MATH3714 Linear Regression and Robustness

Jochen Voss

University of Leeds, Semester 1, 2021–22

Contents

Preface

From previous modules we know how to fit a regression line through points $(x_1, y_1), \dots, (x_n, y_n) \in \mathbb{R}^2$. The underlying model here is described by the equation

$$y_i = \alpha + \beta x_i + \varepsilon_i$$

for all $i \in \{1, 2, \dots, n\}$, and the aim is to find values for the intercept α and the slope β such that the residuals ε_i are as small as possible. This procedure, called simple linear regression, is illustrated in figure 1.

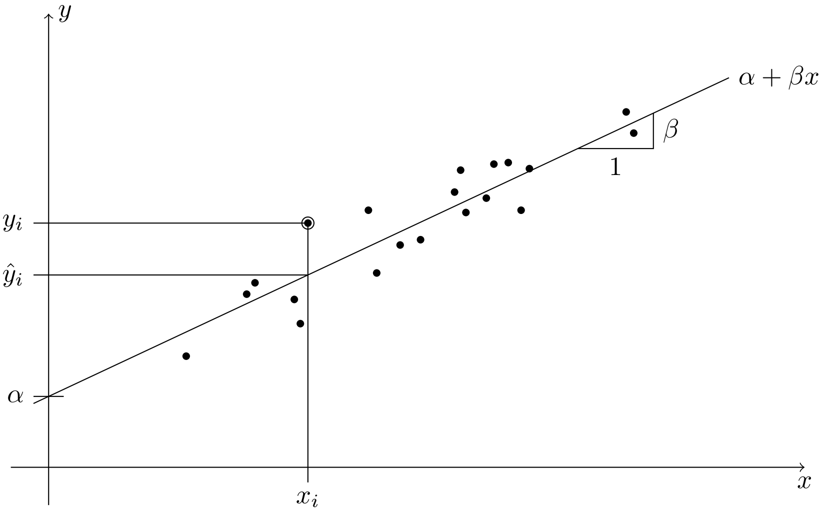


Figure 1: An illustration of linear regression. Each of the black circles in the plot stands for one paired sample (x_i, y_i). The regression line $x \mapsto \alpha + \beta x$, with intercept α and slope β, aims to predict the value of y using the observed value x. For the marked sample (x_i, y_i), the predicted y-value is $\hat{y}$.

In this situation, the variable x is called a **input**, feature, or sometimes the explanatory variable or the "independent variable". The variable y is called **response** or output, or sometimes the "dependent variable", and ε is called the **residual** or error.

Extending the situation of simple linear regression, in this module we will consider multiple linear regression, where the response y is allowed to depend on several input variables. The corresponding model is now

$$y_i = \alpha + \beta_1 x_{i1} + \dots + \beta_p x_{ip} + \varepsilon_i$$

for all $i \in \{1, 2, \dots, p\}$, where n is still the number of observations, and p is now the number of inputs we observe for each sample.

Note that for multiple linear regression, we still consider a single response for each sample, only the number of inputs has been increased. One way to deal with situations where there is more than one output would be to fit separate models for each output.

We will discuss multiple linear regression in much detail; our discussion will be guided by three different aims of linear regression:

1. Prediction: given a not previously observed value x, try to predict the corresponding y.
2. In cases were the residuals ε_i correspond to unwanted noise, the fitted values $\hat{y}_i = \alpha + \beta x_i$ can be considered to be de-noised versions of the observed values y_i.
3. By studying a fitted regression model, sometimes better understanding of the data can be achieved. For example, one could ask whether all of the p input variables carry information about the response y.

We will address these aims by considering different questions, like how to estimate the coefficients $\alpha, \beta_1, \dots, \beta_p$, how to assess model fit, or how to deal with outliers in the data.

About MATH3714

This module is **MATH3714 Linear Regression and Robustness**. The module manager and lecturer is Dr Jochen Voss, and my email address is J.Voss@leeds.ac.uk.

Notes and videos

The main way I expect you to learn the material for this course is by reading these notes and by watching the accompanying videos. I will release two sections of notes each week, for a total of 22 sections.

Reading mathematics is a slow process. Each section roughly corresponds to one traditional lecture, which would have taken 50 minutes. If you find yourself regularly getting through sections in much less than an hour, you're probably not reading carefully enough through each sentence of explanation and each line of mathematics, including understanding the motivation as well as checking the accuracy.

It is possible (but not recommended) to learn the material by only reading the notes and not watching the videos. It is not possible to learn the material by only watching the videos and not reading the notes.

Since we will all be relying heavily on these notes, I'm even more keen than usual to hear about errors mathematical, typographical or otherwise. Please, please email me if think you may have found any.

Lectures

There will be one online synchronous "lecture" session each week, on Mondays at 2-3pm, with me, run through Microsoft Teams. These will not be "lectures" in the traditional sense of the term, but will be an opportunity to re-emphasise material you have already learned from notes and videos, to give extra examples, and to answer common student questions, with some degree of interactivity.

I will assume you have completed all the work for the previous week by the time of the lecture, but I will not assume you've started the work for that week itself.

I am very keen to hear about things you'd like to go through in the lectures; please email me with your suggestions.

Workshops and Problem Sheets

There will be 5 problem sheets, corresponding to workshops in weeks 2, 4, 6, 8 and 10. The main goal of the workshops will be to go over your answers to the problems sheets.

My recommended approach to problem sheets and workshops is the following:

- Work through the problem sheet before the workshop, spending plenty of time on it, and making multiple efforts at questions you get stuck on. I recommend spending *at least three hours* on each problem sheet, in more than one block. Collaboration is encouraged when working through the problems, but I recommend writing up your work on your own.
- Take advantage of the workshops to ask for help or clarification on questions you weren't able to complete.
- After the workshop, attempt again the questions you were previously stuck on.
- If you're still unable to complete a question after this second round of attempts, *then* consult the solutions.

Discussion Board

I have set up a Microsoft Team for the course. I propose to use the "Discussion" channel there as a discussion board. This is a good place to post questions about material from the course, and — even better! — to help answer your colleagues' questions. The idea is that you all as a group should help each other out. I will visit a couple of times a week to clarify if everybody is stumped by a question, or if there is disagreement.

Software

For the module we will use the statistical computing package R. This program is free software, and you can find the program and documentation at the R project homepage. In particular, R will be used in the (assessed) practial.

