Statistics 2 for IBA

Christoph Walsh

Table of contents

Chapter 1

About

Welcome to the online "book" for the second-year IBA course *Statistics 2*. This book accompanies the content covered in the lectures. On Canvas, I will post which chapters/slides we will cover in each lecture.

In this course we cover estimation and inference in both the simple and multiple linear regression models. We provide foundations in theory and do many practical examples with real data. For these practical examples we will using the R programming language in RStudio. This follows on from the first-year IBA course Programming and Quantitative Skills that introduced you to R. You may find it helpful to read through the online book of that course here as a referesher.

As this book is new, it is likely that it will be edited throughout the semester.

Chapter 2

Visualizing the Relationship between Two Variables

Consider a business that is interested in the relationship between the amount it spends on advertising and its sales revenue. This relationship is very important for the business, because if advertising is not very effective at generating more sales, the business could save a lot of money by reducing its advertising.

The business has data on the advertising spending and sales in different media markets (i.e. locations with different TV, radio stations and newspapers) where it sells its products. For example, in New York the business spends $3m on advertising and has revenues of $40m. In Los Angeles it spends $4m on advertising and has revenues of $45m.

In this chapter we will learn how this business can visually assess the relationship between advertising and sales.

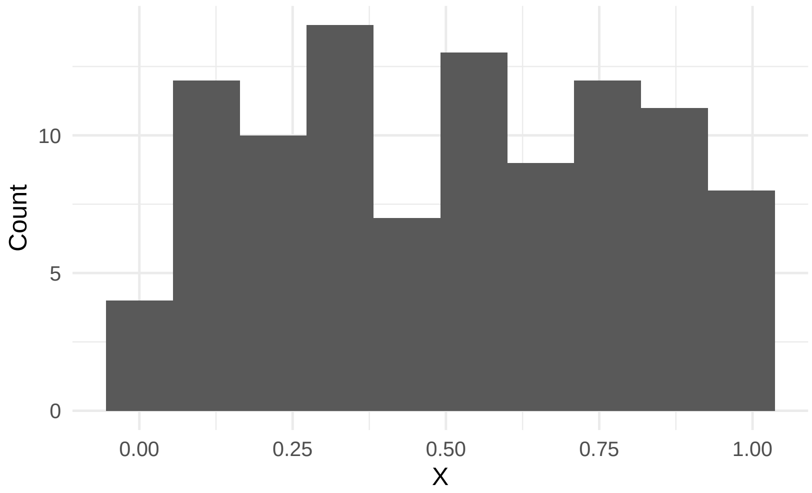
In subsequent chapters, we will learn how this business can quantify this relationship, estimate the impact of advertising on sales, and predict the sales at different advertising levels.

Before we look at an example of real data on advertising and sales, we will first discuss the topic more generally.

2.1 Visualizing a Single Variable

With data on a single variable x, we often visually inspect the data using histograms:

```
# Set seed to get the same random draws each time:
set.seed(30211)
# Generate 100 random observations from the uniform distribution:
df <- data.frame(x = runif(100, 0, 1))
# Load the ggplot2 package:
library(ggplot2)
# Create a histogram of the data using ggplot:
ggplot(df, aes(x)) +
  geom_histogram(bins = 10) +
  xlab("X") +
  ylab("Count") +
  theme_minimal()
```



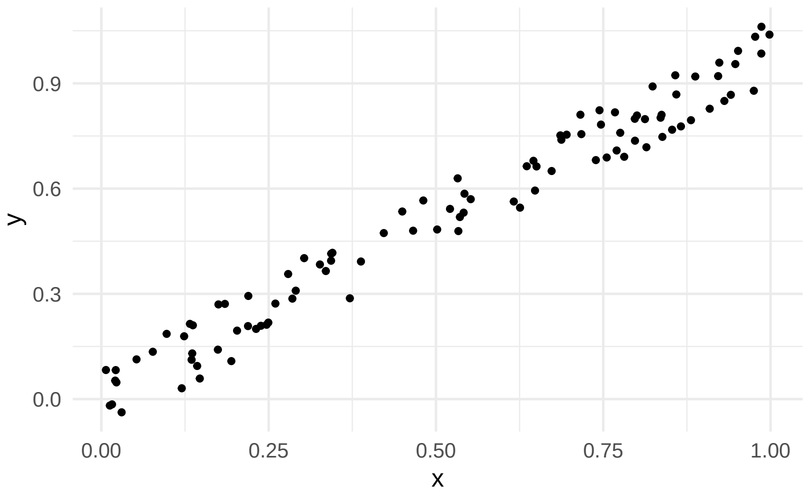
A histogram can tell us about the distribution of a single variable, such as:

- The center of the distribution, i.e. median (here roughly 0.5).
- The range of the data (here 0 and 1).
- The spread of the distribution (here roughly uniformly spread over the range).

2.2 Scatter Plots

With data on two variables x and y, we use *scatter plots* to inspect the relationship. A scatter plot has a dot for each data point (x_i, y_i) for $i = 1, \ldots, n$ on a Cartesian plane.

```
df <- data.frame(x = runif(100, 0, 1))
df$y <- df$x + runif(100, -0.1, 0.1)
ggplot(df, aes(x, y)) +
  geom_point(size = 1) +
  theme_minimal()
```



2.3 Positive Relationship

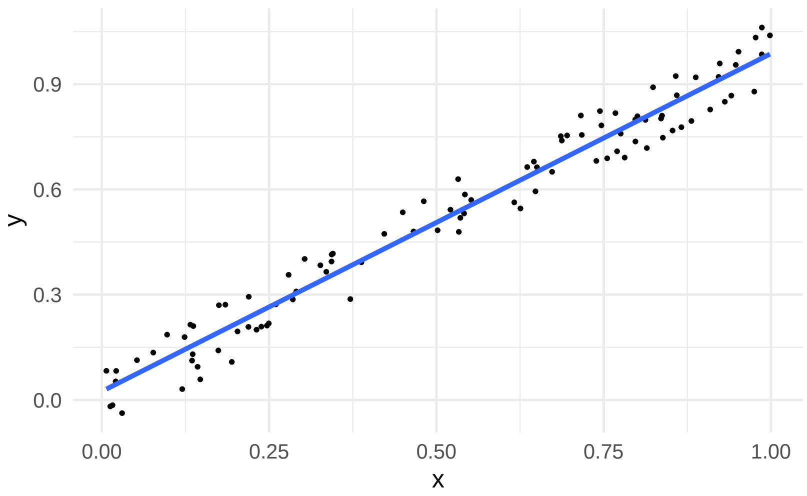
Examining the above scatter plot we notice that:

- When x is high, y is usualy also high.
- When x is low, y is usually also low.

In this case, we say that x and y are *positively linearly related.*

If we draw a line through the cloud of points, the line has a positive slope:

```
ggplot(df, aes(x, y)) +
  geom_point(size = 0.5) +
  geom_smooth(formula = y ~ x, method = "lm", se = FALSE) +
  theme_minimal()
```

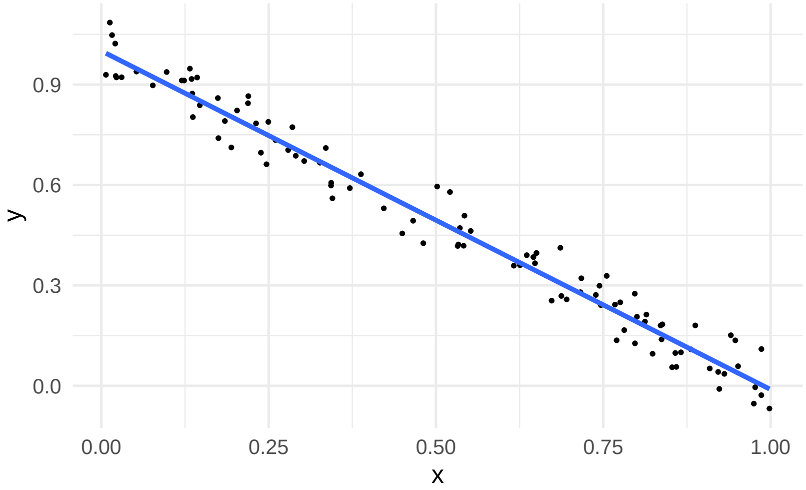


If x was advertising and y was sales, the business observes that it tends to sell more in markets where it advertises more. Therefore advertising may have a positive impact on sales (whether advertising has a causal impact on sales is something we will discuss later).

2.4 Negative Relationship

Suppose the scatter plot instead looked like this:

```
df$y <- 1 + -df$x + runif(100, -0.1, 0.1)
ggplot(df, aes(x, y)) +
  geom_point(size = 0.5) +
  geom_smooth(formula = y ~ x, method = 'lm', se = FALSE) +
  theme_minimal()
```



In this case:

- When x is high, y is usualy low.
- When x is low, y is usually high.

In this case, we say that x and y are *negatively linearly related*. The line through the cloud of points has a negative slope.

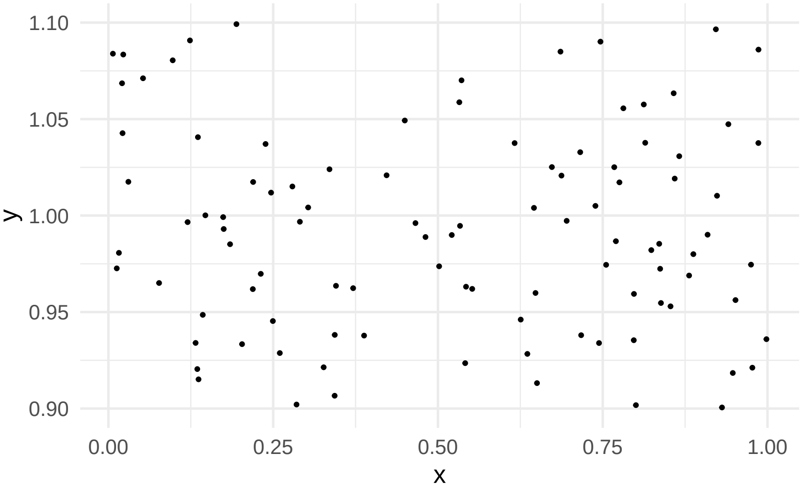
If x was advertising and y was sales, the business may conclude that advertising could be harmful to sales.

2.5 No Relationship

Two variables don't always have to have a positive or negative relationship. Sometimes there is no clear relationship between variables. In this case, we say that x and y are *unrelated*.

Here is an example scatter plot of two variables that are unrelated:

```
set.seed(231)
df$y <- 1 + runif(100, -0.1, 0.1)
ggplot(df, aes(x, y)) +
  geom_point(size = 0.5) +
  scale_y_continuous(limits = c(0.9, 1.1)) +
  theme_minimal()
```



If we were to draw a line through the cloud of points it would be (almost) flat.

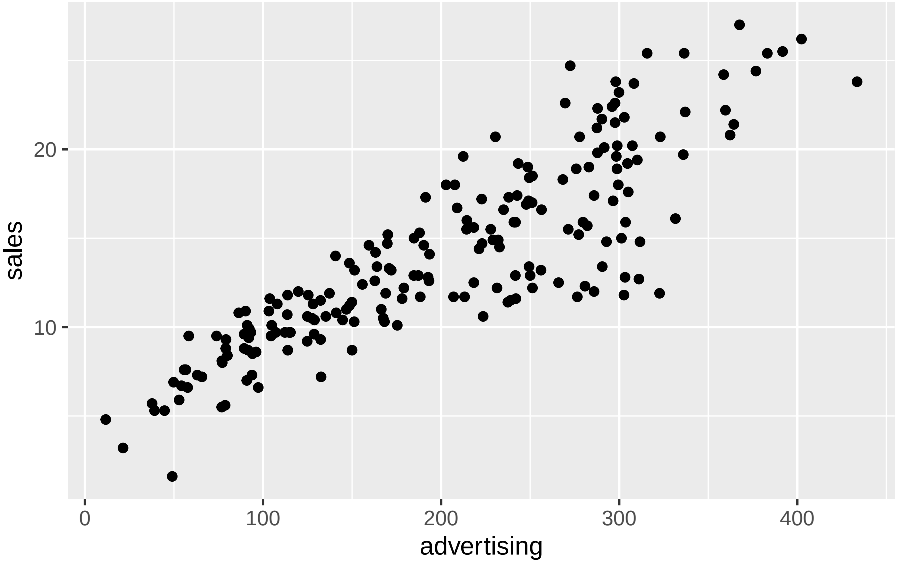
2.6 R Example

We will now learn how to make a scatter plot with a dataset using R. For this we will use this dataset which was downloaded from kaggle.com, which is a website with many datasets used by data scientists. The dataset contains the advertising expenditure across TV, radio and newspapers (measured in thousands of dollars) and sales revenue (measured in millions of dollars) for a company in different media markets.

Following these steps:

- Create a folder on your computer that you will use for datasets and R Scripts for this course (if you don't have one already).
- Download the dataset here.
- Put the `advertising-sales.csv` file you downloaded into your folder for this course.
- Open RStudio and create a new "Project". Select "Use Existing Directory" and navigate to your folder for this course.
- Then create an R Script in RStudio and paste in and run the following code:

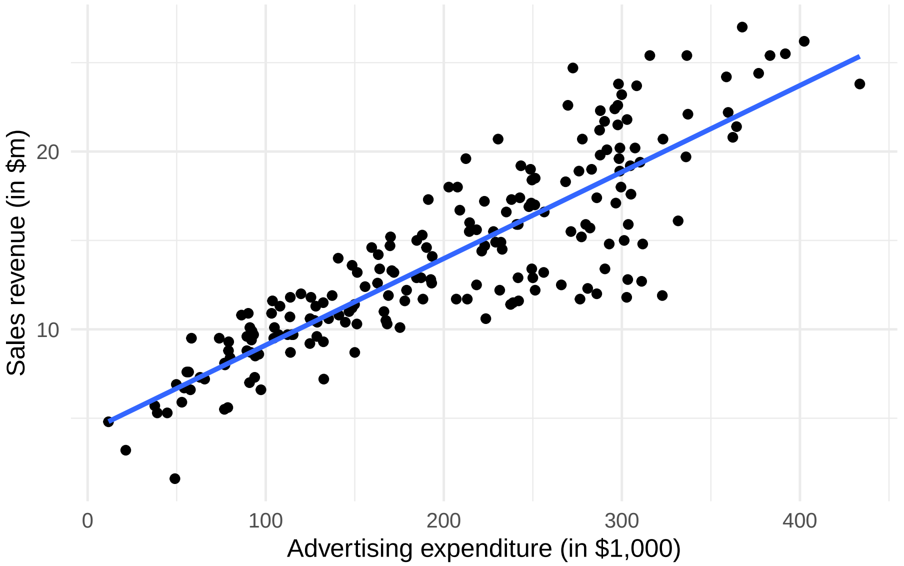
```
library(ggplot2)
df <- read.csv("advertising-sales.csv")
ggplot(df, aes(advertising, sales)) + geom_point()
```



We can see that advertising and sales are positively related.

We can also customize the plot, changing the axis labels and adding a line through the points:

```
ggplot(df, aes(advertising, sales)) +
  geom_point() +
  geom_smooth(formula = y ~ x, method = "lm", se = FALSE) +
  xlab("Advertising expenditure (in $1,000)") +
  ylab("Sales revenue (in $m)") +
  theme_minimal()
```

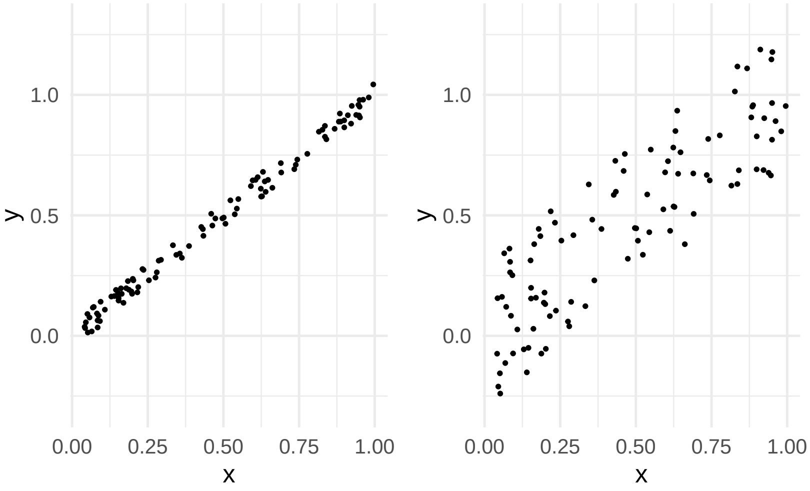


The command creating the blue line through the points is `geom_smooth(formula = y ~ x, method = "lm", se = FALSE)`. We will learn more about this command when we study the simple linear regression model.

2.7 Relationship Strength

Sometimes the relationship between x and y is *stronger* than with other pairs of variables. For example, the relationship between x and y is stronger in the left figure compared to the right figure:

```
library(gridExtra)
df <- data.frame(x = runif(100, 0, 1))
df$y1 <- df$x + runif(100, -0.3, 0.3)
df$y2 <- df$x + runif(100, -0.05, 0.05)
g1 <- ggplot(df, aes(x, y2)) + geom_point(size = 0.5) +
theme_minimal() + scale_y_continuous(limits = c(-0.3, 1.3)) + ylab('y')
g2 <- ggplot(df, aes(x, y1)) + geom_point(size = 0.5) +
theme_minimal() + scale_y_continuous(limits = c(-0.3, 1.3)) + ylab('y')
grid.arrange(g1, g2, nrow = 1)
```



What we would like to do is to be able to *measure* the strength of the linear relationship between x and y. That is what we will do in the next chapter.

Chapter 3

Covariance

In this chapter we will learn about one way to quantify the strength of a linear relationship: the *covariance*.

Because the covariance is a very important measure we repeatedly use throughout this course, we will first motivate where the formula comes from, and then show how to calculate it in R.

3.1 Notation

We first define some notation. We observe a sample with n observations for the variables x and y:

$$((x_1, y_1), (x_2, y_2), \dots, (x_n, y_n))$$

We will often refer to one specific observation as (x_i, y_i). The sample means of x and y are given by:

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad \text{and} \qquad \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

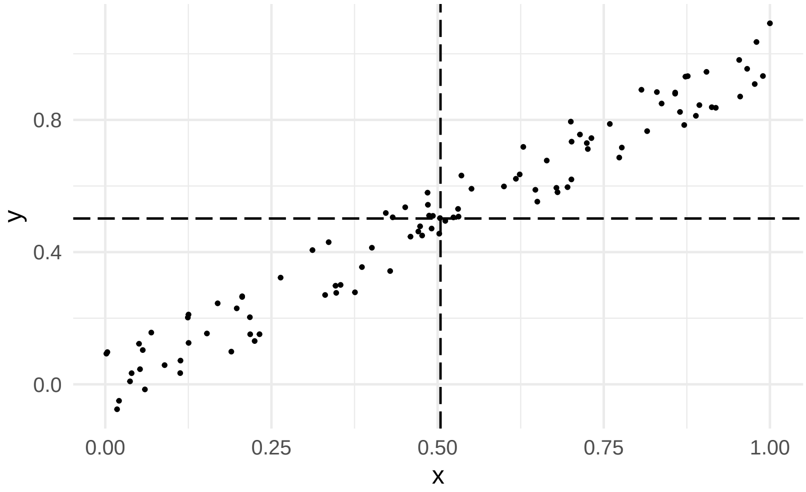
The $\sum_{i=1}^{n}$ term is mathematical notation for “take the sum over i from 1 to n”. It is defined as:

$$\sum_{i=1}^{n} x_i = x_1 + x_2 + \dots + x_n$$

3.2 Quadrants

Consider again two variables x and y that have a positive linear relationship. I plot them below, adding a vertical line at $\bar{x}$ and a horizontal line at $\bar{y}$:

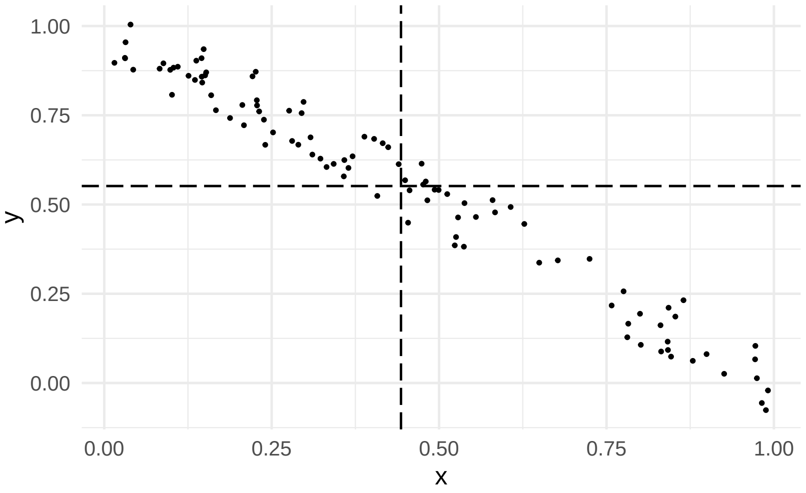
```
library(ggplot2)
set.seed(345345)
df <- data.frame(x = runif(100, 0, 1))
df$y <- df$x + runif(100, -0.1, 0.1)
ggplot(df, aes(x, y)) + geom_point(size = 0.5) +
  theme_minimal() +
  geom_vline(xintercept = mean(df$x), color = 'black', linetype = 'longdash') +
  geom_hline(yintercept = mean(df$y), color = 'black', linetype = 'longdash')
```



What we can see is that most of the data points are in the top-right and bottom-left quadrants. Only a few points are in the top-left or bottom-right quadrants.

Now consider two variables that have a negative linear relationship:

```
df <- data.frame(x = runif(100, 0, 1))
df$y <- 1 - df$x + runif(100, -0.1, 0.1)
ggplot(df, aes(x, y)) + geom_point(size = 0.5) +
  theme_minimal() +
  geom_vline(xintercept = mean(df$x), color = 'black', linetype = 'longdash') +
  geom_hline(yintercept = mean(df$y), color = 'black', linetype = 'longdash')
```



What we can see here is that most of the data points are in the top-left and bottom-right quadrants. Only a few points are in the top-right or bottom-left quadrants.

From this we can conclude is that:

- If most of the data points are in top right-and bottom-left quadrants, we have a positive linear relationship.
- If most of the data points are in top left-and bottom-right quadrants, we have a negative linear relationship.

What we want to do with this is create a formula that captures how often we are in the top-right and bottom-left versus the top-left and bottom-right quadrants.

3.3 Towards a Formula for the Covariance: Intuition

- If an observation x_i is to the *right* of the line, then $x_i - \bar{x} > 0$.
- If an observation x_i is to the *left* of the line, then $x_i - \bar{x} < 0$.
- If an observation y_i is *above* the line, then $y_i - \bar{y} > 0$.
- If an observation y_i is *below* the line, then $y_i - \bar{y} < 0$.

Taken together, in each quadrant it holds that:

	Left	Right
Top	$(x_i - \bar{x})(y_i - \bar{y}) < 0$	$(x_i - \bar{x})(y_i - \bar{y}) > 0$
Bottom	$(x_i - \bar{x})(y_i - \bar{y}) > 0$	$(x_i - \bar{x})(y_i - \bar{y}) < 0$

We call $\left(x_i - \bar{x}\right)\left(y_i - \bar{y}\right)$ the product of x_i and y_i's deviation from their means.

If there is a positive relationship, then *most* points will be in the top-right and bottom-left quadrants, so $\left(x_i - \bar{x}\right)\left(y_i - \bar{y}\right)$ will be positive for most of the observations, but could be negative for some observations. But there will be more positive terms overall and so when we sum over all observations we get:

$$\sum_{i=1}^{n} \left(x_i - \bar{x}\right)\left(y_i - \bar{y}\right) > 0$$

If there is a negative relationship, then *most* points will be in the top-left and bottom-right quadrants. So $\left(x_i - \bar{x}\right)\left(y_i - \bar{y}\right)$ will be negative for most of the observations, but could be positive for some observations. But there will be more negative terms overall and so when we sum over all observations we get:

$$\sum_{i=1}^{n} \left(x_i - \bar{x}\right)\left(y_i - \bar{y}\right) < 0$$

Thus whether the sum $\sum_{i=1}^{n} \left(x_i - \bar{x}\right)\left(y_i - \bar{y}\right)$ is positive or negative can tell us if there is a positive or a negative relationship between x and y.

3.4 Covariance Formula

The formal definition of the covariance is as follows. For two random variables X and Y, the covariance $\sigma_{X,Y}$ is given by:

$$\sigma_{X,Y} = \mathbb{E}\left[\left(X - \mathbb{E}\left[X\right]\right)\left(Y - \mathbb{E}\left[Y\right]\right)\right]$$

where $\mathbb{E}\left[X\right]$ is the expected value of X. In words, the covariance between two random variables is the expectation of the product of each variable's deviation from their expected values.

With data $\left(\left(x_1, y_1\right), \dots, \left(x_n, y_n\right)\right)$, we can estimate $\operatorname{cov}\left(X, Y\right)$ using the sample covariance $s_{X,Y}$:

$$s_{X,Y} = \frac{1}{n-1} \sum_{i=1}^{n} \left(x_i - \bar{x}\right)\left(y_i - \bar{y}\right)$$

Notice that the sum $\sum_{i=1}^{n} \left(x_i - \bar{x}\right)\left(y_i - \bar{y}\right)$ in $s_{X,Y}$ is exactly the same as the one we saw above when analyzing the quadrants. So the covariance formula captures this idea that if the covariance is positive, then most of the points are in the top-right and bottom-left quadrants, and if the covariance is negative, then most of the points are in the top-left and bottom-right quadrants.

The only difference from above is that we divide by $n - 1$. We do this because we are trying to estimate $\sigma_{X,Y}$, which is the expected value of this product of deviations from the means. We divide by $n - 1$ instead of n because it gives less biased estimates of $\sigma_{X,Y}$.

3.5 Relationship to the Variance Formula

Let's compare the formula for the covariance with the variance formula you learned about in Statistics 1. The formal definition of the variance is as follows. For a random variable X, the variance σ_X^2 is given by:

$$\sigma_X^2 = \mathbb{E}\left[(X - \mathbb{E}[X])^2\right]$$

The formula for the sample variance is given by:

$$s_X^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

Imagine we tried to get the covariance between a variable X and itself. We replace X for Y in the covariance formula and we get:

$$\sigma_{X,X} = \mathbb{E}\left[(X - \mathbb{E}[X])(X - \mathbb{E}[X])\right] = \mathbb{E}\left[(X - \mathbb{E}[X])^2\right] = \sigma_X^2$$

which is the same as the variance. We see the same if we replace y_i and $\bar{y}$ with x_i and $\bar{x}$ in the sample covariance formula:

$$s_{X,X} = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(x_i - \bar{x}) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2 = s_X^2$$

So the covariance between a variable and itself is equal to the variance. The variance formula is a special case of the covariance formula when the two variables are the same.

3.6 Calculating the Covariance in R

We can calculate the covariance in R easily using the `cov()` function. We just give it two numeric vectors as arguments. Using our advertising and sales example:

```r
df <- read.csv("advertising-sales.csv")
cov(df$advertising, df$sales)
```

```
[1] 420.9673
```

We can see that the covariance is positive, indicating a positive linear relationship between advertising and sales. This is what we saw in the scatter plot.

For demonstrative purposes[1], let's try to calculate the covariance in R using the

[1] If we wanted to estimate a model $\mathbb{E}[Y_i|x_{i1}, x_{i2}] = \beta_0 + \beta_1(x_{i1} + x_{i2})$, i.e. a simple linear regression model with Y_i explained by the sum of x_{i1} and x_{i2} we can't just do `lm(y ~ x1 + x2, data = df)`. This is because this would actually estimate the model $\mathbb{E}[Y_i|x_{i1}, x_{i2}] = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2}$. To "inhibit" R from "interpreting" the `+` as adding a new variable we can use the `I()` function (the "inhibit interpretation" function). We would use it like this: `lm(y ~ I(x1 + x2), data = df)`.

formula above instead of the built-in `cov()` function.

```
n <- nrow(df)
x <- df$advertising
y <- df$sales
x_bar <- mean(x)
y_bar <- mean(y)
(1 / (n - 1)) * sum((x - x_bar) * (y - y_bar))
```

```
[1] 420.9673
```

We get the same answer.

3.7 Interpreting the Covariance

If the covariance is positive or negative, it can tell us if the relationship is positive or not. But the size of the number we get is difficult to interpret. The covariance formula also depends on the units of the individual variables. For example, if we are interested in the covariance between height and salary, it will matter if we measure height in inches or centimeters or salary in dollars or euros.

To see this, recall that we said that advertising was in thousands of euros, and sales in millions. We can convert both variables to have units in euros as follows:

```
df$advertising <- df$advertising * 1000  # convert from €1000 to €
df$sales <- df$sales * 1000000           # convert from €m to €
head(df)
```

```
  advertising    sales
1      337100 22100000
2      128900 10400000
3      132400  9300000
4      251300 18500000
5      250000 12900000
6      132600  7200000
```

If we get the covariance now, we see that the scale of the number is much much larger:

```
cov(df$advertising, df$sales)
```

```
[1] 420967275126
```

So the covariance is heavily dependent on the units of the variables, and is difficult to tell if a covariance is large or small. It's only able to easily tell us if the relationship is positive or negative.

What we will discuss in the next chapter is another measure of the association between two variables which doesn't depend on the units, and is much easier to interpret the strength of the relationship. This is the *correlation*.

Chapter 4

Correlation

In this chapter we will discuss the correlation, which is a measure of the association between two variables that is easy to interpret and does not depend on the units of the underlying variables.

4.1 Formula

The formula for the sample correlation is very similar to the covariance. The only difference is that we divide by the sample standard deviations of X and Y.

The sample correlation coefficient between X and Y is given by:

$$r_{X,Y} = \frac{s_{X,Y}}{s_X s_Y}$$

where $s_{X,Y}$ is the covariance between X and Y (discussed in Chapter 3) and s_X and s_Y are the sample standard deviations of X and Y (the square root of the variance, which was also discussed in Chapter 3).

Dividing the covariance by the product of the sample standard deviations brings two important benefits:

1. The correlation coefficient must always be between -1 and $+1$ (can be proven mathematically). This makes interpretation easier, as we will see below.
2. The correlation coefficient has no units. If we were to change the units of X, which would scale X proportionally up or down, it would affect $s_{X,Y}$ and s_X the same way and cancel in the formula. We will see an example of this below.

4.2 Interpretation

Similar to the covariance, if the correlation is positive, we say X and Y are positively linearly related. But we can also how close it is to 0 or 1 to quantify the strength of the relationship:

- If the correlation is high (such as 0.8), we can say "there is a strong positive linear relationship between X and Y".
- If the correlation is low (such as 0.2), we can say "there is a weak positive linear relationship between X and Y".

If the correlation is negative, we say X and Y are negatively linearly related. We can use how close it is to 0 or -1 to quantify the strength of the relationship:

- If the correlation is negative and large in magnitude (such as -0.8), we can say "there is a strong negative linear relationship between X and Y".
- If the correlation is negative but small in magnitude (such as -0.2), we can say "there is a weak negative linear relationship between X and Y".

If the correlation is zero or very close to zero, we say "X and Y are uncorrelated".

4.3 Perfect Linear Relationships

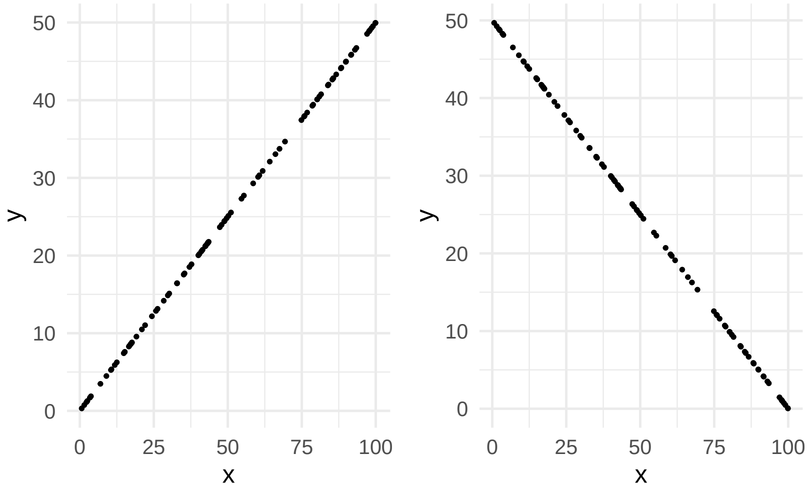
If the correlation is exactly 1, the points fall exactly along a straight upward-sloping line. This is called a *perfect positive linear relationship*. In this case, whenever x increases, then y always increases, and always in the same way.

If the correlation is exactly -1, the points fall exactly along a straight downward-sloping line. This is called a perfect negative linear relationship.

