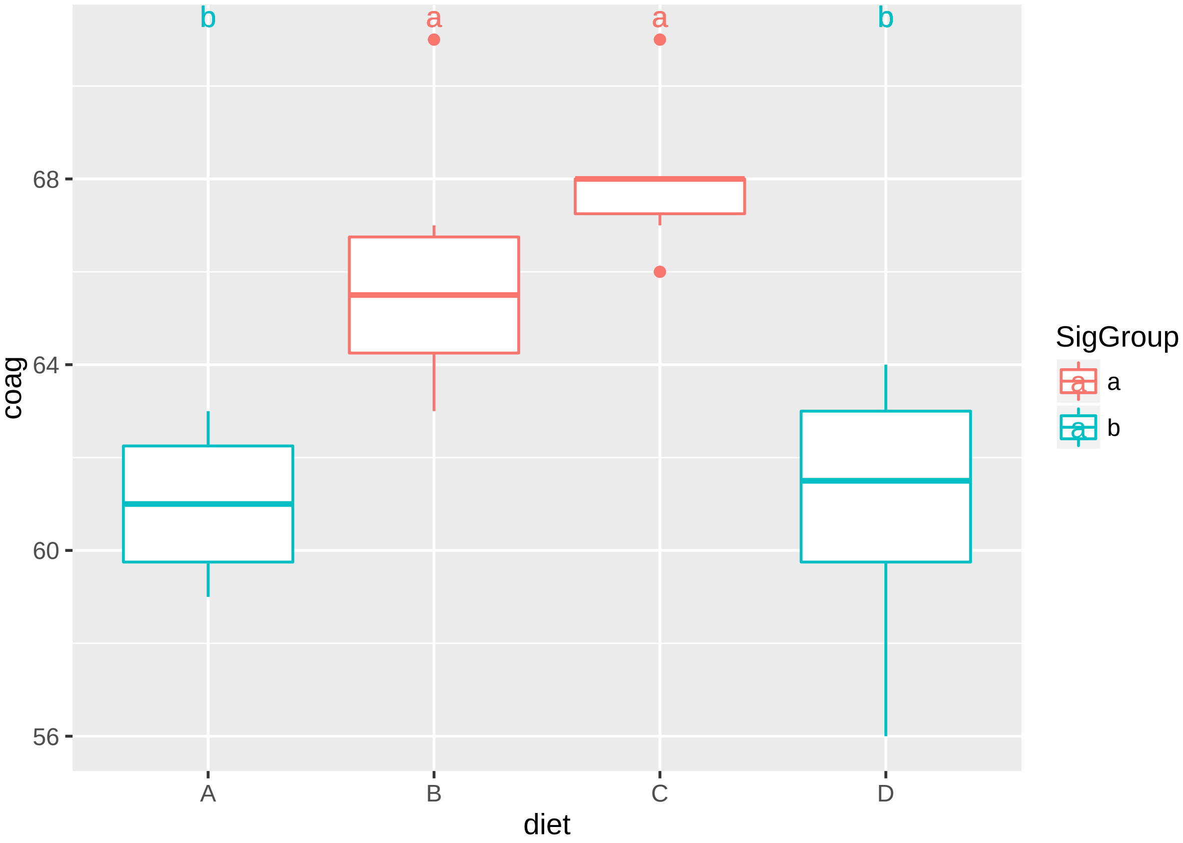
```
##   diet SigGroup
## B    B        a
## C    C        a
## D    D        b
## A    A        b
```

This is useful, but it will be more useful when I merge this small data frame with the original data and then each observation will have an associated significance group.

```
coagulation <- coagulation %>%
  left_join( make_TukeyHSD_letters(m, 'diet') )
```

```
## Joining, by = "diet"
```

```
ggplot(coagulation, aes(x=diet, y=coag, color=SigGroup)) +
  geom_boxplot() +
  geom_text( aes(label=SigGroup), y=71.5)
```



8.5 Exercises

1. Use the dataset `chickwts` in the `faraway` package. This was an experiment to determine which feed types result in the largest chickens. A set of 71 chicks were all randomly assigned one of six feed types and their weight in grams after six weeks was recorded. Determine whether there are differences in the weights of chickens according to their feed. Perform all necessary model diagnostics and examine the contrasts between each pair of feed levels. Summarize these results.

Bibliography