My recommendation would be to install the RStudio environment, which includes R, on your own computer and use this for your work. (Choose the open source version, “RStudio Desktop”, on the download page.) Alternatively you can use RStudio or plain R on the university computers.

Assessments

Your final mark for the module will be based on a computer practical (20%) and a final exam (80%). For the practical (I believe it will take place in week 10) you will need to solve some problem using R and the methods you learned in the course and to present your results in a short report.

1 Simple Linear Regression

As a reminder, we consider simple linear regression in this section. My hope is, that all of you have seen this material before at some stage, *e.g.* in school or in some first or second year modules.

In preparation for notation introduced in the next section, we rename the parameters from α and β to the new names β_0 for the intercept and β_1 for the slope.

1.1 Residual Sum of Squares

In simple linear regression, the aim is to find a regression line $y = \beta_0 + \beta_1 x$, such that the line is "close" to given data points $(x_1, y_1), \ldots, (x_n, y_n) \in \mathbb{R}^2$ for $i \in \{1, 2, \ldots, n\}$. The ususal way to find *alpha* and β_1, and thus the regression line, is by minimising the **residual sum of squares**:

$$r(\beta_0, \beta_1) = \sum_{i=1}^{n} \bigl(y_i - (\beta_0 + \beta_1 x_i)\bigr)^2. \tag{1}$$

For given β_0 and β_1, the value $r(\beta_0, \beta_1)$ measures how close (in vertical direction) the given data points (x_i, y_i) are to the regression line $\beta_0 + \beta_1 x$. By minimising $r(\beta_0, \beta_1)$ we find the regression line which is "closest" to the data. The solution of this minimisation problem is usually expressed in terms of the sample variance s_x and the sample covariance s_{xy}.

Definition 1.1. The **sample covariance** of $x_1, \ldots, x_n \in \mathbb{R}$ and $y_1, \ldots, y_n \in \mathbb{R}$ is given by

$$\mathrm{s}_{xy} := \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y}),$$

where $\bar{x}$ and $\bar{y}$ are the sample means.

The **sample variance** of $x_1, \ldots, x_n \in \mathbb{R}$ is given by

$$\mathrm{s}_x^2 := \mathrm{s}_{xx} = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2,$$

where, again, $\bar{x}$ is the sample mean of the x_i.

Lemma 1.1. *Assume that* $\mathrm{s}_x^2 > 0$. *Then the function* $r(\beta_0, \beta_1)$ *from* (1) *takes its minimum at the point* (β_0, β_1) *given by*

$$\hat{\beta}_1 = \frac{\mathrm{s}_{xy}}{\mathrm{s}_x^2}, \qquad \hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x},$$

where $\bar{x}, \bar{y}$ *are the sample means,* s_{xy} *is the sample covariance and* s_x^2 *is the sample variance.*

Proof. We could find the minimum of r by differentiating and setting the derivatives to zero. Here we follow a different approach which uses a "trick" to simplify the algebra: Let $\tilde{x}_i = x_i - \bar{x}$ and $\tilde{y}_i = y_i - \bar{y}$ for all $i \in \{1, \ldots, n\}$. Then we have

$$\sum_{i=1}^{n} \tilde{x}_i = \sum_{i=1}^{n} x_i - n\bar{x} = 0$$

and, similarly, $\sum_{i=1}^{n} \tilde{y}_i = 0$. Using the new coordinates $\tilde{x}_i$ and $\tilde{y}_i$ we find

$$\begin{aligned}
r(\beta_0, \beta_1) &= \sum_{i=1}^{n} \bigl(y_i - \beta_0 - \beta_1 x_i\bigr)^2 \\
&= \sum_{i=1}^{n} \bigl(\tilde{y}_i + \bar{y} - \beta_0 - \beta_1 \tilde{x}_i - \beta_1 \bar{x}\bigr)^2 \\
&= \sum_{i=1}^{n} \Bigl((\tilde{y}_i - \beta_1 \tilde{x}_i) + (\bar{y} - \beta_0 - \beta_1 \bar{x})\Bigr)^2 \\
&= \sum_{i=1}^{n} \bigl(\tilde{y}_i - \beta_1 \tilde{x}_i\bigr)^2 + 2(\bar{y} - \beta_0 - \beta_1 \bar{x}) \sum_{i=1}^{n} (\tilde{y}_i - \beta_1 \tilde{x}_i) + n\bigl(\bar{y} - \beta_0 - \beta_1 \bar{x}\bigr)^2
\end{aligned}$$

Since $\sum_{i=1}^n \tilde{x}_i = \sum_{i=1}^n \tilde{y}_i = 0$, the second term on the right-hand side vanishes and we get

$$r(\beta_0, \beta_1) = \sum_{i=1}^n \bigl(\tilde{y}_i - \beta_1 \tilde{x}_i\bigr)^2 + n\bigl(\bar{y} - \beta_0 - \beta_1 \bar{x}\bigr)^2. \tag{2}$$

Both of these terms are positive and we can minimise the second term (without changing the first term) by setting $\beta_0 = \bar{y} - \beta_1 \bar{x}$.

To find the value of β_1 which minimises the first term on the right-hand side of (2) we now set the (one-dimensional) derivative w.r.t. β_1 equal to 0. We get the condition

$$\begin{aligned}
0 &\overset{!}{=} \frac{d}{d\beta_1} \sum_{i=1}^n \bigl(\tilde{y}_i - \beta_1 \tilde{x}_i\bigr)^2 \\
&= \sum_{i=1}^n 2(\tilde{y}_i - \beta_1 \tilde{x}_i) \frac{d}{d\beta_1} (\tilde{y}_i - \beta_1 \tilde{x}_i) \\
&= -2 \sum_{i=1}^n (\tilde{y}_i - \beta_1 \tilde{x}_i) \tilde{x}_i \\
&= -2 \sum_{i=1}^n \tilde{x}_i \tilde{y}_i + 2\beta_1 \sum_{i=1}^n \tilde{x}_i^2.
\end{aligned}$$

The only solution to this equation is

$$\begin{aligned}
\beta_1 &= \frac{\sum_{i=1}^n \tilde{x}_i \tilde{y}_i}{\sum_{i=1}^n \tilde{x}_i^2} \\
&= \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2} \\
&= \frac{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2} \\
&= \frac{\mathrm{s}_{xy}}{\mathrm{s}_x^2}.
\end{aligned}$$

Since the second derivative is $2\sum_{i=1}^n \tilde{x}_i^2 \geq 0$, this is indeed a minimum and the proof is complete. □

1.2 Linear Regression as a Parameter Estimation Problem

In statistics, any analysis starts by making a statistical model of the data. This is done by writing random variables which have the same structure as the data, and which are chosen so that the data "looks like" a random sample from these random variables.