Here is what a scatter plot of two variables with a perfect positive linear relationship (left figure) and perfect negative linear relationship (right figure) would look like:

```r
library(ggplot2)
library(gridExtra)
df <- data.frame(x = runif(100, 0, 100))
df$y1 <- 0.5 * df$x
df$y2 <- 50 - 0.5 * df$x
g1 <- ggplot(df, aes(x, y1)) +
  geom_point(size = 0.5) +
  theme_minimal() +
  ylab("y")
g2 <- ggplot(df, aes(x, y2)) +
  geom_point(size = 0.5) +
  theme_minimal() +
  ylab("y")
```

```
grid.arrange(g1, g2, nrow = 1)
```

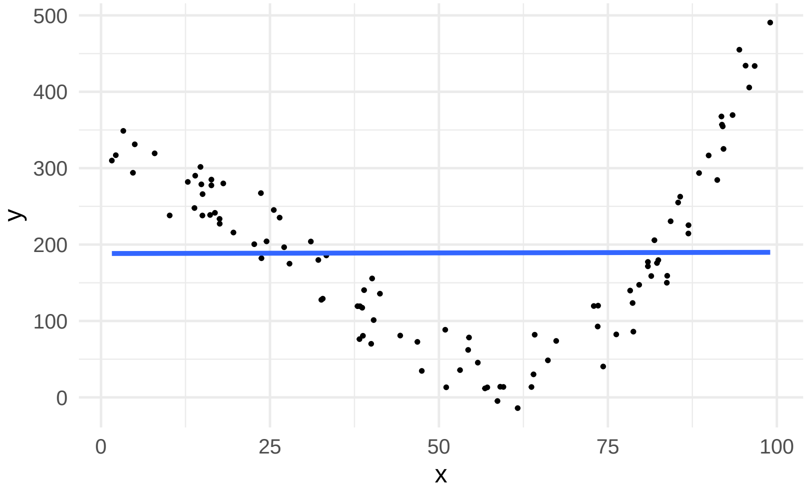


4.4 Non-Linear Relationships

Sometimes x and y may be strongly related, but in a non-linear way. Because the correlation formula only measures the strength of a *linear* relationship, you may get a correlation of close to 0 even if x and y are clearly related.

For example, suppose x and y had a U-shaped relationship, like this:

```
set.seed(2352342)
df <- data.frame(x = runif(100, 0, 100))
df$y <- 300 + df$x - 0.25 * df$x^2 + 0.00266 * df$x^3 + runif(100, -50, 50)
ggplot(df, aes(x, y)) + geom_point(size = 0.5) +
geom_smooth(formula = y ~ x, method = 'lm', se = FALSE) +
theme_minimal() + ylab('y')
```



The correlation coefficient for these data points is only 0.004, very close to zero. This is despite that there is clearly a tight relationship between x and y. Thus the correlation coefficient is only able to tell us about the strength of a *linear* relationship, and does not work for non-linear relationships.

4.5 Calculating the Correlation in R

We can calculate the correlation in R easily using the `cor()` function. Very similar to the `cov()` function, we just give it two numeric vectors as arguments. Using our advertising and sales example:

```r
df <- read.csv("advertising-sales.csv")
cor(df$advertising, df$sales)
```

```
[1] 0.8677123
```

The correlation is positive and close to 1. Thus there is a strong positive linear relationship between advertising and sales.

Let's confirm that the correlation is unaffected by the units of the underlying variables. We convert advertising and sales to euros again and recalculate the correlation:

```r
df <- read.csv("advertising-sales.csv")
df$advertising <- df$advertising * 1000  # convert from €1000 to €
df$sales <- df$sales * 1000000           # convert from €m to €
cor(df$advertising, df$sales)
```

```
[1] 0.8677123
```

We get the same number as before. So the correlation doesn't depend on the units.

Chapter 5

Spurious Relationships

5.1 Introduction

It's possible to measure a very strong correlation between two variables, but it doesn't necessarily mean there is a causal link between the two.

For example, with daily sales data for ice cream (X) and fans (Y) we might measure a very high correlation coefficient. Whenever ice cream sales are high, fan sales are high, and whenever ice cream sales are low, fan sales are also low.

But it would be a strange idea to think that more ice cream causes fan sales to increase, or more fans to cause ice cream sales to increase. A more reasonable explanation for this is that increases in the temperature causes both ice cream sales and fan sales to increase.

We call such a correlation a *spurious correlation.* This is when two variables are correlated but are not causally related. Often some other variable (call it Z) is causing both X and Y to move together. We call such a variable a confounding variable. In the ice cream sales and fan sales example, the temperature is the confounding variable.

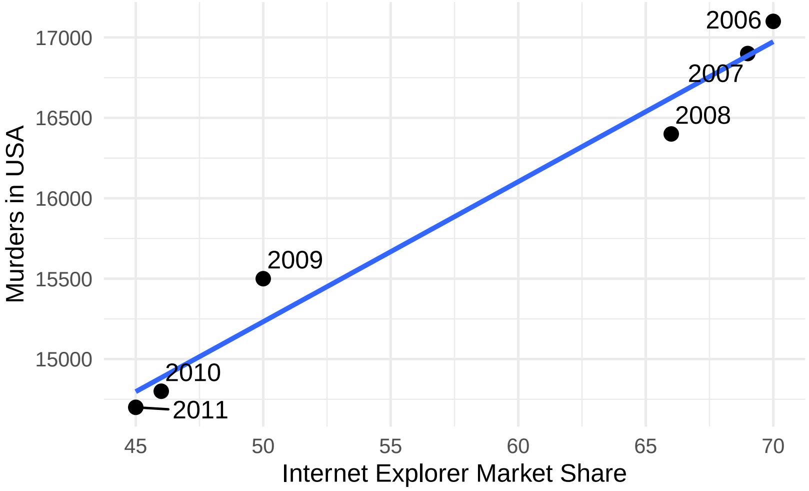
5.2 Examples

We'll now take a look at some examples of spurious correlations.

5.2.1 Internet Explorer and Homocides

The figure below plots the number of murders each year in the US against the annual market share of the web browser Internet Explorer. The correlation is very high and close to 1.

```
library(ggplot2)
library(ggrepel)
df <- data.frame(
  year = 2006:2011,
  x    = c(70, 69, 66, 50, 46, 45),
  y    = c(17100, 16900, 16400, 15500, 14800, 14700)
)
ggplot(df, aes(x, y)) +
  geom_point(size = 2.5) +
  geom_smooth(formula = y ~ x, method = 'lm', se = FALSE) +
  geom_text_repel(aes(label = year)) +
  theme_minimal() +
  xlab("Internet Explorer Market Share") +
  ylab("Murders in USA")
```



Does this mean that the use of Internet Explorer drove people to commit more murders? Although Internet Explorer was very frustrating to use, it is an unlikely explanation. A more reasonable explanation is that both variables saw a declining trend throughout the 2000s from other causes (such as the release of Mozilla Firefox and Google Chrome) and only appear to be correlated.

5.2.2 Chocolate Consumption and Cognitive Function

Another example from this study documented a strong correlation (0.791) between per capita annual chocolate consumption and the number of Noble laureates:

The relationship is surprisingly strong, with only Sweden being an outlier in

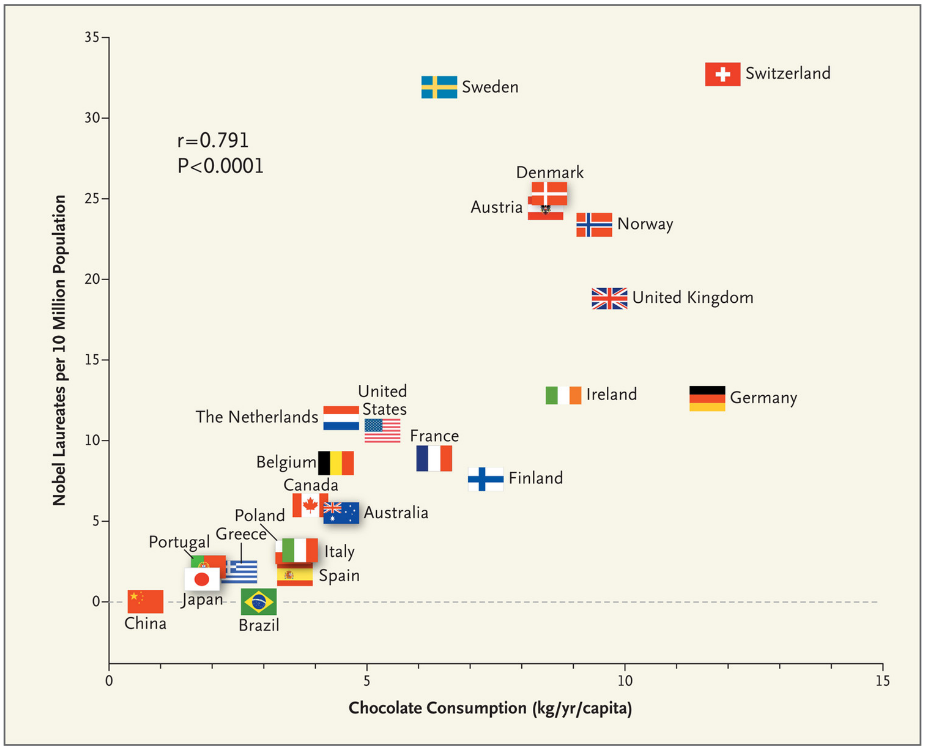


Figure 5.1: Source: Messerli, F.H., 2012. Chocolate consumption, cognitive function, and Nobel laureates. N Engl J Med, 367(16), pp.1562-1564.

having more Nobel laureates than would be predicted by its annual per capita chocolate consumption (interesting, considering Sweden is the country giving out the prize…).

Could it be the chocolate makes your brain function better, leading to more Nobel laureates? After all, we need energy (calories) to think, and chocolate has lots of that! Or is it that nations celebrate winning a Nobel prize by consuming inordinate amounts of chocolate? Both of these explanations are unlikely. What is more likely is that another variable, Z, causes both more chocolate consumption and more Nobel prizes. This is how wealthy and developed a country is, as these lead to both more investment in education and scientific laboratory equipment, and the consumption of more chocolate.

5.2.3 Storks and Babies

When young children ask where babies come from, parents sometimes tell their children that a stork delivered the baby (instead of trying to explain the details of the human reproductive system).

This study found a correlation of 0.62 between the number of stork breeding pairs in a country and the number of births from humans. Is this scientific evidence that storks actually do deliver babies?

A more natural explanation is that big countries (with lots of land) tend to have more people, and hence more births, and big countries also have more storks. Thus the confounder here could be land area.

5.3 Confounders More Generally

The above discussion was for the case where X and Y were causally unrelated. It is also possible that X has a causal impact on Y, but both X and Y are also affected by a confounder Z. In this case we also have to be careful interpreting the correlation between X and Y. In the presence of confounders, it is possible to measure a positive correlation between X and Y but the true causal impact of X on Y is negative.

One example of this is this study, which contributed to one of the authors receiving the Nobel memorial prize in Economics. The authors observe a positive correlation between the number of children in a classroom and how well they do on tests. If we interpret this as having larger classrooms helps children learn, the government might decide to employ fewer teachers and merge more classrooms.

However, there is a confounder here which is the socioeconomic status of students attending the school. Urban areas tend to have a higher socioeconomic status, and students with higher socioeconomic status usually do better on tests despite there being more students in the classroom.

Using a clever trick, the authors determined the true causal effect of class size on test scores and found it to have the opposite sign: more children in the classroom has a negative impact on test scores. Here is the idea behind their approach. There was a rule in Israel that said that you had to go to a particular school depending on where you lived. If there were only 40 students to be enrolled in a particular year, there would be only 1 classroom. But if there were 41 students they would split the students into 2 classrooms (one with 20 students, the other with 21). Because it is more or less random if there are 40 versus 41 students to enroll (as opposed to much bigger or smaller numbers which depend on if it is an urban area or not), if we compare the test scores in schools with exactly 40 students in a year (with big classrooms) and 41 students (with two small classrooms) we can get the causal effect of class size.

Therefore the government may make a totally wrong conclusion by only looking at the correlation.

Chapter 6

The Simple Linear Regression Model (SLR)

Using the correlation coefficient we learned how to measure the strength of the linear relationship between X and Y. We will now introduce the *Simple Linear Regression* model which will allow us to do the following:

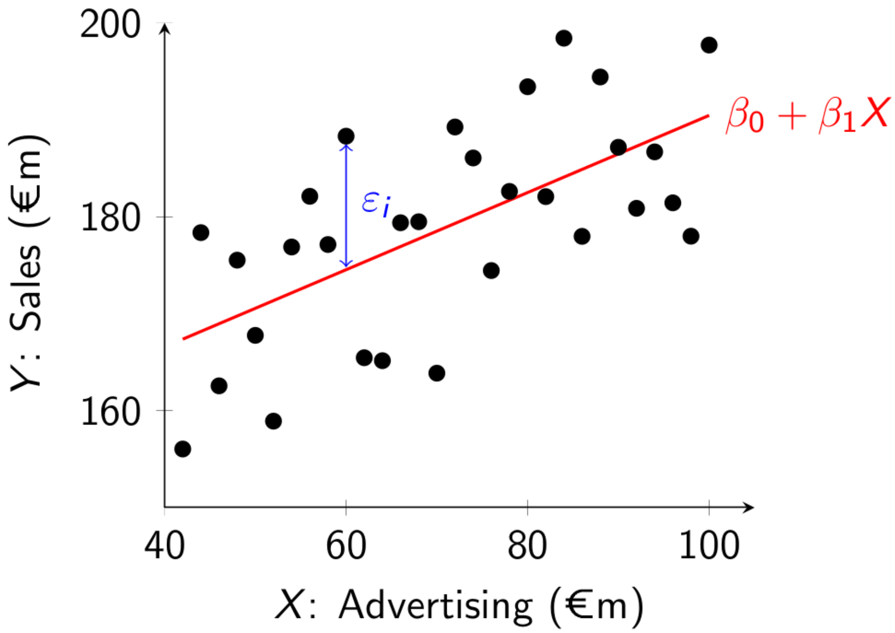
- We will measure what percentage of the variation in Y is explained by the variation in X.
- We will estimate how much Y increases/decreases on average if X increases by 1 unit.
- We will quantify how precise these estimates are.
- We will learn how to predict Y for any value of X , and quantify how precise those predictions are.

6.1 The Model

We model Y as a *linear function* of X. What do we mean by this? It means we assume that Y is linearly related to X. We say that the values of Y are generated according to the line $\beta_0 + \beta_1 X$, where β_0 is the intercept and β_1 is the slope. The intercept β_0 is what Y is when $X = 0$ and the slope β_1 is how much Y increases when X increases by 1 unit. However, for each observation in the data i we won't have that $Y_i = \beta_0 + \beta_1 X_i$ exactly. In fact, the values Y_i will rarely be exactly on the line. Most values will be above it or below it. So we add an error term ε_i to the equation to account for this discrepancy. The model for Y_i is then:

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$$

Graphically, the regression line is given by the red line in the following figure:



The dots represent different data points (X_i, Y_i) from the population, where X is on the horizontal axis and Y is on the vertical axis. In the figure we are trying to model sales as a linear function of advertising. The red regression line is the line that "best fits" the population cloud of points. Because the regression line doesn't match the points exactly, we add an error term ε_i which is the vertical difference between the actual value of Y_i and the corresponding point on the regression line at X_i. The error is positive for points above the line and negative below it.

6.2 Estimation

How do we find the regression line that "best fits" this could of points? That is, how do we find the best β_0 and β_1? Intuitively we want a line that makes the errors as close to zero as possible. Because the errors can be positive or negative, we find the line that makes the sum of *squared* errors the smallest. Taking the square turns the negative errors to positive ones, and also makes the line try to avoid big errors (because when we square them, they get even bigger!).

We won't cover the mathematics here, but it can be shown that in the population, the regression coefficients that minimize the sum of squared errors are given by:

$$\beta_1 = \frac{\sigma_{X,Y}}{\sigma_X^2} \qquad \text{and} \qquad \beta_0 = \mu_Y - \beta_1 \mu_X$$

where μ_X and μ_Y are the population means of X and Y.

With a sample dataset:

$$((x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n))$$

the sample regression coefficients, b_0 and b_1, can be calculated with the sample analogs of this:

$$b_1 = \frac{s_{X,Y}}{s_X^2} \qquad \text{and} \qquad b_0 = \bar{y} - b_1\bar{x}$$

6.3 Predicted Values and Residuals

For any value x_i, the predicted value for y_i is:

$$\hat{y}_i = b_0 + b_1 x_i$$

where the hat ($\hat{}$) denotes that it is a predicted value. This is the value of the Y variable predicted by the model. The difference between the actual value of Y and the one predicted by the model given the corresponding value of the X variable is the prediction error, $y_i - \hat{y}_i$.

We call this prediction error the residual, and denote it by e_i:

$$e_i = y_i - \hat{y}_i$$

Graphically we can represent this in a similar way to above:

6.4 Interpreting Coefficient Estimates

In the next chapter we will learn how to estimate this model in R with real data. But for now, let's consider a simple example and discuss how to interpret the estimates of the intercept, b_0, and the slope, b_1.

Suppose you have a sample of data on advertising (x_i) and sales (y_i), both measured in millions of euros. Suppose you estimate $b_0 = 150$ and $b_1 = 0.4$. The sample regression line is then:

$$150 + 0.4x$$

The intercept gives an estimate of the expected value of Y conditional on $x = 0$. We denote this by $\mathbb{E}[Y_i|x_i = 0]$. This means, it is an estimate of the amount of sales the firm will generate (in millions) if it has zero advertising. Thus if advertising is zero, then the model predicts sales to be €150m. However, if there are no observations for advertising near zero, this prediction is unreliable.

The slope gives an estimate of the expected change in Y when x increases by 1 unit. It is an estimate of

$$\mathbb{E}[Y_i|x_i + 1] - \mathbb{E}[Y_i|x_i]$$

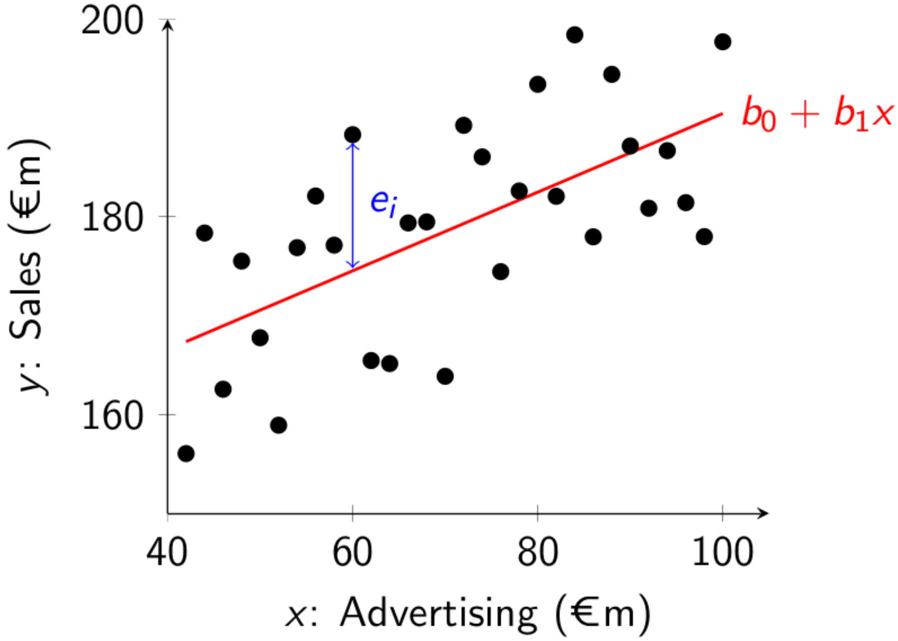


Figure 6.1: Sample Regression Line

If the X variable increases by one unit, the model predicts that the Y variable will on average increase by b_1 units. In this example, if advertising increases by €1m then on average sales increases by €0.4m. We write millions because the units for both variables are in millions.

6.5 Regression Slope Versus Correlation

One thing worth pointing out here is that the regression slope is **not** the same thing as the correlation coefficient. Let's compare the formulas for both of them:

$$b_1 = \frac{s_{X,Y}}{s_X^2}$$
$$r_{X,Y} = \frac{s_{X,Y}}{s_X s_Y}$$

The numerators for both are the same, but the denominators are different. So in general they will be different. The interpretation of the values also differ.

There is one special case when both will have the same value. This is when $s_X = s_Y$. If we *standardize* both the X and Y variable (subtract the mean and divide by the standard deviation), then the sample correlation coefficient and sample regression slope will have the same value. This is because after standardizing the resulting variables both have a standard deviation of 1.

6.6 Why Do We Call it Regression?

The word *regression* comes from the 1886 journal article *Regression towards mediocrity in hereditary stature* by Sir Francis Galton. After collecting data on the heights of many people and their children, he observed that while tall parents on average had tall children (and short parents on average had short children), on average the children's heights were "less extreme" and closer to the mean height of the population than their parents. Thus people with extreme heights (tall or short) did not pass on their traits completely to their children. This phenomenon is called *regression to the mean.*

Chapter 7

SLR Estimation

In this chapter we will learn how to estimate a simple linear regression in R using data. We use the `lm()` function to do this, where LM stands for Linear Model.

We will show 2 examples:

1. The advertising and sales example introduced in Chapter 2.
2. Data on Dutch exports and GDP over time.

7.1 Advertising and Sales Example

To estimate a linear regression with `y` as the dependent variable and `x` as the independent variable (with both variables contained in a dataset `df`), we use the command `lm(y ~ x, data = df)`. Let's try this out with the advertising and sales data:

```r
df <- read.csv("advertising-sales.csv")
lm(sales ~ advertising, data = df)
```

```

Call:
lm(formula = sales ~ advertising, data = df)

Coefficients:
(Intercept)  advertising
    4.24303      0.04869
```

The output shows us the command that was provided (under `Call:`) and the sample regression coefficients, $b_0 = 4.24303$ and $b_1 = 0.04869$. The sample regression line is $4.24303 + 0.04869x$.

Let's interpret these estimates. First let's remind ourselves of what units the variables are in:

- Sales is measured in millions of euros.
- Advertising is measured in thousands of euros.

For the intercept, b_0, recall that it gives an estimate of $\mathbb{E}\left[Y_i|x_i=0\right]$, the expected value of the Y variable when the X variable equals zero. In this example, when the firm does zero advertising, the model predicts that the firm's sales will be 4.24303 units. Because the units of sales are in millions, this means the expected sales will be €4.24303m.

To see if this is a reliable estimate, we check if we have observations x_i at or near zero:

```
summary(df$advertising)
```

```
Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
11.7   123.5   207.3   200.9   281.1   433.6
```

There are no observations at zero. The smallest value is 11.7 (€11,700), so this estimate is potentially unreliable.

For the slope, b_1, recall that it gives an estimate of:

$$\mathbb{E}\left[Y_i|x_i+1\right]-\mathbb{E}\left[Y_i|x_i\right]$$

which is the expected change in units of the Y variable when the X variable increases by 1 unit. Increasing the X by 1 unit corresponds to an increasing in advertising by €1,000. So when advertising increases by €1,000, sales on average increases by 0.04869 × €1,000,000 = €48,690.

7.2 Netherlands Exports and GDP

This advertising-sales dataset is an example of *cross-sectional* data. Cross-sectional data involve observations from different individuals or firms surveyed at the same point in time. We will now consider an example with *time-series* data. Time-series data involve observations from the same individual or firm at different points in time.

The example will we consider uses the dataset nl-exports-gdp.csv which contains two variables measured over 1969-2023:

1. Netherlands GDP (measured in billions of USD).
2. Netherlands total exports of goods and services (measured in billions of USD).

We know from how GDP is calculated that if exports increase by $1bn and nothing else changes, then GDP should also increase by $1bn. Let's check if this is true in the data by estimating the regression model:

$$GDP_i = \beta_0 + \beta_1 Exports_i + \varepsilon_i$$

```r
df <- read.csv("nl-exports-gdp.csv")
lm(gdp ~ exports, data = df)
```

```

Call:
lm(formula = gdp ~ exports, data = df)

Coefficients:
(Intercept)      exports
   287.6114       0.8224
```

The intercept $b_0 = 287.6114$ gives an estimate of the value of GDP (in billions) when exports are zero. So the model predicts that Dutch GDP would be \$287.61bn if it exported zero goods.

Having zero exports is a very strange concept for an open economy like the Netherlands. Let's see if any observations of the X variable are at or near zero:

```r
summary(df$exports)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  62.98  122.05  257.93  332.33  505.88  786.90
```

The smallest value is \$62.98bn, very far from zero. Therefore we should not trust the estimate of the intercept.

The slope, $b_1 = 0.8224$ tells us that if exports increase by 1 unit (\$1bn), on average GDP increases by 0.8224 units (\$822.4m). This is somewhat less than what we expect to see. The reason it is a bit less is because other things are changing at the same time that also affect exports and GDP.

Chapter 8

SLR Model Assumptions

We are now interested in performing *inference* for our model. By inference we mean *inferring* the properties of the population regression model generating our sample data. With model inference we can answer questions like:

- How precise are our estimates b_0 and b_1?
- What are the confidence intervals around b_0 and b_1?
- How can we perform hypothesis tests on b_0 and b_1?

In order to perform model inference, we need to make some assumptions about our model. There are 6 assumptions in total, which we will discuss in the following sections.

8.1 Assumption 1: Linear in Parameters

> Assumption 1: Linear in Parameters
>
> In the population model, the dependent variable Y_i is related to the independent variable X_i and the error ε_i according to:
>
> $$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$$

This assumption means that the process that generates the data in our sample follows this model. That is, Y_i has a linear relationship with X_i and the values Y_i are generated according to the line $\beta_0 + \beta_1 X_i$ plus an error term ε_i.

An example of something that would violate this is if the true population model was something non-linear like:

$$Y_i = \exp(\beta_0) X_i^{\beta_1} \exp(\varepsilon_i)$$

with $X_i > 0$ for all i. If this were the true model, it would not be the end of the world. We could take the natural log of both sides to get:

$$\ln\left(Y_i\right) = \beta_0 + \beta_1 \ln\left(X_i\right) + \varepsilon_i$$

This transformed model satisfies assumption 1. So if we transform our data into logs, we can still use the simple linear regression model.

8.2 Assumption 2: Random Sampling

Assumption 2: Random Sampling

We have a random sample of size n, $\left(\left(x_1, y_1\right), \ldots, \left(x_n, y_n\right)\right)$ following the population model in Assumption 1.

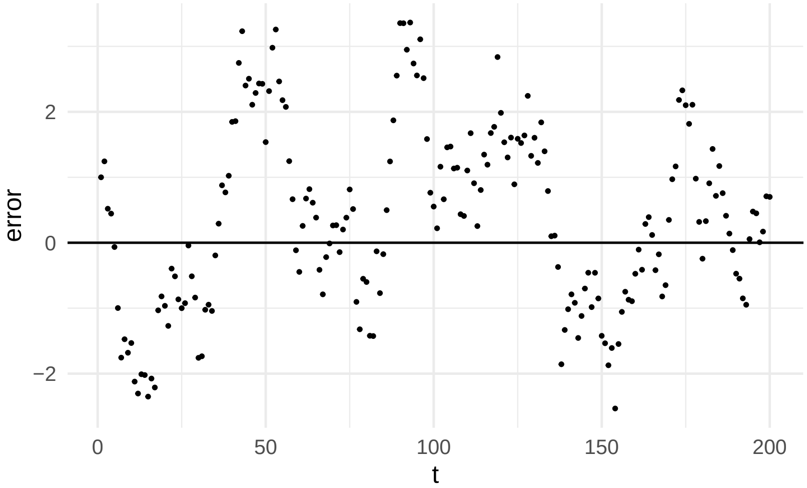
This assumption means that the sample of data we observe were generated according to the model $Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$. The values of y_i that we observe are related to the unknown population parameters, observed x_i and the unobserved error ε_i according to $\beta_0 + \beta_1 x_i + \varepsilon_i$, where ε_i is independent across observation i.

A crucial part of this assumption is the independence of the error terms across observations. For that reason this assumption is also called the independence assumption.

With *cross-section data*, there could be dependence in ε_i between people in the same household/town/industry. With *time-series data*, there could be dependence in ε_i in subsequent time periods.

Violations of this assumption are much more common with time series data. An example of this assumption being violated can be seen in the figure below, which plots the errors against time:

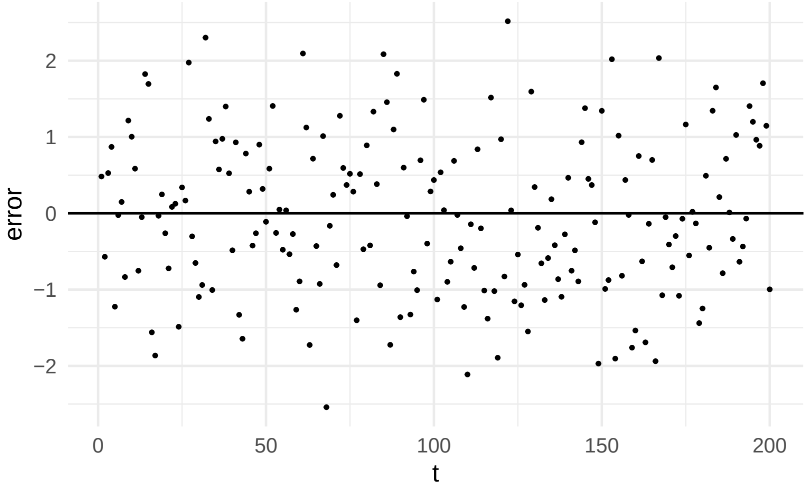
```r
set.seed(5727)
nT <- 200  # number of time periods
df <- data.frame(
  t = 1:nT,
  y = 0,
  x = runif(nT, 10, 12),
  e = 0
)
df$e[1] <- 1
df$y[1] <- 1
# Loop over time periods with first-order autocorrelation in the error:
for (i in 2:nT) {
  df$e[i] <- rnorm(1, 0, 0.5) + 0.95 * df$e[i - 1]
  df$y[i] <- 0.0001 * df$x[i] + df$e[i]
}
library(ggplot2)
ggplot(df, aes(t, y)) +
  geom_point(size = 0.5) +
  geom_abline(intercept = 0, slope = 0) +
  theme_minimal() +
  ylab("error")
```



We observe a clear pattern in the errors: if the error is positive in one time period, it's very likely to be positive in the following time period. If the error is negative in one time period, it's also very likely to be negative in the following time period. If the errors were independent, the value of the error in any given time period should not depend on what the value of the error was in the previous period.

Let's compare what the errors would look like over time if they were independent:

```
set.seed(345346)
nT <- 200
df <- data.frame(
  t = 1:nT,
  e = rnorm(nT)
)
ggplot(df, aes(t, e)) +
  geom_point(size = 0.5) +
  geom_abline(intercept = 0, slope = 0) +
  theme_minimal() +
  ylab("error")
```



Here we just see a random cloud of points. The error in any time period does not appear in any way related to the value of the error in the previous time period.

Later in this course we will learn how to formally test for correlation in the residuals over time. But now, let's learn how to make a plot of the residuals in R in order to visually inspect this model assumption. I would like to stress that this approach only works with time-series data. With cross-sectional data we cannot plot the residuals over time, because all subjects in cross-sectional data are surveyed at the same point in time.

As an example we return to the Netherlands GDP and exports data from Chapter 7.

The first thing we need to do is read in the data and estimate the regression model. This is the same as in Chapter 7. When we estimate the regression model, we will assign it to an object in our environment so we can access the residuals from the model. Let's assign the model to `m` ("M" for model):

```
df <- read.csv("nl-exports-gdp.csv")
m <- lm(gdp ~ exports, data = df)
```

Looking at our environment we can see that `m` is a list. If we click on it in RStudio we can see all the different things stored in this list, such as the coefficients, the residuals and the fitted values. There are 12 objects in total, but we will only use some of these in this course.

We can also list all the things stored in `m` by using the `ls()` command:

```
ls(m)
```

```
 [1] "assign"        "call"          "coefficients"  "df.residual"  
 [5] "effects"       "fitted.values" "model"         "qr"           
 [9] "rank"          "residuals"     "terms"         "xlevels"      
```

Recall that to access objects in a `list` we also use the dollar symbol (the *extraction operator*), just like with a `data.frame`. So to access the residuals, we can use `m$residuals`:

```
m$residuals
```

```
         1          2          3          4          5          6          7 
-76.979824 -66.766015 -60.884810 -56.927320 -48.719213 -40.048630 -37.653082 
         8          9         10         11         12         13         14 
-30.219716 -20.420951 -14.494551 -14.067661 -11.543704 -17.244818 -20.792229 
        15         16         17         18         19         20         21 
-15.474479 -11.867085  -6.794114   1.873095   5.313227   8.821736  16.430221 
        22         23         24         25         26         27         28 
 27.048551  28.452513  31.860574  31.402768  31.051349  29.617104  39.398917 
        29         30         31         32         33         34         35 
 42.211326  52.859591  60.121738  52.430464  62.941840  62.376269  57.683258 
        36         37         38         39         40         41         42 
 44.828888  38.944480  36.803112  42.901967  52.338672  60.545934  33.388814 
        43         44         45         46         47         48         49 
 23.338514   1.345738 -10.829735 -21.366546 -42.487514 -34.390817 -46.009436 
        50         51         52         53         54         55 
-51.404011 -46.927667 -53.610900 -49.250801 -40.059088 -29.095947 
```

This gives us the value of the residuals for all 55 observations in our data. Let's assign these residuals to our dataframe `df` so we can plot them:

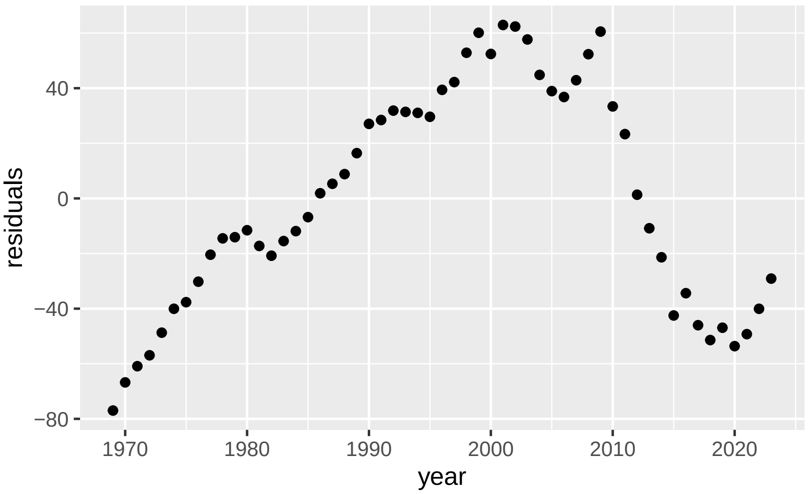
```
df$residuals <- m$residuals
head(df)
```

```
  year      gdp  exports residuals
1 1969 262.4266 62.97751 -76.97982
2 1970 278.5400 70.15091 -66.76601
3 1971 290.5646 77.62057 -60.88481
4 1972 300.8328 85.29375 -56.92732
5 1973 317.2108 95.22753 -48.71921
```

```
6 1974 328.1187 97.94799 -40.04863
```

We then plot the residuals over time. We use the `year` variable as the time variable:

```
ggplot(df, aes(year, residuals)) +
  geom_point()
```



We can see that the residuals in a period clearly depend on the value in the previous period. Therefore the residuals are not independent and violate assumption 2! In Chapter 27 we will learn how to formally test for this violation, and how to correct for it.