To construct a model for the data used in a simple linear regression problem, we use random variables $Y_1, \dots, Y_n$ such that

$$Y_i = \beta_0 + \beta_1 x_i + \varepsilon_i \tag{3}$$

for all $i \in \{1, 2, \dots, n\}$, where $\varepsilon_1, \dots, \varepsilon_n$ are i.i.d. random variables with $\mathbb{E}(\varepsilon_i) = 0$ and $\mathrm{Var}(\varepsilon_i) = \sigma^2$.

- Here we assume that the x-values are fixed and known. The only random quantities in the model are ε_i and Y_i. (There are more complicated models which also allow for randomness of x, but we won't consider such models here.)
- The random variables ε_i are called **residuals** or **errors**. In a scatter plot, the residuals correspond to the vertical distance between the samples and the regression line. Often one assumes that $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$ for all $i \in \{1, 2, \dots, n\}$.
- The values β_0, β_1 and σ^2 are parameters of the model. To fit the model to data, we need to estimate these parameters.

This model is more complex than the models considered in some introductory courses to statistics:

- The data consists now of pairs of numbers, instead of just single numbers.

- We have

$$\mathbb{E}(Y_i) = \mathbb{E}(\beta_0 + \beta_1 x_i + \varepsilon_i) = \beta_0 + \beta_1 x_i + \mathbb{E}(\varepsilon_i) = \beta_0 + \beta_1 x_i.$$

  Thus, the expectation of Y_i depends on x_i and, at least for $\beta_1 \neq 0$, the random variables Y_i are not identically distributed.

In this setup, we can consider the estimates $\hat\beta_0$ and $\hat\beta_1$ from the previous subsection as statistcal parameter estimates for the model parameters β_0 and β_1.

In order to fit a linear model we also need to estimate the residual variance σ^2. This can be done using the estimator

$$\hat\sigma^2 = \frac{1}{n-2}\sum_{i=1}^n \hat\varepsilon_i^2 = \frac{1}{n-2}\sum_{i=1}^n (y_i - \hat\beta_0 - \hat\beta_1 x_i)^2. \tag{4}$$

To understand the form of this estimator, we have to remember that σ^2 is the variance of the ε_i. Thus, using the standard estimator for the variance, we could estimate σ^2 as

$$\sigma^2 \approx \frac{1}{n-1}\sum_{i=1}^n (\varepsilon_i - \bar\varepsilon)^2 \approx \frac{1}{n-1}\sum_{i=1}^n (\hat\varepsilon_i - \bar{\hat\varepsilon})^2, \tag{5}$$

where $\bar\varepsilon$ and $\bar{\hat\varepsilon}$ are the averages of the ε_i and the $\hat\varepsilon_i$, respectively. One can show that $\bar{\hat\varepsilon} = 0$. The estimates of β_0 and β_1 are sensitive to fluctuations in the data, with the effect that the estimated regression line is, on average, slightly closer to the data points than the true regression line would be. This causes the sample variance of the $\hat\varepsilon_i$, on average, to be slightly smaller than the true residual variance σ^2 and the thus the estimator (5) is slightly biased. A more detailed analysis reveals that an unbiased estimator can be obtained if one replaces the pre-factor $1/(n-1)$ in equation (5) with $1/(n-2)$. This leads to the estimator (4).

The main advantage gained by considering a statistical model is, that we now can consider how close the estimators $\hat\beta_0$, $\hat\beta_1$ and $\hat\sigma^2$ are to the true values. Results one can obtain include the following:

- The estimators $\hat\beta_0$, $\hat\beta_1$ and $\hat\sigma^2$ are unbiased: This means that when we plug in random data (x_i, Y_i) from the model (3), on average we get the correc answer: $\mathbb{E}(\hat\beta_0) = \beta_0$, $\mathbb{E}(\hat\beta_1) = \beta_1$, $\mathbb{E}(\hat\sigma^2) = \sigma^2$.
- One can ask about the average distance between the estimated parameters $\hat\beta_0$, $\hat\beta_1$ and $\hat\sigma^2$ and the (unknown) true values β_0, β_1 and σ^2. One measure for these distances is the root mean squared error of the estimators.
- One can consider confidence intervals for the parameters β_0, β_1 and σ^2.
- One can consider statistical hypothesis tests to answer yes/no questions about the parameters. For example, one might ask whether the data could have come from the model with $\beta_0 = 0$.
- One can consider whether the data is compatible with the model at all, irrespective of parameter values. If there is a non-linear relationship between x and y, the model (3) will no longer be appropriate.

We will consider most of these questions over the course of the module.

1.3 Matrix Notation

To conclude this section, we will rewrite the results of this section in a form which we will extensively use for multiple linear regression in the rest of this module. The idea here is to arrange all quantities in the problem as matrices and vectors in order to simplify notation. We write

$$X = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix} \in \mathbb{R}^{n\times 2}, \qquad y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \in \mathbb{R}^n, \qquad \varepsilon = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix} \in \mathbb{R}^n, \qquad \beta = \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} \in \mathbb{R}^2$$

Using this notation, we can rewrite the n equations $y_i = \beta_0 + x_i\beta_1 + \varepsilon_i$ for $i \in \{1, \dots, n\}$ as one vector-valued equation in $\mathbb{R}^n$: we get

$$y = X\beta + \varepsilon,$$

and we want to "solve" this vector-valued equation for β. The sum of squares can now be written as

$$r(\beta) = \sum_{i=1}^{n} \varepsilon_i^2 = \varepsilon^\top \varepsilon = (y - X\beta)^\top (y - X\beta) = y^\top y - 2\beta^\top X^\top y + \beta^\top X^\top X\beta.$$

In the next section we will see that the minimum of r is attained for

$$\hat{\beta} = (XX^\top)^{-1} X^\top y$$

and one can check that the components of this vector $\hat{\beta} = (\hat{\beta}_0, \hat{\beta}_1)$ coincide with the estimates we obtained above.