8.3 Assumption 3: Sample Variation in the Explanatory Variable

Assumption 3: Sample Variation in the Explanatory Variable

The sample outcomes $(x_1, \ldots, x_n)$ are not all the same value.

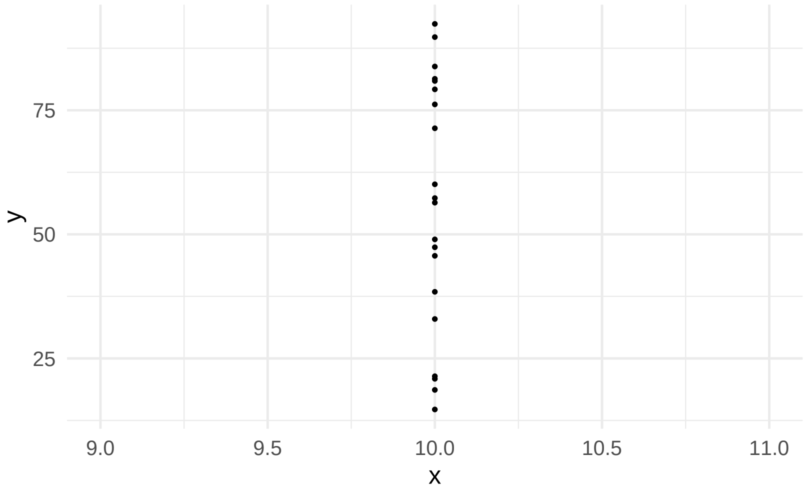
A simple explanation for this assumption is that we need it to avoid dividing by zero when calculating the sample regression coefficients. Recall the formula for the slope coefficient:

$$b_1 = \frac{s_{X,Y}}{s_X^2}$$

If all the values $(x_1, \ldots, x_n)$ in the sample were the same value, then the sample variance of X would be zero, i.e. $s_X^2 = 0$. If $s_X^2 = 0$, we would be dividing by zero in the formula for the sample regression slope.

In the example plot below, all the values of x are equal to 10. The sample variance is zero and we cannot estimate the regression slope.

```
set.seed(3453463)
df <- data.frame(x = rep(10, 20))
df$y <- runif(20, 0, 100)
ggplot(df, aes(x, y)) +
  geom_point(size = 0.5) +
  scale_x_continuous(limits = c(9, 11)) +
  theme_minimal()
```



We can easily check whether this assumption holds with our data in R by calculating the standard deviation of our x variable. If the standard deviation is zero, all values are the same and the assumption is violated. If the standard deviation is positive, there are at least some different values and the assumption holds.

Let's check it in the advertising and sales data:

```
df <- read.csv("advertising-sales.csv")
sd(df$advertising)
```

```
[1] 92.98518
```

This is positive, so it holds.

Although we only need just one value to be different to be able to estimate the regression model, more variation in the x variable will be better for our model.

8.4 Assumption 4: Zero Conditional Mean

Assumption 4: Zero Conditional Mean

The error ε_i has an expected value of zero given any value of the explanatory variable, i.e. $\mathbb{E}\left[\varepsilon_i | X_i\right] = 0$ for all X_i.

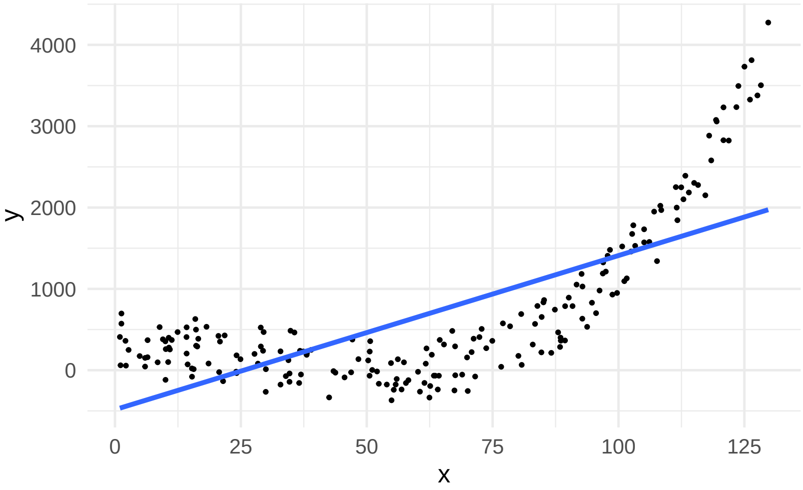
An implication of this is that the error term is uncorrelated with the explanatory variable.

Let's consider some examples of when this assumption would be violated. The first is a model trying to explain ice cream sales (Y) with fan sales (X). Ice cream sales are influenced by temperature. Because temperature is not included in the model (is not the X variable), temperature is included in the error, ε. But temperature is also correlated with fan sales, so ε is correlated with X. Therefore $\mathbb{E}\left[\varepsilon_i | X_i\right] \neq 0$, a violation of the zero conditional mean assumption. This can bias the estimation of β_1. The true β_1 should equal zero: if fan sales increase and nothing else changes (i.e. temperature stays the same), then there should be no increase in ice cream sales. However, if we were to estimate this model with data we would estimate $b_1 > 0$ as we observe a (spurious) correlation between ice cream sales and fan sales.

Another example is the test scores (Y) and classroom size (X) example that we saw in Chapter 5. Test scores are influenced by socioeconomic status, which is higher in urban areas. So the degree of urbanization is included in ε. But urban areas also have classrooms with more students, so ε is correlated with X. The true β_1 should be negative (smaller classrooms improve test scores) but we would estimate $b_1 > 0$. Estimation is biased again!

Non-linearities in the relationship between X and Y can also violate the zero conditional mean assumption $\mathbb{E}\left[\varepsilon_i | X_i\right] = 0$. Consider the following plot:

```r
set.seed(53653)
df <- data.frame(x = runif(200, 0, 130))
df$y <- 300 + df$x - 0.35 * df$x^2 + 0.0043 * df$x^3 + runif(100, -400, 400)
ggplot(df, aes(x, y)) + geom_point(size = 0.5) +
  geom_smooth(formula = y ~ x, method = "lm", se = FALSE) +
  scale_x_continuous(breaks = c(0, 25, 50, 75, 100, 125)) +
  theme_minimal() +
  ylab("y")
```



At $x = 75$, the average value of the error term is negative, whereas at $x = 125$, the average value of the error term is positive. Under the zero conditional mean assumption, the average value of the error should be zero at all values of X, so this would also be a violation of this assumption.

8.5 Assumption 5: Homoskedasticity

Assumption 5: Homoskedasticity

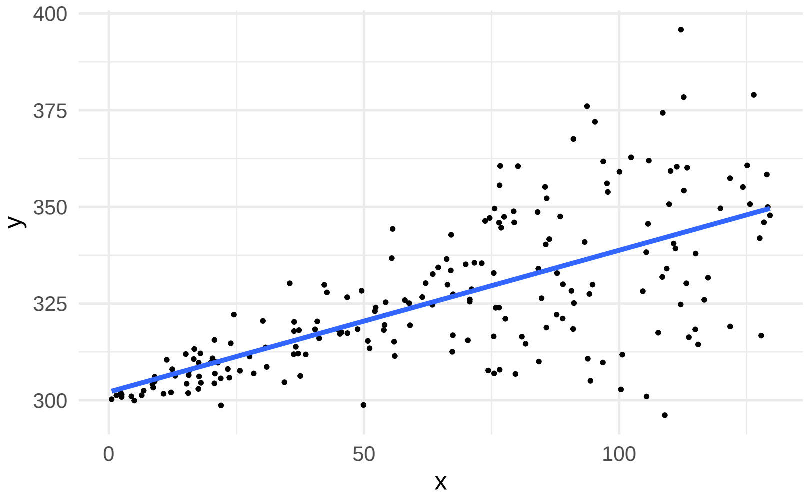
The error ε_i has the same variance given any value of the explanatory variable. In other words:

$$\text{Var}\left(\varepsilon_i|x_i\right) = \sigma_\varepsilon^2$$

Homoskedasticity means that the variance of the errors is the same for small values of x and large values of x. "Skedasticity" comes from the Ancient Greek word (skedánnymi) which means to scatter or disperse. So homoskedasticity litterly means "same dispersion". A violation of homoskedasticity is called *heteroskedasticity*, which means "different dispersion".

Let's take a look at a scatter plot of data that violate the homoskedasticity assumption:

```
set.seed(434634)
df <- data.frame(x = runif(200, 0, 130))
df$y <- 300 + 0.4 * df$x + rnorm(200, 0, 200) * 0.001 * df$x
ggplot(df, aes(x, y)) +
  geom_point(size = 0.5) +
  geom_smooth(formula = y ~ x, method = "lm", se = FALSE) +
  theme_minimal()
```



The variance of the residuals is small at low x and large at high x.

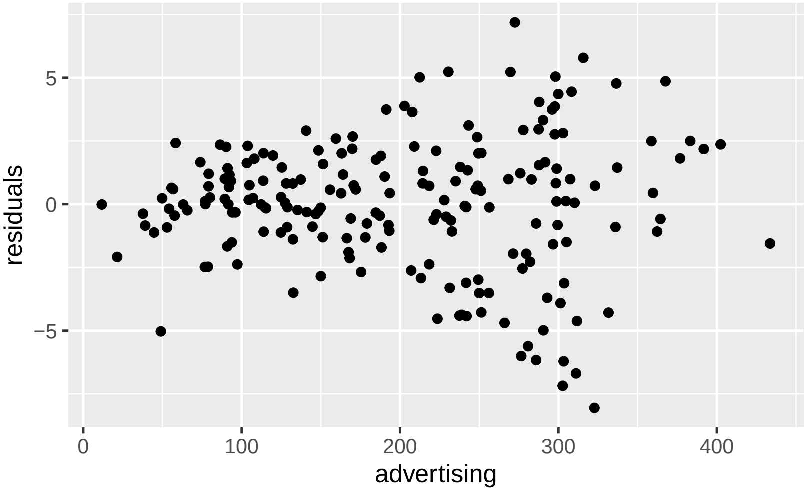
Let's learn how to visually inspect our data for heteroskedasticity in R (in Chapter 26 we will learn how to formally test for it). We will use the advertising and sales data again.

We first obtain the residuals from our estimated model, just like we did when we were testing assumption 2:

```
df <- read.csv("advertising-sales.csv")
m <- lm(sales ~ advertising, data = df)
df$residuals <- m$residuals
```

We then plot the residuals and the x variable against each other:

```
library(ggplot2)
ggplot(df, aes(advertising, residuals)) + geom_point()
```



The dispersion in the residuals appears to be increasing in the x variable. This is evidence of heteroskedasticity, a violation of assumption 5.

When we have a violation of homoskedasticity, our estimates of β_1 are not biased but we can no longer perform inference (obtain confidence intervals or perform hypothesis tests). In Chapter 26 we will learn how we can correct for heteroskedasticity.

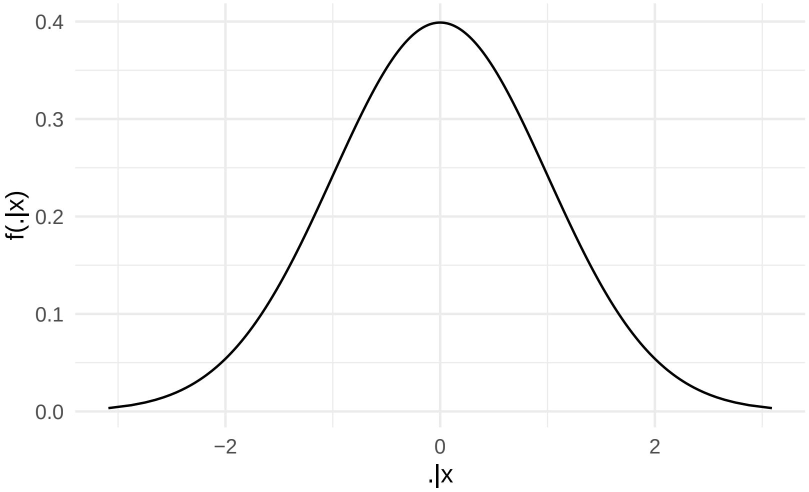
8.6 Assumption 6: Normality

Assumption 6: Normality

The distribution of ε_i conditional on x_i is normally distributed.

This means that the distribution of the error terms conditional on the X variable should have a symmetric bell-curve shape:

```
df <- data.frame(error = qnorm(seq(0.001, 0.999, by = 0.001)))
df$density <- dnorm(df$error)
ggplot(df, aes(error, density)) +
  geom_line() +
  theme_minimal() +
  xlab(" |x") +
  ylab("f( |x)")
```



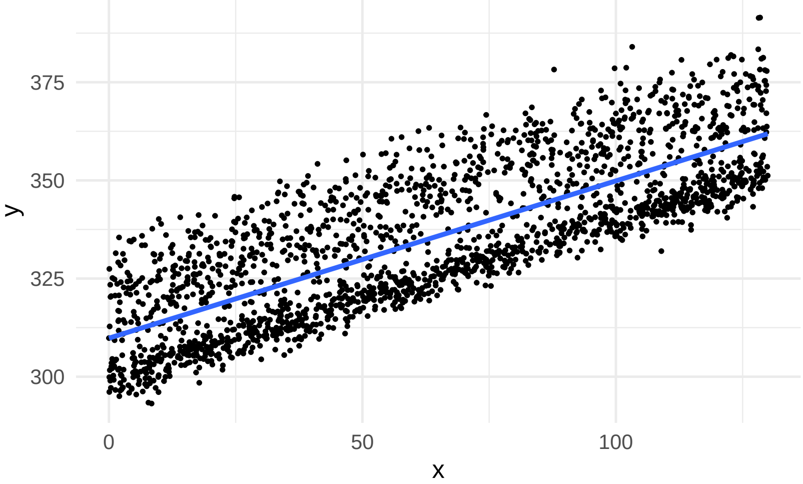
This assumption, combined with assumptions 4 and 5 implies:

$$\varepsilon_i|x_i \sim \mathcal{N}\left(0, \sigma_\varepsilon^2\right)$$

In words: ε_i conditional on x_i follows a normal distribution with a zero mean and variance σ_ε^2.

Let's take a look at an example scatter plot of X and Y of data that violate this assumption:

```r
set.seed(34636)
n <- 2000
df <- data.frame(x = runif(n, 0, 130))
df$y <- 300 + 0.4 * df$x +
  ifelse(rbinom(n, 1, 0.5) == 1, rnorm(n, 20, 8), rnorm(n, -0.5, 3))
ggplot(df, aes(x, y)) +
  geom_point(size = 0.5) +
  geom_smooth(formula = y ~ x, method = 'lm', se = FALSE) +
  theme_minimal()
```



We can see that the errors are not symmetric around the regression line. They are positively skewed. Errors below the regression line (negative values) are closer together, whereas above the regression line (positive values) they are more dispersed.

8.7 Model Assumptions Summary

In short the model assumptions to perform inference are:

- The population model is $Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$.
- We have a random sample of size n, $\left((x_1, y_1), \ldots (x_n, y_n)\right)$ following the population model, with the values $(x_1, \ldots, x_n)$ not all taking the same value.
- The errors conditional on x are normally distributed with a zero mean and constant variance: $\varepsilon_i | x_i \sim \mathcal{N}\left(0, \sigma_\varepsilon^2\right)$.

We can only perform inference when all model assumptions hold. Later in this course we will learn some techniques to deal with some violations of these assumptions.

Chapter 9

SLR Confidence Intervals

We will now learn how to estimate confidence intervals for our estimates b_0 and b_1. Before we do that, we will start by revising how to obtain a confidence interval for the sample mean that you learned in Statistics 1. We will then show the theory behind confidence intervals for the simple linear regression model, followed by how to compute them in R.

9.1 Confidence Interval for the Sample Mean

We first revise how to calculate a (1-α)% confidence interval for the sample mean. We will consider the case where the variance of X is not known and needs to be estimated.

The steps are:

1. We estimate the sample mean with $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$.
2. We estimate the sample variance with $s^2 = \frac{1}{n-1}\sum_{i=1}^{n}\left(x_i - \bar{x}\right)^2$.
3. We compute the standard error of the mean using the formula $\sqrt{\frac{s^2}{n}}$.
4. We look for quantile $1-\frac{\alpha}{2}$ of the Student's t distribution with $n-1$ degrees of freedom using software/tables. Call this number $t_{1-\frac{\alpha}{2},n-1}$.
5. We then calculate the confidence interval using the formula:

$$\bar{x} \pm t_{1-\frac{\alpha}{2},n-1}\sqrt{\frac{s^2}{n}}$$

Let's do a numeric example for a 95% confidence interval. Suppose you have $n = 400$ observations from a random sample. You calculate the sample mean $\bar{x} = 3$ and sample variance $s_X = 16$ from this sample. With $n = 400$ and $\alpha = 0.05$, the quantile of the Student's t distribution is $t_{1-\frac{\alpha}{2},n-1} = 1.96$. $t_{1-\frac{\alpha}{2},n-1} \times \frac{s_X}{\sqrt{n}}$.

Using the formula, the confidence interval is:

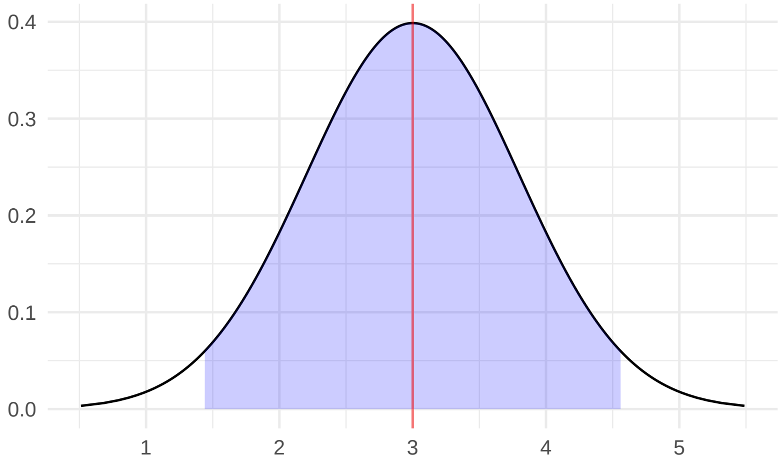
$$3 \pm 1.96 \times \sqrt{\frac{16^2}{400}}$$

Simplifying this gives 3 ± 1.57.

Another way to write the confidence interval is $[1.43, 4.57]$, i.e. $3 - 1.57$ and $3 + 1.57$. What does the confidence interval tell us? It tells us that we are 95% confident that the population mean is between 1.43 and 4.57.

We can illustrate this graphically as follows. We draw the estimated sampling distribution around $\bar{x}$ using the standard error $\sqrt{\frac{16^2}{400}}$. We can see that the distribution is centered around the sample mean of 3 (with the red line). The the 95% confidence interval is shaded in blue which contains 95% of the area under the curve. The area remaining to the left and right are each 2.5% of the total area. We can see that the left edge of the blue area is at 1.43 and the right edge is at 4.57, corresponding to the limits of the 95% confidence interval $[1.43, 4.57]$ we calculated above.

```r
library(ggplot2)
df <- data.frame(x = 3 + (16/sqrt(400))*qt(seq(0.001, 0.999, by = 0.001), 399))
df$y <- dt(qt(seq(0.001, 0.999, by = 0.001), 399), 399)
ci <- qt(0.975, 400 - 1) * (16/sqrt(400))
df$fill <- ifelse(df$x > 3 - ci & df$x < 3 + ci, df$x, NA)
ggplot(df, aes(x, y)) +
  geom_line() +
  xlab("") +
  ylab("") +
  geom_area(aes(x = fill), fill = "blue", alpha = 0.2)  +
  geom_vline(xintercept = 3, color = "red", alpha = 0.5) +
  theme_minimal()
```



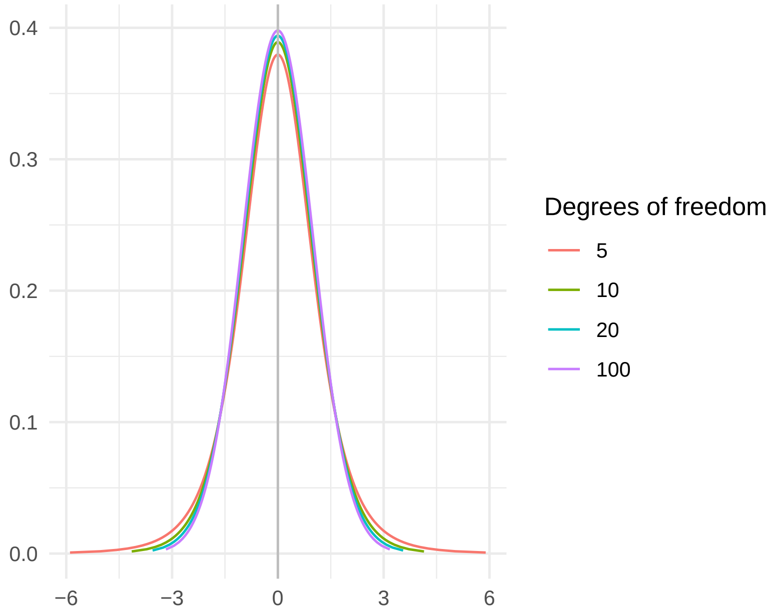
9.2 Who is the “Student” behind the *t* distribution?

Before discussing how to get confidence intervals for the simple linear regression model, let’s just take a quick aside to discuss why we call it the “Student’s” *t* distribution. The “student” is actually William Sealy Gosset (1876-1937), who was the head brewer at the Guinness brewery in Dublin. Gosset wanted to determine the quality of batches of hops by calculating the proportion of soft and hard resins in small samples. Based on these small samples, he wanted to make inference over the entire batch of hops. But because the samples were so small he could not use the normal distribution. Instead he had to come up with a different way to calculate confidence intervals. He figured out how to do this mathematically. Because this discovery was useful beyond brewing (and why we are learning it here) we wanted to publish his discovery. But in order to avoid publishing trade secrets, he published it under a boring title that Guinness’s competitors would never read and wrote about his work under the pseudonym, “Student”.

This is probably Ireland’s greatest contribution to statistics. Unfortunately when you visit the Guinness Brewery in Dublin there is only a tiny plaque stating this. They should definitely make a bigger deal of it!

Let’s take a look at the *t* distribution for different values of the degrees of freedom:

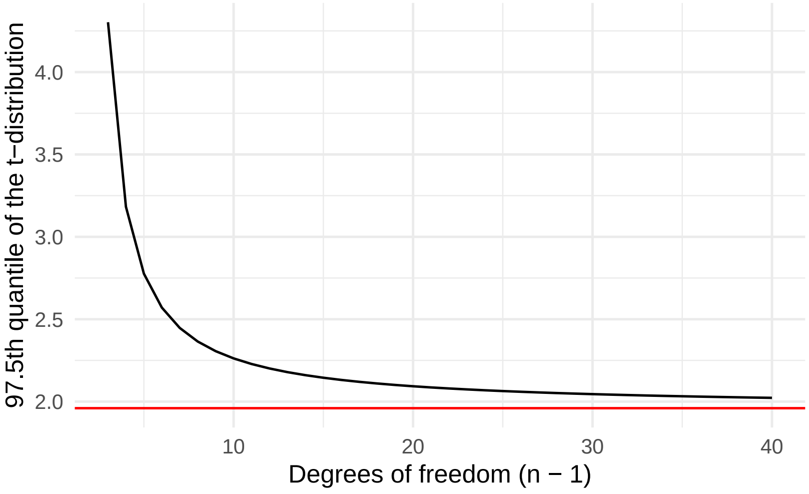
```r
library(ggplot2)
df <- do.call(rbind, lapply(c(5, 10, 20, 100), function(j) {
  out <- data.frame(x = qt(seq(0.001, 0.999, by = 0.001), j), deg_freedom = j)
  out$y <- dt(qt(seq(0.001, 0.999, by = 0.001), j), j)
  return(out)
}))
ggplot(df, aes(x, y, color = factor(deg_freedom))) +
  geom_line() +
  xlab("") +
  ylab("") +
  geom_vline(xintercept = 0, color = "gray") +
  scale_color_discrete(name = "Degrees of freedom") +
  theme_minimal()
```



It has a mean of zero and is almost the same shape as the standard normal distribution. With small n the distribution is wider but as n grows large it converges to the standard normal distribution.

To see how different the quantiles of the t-distribution can be at small sample sizes, we plot the quantile $t_{1-\frac{\alpha}{2},n-1}$ for $\alpha = 0.05$ and n from 3 to 40. We can see that at very small n the quantile is very large. But as n gets larger it approaches the the familiar 1.96 of the normal distribution (shown in red).

```r
library(ggplot2)
df <- data.frame(n = 3:40)
df$t <- qt(0.975, df$n - 1)
ggplot(df, aes(n, t)) +
  geom_line() +
  xlab("Degrees of freedom (n - 1)") +
  ylab("97.5th quantile of the t-distribution") +
  geom_hline(yintercept = qnorm(0.975), color = "red") +
  theme_minimal()
```



9.3 The Standard Errors of the Regression Coefficients

9.3.1 Theory

Above we saw that the standard error of the sample mean was $\sqrt{\frac{s^2}{n-1}}$. To be able to form confidence intervals for the regression coefficients, we need the analog of this for the regression coefficients. To obtain these, we first need to get the sample variance of the estimated model, s_ε^2. The formula for this is:

$$s_\varepsilon^2 = \frac{\sum_{i=1}^n \left(y_i - \hat{y}_i\right)^2}{n-2} = \frac{\sum_{i=1}^n e_i^2}{n-2}$$

The sum $\sum_{i=1}^n e_i^2$ is called the “sum of squared errors”, or SSE for short. We divide by $n-2$ instead of $n-1$ because we had to estimate *two* parameters (β_0 and β_1) to obtain the residuals e_i. When we estimate the sample variance, we only had to estimate *one* parameter (the sample mean), which is why we divided by $n-1$ in that case.

The standard errors for the intercept and slope are then found with the formulas:

$$s_{b_0} = \frac{\sum_{i=1}^{n} x_i^2}{n} \frac{s_\varepsilon}{\sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2}} \qquad s_{b_1} = \frac{s_\varepsilon}{\sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2}}$$

Note: You don't need to know the formula for s_{b_0} or s_{b_1} for the exam. We will always calculate these with R.

9.3.2 Standard Errors in R

Let's see how to calculate these with R. The most straightforward way to do this is to use the `summary()` command with the estimated regression model. Let's try it with the advertising and sales data:

```r
df <- read.csv("advertising-sales.csv")
m <- lm(sales ~ advertising, data = df)
summary(m)
```

```

Call:
lm(formula = sales ~ advertising, data = df)

Residuals:
    Min      1Q  Median      3Q     Max 
-8.0546 -1.3071  0.1173  1.5961  7.1895 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 4.243028   0.438525   9.676   <2e-16 ***
advertising 0.048688   0.001982  24.564   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.6 on 198 degrees of freedom
Multiple R-squared:  0.7529,    Adjusted R-squared:  0.7517 
F-statistic: 603.4 on 1 and 198 DF,  p-value: < 2.2e-16
```

The standard error for the intercept is $s_{b_0} = 0.438525$ and the standard error for the slope is $s_{b_1} = 0.001982$.

The `summary()` command also gives lots of information about the regression. We will learn what all parts of the output means over the coming lectures. If we only want to see the coefficients table with the standard errors, we can do:

```r
coef(summary(m))
```

```
              Estimate  Std. Error   t value     Pr(>|t|)
(Intercept) 4.24302822 0.438525138  9.675678 2.230651e-18
advertising 0.04868788 0.001982108 24.563691 5.059270e-62
```

This table is a `matrix`, which is an R object which is a rectangular array with each element having the same type. Here the array is 2 × 4 (2 rows and 4 columns).

```r
class(coef(summary(m)))
```

```
[1] "matrix" "array"
```

This is different from a `data.frame` because in a `data.frame` columns could have different types (but all elements of each column had to have the same type and length).

To get the standard errors from this matrix, we can extract the column either by its index or its column name:

```r
coef(summary(m))[, 2]
```

```
(Intercept) advertising
0.438525138 0.001982108
```

```r
coef(summary(m))[, "Std. Error"]
```

```
(Intercept) advertising
0.438525138 0.001982108
```

To extract a single value we must also specify the row. We can do this either by its index or its row name. Suppose we want to get s_{b_1}. We can do either:

```r
coef(summary(m))[2, 2]
```

```
[1] 0.001982108
```

or:

```r
coef(summary(m))["advertising", "Std. Error"]
```

```
[1] 0.001982108
```

Of course the first option involves less typing. However, the second is much clearer what the code intends to do: we can read *advertising* and *Std. Error* and know that the number it produces will be the standard error on the advertising coefficient from our model. Therefore the second option is arguably better code.

9.4 Confidence Intervals for Regression Coefficients

9.4.1 Theory

The formula for the regression coefficient confidence intervals is very similar to the one for the sample mean. The formula for the confidence interval for the regression slope is:

$$b_1 \pm t_{1-\frac{\alpha}{2},n-2} \times s_{b_1}$$

Let's compare this to the one for the sample mean we saw above:

$$\bar{x} \pm t_{1-\frac{\alpha}{2},n-1} \times \sqrt{\frac{s^2}{n}}$$

There are 3 differences:

1. We replaced the sample mean $\bar{x}$ with the estimate of the regression slope b_1.
2. We use $n-2$ degrees of freedom instead of $n-1$ when obtaining the $(1-\frac{\alpha}{2})$ quantile of the Student's t distribution.
3. We replaced the standard error of the sample mean $\sqrt{\frac{s^2}{n}}$ with the standard error of the sample regression slope.

Therefore once we know what the standard error is, the formula is essentially the same in both cases: it is the estimate plus or minus the relevant quantile of the Student's t distribution multiplied by the standard error. Only the degrees of freedom argument is different.

9.4.2 Numeric Example

Let's do a numeric example with this formula. Suppose you have a sample with $n = 100$ observations and want a 95% confidence interval for the regression slope. You estimate a slope of $b_1 = 0.3$ and get a standard error of $s_{b_1} = 0.1$. We look up the quantile of the Student's t distribution and obtain $t_{1-\frac{\alpha}{2},n-2} = t_{0.975,98} = 1.984$. To get this quantile in R we can use the `qt()` function:

```r
qt(0.975, 98)
```

```
[1] 1.984467
```

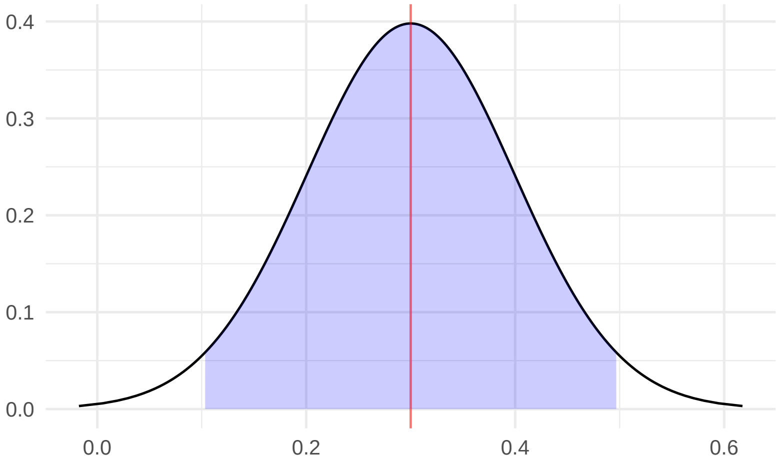
We then use the formula:

$$\begin{aligned}
&b_1 \pm t_{1-\frac{\alpha}{2},n-2} \times s_{b_1}\\
&0.3 \pm 1.984 \times 0.1\\
&0.3 \pm 0.1984
\end{aligned}$$

The confidence interval is then $[0.102, 0.498]$. We are 95% confident that the population regression slope β_1 is between 0.102 and 0.498. The entire confidence interval is above zero so we are 95% confident that X has an effect on Y. That is, the confidence interval does not contain zero, so we are 95% confidence that $\beta_1 \neq 0$.

Graphically the confidence interval is the width of the shaded blue area around the sample estimate at the red line:

```r
library(ggplot2)
df <- data.frame(x = 0.3 + 0.1*qt(seq(0.001, 0.999, by = 0.001), 98))
df$y <- dt(qt(seq(0.001, 0.999, by = 0.001), 98), 98)
ci <- qt(0.975, 98) * 0.1
df$fill <- ifelse(df$x > 0.3 - ci & df$x < 0.3 + ci, df$x, NA)
ggplot(df, aes(x, y)) +
  geom_line() +
  xlab("") +
  ylab("") +
  geom_area(aes(x = fill), fill = "blue", alpha = 0.2)  +
  geom_vline(xintercept = 0.3, color = "red", alpha = 0.5) +
  theme_minimal()
```



The shaded blue area under the curve represents 95% of the total area. The white areas in the tails each make up 2.5% of the area.