Summary

- simple linear regression is the case where there is only one input
- a regression line is fitted by minimising the residual sum of squares
- linear regression is a statistical parameter estimation problem
- the problem can be conveniently written in matrix/vector notation

2 Least Squares Estimates

2.1 Data and Models

For multiple linear regression we assume that there are p inputs and one output. If we have a sample of n obervations, we have np inputs and one output in total. Here we denote the ith observation of the jth input by x_{ij} and the corresponding output by y_j.

As an example, we consider the `mtcars` dataset built into R. This is a small dataset, which contains information about 32 automobiles (1973–74 models). The table lists fuel consumption `mpg`, gross horsepower `hp`, and 9 other aspects of these cars. Here we consider `mpg` to be the output, and the other listed aspects to be inputs. Type `help(mtcars)` in R to learn more about this dataset:

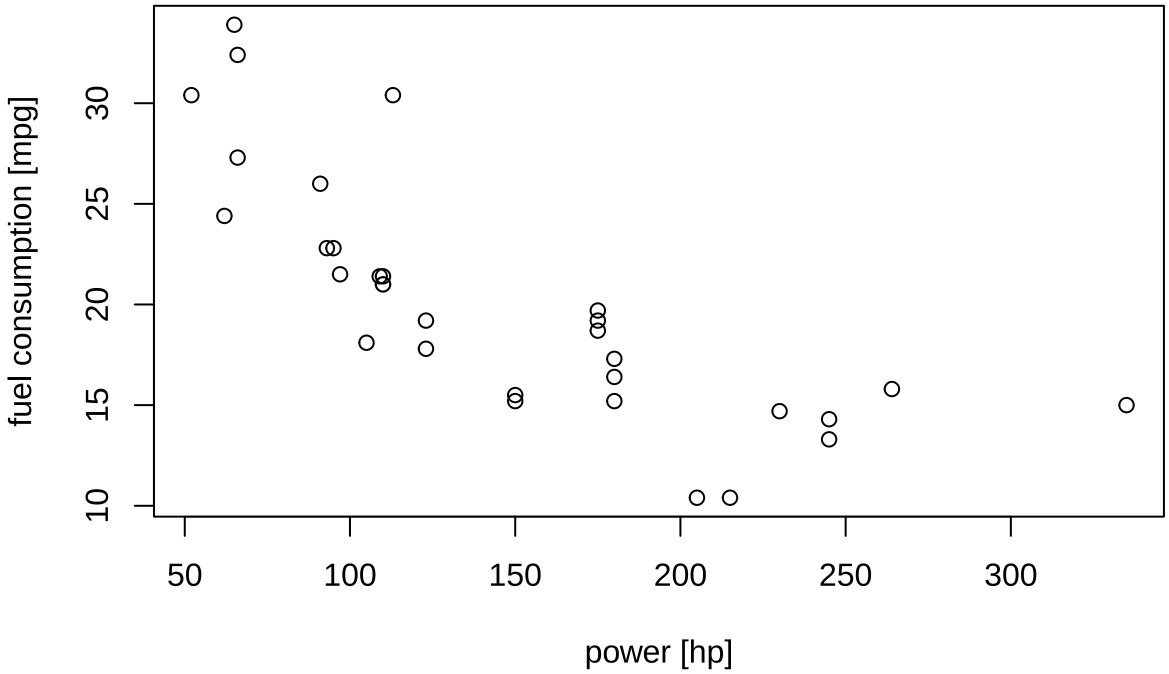
```
mtcars
```

```
##                      mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4           21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag       21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710          22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive      21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout   18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
## Valiant             18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
## Duster 360          14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
## Merc 240D           24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
## Merc 230            22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
## Merc 280            19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
## Merc 280C           17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
## Merc 450SE          16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
## Merc 450SL          17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
## Merc 450SLC         15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
## Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
## Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
## Chrysler Imperial   14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
## Fiat 128            32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
## Honda Civic         30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
## Toyota Corolla      33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
## Toyota Corona       21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
## Dodge Challenger    15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
## AMC Javelin         15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
## Camaro Z28          13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4
## Pontiac Firebird    19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
## Fiat X1-9           27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
## Porsche 914-2       26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
## Lotus Europa        30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
## Ford Pantera L      15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
## Ferrari Dino        19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
## Maserati Bora       15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8
## Volvo 142E          21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
```

For this dataset we have $n = 32$ (number of cars), and $p = 10$ (number of attributes, excluding `mpg`). The values $y_1, \ldots, y_{32}$ are listed in the first column of the table, the values $x_{i,1}$ for $i \in \{1, \ldots, 32\}$ are shown in the second column, and the values $x_{i,10}$ are shown in the last column.

In this data set it is easy to make scatter plots which show how a single input affects the output. For example, we can show how the engine power affects fuel consumption:

```
plot(mtcars$hp, mtcars$mpg,
     xlab = "power [hp]", ylab = "fuel consumption [mpg]")
```



We can see that cars with stronger engines tend to use more fuel (*i.e.* a gallon of fuel lasts for fewer miles; the curve goes down), but leaving out the other inputs omits a lot of information. It is not easy to make a plot which takes all inputs into account. Is is also not immediately obvious which of the variables are most important.

In linear regression, we assume that the output depends on the inputs in a linear (or more precisely, *affine*) way. We write this as

$$y_i = \beta_0 + \beta_1 x_{i,1} + \cdots + \beta_p x_{i,p} + \varepsilon_i \tag{6}$$

where the residuals ε_i are assumed to be “small”.

The parameters β_j can be interpreted as the expected change in the response y per unit change in x_j when all other regressor variables are held fixed. For this reason the parameters β_j (for $j = 1, \ldots, p$) are sometimes called *partial* regression coefficients.

This model describes a hyperplane in the $(p+1)$-dimensional space of the inputs x_j and the output y. The hyperplane is easily visualized when $p = 1$ (as a line in $\mathbb{R}^2$), and visualisation can be attempted for $p = 2$ (as a plane in $\mathbb{R}^3$) but is very hard for $p > 2$.

We defer making a proper statistical model for multiple linear regression until the next section.

2.2 The Normal Equations

Similar to what we did in Section 1.3, we rewrite the model using matrix notation. We define the vectors

$$y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \in \mathbb{R}^n, \qquad \varepsilon = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix} \in \mathbb{R}^n, \qquad \beta = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix} \in \mathbb{R}^{1+p}$$

as well as the matrix

$$X = \begin{pmatrix} 1 & x_{1,1} & \cdots & x_{1,p} \\ 1 & x_{2,1} & \cdots & x_{2,p} \\ \vdots & \vdots & & \vdots \\ 1 & x_{n,1} & \cdots & x_{n,p} \end{pmatrix} \in \mathbb{R}^{n \times (1+p)}.$$

The matrix X is often called the **design matrix**.

Using this notation, the model (6) can be written as

$$y = X\beta + \varepsilon, \tag{7}$$

where again $X\beta$ is a matrix-vector multiplication which "hides" the sums in equation (6), and (7) is an equation of vectors of size n, which combines the n individual equations from (6) for the different values of i.

To simplify notation, we index the columns of X by $0, 1, \ldots, p$ (instead of the more conventional $1, \ldots, p{+}1$), so that we can for example write

$$(X\beta)_i = \sum_{j=0}^{p} x_{i,j}\beta_j = \beta_0 + \sum_{j=1}^{p} x_{i,j}\beta_j.$$

As before, we find the regression coefficients by minimising the residual sum of squares:

$$\begin{aligned} r(\beta) &= \sum_{i=1}^{n} \varepsilon_i^2 \\ &= \sum_{i=1}^{n} \bigl(y_i - (\beta_0 + \beta_1 x_{i,1} + \cdots + \beta_p x_{i,p})\bigr)^2. \end{aligned}$$

In practice, this notation turns out to be cumbersome, and we will use matrix notation in the following proof.

Lemma 2.1. *Assume that the matrix $X^\top X \in \mathbb{R}^{(1+p)\times(1+p)}$ is invertible. Then the function $r(\beta)$ takes its minimum at the vector $\hat\beta \in \mathbb{R}^{p+1}$ given by*

$$\hat\beta = (X^\top X)^{-1} X^\top y.$$

Proof. Using the vector equation $\varepsilon = y - X\beta$, we can also write the residual sum of squares as

$$\begin{aligned} r(\beta) &= \sum_{i=1}^{n} \varepsilon_i^2 \\ &= \varepsilon^\top \varepsilon \\ &= (y - X\beta)^\top (y - X\beta) \\ &= y^\top y - y^\top X\beta - (X\beta)^\top y + (X\beta)^\top (X\beta). \end{aligned}$$

Using the linear algebra rules from Appendix A.2 we find that $y^\top X\beta = (X\beta)^\top y = \beta^\top X^\top y$ and $(X\beta)^\top (X\beta) = \beta^\top X^\top X\beta$. Thus we get

$$r(\beta) = y^\top y - 2\beta^\top X^\top y + \beta^\top X^\top X\beta.$$

Note that in this eqation X is a matrix, y and β are vectors, and $r(\beta)$ is a number.

To find the minimum of this function, we set all partial derivatives $\frac{\partial}{\partial\beta_i} r(\beta)$ equal to 0. Going through the terms in the formula for $r(\beta)$ we find: (1) $y^\top y$ does not depend on β, so we have $\frac{\partial}{\partial\beta_i} y^\top y = 0$ for all i, (2) we have

$$\frac{\partial}{\partial\beta_i} \beta^\top X^\top y = \frac{\partial}{\partial\beta_i} \sum_{j=1}^{p+1} \beta_j (X^\top y)_j = (X^\top y)_i$$

and (3) finally

$$\frac{\partial}{\partial\beta_i} \beta^\top X^\top X\beta = \frac{\partial}{\partial\beta_i} \sum_{j,k=1}^{p+1} \beta_j (X^\top X)_{j,k} \beta_k = 2\sum_{k=1}^{p+1} (X^\top X)_{i,k} \beta_k = 2\bigl((X^\top X)\beta\bigr)_i.$$

(Some care is needed, when checking that the middle equality sign in the previous equation is correct.) Combining these derivatives, we find

$$\frac{\partial}{\partial\beta_i} r(\beta) = 0 - 2(X^\top y)_i + 2\bigl(X^\top X\beta\bigr)_i \tag{8}$$

for all $i \in \{0, 1, \ldots, p\}$. At a local minimum of r, all of these partial derivatives must be zero and using a vector equation we find that a necessary condition for a minimum is

$$X^\top X\beta = X^\top y. \tag{9}$$

Since we assumed that $X^\top X$ is invertible, there is exactly one vector *beta* which solves (9). This vector is given by

$$\hat\beta := (X^\top X)^{-1} X^\top y.$$

As for one-dimensional minimisation, there is a condition on the second derivatives which must be checked to see which local extrema are local minima. Here we are only going to sketch this argument: A sufficient condition for $\hat\beta$ to be a minimum is for the second derivative matrix (the Hessian matrix) to be positive definite (see appendix A.2.6). Using equation (8) we find

$$\frac{\partial}{\partial\beta_i \partial\beta_j} r(\beta) = 2(X^\top X)_{i,j}$$

And thus the Hessian matrix is $H = 2X^\top X$. Using results from linear algebra, one can show that this matrix is indeed positive definite and thus $\hat\beta$ is the unique minimum of r. □

Equation (9) gives a system of $p+1$ linear equations with $p+1$ unknowns. This system of linear equations, $X^\top X\beta = X^\top y$ is called the **normal equations**. If $X^\top X$ is invertible, as assumed in the lemma, this system of equations has $\hat\beta$ as its unique solution. Otherwise, there may be more than one β which leads to the same value $r(\beta)$ and the minimum will no longer be unique. This happens for example, if two of the inputs are identical to each other (or, more generally, one input is linearly dependent on one or more other inputs).

The condition that $X^\top X$ must be invertible in multiple linear regression corresponds to the condition $\mathrm{s}_x^2 > 0$ from lemma 1.1 for simple linear regression.

The value $\hat\beta$ found in the lemma is called the **least squares estimator** for β, or sometimes the ordinary least squares (OLS) estimator.

2.3 Fitted Values

Let us again consider our model

$$y = X\beta + \varepsilon,$$

using the matrix notation introduced above. Here we can think of $X\beta$ as the **true values**, while ε are the errors. The design matrix X (containing the inputs) and the response y are known to us, while the true coefficients β and the errors ε are unknown. Solving for ε we find that the errors satisfy

$$\varepsilon = y - X\beta.$$

Using the least squares estimate $\hat\beta$ we can estimate the true values as

$$\hat y = X\hat\beta. \tag{10}$$

These estimates are called the **fitted values**. Using the definition of $\hat\beta$ we get

$$\hat y = X(X^\top X)^{-1} X^\top y =: Hy.$$

The matrix $H = X(X^\top X)^{-1} X^\top$ is commonly called the **hat matrix** (because it "puts the hat on y").