9.4.3 Confidence Intervals in R

R has a built-in function to easily calculate confidence intervals called `confint()`. We can use it as follows to get a 95% confidence interval for both b_0 and b_1:

```
df <- read.csv("advertising-sales.csv")
m <- lm(sales ~ advertising, data = df)
confint(m, level = 0.95)
```

```
                 2.5 %     97.5 %
(Intercept) 3.37824898 5.10780745
advertising 0.04477913 0.05259663
```

We are 95% confident that the population intercept β_0 is between 3.3782 and 5.1078 and we are 95% confident that the population slope β_1 is between 0.0448 and 0.0526. The entire confidence interval for the slope is above zero so we are 95% confidence that advertising does have an effect on sales (i.e. $\beta_1 \neq 0$).

If we only want to get the confidence interval for the slope and not the intercept we can specify the parameter we want to get:

```
confint(m, parm = "advertising", level = 0.95)
```

```
                 2.5 %     97.5 %
advertising 0.04477913 0.05259663
```

We could alternatively save the confidence interval to an object (let's call it `ci`) and extract elements from it. Suppose I wanted to get just the lower bound of the confidence interval for the slope:

```
ci <- confint(m, level = 0.95)
ci[2, 1]
```

```
[1] 0.04477913
```

9.4.4 Manually Calculating Confidence Intervals in R

Finally, let's check that using the mathematical formula for the confidence interval directly gives the same results as using `confint()`.

```
b_1 <- coef(summary(m))["advertising", "Estimate"]
s_b_1 <- coef(summary(m))["advertising", "Std. Error"]
c(b_1 - qt(0.975, 198) * s_b_1,
    b_1 + qt(0.975, 198) * s_b_1)
```

```
[1] 0.04477913 0.05259663
```

We get the same results as above!

Chapter 10

SLR Hypothesis Testing

We will now learn how to do hypothesis tests for the regression slope. Like confidence intervals, it is very similar to performing hypothesis tests for the sample mean that you learned about in Statistics 1. We will first show the theory behind hypothesis testing for the regression slope. We will then show a numeric example before finally showing how to perform hypothesis tests in R.

10.1 Notation

Suppose we wanted to test if β_1 was different to some number b. For example, if we wanted to test if the slope was not equal to one (i.e. $\beta_1 \neq 1$), then we would have $b = 1$. We call this number b the *hinge*. If we are testing if $\beta_1 \neq b$, we set up the null and alternative hypotheses as follows:

- Null hypothesis: $H_0 : \beta_1 = b$
- Alternative hypothesis: $H_1 : \beta_1 \neq b$

If we are testing a claim like "β_1 is not equal to b" then that is referring to the *alternative* hypothesis. The null hypothesis is then always the exact opposite of the alternative hypothesis.

This type of test is called a two-sided test because the values of β_1 in the alternative hypothesis are on two sides of the null hypothesis.

Denote by B_1 the random variable that estimates β_1 for any random sample drawn from the population. This B_1 is not the regression output for our dataset (our observed sample) in R - that is b_1. Nor is it the true population slope we are trying to estimate - this is β_1. Instead B_1 is a theoretical object that maps a hypothetical random sample that we *could* draw from the population into an estimate of β_1.

When we have a dataset, we have exactly one random sample drawn from the population and our estimate of β_1 from this is b_1. But in principle we could draw another random sample from the population and get a different estimate of β_1. We can think of the possible values of B_1 as all the possible values of estimates of β_1 from different random samples from the population. While b_1 is observed and fixed (because we have observed our sample and estimated the slope), B_1 is neither observed nor fixed: it's a random variable. The estimate b_1 we get from our observed dataset in R is a single realization of this random variable.

So to summarize:

- β_1 is the population regression slope (unobserved and fixed).
- b_1 is the estimated regression slope with our data (observed and fixed).
- B_1 is the random variable that estimates the slope for any random sample (unobserved and a random variable).

We similarly denote by S_{B_1} the random variable that estimates the standard error of the regression slope for any random sample drawn from the population.

10.2 Test Statistic

We will not show the steps but it can be proven mathematically from our model assumptions that:

$$T = \frac{B_1 - \beta_1}{S_{B_1}} \sim t_{n-2}$$

This means that T follows a t distribution with $n-2$ degrees of freedom. This T is also a random variable, because it is a function of two other random variables (B_1 and S_{B_1} and a constant parameter β_1). This formulation is very useful because T has a distribution that does not depend on any unknown parameters. There is no unknown mean or variance for this distribution: it only depends on n which we know. We call T a *pivot* because it follows a distribution that does not depend on unknown parameters.

Now, if the null hypothesis is true ($\beta_1 = b$), then it is the case that:

$$T = \frac{B_1 - b}{S_{B_1}} \sim t_{n-2}$$

So if the null hypothesis were true, then samples drawn from the population should produce values of T that follow this distribution. Denote by small t the realized value of T from our sample. We calculate this using the formula $t = \frac{b_1 - b}{s_{b_1}}$. If the null hypothesis is true, most of the time we should get values of t close to zero, but occasionally (about 5% of the time) we could get more extreme values further away from zero (like greater than $+2$ or less than -2).

How can we use this to test our hypothesis? If we calculate t from our observed sample and find that the value is extreme (like greater than $+2$ or less than -2), then there are 2 possibilities:

- The null hypothesis is true and the sample we observed was a rare case of getting an extreme value of t.
- The null hypothesis is false.

Because the first possibility is rare, then it is more likely that the second possibility holds: it is likely that the null hypothesis is false. Of course it could be possible that the null hypothesis is true and we just observed a freak event. We can't tell these apart. However, it's much more likely that the null hypothesis is false. So if we find that what we observe in our sample to be extremely rare if the null hypothesis were true, then we conclude that the null hypothesis is false. If we find values in the normal range of the null hypothesis we instead conclude that we have no evidence to say the null hypothesis is false.

10.3 Size of the Test

How do we decide whether to reject the null hypothesis or not based on the realized value of T? We first have to decide on the *size of the test.* This is the highest probability that we are willing to accept that we might falsely reject the null hypothesis when it is in fact true. The most common size you see is 5%, but sometimes people use 1% or 10%. The size of the test is denoted by α, where a 5% size is denoted by $\alpha = 0.05$. With $\alpha = 0.05$, there is a 5% chance that we reject null hypothesis when it is in fact true. But that means that 95% of the time we reject the null it is in fact false.

Once we have decided on the size of the test there are two possible ways to proceed, both yielding the same conclusion:

1. The critical value approach (also called the rejection region approach).
2. The p-value approach.

Let's discuss each of these in turn.

10.4 Critical Value Approach for a Two-Sided Test

The critical value approach involves finding a number c that solves the equation:

$$\Pr(|T| \geq c) = \alpha \quad \text{under } H_0$$

In words this means the probability that the absolute value of $T = \frac{B_1 - b}{S_{B_1}}$ from any sample being larger than c is equal to α under the null hypothesis. So if $\alpha = 0.05$, under the null hypothesis the probability of getting a value of $|T|$ larger than the critical value is 5%.

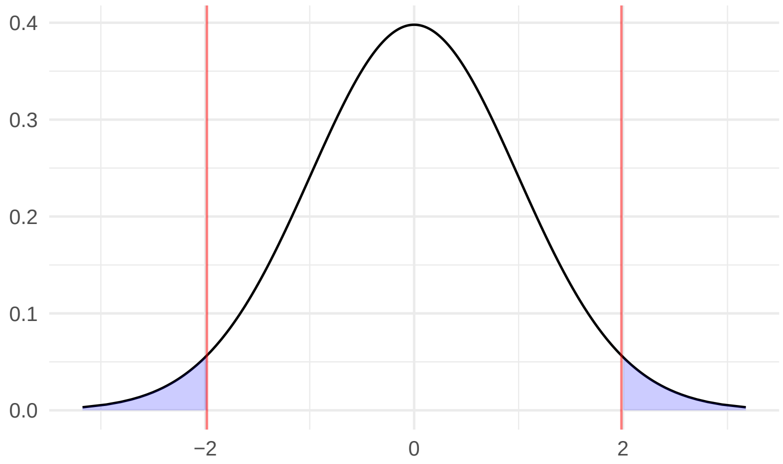
The c that solves this equation is $t_{1-\frac{\alpha}{2},n-2}$, the same quantile from the t distribution that we we use for a $(1-\alpha)\%$ confidence interval. With $\alpha = 0.05$, this is the 97.5th quantile of the t distribution with $n-2$ degrees of freedom.

With this number we compare the realized value of T, $t = \frac{b_1 - b}{s_{b_1}}$ to this critical value $t_{1-\frac{\alpha}{2},n-2}$:

- If $|t| \geq t_{1-\frac{\alpha}{2},n-2}$ we reject the null hypothesis.
- If $|t| < t_{1-\frac{\alpha}{2},n-2}$ we fail to reject the null hypothesis.

Graphically, we calculate t and check if the value lies in one of the shaded regions below:

```r
library(ggplot2)
df <- data.frame(x = qt(seq(0.001, 0.999, by = 0.001), 98))
df$y <- dt(qt(seq(0.001, 0.999, by = 0.001), 98), 98)
cv <- qt(0.975, 98)
df$fill_1 <- ifelse(df$x < -cv, df$x, NA)
df$fill_2 <- ifelse(df$x > cv, df$x, NA)
ggplot(df, aes(x, y)) +
  geom_line() +
  xlab("") +
  ylab("") +
  geom_area(aes(x = fill_1), fill = "blue", alpha = 0.2)  +
  geom_area(aes(x = fill_2), fill = "blue", alpha = 0.2)  +
  geom_vline(xintercept = -cv, color = "red", alpha = 0.5) +
  geom_vline(xintercept = cv, color = "red", alpha = 0.5) +
  theme_minimal()
```



The area of the shaded region is exactly 5% of the total area with $\alpha = 0.05$ (2.5%

on the left and 2.5% on the right). This is the same as with a confidence interval. If the realized value of the test statistic t is in the shaded area, then such a value is unlikely to occur if the null hypothesis is true (occurs with probability 5%). We therefore would reject the null. If the realized value t is in between the two red lines, then the value is not extreme under the null hypothesis and we fail to reject the null hypothesis.

10.5 p-Value Approach for a Two-Sided Test

The other approach is the p-value approach. This approach involves finding a number p that solves the equation:

$$\Pr(|T| \geq |t|) = p \quad \text{under } H_0$$

In words: under H_0, the probability that the absolute value of $T = \frac{B_1 - b}{S_{B_1}}$ from any sample being larger than the absolute value of the observed realization of t is equal to p. It is probability of drawing a sample from the population that is more extreme than the observed one under the null hypothesis.

This p can be calculated with:

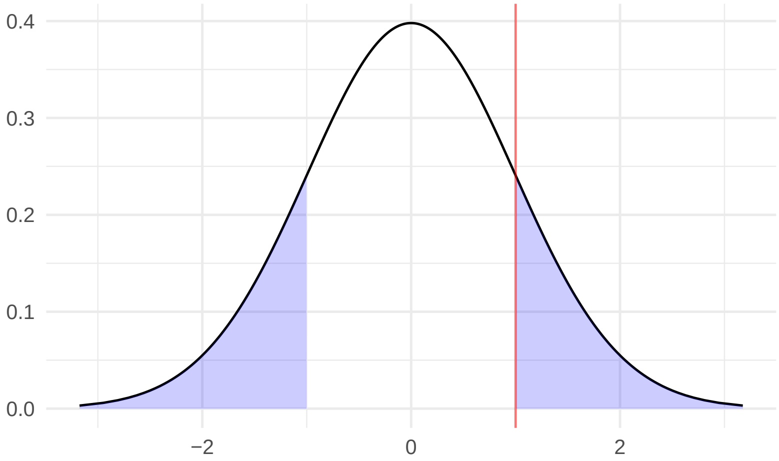
$$p = 2 \times (1 - \Pr(T < |t|))$$

We then compare this number p with the size of the test α:

- If $p \leq \alpha$ we reject the null hypothesis.
- If $p > \alpha$ we fail to reject the null hypothesis.

Graphically, if we calculate $t = 1$, then the p-value is the area to the right of 1 and to the left of -1:

```r
library(ggplot2)
df <- data.frame(x = qt(seq(0.001, 0.999, by = 0.001), 98))
df$y <- dt(qt(seq(0.001, 0.999, by = 0.001), 98), 98)
t <- 1
df$fill_1 <- ifelse(df$x < -t, df$x, NA)
df$fill_2 <- ifelse(df$x > t, df$x, NA)
ggplot(df, aes(x, y)) +
  geom_line() +
  xlab("") +
  ylab("") +
  geom_area(aes(x = fill_1), fill = "blue", alpha = 0.2)  +
  geom_area(aes(x = fill_2), fill = "blue", alpha = 0.2)  +
  geom_vline(xintercept = t, color = "red", alpha = 0.5) +
  theme_minimal()
```



With a sample size of $n = 100$ (98 degrees of freedom), using the formula

$$p = 2 \times (1 - \Pr(T < |t|)) = 2 \times (1 - \Pr(T < 1))$$

the p-value is equal to:

```
2 * (1 - pt(1, 98))
```

```
[1] 0.3197733
```

The function `pt(t, n-2)` is the R function for $\Pr(T < t)$ with a t distribution with $n-2$ degrees of freedom. The shaded blue area is thus 31.97% of the total area. The probability of observing a a sample at least as extreme as $t = 1$ is 31.97%. Because this probability is bigger than 5% we would fail to reject the null hypothesis.

Note that we always end up with the same rejection decision using the critical value approach and the p-value approach. If you do both and end up with different answers, then you know something has gone wrong.

10.6 Making a Conclusion

Once we have rejected or failed to reject the null hypothesis we need to make a conclusion about the initial claim:

- If we reject null hypothesis we conclude that there is sufficient evidence for the claim.
- If we fail to reject, we say there is insufficient evidence for the claim.

If we fail to reject a null hypothesis, we never actually *accept* the null hypothesis. We just say there is not enough evidence to reject it.

To see why we do this, suppose we had a very small sample size (like $n = 5$). With such little data our estimate of β_1 would be very imprecise, leading to a large standard error s_{b_1}. This would lead to a very small value of $t = \frac{b_1 - b}{s_{b_1}}$ even if the null hypothesis is actually false. We would end up failing to reject the null hypothesis because of the small t. To conclude then the null hypothesis is true from a handful of observations would be very naive. Instead, we just don't have enough evidence to say that it is false.

10.7 One-Sided Tests

10.7.1 Hypotheses

If the claim we want to test is "β_1 is larger than b" or "β_1 is smaller than b", then we need to use a one-sided test. These can be either upper-tail or lower-tail tests:

- "β_1 is larger than b" $\Rightarrow$ Upper-tail test.
- "β_1 is smaller than b" $\Rightarrow$ Lower-tail test.

In these cases, the null and alternative hypotheses are:

- Upper-tail test: H_0: $\beta_1 \leq b$, H_1: $\beta_1 > b$
- Lower-tail test: H_0: $\beta_1 \geq b$, H_1: $\beta_1 < b$

Notice that the claim corresponds to the alternative hypothesis. For example, if the claim is "β_1 is larger than b", then this is an upper-tail test and the alternative hypothesis corresponds to the claim: $\beta_1 > b$. The null hypothesis is then just the opposite of the alternative hypothesis. When you are asked to perform a one-sided test you should therefore write down the alternative hypothesis first (which corresponds to the claim you need to test) and then write the null hypothesis as the opposite of this.

10.7.2 Test Statistics

For a one-sided test, the test statistic is the same as before. Under the null hypothesis:

$$T = \frac{B_1 - b}{S_{B_1}} \sim t_{n-2}$$

We also calculate the realized value of the test statistic in our sample the same way:

$$t = \frac{b_1 - b}{s_{b_1}}$$

10.7.3 Critical Values

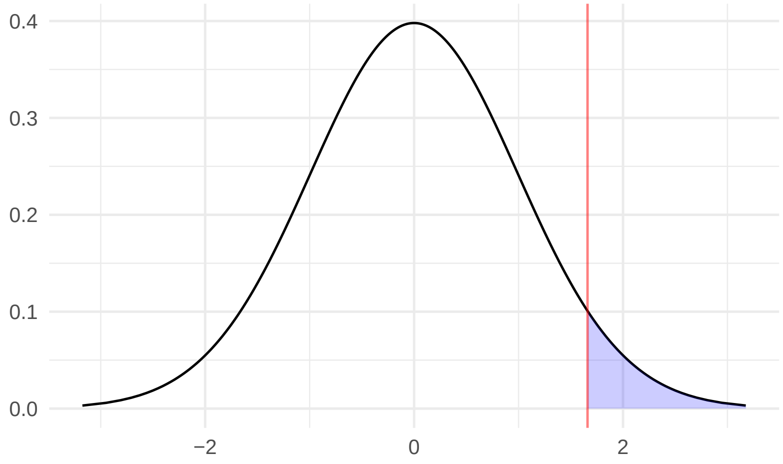
The critical value for an upper-tail test is the value c that solves:

$$\Pr(T \geq c) = \alpha$$

It is the number c such that under H_0 the probability that T exceeds it is equal to α. This is different from the two-sided test because we don't use the absolute value. The critical value here is equal to $t_{1-\alpha,n-2}$. Notice that we find the $1-\alpha$ quantile, and not $1-\frac{\alpha}{2}$ quantile as in the two-sided test.

Graphically, an upper-tail test with the critical value approach involves checking if t in our sample is in the shaded region below:

```r
library(ggplot2)
df <- data.frame(x = qt(seq(0.001, 0.999, by = 0.001), 98))
df$y <- dt(qt(seq(0.001, 0.999, by = 0.001), 98), 98)
cv <- qt(0.95, 98)
df$fill <- ifelse(df$x > cv, df$x, NA)
ggplot(df, aes(x, y)) +
  geom_line() +
  xlab("") +
  ylab("") +
  geom_area(aes(x = fill), fill = "blue", alpha = 0.2)  +
  geom_vline(xintercept = cv, color = "red", alpha = 0.5) +
  theme_minimal()
```



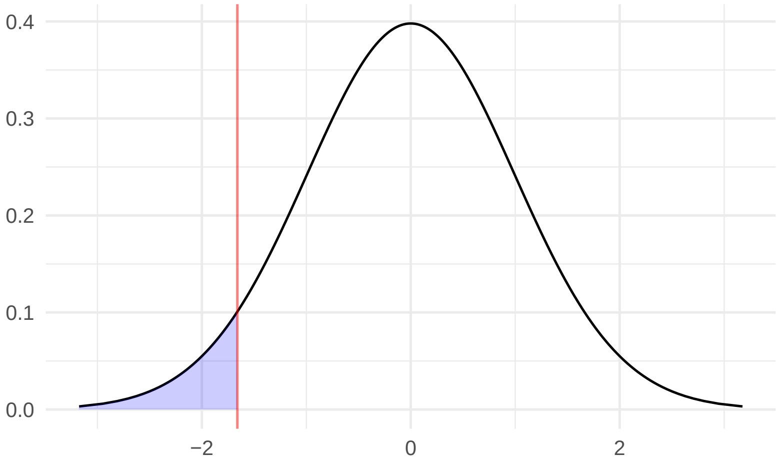
With $\alpha = 0.05$, the shaded region has an area of 5%.

The critical value for an lower-tail test is the value c that solves:

$$\Pr(T \leq c) = \alpha$$

It is the number c such that under H_0 the probability that T is smaller than c is equal to α. The critical value here is equal to $t_{\alpha,n-2}$. This is always equal to $-t_{1-\alpha,n-2}$, the negative of the equivalent upper-tail test critical value. Graphically, an lower-tail test with the critical value approach involves checking if t in our sample is in the shaded region below:

```r
library(ggplot2)
df <- data.frame(x = qt(seq(0.001, 0.999, by = 0.001), 98))
df$y <- dt(qt(seq(0.001, 0.999, by = 0.001), 98), 98)
cv <- -qt(0.95, 98)
df$fill <- ifelse(df$x < cv, df$x, NA)
ggplot(df, aes(x, y)) +
  geom_line() +
  xlab("") +
  ylab("") +
  geom_area(aes(x = fill), fill = "blue", alpha = 0.2)  +
  geom_vline(xintercept = cv, color = "red", alpha = 0.5) +
  theme_minimal()
```



Again, with $\alpha = 0.05$, the shaded region has an area of 5%.

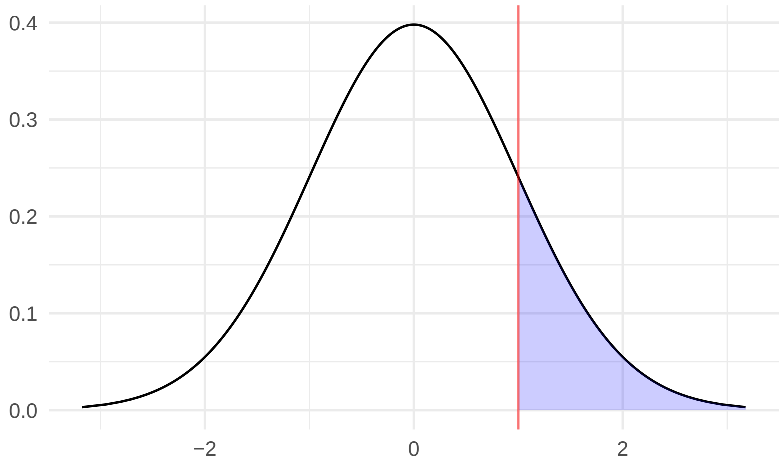
10.7.4 p-Values

To get the p-value for a one-sided test we need to find the probability of obtaining a T at least as extreme as the observed t *in the direction of the test.* For an upper-tail test this is the area under the distribution of T to the right of t:

$$p = \Pr(T \geq t) = 1 - \Pr(T < t)$$

Graphically, if we calculate $t = 1$, then the p-value is the area to the right of 1:

```
library(ggplot2)
df <- data.frame(x = qt(seq(0.001, 0.999, by = 0.001), 98))
df$y <- dt(qt(seq(0.001, 0.999, by = 0.001), 98), 98)
t <- 1
df$fill <- ifelse(df$x > t, df$x, NA)
ggplot(df, aes(x, y)) +
  geom_line() +
  xlab("") +
  ylab("") +
  geom_area(aes(x = fill), fill = "blue", alpha = 0.2)  +
  geom_vline(xintercept = t, color = "red", alpha = 0.5) +
  theme_minimal()
```



With a sample size of $n = 100$ (98 degrees of freedom), using the formula

$$p = 1 - \Pr(T < t) = 1 - \Pr(T < 1)$$

the p-value is equal to:

```r
1 - pt(1, 98)
```

```
[1] 0.1598866
```

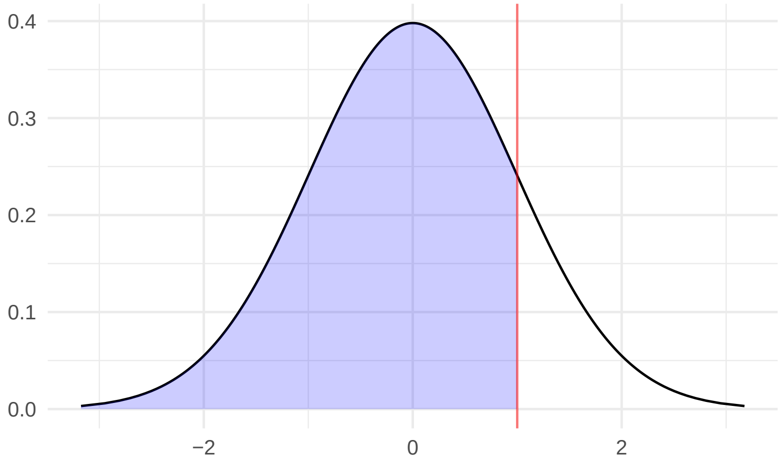
The shaded blue area is thus 15.989% of the total area. The probability of observing a a sample at least as extreme as $t = 1$ in the direction of the test is 15.989%. Because this probability is bigger than 5% we would fail to reject the null hypothesis.

For a lower-tail test the p-value is the area in the distribution of T to the left of t:

$$p = \Pr(T \leq t)$$

Graphically, if we calculate $t = 1$, then the p-value is the area to the left of 1:

```r
library(ggplot2)
df <- data.frame(x = qt(seq(0.001, 0.999, by = 0.001), 98))
df$y <- dt(qt(seq(0.001, 0.999, by = 0.001), 98), 98)
t <- 1
df$fill <- ifelse(df$x < t, df$x, NA)
ggplot(df, aes(x, y)) +
  geom_line() +
  xlab("") +
  ylab("") +
  geom_area(aes(x = fill), fill = "blue", alpha = 0.2)  +
  geom_vline(xintercept = t, color = "red", alpha = 0.5) +
  theme_minimal()
```



With a sample size of $n = 100$ (98 degrees of freedom), using the formula

$$p = \Pr(T \leq t) = \Pr(T \leq 1)$$

the p-value is equal to:

```r
pt(1, 98)
```

```
[1] 0.8401134
```

The shaded blue area is thus 84.01% of the total area. The probability of observing a a sample at least as extreme as $t = 1$ in the direction of the test is 84.01%. Because this probability is bigger than 5% we would fail to reject the null hypothesis.

10.8 Recap

10.8.1 Critical Value Approach

We first decide which type of test we need to use:

- If the claim is "β_1 is different from b" we use a two-sided test.
- If the claim is "β_1 is greater than b" we use an upper-tail test.
- If the claim is "β_1 is less than b" we use a lower-tail test.

We then set up the null and alternative hypotheses:

- Two-sided test: H_0: $\beta_1 = b$, H_1: $\beta_1 \neq b$
- Upper-tail test: H_0: $\beta_1 \leq b$, H_1: $\beta_1 > b$
- Lower-tail test: H_0: $\beta_1 \geq b$, H_1: $\beta_1 < b$

We then form the test statistic. Under the null hypothesis:

$$T = \frac{B_1 - b}{S_{B_1}} \sim t_{n-2}$$

We calculate the realized value of the test statistic using our sample:

$$t = \frac{b_1 - b}{s_{b_1}}$$

We then calculate the critical value and form rejection rules:

- Two-sided test: Reject if $|t| \geq t_{1-\frac{\alpha}{2},n-2}$ otherwise fail to reject.
- Upper-tail test: Reject if $t \geq t_{1-\alpha,n-2}$ otherwise fail to reject.
- Lower-tail test: Reject if $t \leq t_{\alpha,n-2}$ otherwise fail to reject.

Based on whether we reject or not, we make a conclusion about the initial claim.

10.8.2 p-Value Approach

We first decide which type of test we need to use:

- If the claim is "β_1 is different from b" we use a two-sided test.
- If the claim is "β_1 is greater than b" we use an upper-tail test.
- If the claim is "β_1 is less than b" we use a lower-tail test.

We then set up the null and alternative hypotheses:

- Two-sided test: H_0: $\beta_1 = b$, H_1: $\beta_1 \neq b$
- Upper-tail test: H_0: $\beta_1 \leq b$, H_1: $\beta_1 > b$
- Lower-tail test: H_0: $\beta_1 \geq b$, H_1: $\beta_1 < b$

We then form the test statistic. Under the null hypothesis:

$$T = \frac{B_1 - b}{S_{B_1}} \sim t_{n-2}$$

We calculate the realized value of the test statistic using our sample:

$$t = \frac{b_1 - b}{s_{b_1}}$$

We then calculate the p-value:

- Two-sided test: $p = 2 \times (1 - \Pr(T < |t|))$.
- Upper-tail test: $p = 1 - \Pr(T < t)$.
- Lower-tail test: $p = \Pr(T \leq t)$.

We reject if $p \leq \alpha$ otherwise we fail to reject.

Based on whether we reject or not, we make a conclusion about the initial claim.

10.9 Numeric Example

You have a sample with $n = 100$ observations and you estimate $b_1 = 0.3$ and $s_{b_1} = 0.1$. You want to test the claim $\beta_1 > 0.2$ with a p-value approach with $\alpha = 0.05$.

Solution:

This is an upper-tail test. The null and alternative hypotheses are:

- H_0: $\beta_1 \leq 0.2$
- H_1: $\beta_1 > 0.2$.

Under H_0:

$$T = \frac{B_1 - 0.2}{S_{B_1}} \sim t_{98}$$

The value of the test statistic is:

$$t = \frac{b_1 - b}{s_{b_1}} = \frac{0.3 - 0.2}{0.1} = 1$$

The p-value is $p = \Pr(T \geq 1)$. We calculate this in R with:

```
1 - pt(1, 98)
```

```
[1] 0.1598866
```

Conclusion: $p = 0.16 > \alpha = 0.05$ so we cannot reject H_0. There is no evidence that $\beta_1 > 0.2$ at the 5% level.

10.10 Hypothesis Tests in R

Using the advertising and sales data, you are asked to test the following claim at the 5% level: “An increase in advertising of €1,000 on average increases sales by more than €48,000.”

This is an upper-tail test. Recall that advertising is in thousands and sales is in millions. The claim is 1 unit of x increases y by $\frac{48{,}000}{1{,}000{,}000} = 0.048$ units. Thus, the claim is equivalent to testing if $\beta_1 > 0.048$. With this we can form the null and alternative hypotheses:

- H_0: $\beta_1 \leq 0.048$
- H_1: $\beta > 0.48$.

We form the test statistic. Under H_0:

$$T = \frac{B_1 - 0.048}{S_{B_1}} \sim t_{198}$$

We now calculate the value of the test statistic in R:

```
df <- read.csv("advertising-sales.csv")
m <- lm(sales ~ advertising, data = df)
b_1 <- coef(summary(m))["advertising", "Estimate"]
s_b_1 <- coef(summary(m))["advertising", "Std. Error"]
(t <- (b_1 - 0.048) / s_b_1)  # value of the test statistic
```

```
[1] 0.3470444
```

```
(cv <- qt(0.95, 198))         # critical value
```

```
[1] 1.652586
```

```
(pval <- 1 - pt(t, 198))      # p-value
```

```
[1] 0.3644633
```

We reject if $t \geq t_{1-\alpha,n-2}$ with the critical value approach and we reject if $p \leq \alpha$ with the p-value approach:

```
t > cv
```

```
[1] FALSE
```

```
pval < 0.05
```

```
[1] FALSE
```

Both are `FALSE`, so we fail to reject H_0 under both approaches.

There is no evidence for the claim that increasing advertising by €1,000 increases sales by more than €48,000 at the 5% level.

10.11 Summary of R Functions for Hypothesis Tests

Define the following:

- The size of the test, α, is `alpha`.
- The number of observations in the regression, n, is `n`.
- The value of the test statistic, t, is `t`.

Critical Values:

- If two-sided test: `qt(1-alpha/2, n-2)`.
- If upper-tail test: `qt(1-alpha, n-2)`.
- If lower-tail test: `qt(alpha, n-2)`.

p-values:

- If two-sided test: `2*(1-pt(abs(t), n-2))`.
- If upper-tailed test: `1-pt(t, n-2)`.
- If lower-tailed test: `pt(t, n-2)`.

Chapter 11

SLR Statistical Significance

11.1 Test for Model Usefulness

We will now discuss a particular hypothesis test for the regression slope that is so common it has its own name: a test for statistical significance. This is a two-sided test for the regression slope with a zero hinge ($b = 0$):

$$H_0 : \beta_1 = 0 \qquad H_1 : \beta_1 \neq 0$$

Recall that the model is:

$$\mathbb{E}[Y_i|x_i] = \beta_0 + \beta_1 x_i$$

Under the null hypothesis, the model is simply $\mathbb{E}[Y_i|x_i] = \beta_0$. The expected value of Y_i does not depend on x_i. The model trying to predict Y_i using x_i is completely *useless*. Under the alternative hypothesis, $\mathbb{E}[Y_i|x_i] = \beta_0 + \beta_1 x_i$ with $\beta_1 \neq 0$ so Y_i varies with x_i and the model is *useful* (at least to some degree).

Therefore this test is a test of *model usefulness*. If we reject H_0 at the 5% level we say the model is useful at the 5% level.

- If H_0 is rejected, we say the *variable X is significant* and b_1 is *significantly different from zero.*
- If H_0 is not rejected, we say the *variable X is insignificant* and b_1 is *not significantly different from zero.*

Because this test is so common, most statistical software (including R) that estimate the simple linear regression model provide test statistics and p-values for this test by default. We will see this in the next example.

11.2 Example in R

Let's test for model usefulness using the advertising and sales data. We will see that the `summary()` command provides the test statistic and p-value for this test by default:

```
df <- read.csv("advertising-sales.csv")
m <- lm(sales ~ advertising, data = df)
summary(m)
```

```
Call:
lm(formula = sales ~ advertising, data = df)

Residuals:
    Min      1Q  Median      3Q     Max 
-8.0546 -1.3071  0.1173  1.5961  7.1895 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 4.243028   0.438525   9.676   <2e-16 ***
advertising 0.048688   0.001982  24.564   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.6 on 198 degrees of freedom
Multiple R-squared:  0.7529,    Adjusted R-squared:  0.7517 
F-statistic: 603.4 on 1 and 198 DF,  p-value: < 2.2e-16
```

If were to calculate the value of the test statistic from our sample manually, we would calculate it from b_1 and s_{b_1} using:

$$t = \frac{b_1 - 0}{s_{b_1}} = \frac{0.048688 - 0}{0.001982}$$

Let's calculate this in R:

```
b_1 <- coef(summary(m))["advertising", "Estimate"]
s_b_1 <- coef(summary(m))["advertising", "Std. Error"]
b_1 / s_b_1
```

```
[1] 24.56369
```

Looking back at the `summary()` output we see that under `t value` and across from `advertising` in the `summary()` also has this number (rounded to 3 digits after the decimal). The `summary()` command for a regression model always shows the test statistic for a two-sided test with a zero hinge (the test for statistical significance).