Finally, we can estimate the errors using the residuals

$$\hat\varepsilon = y - X\hat\beta = y - \hat y = y - Hy = (I - H)y, \tag{11}$$

where I is the $(p+1) \times (p+1)$ identity matrix.

2.4 Example

To conclude this section, we demonstrate how these methods can be used in R. For this we consider the `mtcars` example from the beginning of the section again. I will first show how to do the analysis "by hand", and later show how the same result can be obtained using R's built-in functions.

We first split `mtcars` into the respons column `y` (the first column) and the design matrix `X` (a column of ones, followed by columns 2 to 11 of `mtcars`):

```
y <- mtcars[, 1]
X <- cbind(1, data.matrix(mtcars[, 2:11]))
```

Next we compute $X^\top X$ and solve the normal equations. Often it is faster, easier, and has lower numerical errors to solve the normal equations rather than inverting the matrix $X^\top X$.

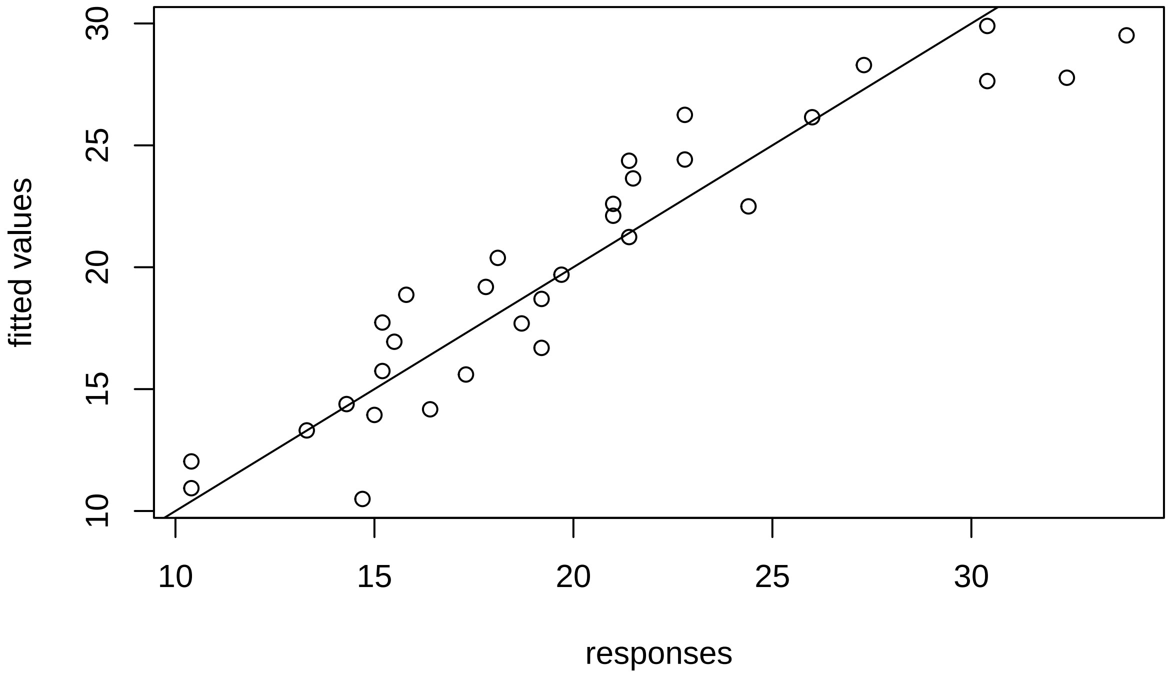
```
XtX <- t(X) %*% X
beta.hat <- solve(XtX, t(X) %*% y)
beta.hat
```

```
##              [,1]
##       12.30337416
## cyl   -0.11144048
## disp   0.01333524
## hp    -0.02148212
## drat   0.78711097
## wt    -3.71530393
## qsec   0.82104075
## vs     0.31776281
## am     2.52022689
## gear   0.65541302
## carb  -0.19941925
```

Without further checks it is hard to know whether the result is correct, or whether we made a mistake somewhere along the lines. One good sign is that we argued earlier that higher `hp` should lead to lower `mpg`, and indeed the corresponding coefficient `-0.02148212` is negative.

Finally, compute the fitted values and generate a plot of fitted values against responses. If everything worked, we would expect the points in this plot to be close to the diagonal.

```
y.hat <- X %*% beta.hat
plot(y, y.hat, xlab = "responses", ylab = "fitted values")
abline(a = 0, b = 1) # plot the diagonal
```



For comparison we now re-do the analysis using built-in R commands. In the `lm()` command below, we use `data=mtcars` to tell R where the data is stored, and the formula `mpg ~ .` states that we want to model `mpg` as a function of all other variable (this is the meaning of `.`).

```
m <- lm(mpg ~ ., data = mtcars) # fit a linear model
coef(m) # get the estimated coefficients
```

```
## (Intercept)         cyl        disp          hp        drat          wt 
## 12.30337416 -0.11144048  0.01333524 -0.02148212  0.78711097 -3.71530393 
##        qsec          vs          am        gear        carb 
##  0.82104075  0.31776281  2.52022689  0.65541302 -0.19941925
```

Comparing these coefficients to the vector `beta.hat` from above shows that we got the same result using both methods. The fitted values can be computed using `fitted.values(m)`. Here we just check that we get the same result as above:

```
max(abs(fitted.values(m) - y.hat))
```

```
## [1] 5.329071e-13
```

This results `5.329071e-13` stands for the number $5.329071 \cdot 10^{-13}$, which is extremely small. The difference between our results and R's result is caused by rounding errors.

Summary

- multiple linear regression allows for more than one input but still has only one output
- the least squared estimate for the coefficients is found by minimising the residual sum of squares
- the estimate can be computed as the solution to the normal equations
- the hat matrix transforms responses into fitted values

Interlude: Linear Regression in R

Fitting a Model

The function `lm()` is used to fit a linear model in R. There are different ways to specify the form of the model and the data to be used for fitting the model.

- The most basic way to call `lm()` is the case where the explanatory variables and the response variable are stored as separate vectors. Assuming, for example, that the explanatory variables are `x1`, `x2`, `x3` and that the response variable is `y` in R, we can tell R to fit the linear model $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \varepsilon$ by using the following command:

  ```
  lm(y ~ x1 + x2 + x3)
  ```

  Note that R automatically added the intercept term β_0 to this model. If we want to fit a model without an intercept, *i.e.* the model $y = \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \varepsilon$, we have to add `0 +` in front of the explanatory variables:

  ```
    lm(y ~ 0 + x1 + x2 + x3)
  ```

  The general form of a model specification is the response variable, followed by `~`, followed by a plus-separated list of explanatory variables. For this form of calling `lm()`, the variables `y`, `x1`, `x2`, and `x3` in the examples above must be already defined before `lm()` is called. It may be a good idea to double-check that the variables have the correct values before trying to call `lm()`.

- Both for the response and for explanatory variables we can specify arbitrary R expressions to compute the numeric values to be used. For example, to fit the model $\log(y) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \varepsilon$ (assuming that all y_i are positive) we can use the following command:

  ```
  lm(log(y) ~ x1 + x2)
  ```

  Some care is needed, because `+`, `*` and `^` have a special meaning inside the first argument of `lm()`; any time we want to compute a variable for `lm()` using these operations, we need to surround the corresponding expression with `I()`, to tell R that `+`, `*` or `^` should have their usual, arithmetic meaning. For example, to fit a model of the form $y \sim \beta_0 + \beta_1 x + \beta_2 x^2 + \varepsilon$, we can use the following R command:

  ```
  lm(y ~ x + I(x^2))
  ```

  Here, the use of `I()` tells R that `x^2` is to be interpreted as the vector $(x_1^2, \dots, x_n^2)$. Similarly, we can fit a model of the form $y = \beta_0 + \beta_1(x_1 + x_2) + \varepsilon$:

  ```
  lm(y ~ I(x1+x2))
  ```

  Here, the use of `I()` tells R that `x1+x2` indicates the vector $(x_{1,1} + x_{2,1}, \dots, x_{1,n} + x_{2,n})$ instead of two separate explanatory variables.

  Details about how to specify models in calls to `lm()` can be found by using the command `help(formula)` in R.

- If the response and the explanatory variables are stored in the columns of a data frame, we can use the `data=...` argument to `lm()` to specify this data frame and then just use the column names to specify the regression model. For example, the `stackloss` data set built into R consists of a data frame with columns `Air.Flow`, `Water.Temp`, `Acid.Conc.`, `stack.loss`. To predict `stackloss$stack.loss` from `stackloss$Air.Flow` we can write

  ```
  lm(stack.loss ~ Air.Flow, data=stackloss)
  ```

  As a special case, a single dot "`.`" can be used in place of the explanatory variables in the model to indicate that all columns except for the given response should be used. Thus, the following two commands are equivalent:

  ```
  lm(stack.loss ~ ., data=stackloss)
  lm(stack.loss ~ Air.Flow + Water.Temp + Acid.Conc., data=stackloss)
  ```

Understanding the Model

The output of the `lm()` function is an R object which can be used the extract information about the fitted model. A good way to work with this object is to store it in a variable and then use commands like the ones listed below to work with this variable. For example, the following R command fits a model for the `stackloss` data set and stores it in the variable `m`:

```
m <- lm(stack.loss ~ ., data=stackloss)
```

Many operations are available to use with this object `m`:

- Printing `m` to the screen:

  ```
  m
  ```

  ```
  ##
  ## Call:
  ## lm(formula = stack.loss ~ ., data = stackloss)
  ##
  ## Coefficients:
  ## (Intercept)     Air.Flow   Water.Temp   Acid.Conc.
  ##    -39.9197       0.7156       1.2953      -0.1521
  ```

  This shows the estimated values for the regression coefficient.

- The command `summary()` can be used to print additional information about the fitted model:

  ```
  summary(m)
  ```

  ```
  ##
  ## Call:
  ## lm(formula = stack.loss ~ ., data = stackloss)
  ##
  ## Residuals:
  ##     Min      1Q  Median      3Q     Max
  ## -7.2377 -1.7117 -0.4551  2.3614  5.6978
  ##
  ## Coefficients:
  ##             Estimate Std. Error t value Pr(>|t|)
  ## (Intercept) -39.9197    11.8960  -3.356  0.00375 **
  ## Air.Flow      0.7156     0.1349   5.307  5.8e-05 ***
  ## Water.Temp    1.2953     0.3680   3.520  0.00263 **
  ## Acid.Conc.   -0.1521     0.1563  -0.973  0.34405
  ## ---
  ## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
  ##
  ## Residual standard error: 3.243 on 17 degrees of freedom
  ## Multiple R-squared:  0.9136, Adjusted R-squared:  0.8983
  ## F-statistic:  59.9 on 3 and 17 DF,  p-value: 3.016e-09
  ```

  We will learn over the course of this module how to interpret most of this output.

- The coefficient vector β can be obtained using `coef(m)`:

  ```
  coef(m)
  ```

  ```
  ## (Intercept)    Air.Flow  Water.Temp  Acid.Conc.
  ## -39.9196744   0.7156402   1.2952861  -0.1521225
  ```

- The fitted values $\hat{y}_i$ can be obtained using the command `fitted(m)`:

  ```
  fitted(m)
  ```

  ```
  ##         1         2         3         4         5         6         7         8
  ## 38.765363 38.917485 32.444467 22.302226 19.711654 21.006940 21.389491 21.389491
  ##         9        10        11        12        13        14        15        16
  ```

```
## 18.144379 12.732806 11.363703 10.220540 12.428561 12.050499  5.638582  6.094949
##        17        18        19        20        21
##  9.519951  8.455093  9.598257 13.587853 22.237713
```

- The estimated residuals $\hat{\varepsilon}_i = y_i - \hat{y}_i$ can be obtained using the command `resid(m)`:

```
resid(m)
```

```
##           1           2           3           4           5           6
##  3.23463723 -1.91748529  4.55553300  5.69777417 -1.71165358 -3.00693970
##           7           8           9          10          11          12
## -2.38949071 -1.38949071 -3.14437890  1.26719408  2.63629676  2.77946036
##          13          14          15          16          17          18
## -1.42856088 -0.05049929  2.36141836  0.90505080 -1.51995059 -0.45509295
##          19          20          21
## -0.59825656  1.41214728 -7.23771286
```