Let’s compare this to the critical value for $\alpha = 0.05$:

```
qt(0.975, 198)
```

```
[1] 1.972017
```

The value of the test statistic 24.564 is greater than the critical value 1.972, so advertising is statistically significant at the 5% level.

The `summary()` table also shows the corresponding p-value for this test in the 4th column. The `<2e-16` means that the number is very very close to zero. `2e-16` here means the number 2 divided by a very large number (a 1 followed by 16 zeros). The `<2e-16` means that the p-value is smaller than this number. Thus the p-value is close to zero, so advertising is statistically significant at the 5% level ($p < 0.05$).

11.3 Significance Stars

The `summary()` command also shows some `***` after the p-value and below the coefficients table it shows `Signif. codes`. This indicates that 3 stars means the p-value is less than 0.001. Here is what all the stars would mean:

- 3 stars (`***`): p-value is between 0 and 0.001.
- 2 stars (`**`): p-value is between 0.001 and 0.01.
- 1 star (`*`): p-value is between 0.01 and 0.05.
- 1 dot (`.`): p-value is between 0.05 and 0.1.
- No star/dot: p-value is between 0.01 and 1.

In the example above, both the intercept and the slope have 3 stars because the p-value for the hypothesis test that the coefficient is different from zero is close to zero in both cases.

The purpose of these stars is for you to be able to quickly see which estimates are significantly different from zero.

Chapter 12

SLR Quantifying Model Usefulness

In Chapter 11 we learned how to test if the model was useful or useless. But we also want to be able to quantify the usefulness of the model. That is, we want to say how much of the variation in the Y-variable we can explain with the X variable.

We do this by comparing the model error before estimating the regression model to the model error after estimating the regression model.

12.1 Total Sum of Squares

Without a regression model, the best way to predict values y_i is to use the sample mean $\bar{y}$. If we do this the sum of squared errors before the regression is

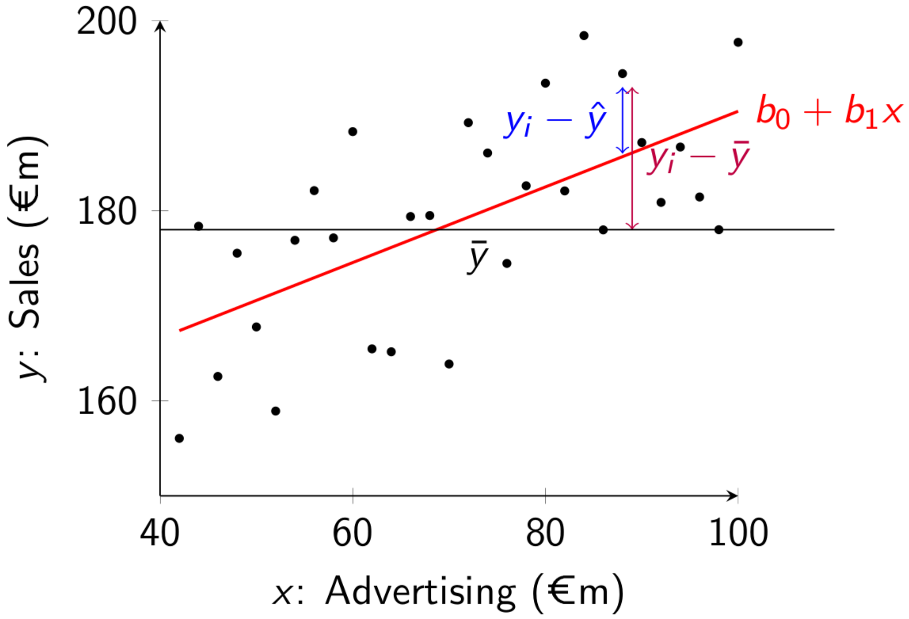
$$SST = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

This is the sum of the squared difference between the actual value of y_i and the predicted value (without the model). We call this the SST, the total sum of squares.

With a regression model, we would predict y_i using the corresponding value x_i and use $\widehat{y}_i = b_0 + b_1 x_i$ to predict y_i. As we already learned in Chapter 9, the sum of squared errors (SSE) is:

$$SSE = \sum_{i=1}^{n} (y_i - \widehat{y}_i)^2$$

Graphically the SST is the sum of squared deviations from the sample mean $\bar{y}$ (the horizontal line) and the SSE is the sum of squared deviations from $\hat{y}$ (the regression line):



12.2 Sum of Squares Due to Regression

We also define a related 3rd term, the sum of squares due to regression:

$$SSR = \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2$$

This measures the variation explained by the regression model. We will not show the steps here but it can be shown that:

$$\underbrace{\sum_{i=1}^{n} (y_i - \bar{y})^2}_{=SST} = \underbrace{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}_{=SSE} + \underbrace{\sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2}_{=SSR}$$

This means that $SST = SSE + SSR$ always.

12.3 Coefficient of Determination: R squared

The coefficient of determination, also called R squared, is given by:

$$R^2 = \frac{SSR}{SST} = 1 - \frac{SSE}{SST}$$

The R^2 is always between 0 and 1 and measures the proportion of the variation in the Y data explained by the X data:

- If R^2 is small (close to 0), the model only explains a small amount of the variation in y-data.
- If R^2 is large (close to 1), the model explains a lot of the variation in y-data.

For example, if $R^2 = 0.75$, then the model explains 75% of the variation in the y-data and 25% is left unexplained.

This can also be explained by considering the two extreme cases:

- Imagine our model was completely useless ($b_1 = 0$). Then our best predictor for y_i is the sample mean: $\hat{y}_i = \bar{y}$. In this case $SSR = 0$ and $SSE = SST$. The R^2 is then equal to $R^2 = \frac{SSR}{SST} = \frac{0}{SST} = 0$.
- Imagine our model was completely perfect and we could perfectly predict y_i with x_i. Then the residuals e_i would all be zero and the sum of squared errors would be zero ($SSE = 0$). Then the R^2 would be $R^2 = 1 - \frac{SSE}{SST} = 1 - \frac{0}{SST} = 1$.

In general we will get an R^2 in between these two extreme cases. When the R^2 is close to zero, the model is close to useless. When the R^2 is close to one, the model is very useful (close to perfect).

For the simple linear regression model, it turns out that the R^2 is the same as the square of the sample correlation coefficient $r_{X,Y}$, so $R^2 = r^2_{X,Y}$. This is why it is called the R squared.

12.4 SSE, SSR and SST in R

We can use the `anova()` function to obtain the SSR, SSR and SST in R. ANOVA here means analysis of variance.

To use this function we first need to estimate a model that tries to explain Y using only an intercept (so no X variable). We can do this in R by replacing the X variable in the `lm()` function with a `1`. Let's do this and let's call the model `m1`:

```r
df <- read.csv("advertising-sales.csv")
m1 <- lm(sales ~ 1, data = df)
summary(m1)
```

```
Call:
lm(formula = sales ~ 1, data = df)

Residuals:
    Min      1Q  Median      3Q     Max 
-12.422  -3.647  -1.123   3.377  12.977 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  14.0225     0.3689   38.01   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.217 on 199 degrees of freedom
```

We get a model with only an intercept. It turns out that this intercept is exactly the same as the sample mean:

```r
mean(df$sales)
```

```
[1] 14.0225
```

This is because if we are only using one parameter to predict Y, the best one to use is the mean.

Now we estimate our model that does include an X variable. Let's call this `m2`. We then use the `anova()` function to compare the variability of the errors before the inclusion of the regressor and afterwards:

```r
m2 <- lm(sales ~ advertising, data = df)
anova(m1, m2)
```

```
Analysis of Variance Table

Model 1: sales ~ 1
Model 2: sales ~ advertising
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)    
1    199 5417.1                                  
2    198 1338.4  1    4078.7 603.37 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

In the table the SST is 5417.1, under `RSS` for model 1. `RSS` here stands for residual sum of squares, which is another name for the sum of squared errors. Because model 1 does not include any regressors, the SST is the same as the residual sum of squares (its SSR is zero).

The SSE for model 2 (our model of interest) is under `RSS` and equals 1338.4. This is the residual sum of squares for model 2, the same as the SSE.

Finally, the SSR is the 4078.7 under `Sum of Sq` for model 2.

If all you need to get is the SSE, a faster way is to use the `deviance()` function on the regression model. We can confirm that it also gives 1338.4:

```
m <- lm(sales ~ advertising, data = df)
deviance(m)
```

```
[1] 1338.444
```

We could also just sum the squared residuals from the model as well:

```
m <- lm(sales ~ advertising, data = df)
sum(m$residuals^2)
```

```
[1] 1338.444
```

Another way to get the SST is to use the formula $SST = (n-1)\,s_y^2$. To see where this formula comes from we write the formula for the sample variance:

$$s_y^2 = \frac{\sum_{i=1}^{n}(y_i - \bar{y})^2}{n-1} = \frac{SST}{n-1}$$

Multiplying across both sides with $(n-1)$ gives the other formula for the SST. Let's test it in R:

```
(nrow(df) - 1) * var(df$sales)
```

```
[1] 5417.149
```

We get the same as above!

We can also calculate it using the formula $SST = \sum_{i=1}^{n}(y_i - \bar{y})^2$:

```
sum((df$sales - mean(df$sales))^2)
```

```
[1] 5417.149
```

Again we get the same as above.

Finally, another way to get the SSR is to calculate the SST and SSR and use the formula $SST = SSE + SSR$ to get:

$$SSR = SST - SSE$$

Let's confirm that also gives the same answer:

```
sst <- sum((df$sales - mean(df$sales))^2)
sse <- sum(m$residuals^2)
ssr <- sst - sse
ssr
```

```
[1] 4078.705
```

12.5 R^2 in R

The R^2 is shown in the standard `summary()` output after `Multiple R-squared`:

```
summary(m)
```

```
Call:
lm(formula = sales ~ advertising, data = df)

Residuals:
    Min      1Q  Median      3Q     Max 
-8.0546 -1.3071  0.1173  1.5961  7.1895 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 4.243028   0.438525   9.676   <2e-16 ***
advertising 0.048688   0.001982  24.564   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.6 on 198 degrees of freedom
Multiple R-squared:  0.7529,	Adjusted R-squared:  0.7517 
F-statistic: 603.4 on 1 and 198 DF,  p-value: < 2.2e-16
```

The R^2 is 0.7529.

But we can also obtain the number directly with:

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

```
[1] 0.7529246
```

The advertising data explains 75.29% of the variation in the sales data.

Finally, to show that we can get the same using $R^2 = 1 - \frac{SSE}{SST}$ we also calculate the R^2 manually:

```
sse <- sum(m$residuals^2)
sst <- sum((df$sales - mean(df$sales))^2)
1 - sse / sst
```

```
[1] 0.7529246
```

We get 0.7529 just like above.

Chapter 13

SLR Prediction Intervals

13.1 Theory

Before we learned how to see what Y the model predicted for each value of X in the data. This was the predicted value:

$$\hat{y}_i = b_0 + b_1 x_i$$

But we can also use the model to predict a value of Y for any value of X, not only values of X in our data.

Suppose we wanted to predict what value Y would be if the independent variable was equal to x_p, some value that we choose (and know). Call this value Y_p.

The population model says that:

$$Y_p = \beta_0 + \beta_1 x_p + \varepsilon_p$$

There are two different objects we may be interested in from this model:

1. An estimate of $\mathbb{E}\left[Y_p | x_p\right]$, the expected value of the dependent variable when the independent variable is equal to x_p.
2. A prediction of Y_p, our best prediction of the value of the dependent variable when the independent variable is equal to x_p.

In our sales and advertising example, the first object could be the average amount of sales if advertising was equal to x_p (not in any particular location; just the average), whereas the second object is the actual value of sales in one location if advertising was set at x_p.

Now, it turns out that the sample statistic $\hat{Y}_p = B_0 + B_1 x_p$ is both the point estimator of $\mathbb{E}\left[Y_p | x_p\right]$ (the first object) and the point predictor of Y_p (the second object).

However, the standard errors for these two estimators will be different:

1. The 95% confidence interval for $\mathbb{E}\left[Y_p|x_p\right]$ should contain the expected value of Y_p given x_p with 95% probability.
2. The 95% prediction interval for Y_p should contain the (still unknown) actual realization of Y_p with 95% probability.

The first object $\mathbb{E}[Y_p|x_p] = \beta_0 + \beta_1 x_p$ does not contain ε_p, whereas $Y_p = \beta_0 + \beta_1 x_p + \varepsilon_p$ does. So the prediction interval for Y_p (which includes the variability in ε_p) should be much wider than the confidence interval for $\mathbb{E}\left[Y_p|x_p\right]$.

We won't discuss the different formulas for these confidence/prediction intervals because we will use R to calculate them. However it is important to be aware why one is wider than the other.

13.2 Example in R

Let's go back to our advertising and sales dataset to show an example of this. Suppose we want to predict sales if €100,000 was spent on advertising. We also want to obtain:

1. A 95% confidence intervals for the expected value of sales given this level of advertising.
2. A 95% confidence intervals for the value sales if we advertised at this level in one market.

If all we were interested in was to get the expectation $\mathbb{E}\left[Y_p|x_p\right]$ or the predicted value $\widehat{Y}_p$, we do the following. We need to make a small `data.frame` with one observation with the appropriate value for x. We then use the `predict()` function in R with our estimated regression model `m`:

```
df <- read.csv("advertising-sales.csv")
m <- lm(sales ~ advertising, data = df)
df_p <- data.frame(advertising = 100)
predict(m, df_p)
```

```
       1
9.111816
```

As we said above, the expectation $\mathbb{E}\left[Y_p|x_p\right]$ and the prediction of Y_p are estimated the same way, so both have the same value. Here, the expected sales is €9.11m if we spend €100,000 on advertising.

Now, suppose we wanted to get a 95% confidence interval for $\mathbb{E}\left[Y_p|x_p\right]$. We can get this by specifying `"confidence"` in the `interval` option in the `predict()` function. We can set the level using the `level` option:

```r
df <- read.csv("advertising-sales.csv")
m <- lm(sales ~ advertising, data = df)
df_p <- data.frame(advertising = 100)
predict(m, df_p, interval = "confidence", level = 0.95)
```

```
       fit     lwr      upr
1 9.111816 8.57622 9.647413
```

This also gives the estimate of $\mathbb{E}\left[Y_p|x_p\right]$ which is `9.111816` (€9.11m). The interpretation of this interval is as follows: We are 95% confident that with a €100,000 spend on advertising, the expected value of sales in the population is between €8.572m and €9.647m.

Now let's get a 95% prediction interval for Y_p. This is almost the same as above, where all we change is `"confidence"` to `"prediction"` in the `interval` option:

```r
df <- read.csv("advertising-sales.csv")
m <- lm(sales ~ advertising, data = df)
df_p <- data.frame(advertising = 100)
predict(m, df_p, interval = "prediction", level = 0.95)
```

```
       fit      lwr      upr
1 9.111816 3.956741 14.26689
```

The interpretation of this interval is as follows: We are 95% confident that with a €100,000 spend on advertising in a local market, the actual value of sales will be between €3.9567 and €14.2669m.

Notice how this interval is much wider than the previous interval for $\mathbb{E}\left[Y_p|x_p\right]$. This is because it also includes the variability in ε_p which is not included in the interval for $\mathbb{E}\left[Y_p|x_p\right]$.

Chapter 14

The Multiple Linear Regression Model (MLR)

In the previous chapters covering the simple linear regression (SLR) model, we studied how Y_i depends on a single variable X_i. However, Y_i may depend on multiple variables $X_{i1}, X_{i2}, \dots, X_{ik}$. For example, sales ($Y_i$) could depend on online advertising (X_{i1}) and offline advertising (X_{i2}), and each type of advertising could have a different impact on Y_i. We can allow for this with the *multiple linear regression* (MLR) model.

We model Y_i as a *linear function* of X_{i1}, X_{i2}, ..., X_{ik} and and error term:

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \dots \beta_k X_{ik} + \varepsilon_i$$

We are going to see that much of what we learned for the simple linear regression model carries directly over to this model. For example, how to obtain confidence intervals and how to perform hypothesis tests on a particular parameter β_j.

14.1 Interpretation of the Parameters

14.1.1 Slope Terms

In the simple linear regression model the regression slope was the average increase in the dependent variable from a unit increase in the independent variable. In the multiple linear regression model this interpretation changes slightly. The coefficient on the first variable β_1 is now how much the expected value of Y_i increases when x_{i1} increases by 1 unit *and all other variables remain unchanged*. This last part about all other variables remaining unchanged was not there before because in the simple linear regression model there were no other variables: there was just the one variable X_i.

The expected value of Y_i given each of the $x_{i1}, \dots x_{ik}$ is:

$$\mathbb{E}\left[Y_i | x_{i1}, x_{i2}, \dots, x_{ik}\right] = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_k x_{ik}$$

If we increase x_{i1} by one unit this becomes:

$$\begin{aligned}\mathbb{E}\left[Y_i | x_{i1}+1, x_{i2}, \dots, x_{ik}\right] &= \beta_0 + \beta_1\left(x_{i1}+1\right) + \beta_2 x_{i2} + \dots + \beta_k x_{ik} \\ \mathbb{E}\left[Y_i | x_{i1}+1, x_{i2}, \dots, x_{ik}\right] &= \beta_1 + \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_k x_{ik}\end{aligned}$$

If we subtract these we see that everything except β_1 cancels:

$$\mathbb{E}\left[Y_i | x_{i1}+1, x_{i2}, \dots, x_{ik}\right] - \mathbb{E}\left[Y_i | x_{i1}, x_{i2}, \dots, x_{ik}\right] = \beta_1$$

So how to interpret β_1 is the left-hand side of this equation: the expected change in Y_i from a unit increase in x_{i1} keep all other variables $x_{i2}, x_{i3}, \dots, x_{ik}$ fixed.

Sometimes to say "keeping all other variables fixed" we say "all else equal" or *ceteris paribus*, which is Latin for "other things equal".

We can use the same logic to interpret the coefficients in front of the other variables. For example, β_2 is the expected change in Y_i from a unit increase in x_{i2} keep all other variables $x_{i1}, x_{i3}, x_{i4}, \dots, x_{ik}$ fixed.

14.1.2 Intercept

To interpret the intercept term β_0 we note that when all variables are exactly equal to zero, $x_{i1} = x_{i2} = \dots = x_{ik} = 0$, we get:

$$\begin{aligned}\mathbb{E}\left[Y_i | x_{i1}=0, x_{i2}=0, \dots, x_{ik}=0\right] &= \beta_0 + \beta_1 \times 0 + \beta_2 \times 0 + \dots \beta_k \times 0 \\ &= \beta_0\end{aligned}$$

So β_0 is the expected value of the dependent variable when all explanatory variables take on a value of zero.

With many explanatory variables (large k), having situations where all explanatory variables equal zero simultaneously becomes increasingly rare. Thus usually the estimate of the intercept β_0 will not make much sense and we won't pay too much attention to it. But we will see some situations where it will.

14.2 Estimation of the Parameters

The parameters β_0, β_1, ..., β_k are estimated by minimizing the sum of squared errors like in the simple linear regression model.

The estimates $b_0, b_1, b_2, \dots, b_k$ that we get are the ones that make the term below as small as possible:

$$\sum_{i=1}^{n} \left(y_i - b_0 - b_1 x_{i1} - b_2 x_{i2} - \dots - b_k x_{ik}\right)^2$$

The mathematical formulas for $b_0, b_1, b_2, \dots, b_k$ involve using matrix algebra so we will not show the formulas for the estimator here. Instead we will use R to estimate the model as in the example in the next subsection.

Just like the simple linear regression model, after estimation we obtain the sample regression line:

$$\hat{y}_i = b_0 + b_1 x_{i1} + \cdots + b_k x_{ik}$$

where $\hat{y}_i$ are the predicted values and e_i are the residuals.

14.3 Example in R

We will now show an example in R. We will move away from the sales and advertising example dataset because that only has one explanatory variable (advertising). We will instead use the dataset `wages1.csv` which contains data on the hourly wage in dollars, years of education, and years of work experience for $n = 526$ people. The data are from the National Longitudinal Survey in the US. We will estimate a model explaining wage (Y) with education (X_1) and experience (X_2).

Estimating the model is almost the same as with the simple linear regression model. The only thing that changes is that we add more explanatory variables to the formula in the `lm()` function using the plus symbol `+`.

```r
df <- read.csv("wages1.csv")
m <- lm(wage ~ educ + exper, data = df)
summary(m)
```

```
Call:
lm(formula = wage ~ educ + exper, data = df)

Residuals:
    Min      1Q  Median      3Q     Max 
-5.5532 -1.9801 -0.7071  1.2030 15.8370 

Coefficients:
            Estimate Std. Error t value       Pr(>|t|)    
(Intercept) -3.39054    0.76657  -4.423 0.000011846645 ***
educ         0.64427    0.05381  11.974        < 2e-16 ***
exper        0.07010    0.01098   6.385 0.000000000378 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.257 on 523 degrees of freedom
```

```
Multiple R-squared:  0.2252,    Adjusted R-squared:  0.2222
F-statistic: 75.99 on 2 and 523 DF,  p-value: < 2.2e-16
```

If we had more variables, we would just add these variables separating each by the plus symbol. For example: `y ~ x1 + x2 + x3 + x4`. We will see examples of this in the upcoming chapters.

The sample regression line in our example is:

$$\hat{y}_i = -3.39 + 0.64 x_{i1} + 0.07 x_{i2}$$

Let's interpret each of these numbers.

The model predicts that an individual with zero years of education and zero years of experience will have an hourly wage of $-\$3.39$. This doesn't make much sense: who would work for a negative wage? If we check the data, neither the variable `educ` nor `exper` have values that equal zero:

```
summary(df)
```

```
      wage             educ           exper      
 Min.   : 0.530   Min.   : 0.00   Min.   : 1.00  
 1st Qu.: 3.330   1st Qu.:12.00   1st Qu.: 5.00  
 Median : 4.650   Median :12.00   Median :13.50  
 Mean   : 5.896   Mean   :12.56   Mean   :17.02  
 3rd Qu.: 6.880   3rd Qu.:14.00   3rd Qu.:26.00  
 Max.   :24.980   Max.   :18.00   Max.   :51.00  
```

For `educ` the smallest value is 9. Because we need (several) observations with values $x_{i1} = x_{i2} = 0$ for b_0 to be reliable, we cannot trust this estimate here.

We now move on to interpreting the coefficients in front of the explanatory variables. All else equal, increasing an individual's education by 1 year while holding experience fixed increases the wage by \$0.64 on average. All else equal, increasing an individual's experience by 1 year while holding education fixed increases the wage by \$0.07 on average.

14.4 Adding and Removing Variables

Suppose now we used the same dataset as above to estimate a model explaining wage with education only, leaving experience out of the model. We use the approach we used with the simple linear regression model:

```
lm(wage ~ educ, data = df)
```

```

Call:
lm(formula = wage ~ educ, data = df)
```

```
Coefficients:
(Intercept)         educ
    -0.9049       0.5414
```

Now let's compare the two sample regression equations, the model with experience included and with experience excluded:

$$\hat{y}_i = -3.39 + 0.64x_{i1} + 0.07x_{i2}$$
$$\hat{y}_i = -0.90 + 0.54x_{i1}$$

In the first model, increasing education by 1 year on average increased wages by \$0.64 holding experience fixed. In the second model, increasing education by 1 year on average increased wages by \$0.54 (without holding experience fixed).

The effect of education on wages is *smaller* in the model without experience. Increasing education by 1 year now only increases wages by \$0.54 on average. Wages depend on experience, so in the simple model experience is included in ε_i. But education and experience are negatively correlated:

```r
cor(df$educ, df$exper)
```

```
[1] -0.2995418
```

When education is higher for someone that often means they spent more time in school/college and got less experience. So when we increase education for someone and *not* hold experience fixed, it has a smaller effect on wages because that usually means that person has less experience. Thus in the simpler model we have a violation of the $\mathbb{E}[\varepsilon_i|X_i] = 0$ assumption. The error term which includes experience is negatively correlated with the education variable. The negative correlation biases the estimates of β_1 downward. This kind of bias is called *omitted variable bias*.

For this reason we prefer models that include more variables that can impact the Y variable that are correlated with our X variables of interest.

Chapter 15

MLR Model Assumptions

We now discuss the model assumptions that we require to perform inference, which we will discuss for the single variable case in the next chapter (Chapter 16). These assumptions are almost the same as the simple linear regression model, except for assumption 3. For completeness we go through each individual assumption again here.

15.1 Assumption 1: Linear in Parameters

Assumption 1: Linear in Parameters

In the population model, the dependent variable Y_i is related to the independent variables $X_{i1}, \dots, X_{ik}$ and the error ε_i according to:

$$Y_i = \beta_0 + \beta_1 X_{i1} + \cdots + \beta_k X_{ik} + \varepsilon_i$$

Again, this assumption means that the process that generates the data in our sample follows this model. That is, Y_i is linear in $X_{i1}, X_{i2}, \dots, X_{ik}$ and the values Y_i are generated according to the model.

15.2 Assumption 2: Random Sampling

Assumption 2: Random Sampling

We have a random sample of size n, $((x_{11}, \dots, x_{1k}, y_1), \dots (x_{n1}, \dots, x_{nk}, y_n))$ following the population model in Assumption 1.

This assumption means that the sample of data we observe were generated according to the model $Y_i = \beta_0 + \beta_1 X_{i1} + \dots + \beta_k X_{ik} + \varepsilon_i$. The values of y_i that we observe are related to the unknown population parameters, observed $x_{i1}, \dots, x_{ik}$ and the unobserved error ε_i according to $\beta_0 + \beta_1 x_{i1} + \dots + \beta_k x_{ik} + \varepsilon_i$, where ε_i is independent across observation i.

15.3 Assumption 3: No Perfect Collinearity

This assumption is now different from the SLR model:

> Assumption 2: Random Sampling
>
> In the sample, none of the independent variables are constant and there are no *exact linear* relationships among the independent variables.

The first part of this assumption is the same as before, holding for each individual x variable. It requires each variable in the regression to have a standard deviation greater than zero.

The second part means that we should not be able to write one of the variables as a linear function of one (or more) of the other variables, holding exactly for every observation.

We will explain this second part using an example dataset. We will use dataset clothing-exp.csv which contains data on a random sample of households with the following variables:

- `clothing_exp`: Annual clothing expenditure of the household (in €000).
- `hh_exp`: Annual household income household (in €000).
- `num_kids`: Number of children in the household.
- `hh_size`: Total number of people in the household.

Let's estimate a regression model trying to explain clothing expenditure with the household size, the number of children and the total number of people in the household:

```r
df <- read.csv("clothing-exp.csv")
m <- lm(clothing_exp ~ hh_inc + num_kids + hh_size, data = df)
summary(m)
```

```

Call:
lm(formula = clothing_exp ~ hh_inc + num_kids + hh_size, data = df)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.27225 -0.05878 -0.00765  0.05767  0.43981 
```

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.0125930  0.0232879  -0.541    0.589
hh_inc       0.0822021  0.0004423 185.861   <2e-16 ***
num_kids     0.0108057  0.0137232   0.787    0.432
hh_size      0.0119808  0.0116495   1.028    0.305
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1031 on 296 degrees of freedom
Multiple R-squared:  0.9921,    Adjusted R-squared:  0.9921
F-statistic: 1.246e+04 on 3 and 296 DF,  p-value: < 2.2e-16
```

For practice, let's interpret the coefficients from the sample regression equation:

$$\hat{y}_i = -0.01259 + 0.08220x_{i1} + 0.01081x_{i2} + 0.01198x_{i3}$$

The estimate of the intercept (b_0) says that a household with zero income and nobody in it spends on average −€12.59 per year on clothing. Let's check the summary statistics of the explanatory variables:

```
summary(df)
```

```
  clothing_exp       hh_inc         num_kids         hh_size     
 Min.   :0.910   Min.   :10.90   Min.   :0.0000   Min.   :1.000  
 1st Qu.:1.677   1st Qu.:20.50   1st Qu.:0.0000   1st Qu.:2.000  
 Median :2.270   Median :27.33   Median :0.0000   Median :2.000  
 Mean   :2.531   Mean   :30.43   Mean   :0.8733   Mean   :2.743  
 3rd Qu.:3.085   3rd Qu.:36.82   3rd Qu.:2.0000   3rd Qu.:4.000  
 Max.   :6.690   Max.   :83.38   Max.   :5.0000   Max.   :7.000  
```

Household income and household size are never zero in the data. Because we don't have $x_{i1} = x_{i2} = x_{i3} = 0$ for any observation, this estimate is not reliable. It also doesn't make much sense either, because an unoccupied house does not have anyone in it to buy clothes (especially if they have no income!).

For b_1, increasing household income by €1,000, holding family composition fixed, increases clothing expenditure by €82.20 on average. For b_2, increasing the number of children by 1, holding income and the total household size fixed (i.e. replacing an adult with a child), increases clothing expenditure by €10.81 on average. For b_3, increasing the household size by 1, holding income and the number of children fixed (i.e. adding an adult), increases clothing expenditure by €11.98 on average.

Suppose now we wanted to create a new variable to add to this model: the number of adults. We can create this variable in R by subtracting the number of children from the total household size. Let's try this:

```r
df$num_adults <- df$hh_size - df$num_kids
m <- lm(clothing_exp ~ hh_inc + num_kids + hh_size + num_adults, data = df)
summary(m)
```

```

Call:
lm(formula = clothing_exp ~ hh_inc + num_kids + hh_size + num_adults, 
    data = df)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.27225 -0.05878 -0.00765  0.05767  0.43981 

Coefficients: (1 not defined because of singularities)
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -0.0125930  0.0232879  -0.541    0.589    
hh_inc       0.0822021  0.0004423 185.861   <2e-16 ***
num_kids     0.0108057  0.0137232   0.787    0.432    
hh_size      0.0119808  0.0116495   1.028    0.305    
num_adults          NA         NA      NA       NA    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1031 on 296 degrees of freedom
Multiple R-squared:  0.9921,	Adjusted R-squared:  0.9921 
F-statistic: 1.246e+04 on 3 and 296 DF,  p-value: < 2.2e-16
```

Notice that we don't get an estimate for `num_adults`. This is because of perfect collinearity. It's possible to write: $x_{i4} = x_{i3} - x_{i2}$ for all i which means there is an *exact linear relationship* between some of the independent variables.

To satisfy assumption 3 we should not be able to write one variable as a linear function of other explantory variables with the relationship holding exactly for every observation in the dataset.

15.4 Assumption 4: Zero Conditional Mean

Assumption 4: Zero Conditional Mean

The error ε_i has an expected value of zero given any value of the explanatory variables, i.e. $\mathbb{E}\left[\varepsilon_i | X_{i1}, \dots, X_{ik}\right] = 0$ for all $X_{i1}, \dots, X_{ik}$.

This assumption, like before, implies that the error term cannot be correlated with any of the explanatory variables. It also rules out any nonlinear relationships.

15.5 Assumption 5: Homoskedasticity

Assumption 5: Homoskedasticity

The error ε_i has the same variance given any value of the explanatory variables. In other words:

$$\mathrm{Var}\left(\varepsilon_i | x_{i1}, \ldots, x_{ik}\right) = \sigma_\varepsilon^2$$

Just like in the SLR model, this means that the dispersion of the error terms should not vary with any of the explanatory variables.

15.6 Assumption 6: Normality

Assumption 6: Normality

The distribution of ε_i conditional on $x_{i1}, \ldots, x_{ik}$ is normally distributed.

This assumption, combined with assumptions 4 and 5 implies:

$$\varepsilon_i | x_{i1}, \ldots, x_{ik} \sim \mathcal{N}\left(0, \sigma_\varepsilon^2\right)$$

In words: ε_i conditional on $x_{i1}, \ldots, x_{ik}$ follows a normal distribution with a zero mean and variance σ_ε^2.

Chapter 16

MLR Inference on a Single Variable

In Chapter 15 we discussed the model assumptions for the multiple linear regression model. When all of these assumptions hold we are able to perform inference. In this chapter we will discuss inference on a single variable: how to obtain confidence intervals and how to perform hypothesis tests on one variable. We will see that this is very similar to the SLR model.

16.1 Model Variance

To obtain standard errors for the regression coefficients, we first require an estimate of the model variable σ_ε^2.

The sum of squared errors SSE is the same as before:

$$SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} e_i^2$$

To obtain the sample variance of the estimated model, s_ε^2, we use the formula:

$$s_\varepsilon^2 = \frac{SSE}{n-k-1}$$

Notice that now we divide by $n - k - 1$ instead of $n - 2$ in the simple linear regression model. Because we are now estimating $k + 1$ parameters (the k coefficients on the variables plus the intercept) we have only $n-(k+1) = n-k-1$ degrees of freedom.

The simple linear regression model is a special case of the multiple linear regression model when $k = 1$. So $n - k - 1 = n - 1 - 1 = n - 2$, like what we had in the simple linear regression model.

The *standard error of the estimated model* is then $s_\varepsilon = \sqrt{s_\varepsilon^2}$.

16.2 Confidence Intervals

Obtaining a confidence interval for the multiple linear regression model is very similar to obtaining one for the simple linear regression model.

The formula for the confidence interval for b_j is:

$$b_j \pm t_{1-\frac{\alpha}{2}, n-k-1} \times s_{b_j}$$

where j is one of the variables $1, \dots, k$. We will not write the formula for s_{b_j} here, but will calculate it in R. The only difference is we use the Student's t distribution with $n-k-1$ degrees of freedom instead of $n-2$.

Obtaining the confidence interval is also the same in R and we interpret it the same way. If we have a 95% confidence interval $\left[L_j, U_j\right]$ around b_j we say "we are a 95% confident that the population β_j is between L_j and U_j."

Let's show a quick example in R using the wages, education and experience data. If we want to get a 95% confidence interval around b_1, we do:

```r
df <- read.csv("wages1.csv")
m <- lm(wage ~ educ + exper, data = df)
confint(m, "educ", level = 0.95)
```

```
         2.5 %    97.5 %
educ 0.5385695 0.7499747
```

We are 95% confident that the average impact on wages of one additional year of education is between \$0.54 and \$0.75 holding experience fixed.

16.3 Hypothesis Testing

Hypothesis tests for individual parameters are also done the same way as with the simple linear regression model. The only difference again is that we use $n-k-1$ degrees of freedom instead of $n-2$ when finding the quantiles of the t distribution and finding p-values.

We will do an example with the wages, education and experience data. Suppose you want to test the claim that increasing your experience by 1 year on average increases your wage by more than \$0.05, holding education fixed. You will use $\alpha = 0.05$.

The model is:

$$\mathbb{E}[Y_i | x_{i1}, x_{i2}] = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2}$$

with x_{i1} being education and x_{i2} being experience. The claim is equivalent to testing if $\beta_2 > 0.05$.

The null and alternative hypotheses are then:

$$\begin{aligned} H_0 &: \beta_2 \leq 0.05 \\ H_1 &: \beta_2 > 0.05 \end{aligned}$$

Recall that the claim is the alternative hypothesis and the null hypothesis is the opposite to the alternative. It is therefore usually helpful to write down the alternative hypothesis first.

Under the null hypothesis the test statistic $T = \frac{B_2 - 0.05}{S_{B_2}}$ follows a t distribution with $n - k - 1$ degrees of freedom.

We can use R to calculate the value of the test statistic:

```r
df <- read.csv("wages1.csv")
m <- lm(wage ~ educ + exper, data = df)
b_2 <- coef(summary(m))["exper", "Estimate"]
s_b_2 <- coef(summary(m))["exper", "Std. Error"]
t <- (b_2 - 0.05) / s_b_2
t
```

```
[1] 1.830576
```

If we are using the critical value approach we check if $t \geq t_{1-\alpha,n-k-1}$. Let's calculate this in R:

```r
qt(0.95, m$df.residual)
```

```
[1] 1.647772
```

Notice that I used `m$df.residual` to get the degrees of freedom. This number stored in the model output always contains the number $n - k - 1$. Let's check that this matches what we would get if we wanted to get $n - k - 1$ manually. We can get the number of observations (number of rows in our dataset) minus the number of regressors (2) minus 1:

```r
nrow(df) - 2 - 1
```

```
[1] 523
```

```r
m$df.residual
```

```
[1] 523
```

Both give 523. However, using `m$df.residual` is more reliable because if the dataset contains missing observations then the number of rows of `df` does not equal the number of observations n used in the regression.

Because the test statistic 1.831 is greater than the critical value 1.648 we reject the null hypothesis.

We can also do this using the p-value method. To get the p-value in R we do:

```
1 - pt(t, m$df.residual)
```

```
[1] 0.03386635
```

The p-value is less than the significane level $\alpha = 0.05$ so we reject the null hypothesis. This is the same result as with the critical value approach.

To conclude, because we reject the null hypothesis there is sufficient evidence for the claim that increasing your experience by one year holding education fixed increases your wage by more than \$0.05 on average.

Here is a summary of the R functions we use for hypothesis testing in the MLR model. First define the following:

- The size of the test, α, is `alpha`.
- The regression is stored as `m` so that `m$df.residual` equals $n - k - 1$.
- The value of the test statistic, t, is `t`.

Critical Values:

- If upper-tail test: `qt(1-alpha, m$df.residual)`.
- If lower-tail test: `qt(alpha, m$df.residual)`.
- If two-sided test: `qt(1-alpha/2, m$df.residual)`.

p-values:

- If upper-tailed test: `1-pt(t, m$df.residual)`.
- If lower-tailed test: `pt(t, m$df.residual)`.
- If two-sided test: `2*(1-pt(abs(t), m$df.residual))`.

16.4 Statistical Significance

Finally, just like with the simple linear regression model, the most common hypothesis test for the multiple linear regression model is the test for statistical significance:

$$H_0 : \beta_j = 0 \text{ vs } H_1 : \beta_j \neq 0$$

Under H_0, the variable j is useless within the model:

$$\mathbb{E}\left[Y_i | x_{i1}, \dots, x_{ik}\right] = \beta_0 + \beta_1 x_{i1} + \dots + 0 \cdot x_{ij} + \dots + \beta_k x_{ik}$$

That is, under H_0, variable j does not contribute to explaining Y_i.

In contrast, under H_1, the variable j is useful within the model. If we reject H_0, β_j is statistically different from zero. We say variable j is individually statistically significant.

Thus how we interpret it is slightly different to the SLR model. There we said that the model was useful (because there was only one variable), whereas here we say that an individual variable is useful within the model. Later in Chapter 19 we will learn how to test for model usefulness more generally in the MLR model.

To check for statistical significance in R we can use the `summary()` function and quickly check the p-values of the included regressors:

```r
df <- read.csv("wages1.csv")
m <- lm(wage ~ educ + exper, data = df)
summary(m)
```

```

Call:
lm(formula = wage ~ educ + exper, data = df)

Residuals:
    Min      1Q  Median      3Q     Max 
-5.5532 -1.9801 -0.7071  1.2030 15.8370 

Coefficients:
            Estimate Std. Error t value       Pr(>|t|)    
(Intercept) -3.39054    0.76657  -4.423 0.000011846645 ***
educ         0.64427    0.05381  11.974        < 2e-16 ***
exper        0.07010    0.01098   6.385 0.000000000378 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.257 on 523 degrees of freedom
Multiple R-squared:  0.2252,	Adjusted R-squared:  0.2222 
F-statistic: 75.99 on 2 and 523 DF,  p-value: < 2.2e-16
```

If we see that the p-values are below our desired significance level (such as 0.05) we say that variable is individually statistically significant. In this model both variables are individually significant. We can also see this by looking at the `***` next to the p-values.

Chapter 17

MLR Quantifying Model Usefulness

The formulas for SSE, SSR, SST and R^2 are also the exact same as the SLR model. We show here how to calculate them in R.

17.1 SSE, SSR, SST

$$SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$
$$SSR = \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2$$
$$SST = \sum_{i=1}^{n} (y_i - \bar{y})^2$$
$$R^2 = SSR/SST = 1 - SSE/SST$$

We also calculate them in R using the approaches we saw in Chapter 12. Let's show how to do this with the wages, education and experience model.

We first estimate a model using only an intercept and call it `m1`. We then estimate our full model and call it `m2`. We then use the `anova()` function to compare the two models:

```
df <- read.csv("wages1.csv")
m1 <- lm(wage ~ 1, data = df)
m2 <- lm(wage ~ educ + exper, data = df)
anova(m1, m2)
```

```
Analysis of Variance Table

Model 1: wage ~ 1
Model 2: wage ~ educ + exper
  Res.Df    RSS Df Sum of Sq     F    Pr(>F)    
1    525 7160.4                                  
2    523 5548.2  2    1612.2 75.99 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The SST is 7160.4, the SSE is 5548.2 and the SSR is 1612.2.

We can also use the other approaches we saw in Chapter 12:

```
m <- lm(wage ~ educ + exper, data = df)
```

For the SST we can do either:

```
(nrow(df) - 1) * var(df$wage)
```

```
[1] 7160.414
```

```
sum((df$wage - mean(df$wage))^2)
```

```
[1] 7160.414
```

For the SSE we can do either:

```
deviance(m)
```

```
[1] 5548.16
```

```
sum(m$residuals^2)
```

```
[1] 5548.16
```

For the SSR we can use the above results to get:

```
sst <- (nrow(df) - 1) * var(df$wage)
sse <- deviance(m)
ssr <- sst - sse
ssr
```

```
[1] 1612.255
```

In each case we get the same numbers as the `anova()` function.

17.2 R^2

The R^2 is also shown in the default `summary()` output:

```
df <- read.csv("wages1.csv")
m <- lm(wage ~ educ + exper, data = df)
summary(m)
```

```

Call:
lm(formula = wage ~ educ + exper, data = df)

Residuals:
    Min      1Q  Median      3Q     Max 
-5.5532 -1.9801 -0.7071  1.2030 15.8370 

Coefficients:
            Estimate Std. Error t value       Pr(>|t|)    
(Intercept) -3.39054    0.76657  -4.423 0.000011846645 ***
educ         0.64427    0.05381  11.974        < 2e-16 ***
exper        0.07010    0.01098   6.385 0.000000000378 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.257 on 523 degrees of freedom
Multiple R-squared:  0.2252,	Adjusted R-squared:  0.2222 
F-statistic: 75.99 on 2 and 523 DF,  p-value: < 2.2e-16
```

The R^2 is 0.2252. In our model education and experience explain 22.52% of the variation in the wages data. The remaining 77.48% remains unexplained.

We can also extract this number from the output with:

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

```
[1] 0.2251622
```

One thing to note is that the R^2 is no longer the square of the sample correlation coefficient. That is only true for the simple linear regression.

17.3 Adjusted R^2

Recall that the formula for the R squared is $R^2 = 1 - \frac{SSE}{SST}$ and measures the % of the variation in the y-data that is explained by the independent variables. If we add more and more variables to our model, the sum of squared errors always falls with each variable added and so will always increase the R^2. This could lead us to add *too many* variables to our model (a problem called “overfitting”).

You may have noticed that the `summary()` output also gives another number called the `Adjusted R-squared`. In our example it is 0.2222. This adjusted R squared is one way to help us building models to avoid this overfitting problem.[1]

The formula for the adjusted R^2 is:

$$R^2_{adj} = 1 - \frac{SSE/\left(n-k-1\right)}{SST/\left(n-1\right)}$$

The adjusted R^2 will decrease if adding a new variable does not explain much of the variation in the y-data.

If we want to extract the adjusted R^2 from the R output we can use the command:

```r
summary(m)$adj.r.squared
```

```
[1] 0.2221991
```

The adjusted R^2 is always smaller than the ordinary R squared and can be negative if the explanatory power of the model is very poor.

[1] If we wanted to estimate a model $\mathbb{E}[Y_i|x_{i1}, x_{i2}] = \beta_0 + \beta_1(x_{i1} + x_{i2})$, i.e. a simple linear regression model with Y_i explained by the sum of x_{i1} and x_{i2} we can’t just do `lm(y ~ x1 + x2, data = df)`. This is because this would actually estimate the model $\mathbb{E}[Y_i|x_{i1}, x_{i2}] = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2}$. To “inhibit” R from “interpreting” the `+` as adding a new variable we can use the `I()` function (the “inhibit interpretation” function). We would use it like this: `lm(y ~ I(x1 + x2), data = df)`.

Chapter 18

MLR Prediction Intervals

Just like we saw in Chapter 13, with chosen values $x_{p1}, \ldots, x_{pk}$ for each of the independent variables, we can use our regression model to estimate both the expected value of the dependent variable at those values $\mathbb{E}\left[Y_p|x_{p1}, \ldots, x_{pk}\right]$ and make a prediction of the realized value of Y_p. We can also obtain a confidence interval for $\mathbb{E}\left[Y_p|x_{p1}, \ldots, x_{pk}\right]$ and a prediction interval for Y_p. Doing these in R is very similar to how we did it for the simple linear regression model in Chapter 13. We will show examples of these using the wages, education and experience data.

18.1 Confidence Interval for $\mathbb{E}\left[Y_p|x_{p1}, \ldots, x_{pk}\right]$

You want to estimate the average wage of people with 12 years of education and 13 years of experience and also obtain a 95% confidence interval for that estimate.

Just like in Chapter 13 we perform the following steps:

1. Estimate the regression model.
2. Create a `data.frame` with one row containing the values for each of the independent variables.
3. Use the `predict()` function with the estimated model and this one-row `data.frame`, specifying whether you want a confidence or prediction interval and the level of confidence.

Here are the steps for our example:

```r
df <- read.csv("wages1.csv")
m <- lm(wage ~ educ + exper, data = df)
df_p <- data.frame(educ = 12, exper = 13)
```

```
predict(m, df_p, interval = "confidence", level = 0.95)
```

```
       fit      lwr      upr
1 5.251966 4.948709 5.555222
```

The model estimates that the average wage of people with 12 years of education and 13 years of experience is \$5.25.

To interpret the confidence interval we say that we are 95% confident that the population average wage of people with 12 years of education and 13 years of experience is between \$4.95 and \$5.56.

18.2 Confidence Interval for Y_p given \$x_{p1}, ..., x_{pk} \$

Now you want to predict the wage of an individual with 12 years of education and 13 years of experience and obtain a 95% confidence interval for that prediction.

We follow almost the same steps as before, but now we use the `"prediction"` option for `interval` in the `predict()` function instead of `"confidence"`:

```
df <- read.csv("wages1.csv")
m <- lm(wage ~ educ + exper, data = df)
df_p <- data.frame(educ = 12, exper = 13)
predict(m, df_p, interval = "prediction", level = 0.95)
```

```
       fit       lwr      upr
1 5.251966 -1.153713 11.65764
```

The model predicts that the wage of an individual with 12 years of education and 13 years of experience is \$5.25.

We are 95% confident that for any individual drawn from the population 12 years of education and 13 years of experience will have a wage between $-\$1.15$ and \$11.66.

Notice that the prediction is the same as the estimate of $\mathbb{E}\left[Y_p|x_{p1}=12, x_{p2}=13\right]$ but the confidence interval is much wider. This is because we are more uncertain about the wage of one individual (which contains the variability of the error) compared to the average wage (where the errors are averaged out across individuals). The lower bound of this confidence interval is even negative! The upper bound is also very large in the distribution of wages:

```
summary(df$wage)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  0.530   3.330   4.650   5.896   6.880  24.980 
```

We can check what quantile the upper bound is in:

```r
mean(df$wage < 11.65764)
```

```
[1] 0.9220532
```

This means that our prediction interval is extremely wide. We are 95% confident that the wage of this individual will be between \$0 (smaller than the lowest observed wage in the data) and \$11.66 (larger than 92.2% of observed wages in the data).

Chapter 19

F-test

In Chapter 11 we learned how to test for model usefulness for the simple linear regression model. In Chapter 16 (Section 16.4) we learned how to test the usefulness of individual variables in the multiple linear regression model. In this chapter we will learn how to test for the usefulness of the model as a whole in the multiple linear regression model.

To do this we are going to use an F-test which makes use of the F distribution.

19.1 F-Test Theory

We want to test if the whole model is useful or not:

$$\begin{aligned}
&H_0 : \beta_1 = \beta_2 = \cdots = \beta_k = 0 \\
&H_1 : \text{at least one of } \beta_j \neq 0 \text{ for } j = 1, \ldots, k
\end{aligned}$$

Under H_0, the whole model is useless. Under H_1, there is at least one useful variable in the model. Notice that these only include the parameters in front of regressors and not the intercept β_0.

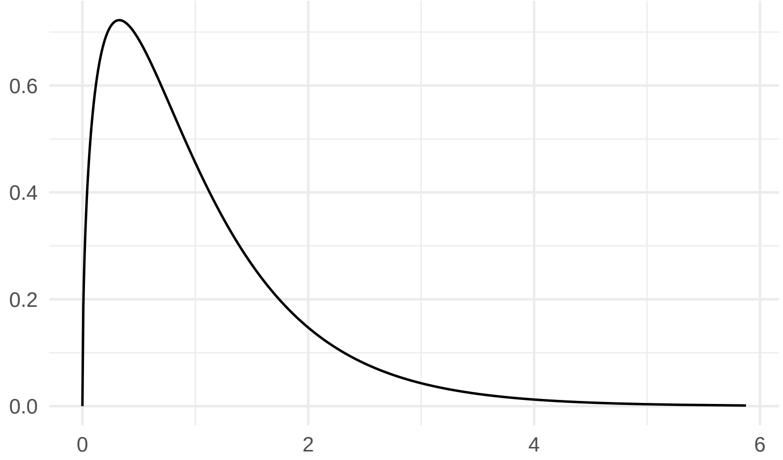
Under the null hypothesis, the test statistic:

$$F = \frac{SSR/k}{SSE/(n-k-1)}$$

follows an F distribution with k numerator and $n-k-1$ denominator degrees of freedom. We use $F_{k,n-k-1}$ to denote this distribution.

Let's take a look at what $F_{k,n-k-1}$ looks like. For $k = 3$ and $n = 100$, the density of the distribution looks like this:

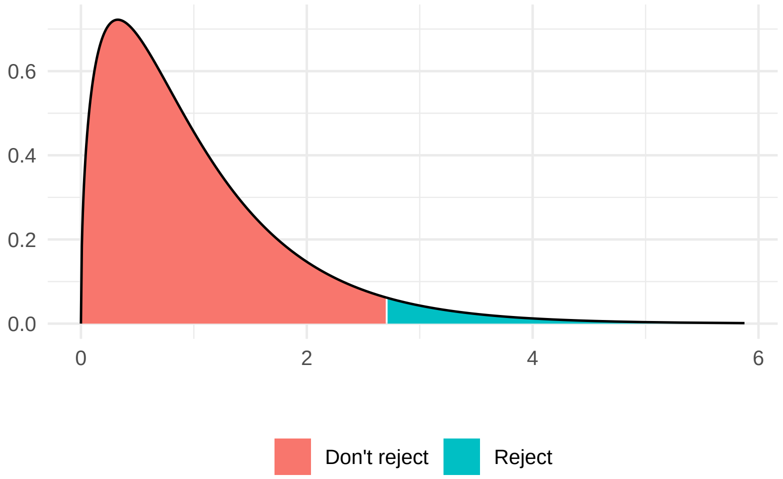
```
library(ggplot2)
k <- 3
n <- 100
df <- data.frame(x = qf(seq(0.000, 0.999, by = 0.001), k, n - k - 1))
df$y <- df(df$x, k, n - k - 1)
ggplot(data = df, aes(x = x, y = y)) +
  geom_line() +
  xlab("") +
  ylab("") +
  theme_minimal()
```



This means that if the null hypothesis is true, then the F-ratio $F = \frac{SSR/k}{SSE/(n-k-1)}$ from samples drawn from the population should usually be less then 2.5 because that's where the bulk of the mass of the distribution is. If in our realized sample we see a value of F larger than, say, 3, then that is something that is very rare under the null hypothesis. If we find that the F we get in our model is large, then it is less likely that we just happened to observe a very extreme sample drawn from the population and more likely that that the null hypothesis is false. That is, it is unlikely that all $\beta_1 = \beta_2 = \cdots = \beta_k = 0$ and that the model is useful.

If we are using a critical value approach with a 5% level then we find the point in the distribution with 95% of the area to the left and 5% of the area to the right. This point turns out to be at 2.6994. So if we find the F-ratio in our sample to be bigger than 2.6994 we reject the null hypothesis and conclude that our model is useful. If we find it to be smaller than 2.6994 then we say that we have insufficient evidence to suggest our model is useful. We show the rejection region and non-rejection region in the same plot:

```r
library(ggplot2)
k <- 3
n <- 100
df <- data.frame(x = qf(seq(0.000, 0.999, by = 0.001), k, n - k - 1))
df$y <- df(df$x, k, n - k - 1)
df$reject <- ifelse(df$x > qf(0.95, k, n - k - 1), "Reject", "Don't reject")
ggplot(data = df) +
  geom_ribbon(aes(x = x, ymin = 0, ymax = y, fill = reject)) +
  geom_line(aes(x = x, y = y)) +
  xlab("") +
  ylab("") +
  theme_minimal() +
  theme(legend.title = element_blank(), legend.position = "bottom")
```

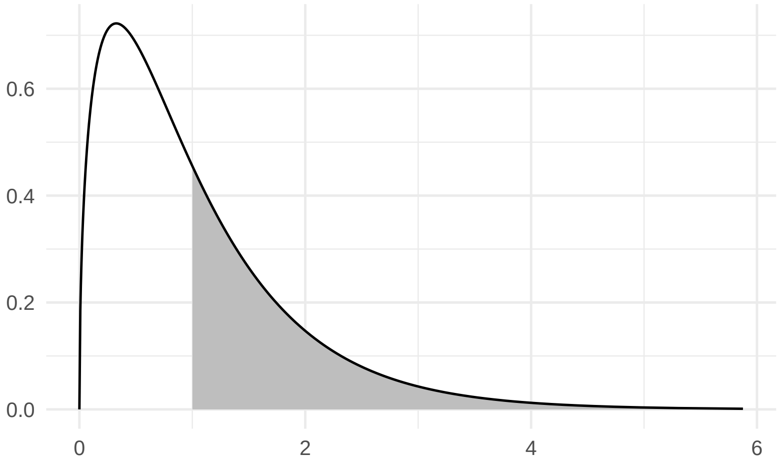


If our sample is in the red area we don’t reject; if it is in the blue area we do. Notice that the test is a one-sided test.

The p-value is the probability of obtaining a sample at least as extreme as the observed sample. It is the area under the distribution to the right of the observed F-ratio. If we obtained an F-ratio of 1 in our sample, the p-value would be the area to the right of 1, which is equal to 0.396 and indicated by the gray area in the figure below:

```r
library(ggplot2)
k <- 3
```

```
n <- 100
df <- data.frame(x = qf(seq(0.000, 0.999, by = 0.001), k, n - k - 1))
df$y <- df(df$x, k, n - k - 1)
df$fill <- df$x > 1
ggplot() +
  geom_ribbon(data = df[df$x > 1,], aes(x = x, ymin = 0, ymax = y),
              fill = "gray") +
  geom_line(data = df, aes(x = x, y = y)) +
  xlab("") +
  ylab("") +
  theme_minimal() +
  theme(legend.title = element_blank(), legend.position = "bottom")
```



19.2 F-Test in R

We will do an F test with the wages, education and experience example.

- We construct the null and alternative hypotheses:

$$H_0 : \beta_1 = \beta_2 = 0$$
$$H_1 : \text{at least one of } \beta_j \neq 0 \text{ for } j = 1, 2$$

- Under H_0:

$$F = \frac{SSR/k}{SSE/\left(n-k-1\right)} \sim F_{k,n-k-1}$$

  We now have to calculate the realized value of F in our sample, f. We estimate the model:

```
df <- read.csv("wages1.csv")
m <- lm(wage ~ educ + exper, data = df)
summary(m)
```

```
Call:
lm(formula = wage ~ educ + exper, data = df)

Residuals:
    Min      1Q  Median      3Q     Max 
-5.5532 -1.9801 -0.7071  1.2030 15.8370 

Coefficients:
            Estimate Std. Error t value       Pr(>|t|)    
(Intercept) -3.39054    0.76657  -4.423 0.000011846645 ***
educ         0.64427    0.05381  11.974        < 2e-16 ***
exper        0.07010    0.01098   6.385 0.000000000378 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.257 on 523 degrees of freedom
Multiple R-squared:  0.2252,	Adjusted R-squared:  0.2222 
F-statistic: 75.99 on 2 and 523 DF,  p-value: < 2.2e-16
```

We see that the output of `summary()` already gives the value of the F-test test statistic and the associated p-value. The realized value of f in our sample is 75.99 and the associated p-value of the F-test is very close to zero (`< 2.2e-16`). We can extract the value from the first value of `summary(m)$fstatistic`. The 2nd and 3rd values are the numerator and denominator degrees of freedom, respectively.

```
summary(m)$fstatistic
```

```
    value     numdf     dendf 
 75.98998   2.00000 523.00000 
```

```
(f <- summary(m)$fstatistic[1])
```

```
   value 
75.98998 
```

To obtain the critical value we use the `qf()` function with 3 arguments: (i) one minus the size of the test, $1 - \alpha$, (ii) the numerator degrees of freedom, k and (iii) the denominator degrees of freedom, $n - k - 1$:

```r
qf(0.95, 2, 523)
```

```
[1] 3.012957
```

We can also get these degrees of freedom from the model output to make sure we use the right numbers. We can do:

```r
(numdf <- summary(m)$fstatistic[2])
```

```
numdf
    2
```

```r
(dendf <- summary(m)$fstatistic[3])
```

```
dendf
  523
```

```r
qf(0.95, numdf, dendf)
```

```
[1] 3.012957
```

We reject H_0 if the observed f is greater than the critical value, 3.013. Indeed in our case $f = 75.99$ so we reject the null hypothesis.

If we were using the p-value approach we could just read the p-value right from the `summary()` output. To obtain it manually we can do:

```r
1 - pf(f, 2, 523)
```

```
value
    0
```

The p-value is numerically zero, so we reject the null hypothesis.

We conclude by saying our model is useful at the 5% level.

19.3 Summary of Steps

19.3.1 Critical Value Method for Testing Model Usefulness

- Construct null and alternative hypotheses:

$$\begin{aligned} &H_0 : \beta_1 = \beta_2 = \cdots = \beta_k = 0 \\ &H_1 : \text{at least one of } \beta_j \neq 0 \text{ for } j = 1, \dots, k \end{aligned}$$

- Under H_0:
$$F = \frac{SSR/k}{SSE/\left(n-k-1\right)} \sim F_{k,n-k-1}$$
- Calculate the value of the test statistic, f.
  - Extract from R output with `summary(m)$fstatistic[1]`.
- Reject H_0 if $f \geq F_{1-\alpha,k,n-k-1}$.
  - We find $F_{1-\alpha,k,n-k-1}$ in R with `qf(1-alpha, k, n-k-1)`.
- Draw a conclusion.

19.3.2 p-Value Method for Testing Model Usefulness

- Construct null and alternative hypotheses:
$$\begin{aligned} &H_0 : \beta_1 = \beta_2 = \cdots = \beta_k = 0 \\ &H_1 : \text{at least one of } \beta_j \neq 0 \text{ for } j = 1, \ldots, k \end{aligned}$$
- Under H_0:
$$F = \frac{SSR/k}{SSE/\left(n-k-1\right)} \sim F_{k,n-k-1}$$
- Calculate the value of the test statistic, f.
  - Extract from R output with `summary(m)$fstatistic[1]`.
- Calculate the p-value and reject if $p \leq \alpha$.
  - Find the p-value in R with: `1-pf(f, k, n-k-1)`.
  - However, we will see that `summary()` always gives this, so it's not necessary to calculate.
- Draw a conclusion.

Chapter 20

Partial F-Test

In the multiple linear regression model we learned that:

1. We use a t-test to test if a single variable X_j is useful in the model.
2. We use an F-test to test if all X_1, X_2, ..., X_k were jointly useful in the model.

What we will learn in this chapter is how to test if a subset of X_1, X_2, ..., X_k are jointly useful. This test is called a *Partial F-test*.

20.1 Complete and Reduced Model

In terms of definitions, we call the *complete* model the model with k independent variables:

$$\mathbb{E}\left[Y_i | x_{i1}, \dots, x_{ik}\right] = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_g x_{ig} + \beta_{g+1} x_{i,g+1} + \cdots + \beta_k x_{ik}$$

The *reduced* model is the model with $g < k$ independent variables:

$$\mathbb{E}\left[Y_i | x_{i1}, \dots, x_{ig}\right] = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_g x_{ig}$$

20.2 Null and Alternative Hypotheses

What the partial F-test does is test if the $k - g$ variables X_{g+1}, X_{g+2}, ..., X_k are jointly useful in the model.

The null and alternative hypotheses are:

$$\begin{aligned} &H_0 : \beta_{g+1} = \beta_{g+2} = \cdots = \beta_k = 0 \\ &H_1 : \text{ at least one of } \beta_j \neq 0 \text{ for } j = g+1, \dots, k \end{aligned}$$

20.2.0.0.0.1 * Example

This is quite general notation so to help fix ideas suppose the complete model has 5 variables and the reduced model has 3 variables. The complete model is then:

$$\mathbb{E}\left[Y_i|x_{i1},\dots,x_{ik}\right]=\beta_0+\beta_1x_{i1}+\beta_2x_{i2}+\beta_3x_{i3}+\beta_4x_{i4}+\beta_5x_{i5}$$

The reduced model is then:

$$\mathbb{E}\left[Y_i|x_{i1},\dots,x_{ig}\right]=\beta_0+\beta_1x_{i1}+\beta_2x_{i2}+\beta_3x_{i3}$$

The null and alternative hypotheses of the partial F-test are then:

$$\begin{aligned}&H_0:\beta_4=\beta_5=0\\&H_1:\text{ at least one of }\beta_j\neq 0\text{ for }j=4,5\end{aligned}$$

20.3 The Test Statistic

We first discuss the intuition for the partial F-test test statistic.

The total variation in the y-data is measured by the total sum of squared $SST=\sum_{i=1}^{n}(y_i-\bar{y})^2$. This is like the sum of squared errors without any model: where we just use the mean $\bar{y}$ to predict y_i. With the reduced model, the sum of squared errors is SSE_r and with the complete model the sum of squared errors is SSE_c.

By adding more variables to our model, we always reduce the sum of squared errors, so it holds that $SSE_r\geq SSE_c$ always. If the complete model reduces the sum of squared errors “a lot” (i.e. SSE_r-SSE_c is large), then the new $k-g$ variables are useful additions to the model. If the complete model's sum of square errors is “very similar” to the reduced model (i.e. SSE_r-SSE_c is small), then the new $k-g$ variables are not very useful additions to the model.

The partial F test statistic captures the size of this difference. Under H_0:

$$F=\frac{\left(SSE_r-SSE_c\right)/\left(k-g\right)}{SSE_c/\left(n-k-1\right)}\sim F_{k-g,n-k-1}$$

That is, this test statistic follows an F distribution with $k-g$ numerator and $n-k-1$ denominator degrees of freedom.

20.4 Carrying out the Test

To show how to carry out the rest of the test we will use an example. Because a partial F-test only makes sense in models with at least 3 variables, we will return to the clothing expenditure model we saw in Chapter 15.

Why do we need at least 3 variables? With only 2 variables, we can use a regular t-test to test the usefulness of one variable, and we can use a regular F-test to test the usefulness of both variables. To have a subset with at least 2 variables we need a complete model with at least 3 variables.

Using the clothing expenditure model, we ask the following question: Does a household's clothing expenditure depend on the household composition (number of people and number of children) after we control for household income (using $\alpha = 0.05$)?

The question is asking if the number of children and the household size are useful additions to the model.

The complete model is:

$$\mathbb{E}\left[Y_i | x_{i1}, x_{i2}, x_{i3}\right] = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3}$$

where x_{i1} is household income, x_{i2} is the number of children and x_{i3} is the household size.

The reduced model does not include the household composition variables:

$$\mathbb{E}\left[Y_i | x_{i1}\right] = \beta_0 + \beta_1 x_{i1}$$

The null & alternative hypotheses are:

- $H_0 : \beta_2 = \beta_3 = 0$.
- $H_1 : \beta_2 \neq 0$ or $\beta_3 \neq 0$ or both.

We form the test statistic: Under H_0:

$$F = \frac{\left(SSE_r - SSE_c\right) / \left(k - g\right)}{SSE_c / \left(n - k - 1\right)} \sim F_{k-g,n-k-1}$$

We now need to calculate the realized value of the test statistic in our sample. To do this we follow a very similar approach to our we calculate the SSE, SSR and SST. We estimate both the reduced model and the complete model and call then `m1` and `m2`, respectively, and then use the `anova()` function.

```r
df <- read.csv("clothing-exp.csv")
m1 <- lm(clothing_exp ~ hh_inc, data = df)
m2 <- lm(clothing_exp ~ hh_inc + num_kids + hh_size, data = df)
anova(m1, m2)
```

```
Analysis of Variance Table

Model 1: clothing_exp ~ hh_inc
Model 2: clothing_exp ~ hh_inc + num_kids + hh_size
  Res.Df    RSS Df Sum of Sq      F     Pr(>F)
1    298 3.3809
```

```
2    296 3.1442  2   0.23671 11.142 0.00002161 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The SSE_r is 3.3809 under RSS for model 1. The SSE_c is 3.1442 under RSS for model 2. Notice that the SSE_c is smaller than SSE_r. This will always be the case because additional variables will always reduce the sum of squared errors in the model. What the partial F-test is testing is whether this reduction is relatively large. The 0.23671 under `Sum of Sq` is the difference $SSE_r - SSE_c$ in the numerator of the partial F-test statistic formula.

The resulting partial F-test test statistic is under `F` and is 11.142. The associated p-value is next to it and equals 0.00002161. The table also shows the numerator and denominator degrees of freedom for the partial F-test: The 2 under `Df` is $k - g$ and the 296 under `Res.Df` for model 2 is $n - k - 1$. Putting everything together from the formula $\frac{(SSE_r - SSE_c)/(k-g)}{SSE_c/(n-k-1)}$ we can confirm that these give the same F as in the 5th column:

```
(0.23671 / 2) / (3.1442 / 296)
```

```
[1] 11.14213
```

If we want to extract the value of the test statistic we can do:

```
anova(m1, m2)$F[2]
```

```
[1] 11.14205
```

If we are following the critical value approach we compare this test statistic to the critical value. Similar to the regular F test we get the critical value using the `qf()` function but instead of $k - g$ numerator degrees of freedom and $n - k - 1$ denominator degrees of freedom:

```
qf(0.95, 2, 296)
```

```
[1] 3.026257
```

The test statistic is greater than the critical value so we reject the null hypothesis. We also get the same conclusion looking at the critical value, which is less than 0.05. Thus the household composition variables are useful additions to the model after controlling for household income.

20.5 Relationship between the Partial F-test the F-test

The partial F-test is actually a generalization of the regular F test we learned about before. If the reduced model is simply:

$$\mathbb{E}\left[Y_i\right] = \beta_0$$

then the partial F-test turns into a regular F-test. This is the case with $g = 0$.

To see this, consider the test statistic

$$F = \frac{\left(SSE_r - SSE_c\right)/\left(k-g\right)}{SSE_c/\left(n-k-1\right)} \sim F_{k-g,n-k-1}$$

With a reduced model of just a constant, the SSE_r is the same as the SST. This is exactly how we learned how to obtain the SSE, SSR and SST in Chapter 17. We had to estimate a reduced model with just a constant. With $g = 0$ and calling SSE_c simply SSE, the test statistic becomes:

$$F = \frac{\left(SST - SSE\right)/k}{SSE/\left(n-k-1\right)} \sim F_{k,n-k-1}$$

Finally, using the identity $SST = SSE + SSR$ we can write:

$$F = \frac{SSR/k}{SSE/\left(n-k-1\right)} \sim F_{k,n-k-1}$$

The is exactly the same as the standard F-test!

20.6 Summary of Steps

20.6.1 Critical Value Method for the Partial F-Test}

- Construct null and alternative hypotheses:

$$\begin{aligned} &H_0 : \beta_{g+1} = \beta_{g+2} = \dots = \beta_k = 0 \\ &H_1 : \text{ at least one of } \beta_j \neq 0 \text{ for } j = g+1, \dots, k \end{aligned}$$

- Under H_0:

$$F = \frac{\left(SSE_r - SSE_c\right)/\left(k-g\right)}{SSE_c/\left(n-k-1\right)} \sim F_{k-g,n-k-1}$$

- Calculate the value of the test statistic, f.
- Reject H_0 if $f \geq F_{1-\alpha,k-g,n-k-1}$
  - Find $F_{1-\alpha,k-g,n-k-1}$ in R with `qf(1-alpha, k-g, n-k-1)`.
- Draw a conclusion.

20.6.2 p-Value Method for the Partial F-Test

- Construct null and alternative hypotheses:

$$\begin{aligned} &H_0: \beta_{g+1} = \beta_{g+2} = \cdots = \beta_k = 0 \\ &H_1: \text{ at least one of } \beta_j \neq 0 \text{ for } j = g+1, \ldots, k \end{aligned}$$

- Under H_0:

$$F = \frac{\left(SSE_r - SSE_c\right) / \left(k-g\right)}{SSE_c / \left(n-k-1\right)} \sim F_{k-g, n-k-1}$$

- Calculate the value of the test statistic, f.
- Reject H_0 if $p = \Pr\left(F \geq f\right) \leq \alpha$
  - Find p in R with `1-pf(f, k-g, n-k-1)`.
- Draw a conclusion.

Chapter 21

Collinearity

21.1 Introduction

In Chapter 20 we learned how to test if a subset of variables were useful in a model. We showed an example with the clothing expenditure data and determined that the “household composition” variables (number of children and the household size) were jointly useful in the model. Here are the steps again:

```r
df <- read.csv("clothing-exp.csv")
m1 <- lm(clothing_exp ~ hh_inc, data = df)
m2 <- lm(clothing_exp ~ hh_inc + num_kids + hh_size, data = df)
anova(m1, m2)
```

```
Analysis of Variance Table

Model 1: clothing_exp ~ hh_inc
Model 2: clothing_exp ~ hh_inc + num_kids + hh_size
  Res.Df    RSS Df Sum of Sq      F     Pr(>F)    
1    298 3.3809                                   
2    296 3.1442  2   0.23671 11.142 0.00002161 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p-value of the partial F-test is close to zero (`0.00002161`) so we reject the null hypothesis that the variables were useless in the model and conclude that they are useful.

Let’s take a look at the individual significance of each variable:

```r
summary(m2)
```

```
Call:
lm(formula = clothing_exp ~ hh_inc + num_kids + hh_size, data = df)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.27225 -0.05878 -0.00765  0.05767  0.43981 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -0.0125930  0.0232879  -0.541    0.589    
hh_inc       0.0822021  0.0004423 185.861   <2e-16 ***
num_kids     0.0108057  0.0137232   0.787    0.432    
hh_size      0.0119808  0.0116495   1.028    0.305    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1031 on 296 degrees of freedom
Multiple R-squared:  0.9921,	Adjusted R-squared:  0.9921 
F-statistic: 1.246e+04 on 3 and 296 DF,  p-value: < 2.2e-16
```

Here only household income is individually statistically significant at the 5% level. The *p*-values for number of children and household size are both greater than 0.05 (0.432 and 0.305 respectively) and thus insignificant.

How can it be that neither of these two variables are individually significant, but together they are jointly significant? We will see that this can happen when we face the problem of *collinearity*.

21.2 Collinearity versus Strictly Collinearity

Finding variables to be jointly significant but individually insignificant can sometimes occur if variables are strongly (but not perfectly) correlated with each other. Let's check the correlation between the two variables:

```
cor(df$num_kids, df$hh_size)
```

```
[1] 0.9270981
```

A correlation of 0.927 indicates a very strong positive linear relationship between the two variables. This makes sense, because more children in a household usually means there are more people in the household in total!

When there is a strong correlation (close to $+1$ or -1) between the independent variables, we encounter a problem known as *collinearity*. This problem is related but different to the no strict collinearity assumption we encountered in Chapter 15.

Strict collinearity is when one of the independent variables is an exact linear combination of one or more independent variables. This would occur if two variables have a perfect linear relationship (a correlation of +1 or −1). In this case R will not estimate a regression coefficient for one of the two perfectly correlated variables and will return `NA` for that variable.

Collinearity, on the other hand, is when one of the independent variables is strongly related to another variable (or a linear combination of other variables) but not perfectly so. A correlation of 0.927 in our example above is an example of two variables that are the strongly but not perfectly related. In the presence of collinearity R will estimate the model but two problems can occur:

1. The interpretation of the parameter estimates can become difficult. It is unclear if the number of children or the number of adults or both are increasing the clothing expenditure.
2. The standard errors on the estimated parameters can increase. This results in wider confidence intervals and smaller p-values in individual significance tests.

21.3 Possible Remedies for Collinearity

When you face a collinearity problem there are a number of different possible remedies.

One solution is to remove the offending variable. If two variables are highly correlated, then including both does not offer very much additional information when one variable is already included. In the clothing expenditure example, we might decide to drop the `num_kids` variable, because once we know the household size, knowing how many children there are in the household does not contain very much additional information because we know that large households usually contain lots of children. Let's try this out:

```
summary(lm(clothing_exp ~ hh_inc + hh_size, data = df))
```

```

Call:
lm(formula = clothing_exp ~ hh_inc + hh_size, data = df)

Residuals:
     Min       1Q   Median       3Q      Max
-0.27475 -0.05785 -0.00393  0.05942  0.43730

Coefficients:
              Estimate Std. Error t value   Pr(>|t|)
(Intercept) -0.0252035  0.0168961  -1.492      0.137
hh_inc       0.0821609  0.0004389 187.202    < 2e-16 ***
```

```
hh_size      0.0204746  0.0043961   4.657 0.00000484 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.103 on 297 degrees of freedom
Multiple R-squared:  0.9921,    Adjusted R-squared:  0.9921
F-statistic: 1.871e+04 on 2 and 297 DF,  p-value: < 2.2e-16
```

We can see that the household size variable is now individually statistically significant.

Another solution that is sometimes available is to create a new variable from the two problematic variables to solve the problem. For example, we could create a variable **num_adults** from the **hh_size** and **num_kids** variables. We could then change the model to use **num_adults** and **num_kids** instead of the household size variable. Unlike the previous solution which throws away the information about the household composition, this approach allows us to see the effects of adults and children separately.

Let's create the variable and check their correlation:

```
df$num_adults <- df$hh_size - df$num_kids
cor(df$num_adults, df$num_kids)
```

```
[1] 0.2301997
```

This correlation, although sizeable, is much smaller than before and not large enough to create a collinearity problem in the regression. To better understand this correlation, let's cross-tabulate the two variables:

```
table(df$num_kids, df$num_adults)
```

```
      1   2   3
  0  54 101  12
  1   9  30   7
  2   2  50   6
  3   0  18   1
  4   0   7   0
  5   0   3   0
```

Here the number of adults is shown left to right (1 to 3) and the number of children is shown top to bottom (0 to 5). The numbers in the table show the number of observations with that number of adults and number of children combination. For example, the 54 indicates that there are 54 observations (out of 300) with 1 adult and 0 children in the household. The 101 indicates that there are 101 observations with 2 adults and 0 children.

Looking at the relationship between the number of adults and number of children, we see there are no houses with no adults (each house has at least 1, 2 or 3 adults). Houses with children generally have at least 2 adults. Only 11 houses have 1-2 children and only 1 adult. So the positive correlation comes from children mostly living in houses with 2-3 adults and most of the single-adult houses have no children.

Let's now run the regression with these two variables:

```
summary(lm(clothing_exp ~ hh_inc + num_adults + num_kids, data = df))
```

```

Call:
lm(formula = clothing_exp ~ hh_inc + num_adults + num_kids, data = df)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.27225 -0.05878 -0.00765  0.05767  0.43981 

Coefficients:
              Estimate Std. Error t value  Pr(>|t|)    
(Intercept) -0.0125930  0.0232879  -0.541     0.589    
hh_inc       0.0822021  0.0004423 185.861   < 2e-16 ***
num_adults   0.0119808  0.0116495   1.028     0.305    
num_kids     0.0227865  0.0052888   4.308 0.0000224 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1031 on 296 degrees of freedom
Multiple R-squared:  0.9921,	Adjusted R-squared:  0.9921 
F-statistic: 1.246e+04 on 3 and 296 DF,  p-value: < 2.2e-16
```

We now see that `num_kids` is significant, while `num_adults` is insignificant. The size of the coefficient for `num_kids` is now similar to `hh_size` in the previous regression. The previous regression told us that more people in the household increased clothing expenditure, but we did not know if it was the children or the adults that were driving this. This regression now makes this clear: adding a child to a household increases the clothing expenditure on average more than adding an adult (holding all else constant).

Chapter 22

Higher-Order Terms

22.1 Theory

In this chapter we will discuss how to model non-linear relationships between X and Y.

In the simple linear regression model with $\mathbb{E}[Y_i|x_i] = \beta_0 + \beta_1 x_i$, if x_i increases by 1 unit, $\mathbb{E}[Y_i|x_i]$ increases by β_1 units no matter the value of x.

With the multiple linear regression model, we can use x_i^2 as a second variable in the model to make $\mathbb{E}[Y_i|x_i]$ a quadratic function of x_i:

$$\mathbb{E}[Y_i|x_i] = \beta_0 + \beta_1 x_i + \beta_2 x_i^2$$

Now as x_i changes, the change in $\mathbb{E}[Y_i|x_i]$ depends on the initial value of x_i. Let's look at $\mathbb{E}[Y_i|x_i]$ for different values of x_i:

$$\begin{aligned}
&\mathbb{E}[Y_i|x_i = 0] = \beta_0 \\
&\mathbb{E}[Y_i|x_i = 1] = \beta_0 + \beta_1 \cdot 1 + \beta_2 \cdot 1 \\
&\mathbb{E}[Y_i|x_i = 2] = \beta_0 + \beta_1 \cdot 2 + \beta_2 \cdot 4 \\
&\mathbb{E}[Y_i|x_i = 3] = \beta_0 + \beta_1 \cdot 3 + \beta_2 \cdot 9
\end{aligned}$$

As x_i goes from 0 to 1, $\mathbb{E}[Y_i|x_i]$ increases by $\beta_1 + \beta_2$. But when x_i goes from 1 to 2, $\mathbb{E}[Y_i|x_i]$ increases by $\beta_1 + 3\beta_2$. The change depends on the value of x_i!

This modeling approach is useful if the underlying relationship between X and Y is non-linear and a quadratic function is better suited to fit the relationship.

22.2 Estimation in R

We will now learn how we can estimate a model like this in R. The model we want to estimate is:

$$\mathbb{E}\left[Y_i|x_i\right] = \beta_0 + \beta_1 x_i + \beta_2 x_i^2$$

One way to do this is to create a new variable that is the square of the x variable and add it to the model.

Let's try this with the clothing expenditure data and a quadratic term for household income:

```
df <- read.csv("clothing-exp.csv")
df$hh_inc_sq <- df$hh_inc^2
summary(lm(clothing_exp ~ hh_inc + hh_inc_sq, data = df))
```

```

Call:
lm(formula = clothing_exp ~ hh_inc + hh_inc_sq, data = df)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.31144 -0.05935 -0.00628  0.06120  0.40051 

Coefficients:
               Estimate  Std. Error t value Pr(>|t|)    
(Intercept) -0.06057699  0.03156091  -1.919  0.05590 .  
hh_inc       0.08738403  0.00177497  49.231  < 2e-16 ***
hh_inc_sq   -0.00006019  0.00002177  -2.764  0.00606 ** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1053 on 297 degrees of freedom
Multiple R-squared:  0.9918,    Adjusted R-squared:  0.9917 
F-statistic: 1.788e+04 on 2 and 297 DF,  p-value: < 2.2e-16
```

There is also a way in R to include squared terms (or any other function) of a variable without having to create a new variable. We can just make the transformation directly within the formula in the `lm()` function. We just have to put it inside the `I()` function (where `I` stands for "inhibit interpretation").[1]

```
df <- read.csv("clothing-exp.csv")
summary(lm(clothing_exp ~ hh_inc + I(hh_inc^2), data = df))
```

[1] If we wanted to estimate a model $\mathbb{E}[Y_i|x_{i1}, x_{i2}] = \beta_0 + \beta_1(x_{i1} + x_{i2})$, i.e. a simple linear regression model with Y_i explained by the sum of x_{i1} and x_{i2} we can't just do `lm(y ~ x1 + x2, data = df)`. This is because this would actually estimate the model $\mathbb{E}[Y_i|x_{i1}, x_{i2}] = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2}$. To "inhibit" R from "interpreting" the `+` as adding a new variable we can use the `I()` function (the "inhibit interpretation" function). We would use it like this: `lm(y ~ I(x1 + x2), data = df)`.

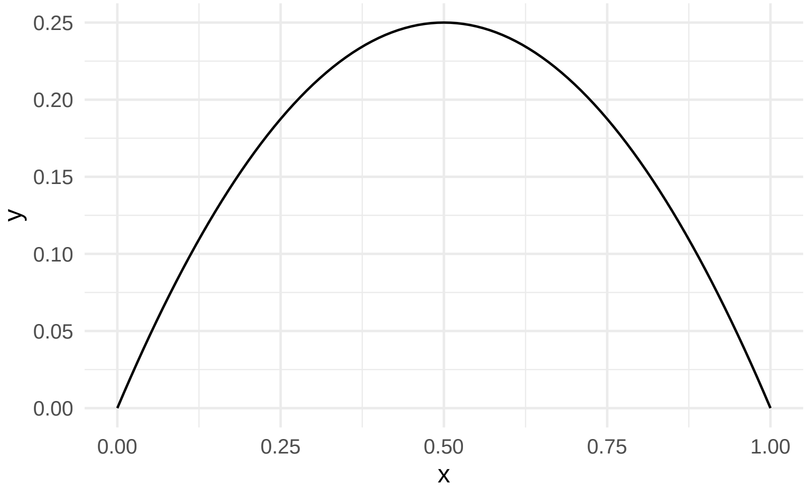
```
Call:
lm(formula = clothing_exp ~ hh_inc + I(hh_inc^2), data = df)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.31144 -0.05935 -0.00628  0.06120  0.40051 

Coefficients:
               Estimate  Std. Error t value Pr(>|t|)    
(Intercept) -0.06057699  0.03156091  -1.919  0.05590 .  
hh_inc       0.08738403  0.00177497  49.231  < 2e-16 ***
I(hh_inc^2) -0.00006019  0.00002177  -2.764  0.00606 ** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1053 on 297 degrees of freedom
Multiple R-squared:  0.9918,    Adjusted R-squared:  0.9917 
F-statistic: 1.788e+04 on 2 and 297 DF,  p-value: < 2.2e-16
```

We get the same result and saved one line of code. More importantly we can keep our data frame `df` “cleaner” by not having an extra variable in it that we only need for this regression.

Now let’s interpret the results. The both the level term `hh_inc` and the squared term `I(hh_inc^2)` are statistically significant. The *p*-value for the first term is very close to zero and is 0.00606 for the second term. Therefore there is statistical evidence of a non-linear relationship between household income and clothing expenditure.

The level term is positive and the quadratic term is negative. When this occurs the functional form has an inverse-U shape:

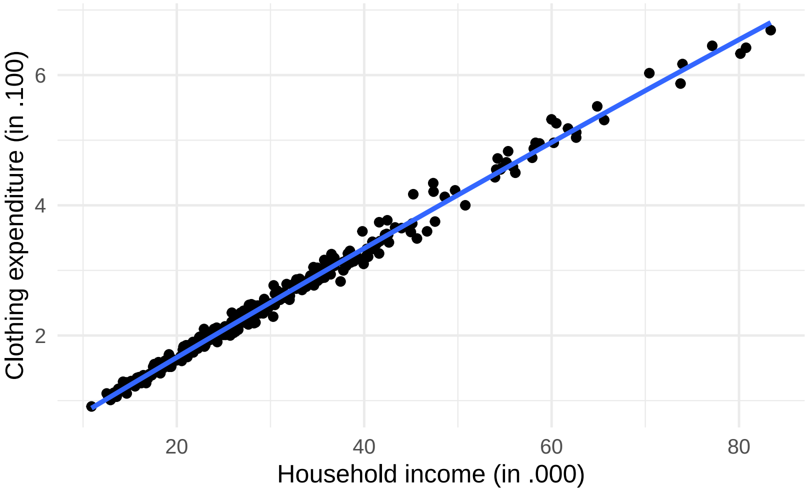
```r
library(ggplot2)
df <- data.frame(x = seq(0, 1, by = 0.01))
df$y <- df$x - df$x^2
ggplot(df, aes(x, y)) +
  geom_line() +
  theme_minimal()
```



This means for small income levels, as income increases clothing expenditure increases on average. But as income rises, unit increases in income has a smaller effect on clothing expenditure. For very high levels of income, eventually increases in income leads to a decrease in clothing expenditure, but this might be outside the range of our data.

Let's take a look at this in the data.

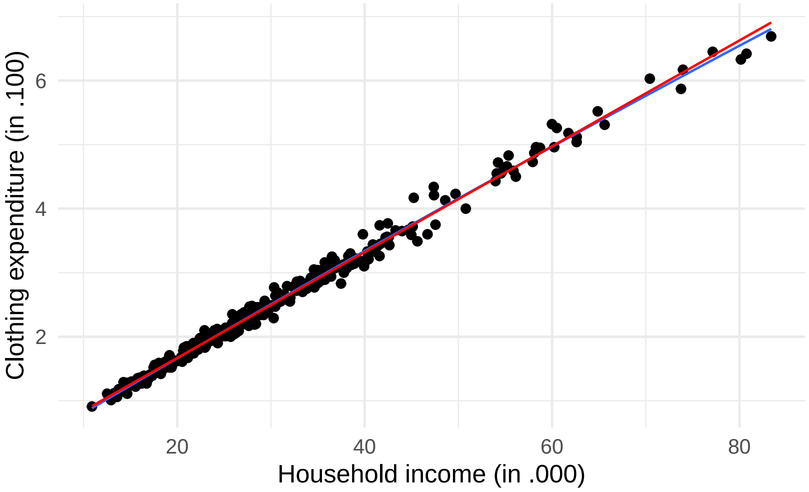
```r
library(ggplot2)
df <- read.csv("clothing-exp.csv")
ggplot(df, aes(hh_inc, clothing_exp)) +
  geom_point() +
  geom_smooth(method = "lm", formula = y ~ x + I(x^2), se = FALSE) +
  xlab("Household income (in €000)") +
  ylab("Clothing expenditure (in €100)") +
  theme_minimal()
```



When we fit the quadratic function to the data the function almost appears linear! This is because the estimate of β_2 is very very small (-0.0000601). Although the coefficient estimate is statistically significant, the fact that it is so small it has very little impact on the predictions.

Let's compare it to a standard linear model without a quadratic term, which we add to the plot in red.

```
library(ggplot2)
df <- read.csv("clothing-exp.csv")
ggplot(df, aes(hh_inc, clothing_exp)) +
  geom_point() +
  geom_smooth(method = "lm", formula = y ~ x + I(x^2), se = FALSE,
              lwd = 0.5) +
  geom_smooth(method = "lm", formula = y ~ x, color = "red", se = FALSE,
              lwd= 0.5) +
  xlab("Household income (in €000)") +
  ylab("Clothing expenditure (in €100)") +
  theme_minimal()
```



It looks almost identical. The blue line is only slightly different at very high levels in income.

If we compare the R^2s from both models we will also see that adding the quadratic term only explains a very small amount more of the variation in clothing expenditure:

```r
df <- read.csv("clothing-exp.csv")
summary(lm(clothing_exp ~ hh_inc, data = df))$r.squared
```

```
[1] 0.9915508
```

```r
summary(lm(clothing_exp ~ hh_inc + I(hh_inc^2), data = df))$r.squared
```

```
[1] 0.9917628
```

The linear model can explain 99.155%, while the quadratic model can explain 99.176%. So although the quadratic model has more explanatory power, we may prefer the simpler model for ease of interpretation because it is almost as good.

Chapter 23

Interaction Terms

23.1 Theory

Sometimes the effect of one X variable on Y depends on the value of an other X variable. For example, in our clothing expenditure example, the impact of household size on clothing expenditure might depend on the household income. Increasing the household size by one more person might have a larger effect on clothing expenditure for richer households compared to poorer households.

To model such relationships we can use *interaction terms*. To interact two variables we can estimate the model:

$$\mathbb{E}\left[Y_i|x_1,x_2\right] = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2$$

The 3rd term $x_{i1}x_{i2}$ is called an *interaction term*. When we include this, the expected value of Y_i given x_{i1} now depends on the *level* of x_{i2} (and vice versa). To see this, let's look at $\mathbb{E}\left[Y_i|x_{i1},x_{i2}\right]$ for different values of x_{i2}:

$$\begin{aligned}
\mathbb{E}\left[Y_i|x_{i1},x_{i2}=0\right] &= \beta_0 + \beta_1 x_{i1}\\
\mathbb{E}\left[Y_i|x_{i1},x_{i2}=1\right] &= \beta_0 + \beta_1 x_{i1} + \beta_2 + \beta_3 x_{i1}\\
\mathbb{E}\left[Y_i|x_{i1},x_{i2}=2\right] &= \beta_0 + \beta_1 x_{i1} + 2\beta_2 + 2\beta_3 x_{i1}
\end{aligned}$$

For each case, let's increase x_{i1} by one unit:

$$\begin{aligned}
\mathbb{E}\left[Y_i|x_{i1}+1,x_{i2}=0\right] &= \beta_0 + \beta_1 x_{i1} + \beta_1\\
\mathbb{E}\left[Y_i|x_{i1}+1,x_{i2}=1\right] &= \beta_0 + \beta_1 x_{i1} + \beta_1 + \beta_2 + \beta_3 x_{i1} + \beta_3\\
\mathbb{E}\left[Y_i|x_{i1}+1,x_{i2}=2\right] &= \beta_0 + \beta_1 x_{i1} + \beta_1 + 2\beta_2 + 2\beta_3 x_{i1} + 2\beta_3
\end{aligned}$$

Taking the difference for each case:

$$\begin{aligned}
\mathbb{E}\left[Y_i|x_{i1}+1,x_{i2}=0\right] - \mathbb{E}\left[Y_i|x_{i1},x_{i2}=0\right] &= \beta_1\\
\mathbb{E}\left[Y_i|x_{i1}+1,x_{i2}=1\right] - \mathbb{E}\left[Y_i|x_{i1},x_{i2}=1\right] &= \beta_1 + \beta_3\\
\mathbb{E}\left[Y_i|x_{i1}+1,x_{i2}=2\right] - \mathbb{E}\left[Y_i|x_{i1},x_{i2}=2\right] &= \beta_1 + 2\beta_3
\end{aligned}$$

So:

- When $x_{i2} = 0$, a unit increase in x_{i1} increases Y_i by β_1 on average.
- When $x_{i2} = 1$, a unit increase in x_{i1} increases Y_i by $\beta_1 + \beta_3$ on average.
- When $x_{i2} = 2$, a unit increase in x_{i1} increases Y_i by $\beta_1 + 2\beta_3$ on average.

When we include an interaction term, we therefore need to look at both β_1 and β_3 to learn about the impact of x_{i1} on Y_i.

23.2 Interaction Terms in R

Let's try this out with the clothing expenditure data. We want to interact household income (X_{i1}) with household size (X_{i2}) and estimate the model:

$$\mathbb{E}\left[Y_i|x_{i1}, x_{i2}\right] = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i1} x_{i2}$$

One way to do this is to create a new variable with the interaction and add it to the model. Let's try this first:

```r
df <- read.csv("clothing-exp.csv")
df$hh_inc_hh_size <- df$hh_inc * df$hh_size
summary(lm(clothing_exp ~ hh_inc + hh_size + hh_inc_hh_size, data = df))
```

```

Call:
lm(formula = clothing_exp ~ hh_inc + hh_size + hh_inc_hh_size,
    data = df)

Residuals:
     Min       1Q   Median       3Q      Max
-0.26738 -0.05882 -0.00592  0.05793  0.44451

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)    0.0090748  0.0312002   0.291    0.771
hh_inc         0.0809904  0.0009976  81.189   <2e-16 ***
hh_size        0.0082387  0.0103454   0.796    0.426
hh_inc_hh_size 0.0003971  0.0003040   1.306    0.192
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1029 on 296 degrees of freedom
Multiple R-squared:  0.9922,    Adjusted R-squared:  0.9921
F-statistic: 1.25e+04 on 3 and 296 DF,  p-value: < 2.2e-16
```

But because interaction terms are so common in linear regression models, R has a shortcut to do this. All we have to do is include `x1 * x2` in the formula and

R will include the two level terms and the interaction term. So when we do this we don't even need to add the individual x1 and x2 variables. Let's try this out:

```
summary(lm(clothing_exp ~ hh_inc * hh_size, data = df))
```

```
Call:
lm(formula = clothing_exp ~ hh_inc * hh_size, data = df)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.26738 -0.05882 -0.00592  0.05793  0.44451 

Coefficients:
                Estimate Std. Error t value Pr(>|t|)    
(Intercept)    0.0090748  0.0312002   0.291    0.771    
hh_inc         0.0809904  0.0009976  81.189   <2e-16 ***
hh_size        0.0082387  0.0103454   0.796    0.426    
hh_inc:hh_size 0.0003971  0.0003040   1.306    0.192    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1029 on 296 degrees of freedom
Multiple R-squared:  0.9922,    Adjusted R-squared:  0.9921 
F-statistic: 1.25e+04 on 3 and 296 DF,  p-value: < 2.2e-16
```

We get the same as above! The term hh_inc:hh_size is the interaction term (*Note:* we can add an interaction term without the level terms to the model using x1:x2, but you should always include the level terms when doing an interaction).

Let's interpret this. All terms, including the interaction term, are positive. With this model neither household size nor the interaction term are statistically significant. Ignoring statistical significance, we can interpret the parameter estimates as follows:

- The larger the household size, the larger the effect of a unit increase in income on clothing expenditure.
  - This makes sense because if a large household gets more income they have more people they can buy clothes for.
- The higher the household income, the larger the effect of a unit increase in household size on clothing expenditure.
  - This makes sense because if a richer household gets one more member in it, they have more money to buy clothes for the additional person.

Chapter 24

Dummy Variables

24.1 Introduction

Very often we have categorical variables that can take on two values. Examples of this are:

- Yes/no questions in a survey.
- Gender (at birth).
- Whether you have a college degree or not.

Because these are categorical variables and not numeric variables, we cannot include them in our regression model directly. However we can code a numeric variable that contains the information from the categorical variable. Such a variable is called a *dummy variable*.

A dummy variable is a variable that = 1 if something is true and = 0 if it is false:

- For the yes/no questions, we can create a variable that = 1 for "yes" responses and = 0 for "no" responses.
- For the gender variable, we can create a variable that = 1 if observation is female and = 0 if male. Such a variable is called a "female dummy".
  - We could alternatively create a "male dummy" that = 1 for male and = 0 for female.
- For the college degree variable, we can create a variable that = 1 if the observation has a college degree and = 0 if not. Such a variable is called a "college degree dummy".

24.2 Theory

Consider a simple linear regression model with a dummy variable:

$$\mathbb{E}\left[Y_i|x_i\right] = \beta_0 + \beta_1 x_i$$

- When the dummy variable equals 0, the expected value of Y_i is $\mathbb{E}\left[Y_i|x_i = 0\right] = \beta_0$. Call this μ_0. - When the dummy variable equals 1, the expected value of Y_i is $\mathbb{E}\left[Y_i|x_i = 1\right] = \beta_0 + \beta_1$. Call this μ_1.

The difference in means between the two groups, $\mu_1 - \mu_0$, is equal to β_1. Therefore we can estimate this regression model to estimate the difference in means, and hypothesis tests on β_1 are equivalent to hypothesis tests for the difference in means.

24.3 Dummy Variable Trap

Suppose we created two variables:

1. x_{i1} is a female dummy that $= 1$ for females and $= 0$ for males.
2. x_{i2} is a male dummy that $= 1$ for males and $= 0$ for females.

We could use either one of these to estimate the model above to get the difference in means. But what we cannot do is estimate a model with both variables. This is because $x_{i1} = 1 - x_{i2}$ for every observation (when $x_{i1} = 0$, $x_{i2} = 1$ and vice versa). If we include both variables we run into the problem of strict collinearity and R will drop one of the variables. This problem is called the *dummy variable trap*. When we have a qualitative variable with two values we need to choose one value for zero (what we call the base level or base category) and the other for one and not include both.

24.4 Dummy Variables in R

The example datasets we worked with so far do not have categorical variables. We therefore will employ a new dataset to illustrate how to estimate and interpret a model with a dummy variable.

The dataset `wages2.csv` contains wage data for $n = 526$ people from the 1976 Current Population Survey in the US.

The variables are:

- `wage`: Average hourly earnings (in USD).
- `educ`: Years of education.
- `female`: Female dummy.
- `married`: Married dummy.

We will use these data to test (with $\alpha = 0.05$) if the average hourly wage of men is more than \$2.00 larger than the mean hourly wage of women.

Mathematically, we want to test if $\mu_0 - 2 > \mu_1$. In words: the population mean hourly wage for men minus 2 is greater than the mean hourly wage for women. This will be our H_1. Rewriting this as $\mu_1 - \mu_0 < -2$ means we can use a simple linear regression model with a female dummy to test if $\beta_1 < -2$.

Let's estimate the regression model in R:

```r
df <- read.csv("wages2.csv")
m <- lm(wage ~ female, data = df)
summary(m)
```

```

Call:
lm(formula = wage ~ female, data = df)

Residuals:
    Min      1Q  Median      3Q     Max 
-5.5995 -1.8495 -0.9877  1.4260 17.8805 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   7.0995     0.2100  33.806  < 2e-16 ***
female       -2.5118     0.3034  -8.279 1.04e-15 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.476 on 524 degrees of freedom
Multiple R-squared:  0.1157,    Adjusted R-squared:  0.114 
F-statistic: 68.54 on 1 and 524 DF,  p-value: 1.042e-15
```

Let's interpret the coefficient estimates before running the test. The intercept is the estimate of $\mathbb{E}\left[Y_i | x_{i1} = 0\right] = \beta_0$. It means the average wage of men in the data is \$7.10. The estimate of the slope β_1 is the difference between the mean hourly wage of women and the mean hourly wage of men. Thus women on average earn \$2.51 less than men in the data.

We can also get these numbers by calculating the means by group directly:

```r
mean(df$wage[df$female == 0])
```

```
[1] 7.099489
```

```r
mean(df$wage[df$female == 1])
```

```
[1] 4.587659
```

The difference in means is then:

```r
mean(df$wage[df$female == 1]) - mean(df$wage[df$female == 0])
```

```
[1] -2.51183
```

which corresponds to the estimate of the slope.

We could also get the means by group using the `aggregate()` function:

```r
aggregate(wage ~ female, data = df, FUN = mean)
```

```
  female     wage
1      0 7.099489
2      1 4.587659
```

We are now ready to perform the hypothesis test. We set up the null and alternative hypothesis:

$$H_0 : \beta_1 \geq -2$$
$$H_1 : \beta_1 < -2$$

Under H_0, the test statistic $T = \frac{B_1-(-2)}{S_{B_1}}$ follows a t distribution with $n-2$ degrees of freedom (524).

Let's calculate the value of the test statistic in R:

```r
b_1 <- coef(summary(m))["female", "Estimate"]
s_b_1 <- coef(summary(m))["female", "Std. Error"]
(t <- (b_1 + 2) / s_b_1)
```

```
[1] -1.686931
```

This is a lower tail test. If we are using the critical value method, we reject H_0 if $t \leq t_{\alpha,n-2}$. We can calculate the critical value in R with:

```r
(cv <- qt(0.05, m$df.residual))
```

```
[1] -1.647767
```

```r
t < cv
```

```
[1] TRUE
```

The test statistic is smaller than the critical value (lies in the rejection region) so we reject the null hypothesis.

If we are using the p-value method we can calculate the p-value with:

```
(pval <- pt(t, m$df.residual))
```

```
[1] 0.04610592
```

```
pval < 0.05
```

```
[1] TRUE
```

The p-value (0.0461) is smaller than the significance level (0.05) so we reject the null hypothesis.

In both cases we reject the null hypothesis. Thus there is sufficient evidence for the claim that men earn more than $2 more than women at the 5% level.

24.5 Multiple Linear Regression with Dummy Variables

We can also use dummy variables in a multiple linear regression model. Using the same data, let's see if these differences in wages be explained by different levels of educational attainment. To do this we want to compare the average wages of women and men *of the same education level*.

Let x_{i1} be years of education and x_{i2} be the female dummy. The expected wage for men given the education level is:

$$\mathbb{E}\left[Y_i | x_{i1}, x_{i2} = 1\right] = \beta_0 + \beta_1 x_{i1} + \beta_2 \times 1 = \beta_0 + \beta_1 x_{i1} + \beta_2$$

The expected wage for women given the education level is:

$$\mathbb{E}\left[Y_i | x_{i1}, x_{i2} = 0\right] = \beta_0 + \beta_1 x_{i1} + \beta_2 \times 0 = \beta_0 + \beta_1 x_{i1}$$

Taking differences yields β_2. This is the difference in mean wages holding education fixed.

Let's estimate the model in R:

```
df <- read.csv("wages2.csv")
m <- lm(wage ~ educ + female, data = df)
summary(m)
```

```

Call:
lm(formula = wage ~ educ + female, data = df)

Residuals:
    Min      1Q  Median      3Q     Max 
-5.9890 -1.8702 -0.6651  1.0447 15.4998 
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.62282    0.67253   0.926    0.355
educ         0.50645    0.05039  10.051  < 2e-16 ***
female      -2.27336    0.27904  -8.147 2.76e-15 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.186 on 523 degrees of freedom
Multiple R-squared:  0.2588,    Adjusted R-squared:  0.256
F-statistic: 91.32 on 2 and 523 DF,  p-value: < 2.2e-16
```

The estimated coefficient on the female dummy is now −2.27, compared to −2.51 before. This means that women in this sample on average earned $2.27 less than men *of the same education level*.

Chapter 25

Qualitative Variables with Multiple Levels

25.1 Introduction

In Chapter 24 we learned how to use qualitative variables with two values - such as gender - in a regression model. By including a dummy variable for one of the gender values, we were able to cover all the possible values that the gender could take: $= 1$ for females and $= 0$ for males.

But often we have data with qualitative variables that can take on more than two values. For example we could have variables like:

- Educational attainment: High school, Bachelor, Master, PhD.
- Industry: Primary sector (e.g.~agriculture), Manufacturing, Services.
- Region: US States, Provinces of the Netherlands.

What we will learn in this chapter is how to include this kind of information in a linear regression model.

25.2 Theory

Suppose we are interested in the impact of industry sector on wages. We have a sample of wages Y_i for n individuals and what sector they work in ($sector_i$): primary, manufacturing or services.

25.2.1 The Incorrect Approach

Suppose for the moment we decided to follow the logic in Chapter 24 and created a numeric variable x_{i1} with the sector information as follows:

- $= 0$ if $sector_i =$ `primary`.
- $= 1$ if $sector_i =$ `manufacturing`.
- $= 2$ if $sector_i =$ `services`.

Suppose also we used this variable to estimate the regression model:

$$\mathbb{E}[Y_i|x_{i1}] = \beta_0 + \beta_1 x_{i1}$$

We will see now that this approach is incorrect.

For individuals in the primary sector we have:

$$\mathbb{E}[Y_i|x_{i1} = 0] = \beta_0$$

Therefore β_0 is the average wage of people in the primary sector.

For individuals in the manufacturing sector we have:

$$\mathbb{E}[Y_i|x_{i1} = 1] = \beta_0 + \beta_1$$

This means that β_1 is the average difference in wages between people in the manufacturing sector and the primary sector.

But then for individuals in the services sector we have:

$$\mathbb{E}[Y_i|x_{i1} = 2] = \beta_0 + 2\beta_1$$

This means that β_1 is also the average difference in wages between people in the services sector and the manufacturing sector! It also means that the difference in wages between services and the primary sector is $2\beta_1$.

Using a variable like this means that going from one sector to the next leads to an increase in wage of β_1 on average for all sectors. But there is no reason to think that going from primary to manufacturing *and* manufacturing to services will lead to the *same* average increase in wage. This is a very restrictive way to use this variable. We want a more flexible way to use the information about the sector in the model.

25.3 The Correct Approach

Instead of creating *one* numeric variable with the information from the qualitative variable what we should do is create a dummy variable *for each value* of the categorical variable. For the sector example we create 3 variables:

1. $D_{i1} = 1$ if primary sector and $x_{i1} = 0$ otherwise.
2. $D_{i2} = 1$ if manufacturing sector and $x_{i2} = 0$ otherwise.
3. $D_{i3} = 1$ if services sector and $x_{i3} = 0$ otherwise.

We then estimate a regression model using these dummy variables. We cannot include all 3 dummy variables because otherwise we run into the dummy variable trap we encountered in Chapter 24. This is because $D_{i1} = 1 - D_{i2} - D_{i3}$ always:

- If $D_{i1} = 0$ then one of D_{i2} or D_{i3} equals 1.
- If $D_{i1} = 1$ then $D_{i2} = D_{i3} = 0$.

We need to choose one category to be the *base category*. Let's let this be the primary sector. The model we would estimate is then:

$$\mathbb{E}[Y_i|sector_i] = \beta_0 + \beta_1 D_{i2} + \beta_2 D_{i3}$$

For the primary sector we have:

$$\mathbb{E}[Y_i|sector_i = \texttt{primary}] = \beta_0$$

For the manufacturing sector we have:

$$\mathbb{E}[Y_i|sector_i = \texttt{manufacturing}] = \beta_0 + \beta_1$$

For the services sector we have:

$$\mathbb{E}[Y_i|sector_i = \texttt{services}] = \beta_0 + \beta_2$$

So:

- β_1 is the average difference between the manufacturing and primary sectors.
- β_2 is the average difference between the services and primary sectors.
- $\beta_2 - \beta_1$ is the average difference between the services and manufacturing sectors.

Now the model is much more flexible.

25.4 Qualitative Variables in R

To show how to do this in R we will use a dataset on the average house prices Y_i by municipality (*gemeente*) in the Netherlands in 2022 and the province each municipality is in, $prov_i$. We will use this dataset to see how much location (province) impacts house prices.

To do this we create 12 dummy variables, one for each province:

- $D_{i1} = 1$ if $prov_i =$ `Drenthe` and zero otherwise.
- $D_{i2} = 1$ if $prov_i =$ `Flevoland` and zero otherwise.
- $\vdots$
- $D_{i12} = 1$ if $prov_i =$ `Zuid-Holland` and zero otherwise.

Because $D_{i1} = 1 - D_{i2} - D_{i3} - \cdots - D_{i12}$ for all i, we need to exclude one province to avoid the dummy variable trap. Let's choose Drenthe (D_{i1}) to be the base level.

The model is then:

$$\mathbb{E}\left[Y_i|prov_i\right] = \beta_0 + \beta_1 D_{i2} + \beta_2 D_{i3} + \cdots + \beta_{11} D_{i12}$$

To get the data ready we merge the following two datasets by municipality:

- cpb-house-prices.csv
- municipality-province.csv

We need to be careful that the house prices data uses ; for separators and commas for decimal points:

```
df1 <- read.csv("cpb-house-prices.csv", sep = ";", dec = ",")
names(df1) <- c("municipality", "house_price_2022", "house_price_2021")
df2 <- read.csv("municipality-province.csv")
names(df2) <- c("municipality", "province")
df <- merge(df1, df2, by = "municipality")
```

Next, what we could do is spend a lot of time creating 11 dummy variables, one for each province, and then typing all 11 into the formula in the `lm()` function. The good news is that there is no need to do this with R. If we provide a character vector into the `lm()` function, R will interpret it as a factor variable (a qualitative variable), and automatically create these dummies. R will also choose one level to be the base level automatically. Unless the variable is already a factor R will always choose the first value alphabetically (here Drenthe) to be the base level.

So estimating this model is as simple as:

```
m <- lm(house_price_2022 ~ province, data = df)
summary(m)
```

```

Call:
lm(formula = house_price_2022 ~ province, data = df)

Residuals:
    Min      1Q  Median      3Q     Max 
-270.55  -51.18   -8.15   33.68  577.65 

Coefficients:
                      Estimate Std. Error t value    Pr(>|t|)    
(Intercept)            381.308     27.768  13.732     < 2e-16 ***
provinceFlevoland        7.242     48.095   0.151      0.8804    
provinceFryslân        -10.419     35.848  -0.291      0.7715    
provinceGelderland      53.907     30.862   1.747      0.0816 .  
provinceGroningen      -84.728     41.186  -2.057      0.0405 *  
provinceLimburg        -38.889     32.704  -1.189      0.2352    
provinceNoord-Brabant   61.476     30.599   2.009      0.0453 *  
provinceNoord-Holland  159.944     31.326   5.106 0.000000559 ***
provinceOverijssel      -4.944     33.781  -0.146      0.8837    
provinceUtrecht        137.172     33.570   4.086 0.000055142 ***
```

```
provinceZeeland        -43.208     38.507  -1.122      0.2626
provinceZuid-Holland    62.294     30.982   2.011      0.0452 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 96.19 on 329 degrees of freedom
Multiple R-squared:  0.3251,    Adjusted R-squared:  0.3026
F-statistic: 14.41 on 11 and 329 DF,  p-value: < 2.2e-16
```

Let's interpret these estimates. We first note that:

$$\begin{aligned}
\mathbb{E}\left[Y_i|prov_i = \texttt{Drenthe}\right] =& \beta_0 + \beta_1 \times 0 + \beta_2 \times 0 + \dots + \beta_{10} \times 0 + \beta_{11} \times 0 = \beta_0 \\
\mathbb{E}\left[Y_i|prov_i = \texttt{Flevoland}\right] =& \beta_0 + \beta_1 \times 1 + \beta_2 \times 0 + \dots + \beta_{10} \times 0 + \beta_{11} \times 0 = \beta_0 + \beta_1 \\
&\vdots \\
\mathbb{E}\left[Y_i|prov_i = \texttt{Zuid-Holland}\right] =& \beta_0 + \beta_1 \times 0 + \beta_2 \times 0 + \dots + \beta_{10} \times 0 + \beta_{11} \times 1 = \beta_0 + \beta_{11}
\end{aligned}$$

So our estimate of β_0 is the average house price in Drenthe in our sample. Because house prices are in thousands of euros, the average house price in Drenthe in our sample is €381,308.33. Our estimate of $\beta_0 + \beta_1$ is the average house price in Flevoland in our sample. This is €381,308.33+€7,241.67=€388,550.00. Our estimate of β_1 is therefore the difference in average house price between Flevoland and Drenthe in our sample.

We will now do some example questions with this output.

One example is: "are there any differences in average house prices across provinces (at the 5% level)?"

To do this, let μ_j be the population average house price in province $j = 1, 2, \dots, 12$. This question is essentially asking to test:

$$\begin{aligned}
H_0 :& \mu_1 = \mu_2 = \dots = \mu_{12} \\
H_1 :& \text{ at least one } \mu_j \neq \mu_k \text{ for } j, k = 1, \dots, 12
\end{aligned}$$

Using our model with 11 dummy variables (with Drenthe as the base category), this is the same as:

$$\begin{aligned}
H_0 &: \beta_1 = \beta_2 = \dots = \beta_{11} = 0 \\
H_1 &: \text{ at least one } \beta_j \neq 0 \text{ for } j = 1, 2, \dots, 11
\end{aligned}$$

This is just an F-test for testing the model's usefulness!

Let's do the F-test as a recap. Under H_0, $F \sim F_{k,n-k-1}$. We can get the value of the test statistic from the model summary:

```
summary(m)$fstat
```

```
   value     numdf     dendf 
14.40891  11.00000 329.00000 
```

The critical value can be found with (using `numdf` and `dendf` from above to get the numerator and denominator degrees of freedom):

```
qf(0.95, 11, 329)
```

```
[1] 1.817809
```

Because the test statistic (14.409) is larger than the critical value (1.818) we reject the null hypothesis. There is sufficient evidence to suggest that the average house prices are different across provinces.

We can also use the p-value approach. The p-value for the F-test is already shown in the summary output, but we could also obtain it manually using:

```
f_stat <- summary(m)$fstat[1]
(p_val <- 1 - pf(f_stat, 11, 329))
```

```
value
    0
```

The p-value (0) is smaller than the significance level (0.05), so we also reject H_0 with this approach.

25.5 Specifying the Base Level

The coefficient estimates b_1, ..., b_{11} are always interpreted as the differences with respect to the base level. Because of this, we may want to specify the base level to help us interpret the results. By default, R chooses Drenthe as the base level. But we may want to use Noord-Brabant or another province as the base level. How can we do this?

To do this we first convert the variable to a `factor` and then "relevel" the factor variable using the `relevel()` function specifying the base level. Let's do this making Noord-Brabant the base level:

```
df$province <- factor(df$province)
df$province <- relevel(df$province, ref = "Noord-Brabant")
summary(lm(house_price_2022 ~ province, data = df))
```

```

Call:
lm(formula = house_price_2022 ~ province, data = df)

Residuals:
    Min      1Q  Median      3Q     Max 
-270.55  -51.18   -8.15   33.68  577.65 
```

```
Coefficients:
                       Estimate Std. Error t value    Pr(>|t|)    
(Intercept)            442.7839    12.8540  34.447     < 2e-16 ***
provinceDrenthe        -61.4756    30.5986  -2.009    0.045343 *  
provinceFlevoland      -54.2339    41.3198  -1.313    0.190252    
provinceFryslân        -71.8950    26.0626  -2.759    0.006130 ** 
provinceGelderland      -7.5682    18.6185  -0.406    0.684647    
provinceGroningen     -146.2039    33.0225  -4.427 0.000012994 ***
provinceLimburg       -100.3646    21.5336  -4.661 0.000004582 ***
provinceNoord-Holland   98.4683    19.3781   5.081 0.000000629 ***
provinceOverijssel     -66.4199    23.1372  -2.871    0.004361 ** 
provinceUtrecht         75.6968    22.8275   3.316    0.001015 ** 
provinceZeeland       -104.6839    29.6136  -3.535    0.000466 ***
provinceZuid-Holland     0.8181    18.8163   0.043    0.965346    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 96.19 on 329 degrees of freedom
Multiple R-squared:  0.3251,    Adjusted R-squared:  0.3026 
F-statistic: 14.41 on 11 and 329 DF,  p-value: < 2.2e-16
```

Now the intercept is the average house price in Noord-Brabant and all the coefficient estimates are differences between Noord-Brabant. For example, houses in Drenthe are on average €61,475.6 cheaper than Noord-Brabant while houses in Noord-Holland are on average €98,468.3 more expensive.

25.6 Interaction Terms with Dummy Variables

We can also combine dummy variables with interaction terms. Consider the following model with the wages2.csv data, where Y_i is the hourly wage:

$$\mathbb{E}\left[Y_i|educ_i, female_i, married_i\right] = \beta_0+\beta_1 educ_i+\beta_2 female_i+\beta_3 married_i+\beta_4 female_i \times married_i$$

Holding $educ_i$ fixed, there are 4 possible combinations for the female and married dummies:

$$\begin{aligned}
\text{Unmarried men:} &\quad \mathbb{E}\left[Y_i|educ_i, female_i=0, married_i=0\right] = \beta_0+\beta_1 educ_i \\
\text{Unmarried women:} &\quad \mathbb{E}\left[Y_i|educ_i, female_i=1, married_i=0\right] = \beta_0+\beta_1 educ_i+\beta_2 \\
\text{Married men:} &\quad \mathbb{E}\left[Y_i|educ_i, female_i=0, married_i=1\right] = \beta_0+\beta_1 educ_i+\beta_3 \\
\text{Married women:} &\quad \mathbb{E}\left[Y_i|educ_i, female_i=1, married_i=1\right] = \beta_0+\beta_1 educ_i+\beta_2+\beta_3+\beta_4
\end{aligned}$$

Holding education fixed:

- β_2 is the average difference in wage between unmarried women and unmarried men.

- β_3 is the average difference in wage between married men and unmarried men.
- $\beta_2 + \beta_4$ is the average difference in wage between married women and married men.
- $\beta_3 + \beta_4$ is the average difference in wage between married women and unmarried women.

So:

- β_2 is the wage gap for unmarried women.
- $\beta_2 + \beta_4$ is the wage gap for married women.
- β_4 can therefore be interpreted as the difference in wage gap between married and unmarried women.

Let's estimate it in R:

```
df <- read.csv("wages2.csv")
m <- lm(wage ~ educ + female * married, data = df)
summary(m)
```

```
Call:
lm(formula = wage ~ educ + female * married, data = df)

Residuals:
    Min      1Q  Median      3Q     Max 
-6.5907 -1.6293 -0.7337  1.1014 14.6606 

Coefficients:
               Estimate Std. Error t value        Pr(>|t|)    
(Intercept)    -1.02442    0.69311  -1.478           0.140    
educ            0.49356    0.04856  10.164         < 2e-16 ***
female         -0.36896    0.43341  -0.851           0.395    
married         2.64107    0.39936   6.613 0.0000000000933 ***
female:married -2.82883    0.55556  -5.092 0.0000004962244 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.065 on 521 degrees of freedom
Multiple R-squared:  0.3165,    Adjusted R-squared:  0.3113 
F-statistic: 60.32 on 4 and 521 DF,  p-value: < 2.2e-16
```

We now interpret the estimates:

According to the model, $b_0 = -1.02442$ means the expected wage for an unmarried man with zero years of education is $-\$1.02$.

Let's see how many observations have $educ_i = female_i = married_i = 0$:

```
nrow(df[df$educ == 0 & df$female == 0 & df$married == 0, ])
```

```
[1] 0
```

No observations satisfy this. Therefore we should not trust this estimate.

Interpreting b_1 i done as normal. Holding gender and marital status fixed, increasing education by one year on average increases the wage by 49 cents.

To interpret b_2 we need to be careful because `female` also appears in the interaction. When the variable married equals zero, then this term drops out and we can interpret the variable as normal. So b_2 is the average difference in wage between unmarried women and unmarried men, holding education fixed. So holding education fixed, unmarried women on average earn 37 cents less than unmarried men. The wage gap is therefore 37 cents for married women.

To interpret b_3 we need to be careful because `married` also appears in the interaction. When the female dummy equals zero, then this term drops out and we can interpret the variable as normal. So b_3 is the average difference in wage between married men and unmarried men, holding education fixed. Holding education fixed, married men on average earn \$2.64 more than unmarried men.

Finally, for b_4 we recall that above we showed that β_4 can be interpreted as the difference in wage gap between married and unmarried women. So holding education fixed, the gender wage gap is \$2.83 larger for married women compared to unmarried women.

Let's consider an example question from this output.

Holding education fixed, do unmarried women earn less than unmarried men (at the 5% level)? Use a p-value approach.

This question is asking if $\beta_2 < 0$, so the null and alternative hypotheses are $H_0 : \beta_2 \geq 0$ and $H_1 : \beta_2 < 0$. Under H_0, the test statistic $T = B_2/S_{B_2} \sim t_{n-k-1}$. Because the hinge is zero, we can read the test statistic directly from the table: $t = -0.8513$. However, the p-value in the table is for a two-sided test. We can get the p-value with:

```
(t <- coef(summary(m))["female", "t value"])
```

```
[1] -0.8513
```

```
pt(t, m$df.residual)
```

```
[1] 0.1974969
```

The p-value (0.1975) is greater than the significance level (0.05), so we we do not reject the null hypothesis. There is not enough evidence to show that unmarried women earn less than unmarried men of equal education levels.

Chapter 26

Testing and Correcting for Heteroskedasticity

In the final three chapters we will revisit some of the model assumptions and introduce formal tests and corrections for two of these. This chapter will discuss testing and correcting for heteroskedasticity.

Under the homoskedasticity assumption $\mathrm{Var}\left(\varepsilon_i | x_{i1}, \dots, x_{ik}\right) = \sigma_\varepsilon^2$ for all $x_{i1}, \dots, x_{ik}$.

Heteroskedasticity is when the variance of the errors varies with the values of the explanatory variables. In the presence of heteroskedasticity, the standard errors may not be reliable. This frequently occurs in practice. We will now learn:

- How to formally test for the prescence of heteroskedasticity.
- How to adjust the model's standard errors for heteroskedasticity.

26.1 Formal Test for Heteroskedasticity

We can formally test for heteroskedasticity as follows.

1. Estimate the original model $\mathbb{E}\left[Y_i | x_{i1}, \dots, x_{ik}\right] = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_k x_{ik}$ and save the residuals, e_i.
2. Estimate the auxiliary model which uses e_i^2 as the dependent variable:

$$\mathbb{E}\left[e_i^2 | x_{i1}, \dots, x_{ik}\right] = \gamma_0 + \gamma_1 x_{i1} + \cdots + \gamma_k x_{ik}$$

3. Apply the F-test for the usefulness of this model.

Under H_0, $\gamma_1 = \cdots = \gamma_k = 0$ and we have homoskedasticity (the dispersion of the residuals does not vary with the independent variables). Under H_1, at least one $\gamma_j \neq 0$ and we have heteroskedasticity (the dispersion of the residuals does not vary with the independent variables).

The logic of the test is that if the independent variables are useful at explaining e_i^2, then the variance of the residuals does depend on the values of the independent variables, violating homoskedasticity.

Let's try this out with a regression model:

```
# Step 1: Estimate original model and save the residuals
df <- read.csv("wages2.csv")
m <- lm(wage ~ educ + female * married, data = df)
df$e <- m$residuals
# Step 2: Estimate the auxialiary model with the square of residuals
aux <- lm(e^2 ~ educ + female * married, data = df)
# Step 3: Apply the F-test:
summary(aux)$fstat
```

```
    value     numdf     dendf 
 10.72187   4.00000 521.00000 
```

```
qf(0.95, 4, 521)
```

```
[1] 2.389045
```

- *Critical value approach:* The F statistic (10.722) is larger than the critical value (2.389). Therefore we reject the null hypothesis. There is evidence of heteroskedasticity.
- *p-value approach:* The F test p-value (0.000) is smaller than the significance level (0.05). Therefore we reject the null hypothesis. There is evidence of heteroskedasticity.

26.2 Correcting Standard Errors for Heteroskedasticity in R

The standard formula for the standard errors of the regression coefficients assumes homoskedasticity. In the presence of heteroskedasticity there is another formula that accounts and corrects for this. We won't go into the details of this formula, but we will learn how to get R to use these corrected standard errors.

To do this we use the function `vcovHC()` from the `sandwich` package. This function name is from **V**ariance **Cov**ariance **H**eteroskedasticity **C**onsistent. The package is called `sandwich` because the mathematical formula for the standard errors has a "bread" component and a "meat" component with the form *bread* × *meat* × *bread*. Again, we won't go into the details of this.

The function `vcovHC()` by itself doesn't give us the corrected regression table. We will use the `coeftest()` function from the package `lmtest` to do this.

In practice, many people use these standard errors by default without even doing a formal test for heteroskedasticity. This is because heteroskedasticity is so common that the safe approach is to use heteroskedasticity-robust standard errors all the time. However, in the exam you should only use these standard errors if specifically instructed to use them. In normal cases you should use the default standard errors from the `summary()` function.

Let's get the regression table with the corrected standard errors in R:

```
library(lmtest)
library(sandwich)
coeftest(m, vcov = vcovHC(m))
```

```
t test of coefficients:

                Estimate Std. Error t value  Pr(>|t|)    
(Intercept)    -1.024421   0.787960 -1.3001    0.1941    
educ            0.493559   0.059092  8.3524 6.088e-16 ***
female         -0.368964   0.374822 -0.9844    0.3254    
married         2.641066   0.404064  6.5363 1.505e-10 ***
female:married -2.828826   0.501106 -5.6452 2.714e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Notice that the coefficient estimates are the same as before, but the standard errors are slightly different. Because the test statistics for individual significance and associated *p*-values depend on the standard errors, these also change.

Chapter 27

Testing and Correcting for Serial Correlation

27.1 Introduction

With time-series data, serial correlation in the error terms is very common. If e_t is positive, e_{t+1} is often positive in the following period. This is called first-order autocorrelation. If this occurs, the default standard errors are no longer reliable.

Sometimes changing the regression specification helps remove the problem. For example:

- Using differences $x_t - x_{t-1}$ instead of levels x_t.
- Using growth rates $\frac{x_t - x_{t-1}}{x_{t-1}}$ instead of levels x_t.
- Adding a time trend term to the model.

In this chapter we will learn how to formally test for first-order autocorrelation and how to correct the standard errors for it.

27.2 Formal Test for First-Order Autocorrelation

We can formally test for first-order autocorrelation as follows.

1. Estimate the original model:

$$\mathbb{E}\left[Y_t | x_{t1}, \dots, x_{tk}\right] = \beta_0 + \beta_1 x_{t1} + \dots + \beta_k x_{tk}$$

   and save the residuals, e_t.
2. Create a new variable which is the lag of the residuals, e_{t-1}.

3. Estimate the auxiliary model:

$$e_t = \gamma_0 + \gamma_1 e_{t-1} + \nu_t$$

4. Apply the t-test (significance test) on γ_1. Under H_0 there is *no* first-order autocorrelation and under H_1 is there is first-order autocorrelation.

In this auxiliary regression, γ_1 is the correlation coefficient between e_t and e_{t-1}. The logic behind the test is that if the previous period's residual can predict the current period's one, then the residuals are not independent across time.

27.3 Testing for First-Order Autocorrelation in R

Let's see how to do these steps in R. We will use the Dutch GDP and exports data we encountered in Chapter 8.

```
# Step 1: Estimate the original model and save the residuals:
df <- read.csv("nl-exports-gdp.csv")
m <- lm(gdp ~ exports, data = df)
df$e <- m$residuals
# Step 2: Create a new variable which is the lag of the residuals:
df$lag_e <- c(NA, df$e[1:(nrow(df)-1)])
# Step 3: Estimate the auxiliary model:
aux <- lm(e ~ lag_e, data = df)
# Step 4: Apply an individual significance test on the lagged residual term:
summary(aux)
```

```

Call:
lm(formula = e ~ lag_e, data = df)

Residuals:
    Min      1Q  Median      3Q     Max 
-24.806  -3.847   1.886   5.140  12.424 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  0.91581    1.19398   0.767    0.447    
lag_e        0.94605    0.02968  31.878   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 8.773 on 52 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  0.9513,    Adjusted R-squared:  0.9504 
```

```
F-statistic:  1016 on 1 and 52 DF,  p-value: < 2.2e-16
```

The t-test for the individual significance of the lagged residual has a p-value close to zero. This is very strong evidence for first-order serial correlation.

27.4 Taking Growth Rates

Before learning how to correct the standard errors for serial correlation, let's first try taking growth rates of both GDP and exports to see if the first-order serial correlation problem goes away. Note that by taking growth rates we lose the first observation because we do not know what the lagged value is in the first period in the data. This is why we need to use the `na.omit()` function to drop the missing observations.

```r
df <- read.csv("nl-exports-gdp.csv")
df$lag_gdp <- c(NA, df$gdp[1:(nrow(df)-1)])
df$lag_exports <- c(NA, df$exports[1:(nrow(df)-1)])
df$gdp_growth <- (df$gdp - df$lag_gdp) / df$lag_gdp
df$exports_growth <- (df$exports - df$lag_exports) / df$lag_exports
df <- na.omit(df)
m <- lm(gdp_growth ~ exports_growth, data = df)
summary(m)
```

```

Call:
lm(formula = gdp_growth ~ exports_growth, data = df)

Residuals:
       Min         1Q     Median         3Q        Max 
-0.0283831 -0.0084130  0.0006188  0.0099133  0.0268511 

Coefficients:
               Estimate Std. Error t value         Pr(>|t|)    
(Intercept)    0.004750   0.002726   1.742           0.0873 .  
exports_growth 0.380987   0.042394   8.987 0.00000000000364 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.01317 on 52 degrees of freedom
Multiple R-squared:  0.6083,	Adjusted R-squared:  0.6008 
F-statistic: 80.76 on 1 and 52 DF,  p-value: 0.000000000003638
```

We now repeat the formal test for serial autocorrelation to see if the problem remains:

```
df$e <- m$residuals
df$lag_e <- c(NA, df$e[1:(nrow(df)-1)])
aux <- lm(e ~ lag_e, data = df)
summary(aux)
```

```

Call:
lm(formula = e ~ lag_e, data = df)

Residuals:
      Min         1Q    Median         3Q        Max 
-0.028405 -0.007534 -0.002025  0.008030  0.032830 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.0002411  0.0017667  -0.136    0.892
lag_e        0.2116660  0.1354665   1.562    0.124

Residual standard error: 0.01286 on 51 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  0.04568,	Adjusted R-squared:  0.02697 
F-statistic: 2.441 on 1 and 51 DF,  p-value: 0.1244
```

Now the lagged residual has a p-value greater than 0.05. There is no longer evidence of first-order serial correlation.

27.5 Correcting for First-Order Autocorrelation in R

If taking growth rates, differences or adding a trend term does not remove the problem, you can correct the standard errors for serial correlation in a similar way to how we corrected for heteroskedasticity. To do this we use the function `vcovHAC()`, which corrects for both heteroskedasticity and autocorrelation.

We will now show how to do this in R. Let's suppose for the moment that our model with growth rates still suffered from serial correlation and we wanted to correct for it.

```
library(lmtest)
library(sandwich)
coeftest(m, vcov = vcovHAC(m))
```

```

t test of coefficients:
```

```
                Estimate Std. Error t value         Pr(>|t|)
(Intercept)    0.0047496  0.0030884  1.5379           0.1301
exports_growth 0.3809872  0.0437884  8.7006 0.00000000001012 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Notice that the coefficient estimates are the same as before but the standard errors are slightly different (e.g. 0.0437884 instead of 0.042394 for the slope).

Chapter 28

The Zero Conditional Mean Assumption

28.1 Introduction

A crucial assumption in the linear regression model is that $\mathbb{E}\left[\varepsilon_i | x_{i1}, \dots, x_{ik}\right] = 0$. This assumption implies no correlation between the error term and the explanatory variables. A violation of this assumption means our estimates of β_j are either too big or too small, sometimes even turning the opposite sign! Recall the class size and test scores example we saw in Chapter 8 where we saw that a regression of test scores on class size can yield a positive coefficient estimate on class size, even though we expect a negative one. Naturally this is much more serious than having standard errors that are too small.

A common remedy to this problem is to add more explanatory variables to the model that we suspect are correlated with our X variables of interest and the outcome variable Y. For example, adding the average socioeconomic status of the students to the class size and test scores model.

In this chapter we will briefly discuss some other solutions to the problem. At the very end of this chapter we will also have a brief discussion on some other model assumptions.

28.2 Experiments and Natural Experiments

Often adding more explanatory variables does not solve the problem. This is usually because there are variables which we would like to include but we do not have data on them (they are unobserved).

One way to solve this is to run an experiment: If we change X for individuals randomly and observe their outcomes Y, the randomness guarantees no correlation with the error. For example, we could randomly put students into classes of different sizes and observe their test scores afterwards.

But often we can't run an experiment because it's too expensive or unethical. For example, if we want to know the effect of a college degree on future wages, it would be unethical to stop people who would otherwise have went to college from obtaining a degree just to see how much less income they would make.

When an experiment is too expensive or unethical, sometimes we can use a "natural experiment". This is when there is an institutional feature that generates randomness in a variable. Returning to the class size and test scores example. In Israel, you have to go to a particular school based on where you live. There are strict rules that determine the number of classrooms in a school district:

- If there are 40 students to be enrolled, there is only 1 classroom.
- If there are 41 students to be enrolled, they are split into 2 classrooms (one with 20 and one 21 students).

Having 40 versus 41 students enrolled in a year is effectively random. Therefore if we compare test scores only between schools with 40 students (big classrooms) and 41 students (small classrooms), we can get the causal effect of classroom size on test scores.

Here is another example of a natural experiment. Suppose we want to estimate the effect of attending an elite secondary school (a dummy variable X) on future earnings (Y):

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$$

Often people who attend these schools are very able and productive. But able and productive people can find better jobs, regardless of where they go to school. So the X variable is correlated with the error term.

Ability/productivity is a difficult variable to measure precisely, so we can't add it to our model. It would also be both prohibitively expensive and unethical to randomly force some people to attend an elite school and others not.

So we rely on the natural experiment approach. We can make use of the fact that some elite schools have an entrance exam where you can enter if you achieve a minimum score on the exam. Students that just barely passed and just barely failed scored very similar on the exam, and on average should be similar to each other. Just some people were lucky and just about passed, while others had some bad luck and scored just below the required grade. By comparing future earnings only between those just above and just below the passing grade, we can get the causal effect.

28.3 Other Model Assumptions

We end this chapter with a very brief discussion of the other model assumptions and possible remedies for violations.

28.3.1 Non-Linearities or Non-Normal Error Terms

We will not discuss a formal test for these. If based on an analysis of scatter plots you suspect a violation of either of these, a change in the model specification can help. For example:

- Taking the natural logarithm of either the Y variable, the X variable, or both.
- Transforming levels of a variable X_t to either:
  - Changes: $X_t - X_{t-1}$.
  - Growth rates $\left(X_t - X_{t-1}\right) / X_{t-1}$.
- Add higher-order terms (such as X^2) to the model.

28.3.2 Perfect Collinearity

We won't discuss a formal test for this because R automatically "drops" variables that suffer from it. We will know immediately if it is present in our model. The remedy is simple: we just have to drop the offending variables from the model.