- The design matrix X can be found using `model.matrix(m)`:

```
model.matrix(m)
```

```
##    (Intercept) Air.Flow Water.Temp Acid.Conc.
## 1            1       80         27         89
## 2            1       80         27         88
## 3            1       75         25         90
## 4            1       62         24         87
## 5            1       62         22         87
## 6            1       62         23         87
## 7            1       62         24         93
## 8            1       62         24         93
## 9            1       58         23         87
## 10           1       58         18         80
## 11           1       58         18         89
## 12           1       58         17         88
## 13           1       58         18         82
## 14           1       58         19         93
## 15           1       50         18         89
## 16           1       50         18         86
## 17           1       50         19         72
## 18           1       50         19         79
## 19           1       50         20         80
## 20           1       56         20         82
## 21           1       70         20         91
## attr(,"assign")
## [1] 0 1 2 3
```

Making Predictions

One of the main aims of fitting a linear model is to use the model to make predictions for new, not previously observed x-values, *i.e.* to compute $y_{\text{new}} = X_{\text{new}}\hat{\beta}$. The general form of the command for prediction is `predict(m, newdata)`, where `m` is the model previously fitted using `lm()`, and `newdata` specifies the new x-values to predict responses for. The argument `new.data` should be a `data.frame` and for each variable in the original model there should be a column in `newdata` which has the name of the original variable and contains the new values. For example, if the model was fitted using

```
m <- lm(y ~ x + I(x^2))
```

and if the new samples are stored in `x.new`, then responses for the x-values in `x.new` can be predicted using the following command:

```
predict(m, data.frame(x=x.new))
```

As a second example, for the `stackloss` data set, the following commands can be used to predict `stack.loss` for two new x-values:

```
m <- lm(stack.loss ~ ., data=stackloss)
new.data <- data.frame(Air.Flow=c(70, 73), Water.Temp=c(25,24), Acid.Conc.=c(78,90))
predict(m, new.data)
```

```
##        1        2
## 30.69174 29.71790
```

More information about `predict()` can be found by reading the output of `help(predict.lm)`.

Problem Sheet 1

You should attempt all these questions and write up your solutions in advance of your workshop in week 3 where the answers will be discussed.

1 Consider the simple linear regression model $y_i = \beta_0 + x_i\beta_1 + \varepsilon_i$ for $i \in \{1, 2, \dots, n\}$ and let X be the design matrix.

a. Show that $X^\top X = \begin{pmatrix} n & \sum_{i=1}^n x_i \\ \sum_{i=1}^n x_i & \sum_{i=1}^n x_i^2 \end{pmatrix} \in \mathbb{R}^{2\times 2}$.

b. Using the formula

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} = \frac{1}{ad - bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix},$$

find $(X^\top X)^{-1}$.

c. Find $X^\top y$ and use this to derive an explicit formula for the least squares estimate $\hat\beta = (X^\top X)^{-1} X^\top y$.

2 Let $H = X(X^\top X)^{-1}X^\top \in \mathbb{R}^{n\times n}$ be the hat matrix and $\mathbf{1} = (1, 1, \dots, 1) \in \mathbb{R}^n$. Show that $H\mathbf{1} = \mathbf{1}$.

3 For the `stackloss` data set built into R, predict a value for `stack.loss` when the inputs are `Air.Flow = 60`, `Water.Temp = 21` and `Acid.Conc = 87`.

4 Let $\varepsilon_1, \dots, \varepsilon_n \sim \mathcal{N}(\mu, \sigma^2)$ be independent. Then $\varepsilon = (\varepsilon_1, \dots, \varepsilon_n)$ is a random vector. Determine $\mathbb{E}(\varepsilon)$, $\mathrm{Cov}(\varepsilon)$ and $\mathbb{E}(\|\varepsilon\|^2)$.

3 Random Vectors and Covariance

Like in the one-dimensional case, we can build a **statistical model** for the data where we assume that the errors are random. More precisely we will assume

$$Y_i = \beta_0 + \beta_1 x_{i,1} + \cdots + \beta_p x_{i,p} + \varepsilon_i \tag{12}$$

for all $i \in \{1, 2, \dots, n\}$, where $\varepsilon_1, \dots, \varepsilon_n$ are now independent and identically distributed (i.i.d.) random variables with $\mathbb{E}(\varepsilon_i) = 0$ and $\mathrm{Var}(\varepsilon_i) = \sigma^2$. As in (7), the statistical model can be written in vector form as

$$Y = X\beta + \varepsilon. \tag{13}$$

This is a vector-valued equation which contains two “random vectors”, Y and ε.

A **random vector** is a vector $Z = (Z_1, \dots, Z_n)$ where each component Z_i is a random variable.

3.1 Expectation

The expectation of a random vector is taken for each component separately. This is formalised in the following definition.

Definition 3.1. Let $Z = (Z_1, \dots, Z_n) \in \mathbb{R}^n$ be a random vector. Then the expectation of Z is the (non-random) vector

$$\mathbb{E}(X) = \begin{pmatrix} \mathbb{E}(Z_1) \\ \vdots \\ \mathbb{E}(Z_n) \end{pmatrix} \in \mathbb{R}^n.$$

The same convention is sometimes used for random matrices M, as $\mathbb{E}(M)_{ij} = \mathbb{E}(M_{ij})$.

Example 3.1. The random vector ε in (13) has

$$\mathbb{E}(\varepsilon)_i = \mathbb{E}(\varepsilon_i) = 0$$

for all $i \in \{1, \dots, n\}$ and thus $\mathbb{E}(\varepsilon) = 0 \in \mathbb{R}^n$, where 0 here denotes the zero-vector $(0, \dots, 0) \in \mathbb{R}^n$.

Since the expectation of a random vector is defined in term of the usual expectation, most rules we know for expectations still hold. For example, if Y and Z are two random vectors, we have $\mathbb{E}(Y + Z) = \mathbb{E}(Y) + \mathbb{E}(Z)$.

Example 3.2. The random vector Y in (13) has

$$\mathbb{E}(Y)_i = \mathbb{E}(Y_i) = \mathbb{E}\big((X\beta)_i + \varepsilon_i\big) = (X\beta)_i + \mathbb{E}(\varepsilon_i) = (X\beta)_i$$

for all $i \in \{1, \dots, n\}$ and thus $\mathbb{E}(Y) = X\beta \in \mathbb{R}^n$. We often will write the above derivation in vector form as

$$\mathbb{E}(Y) = \mathbb{E}(X\beta + \varepsilon) = X\beta + \mathbb{E}(\varepsilon) = X\beta.$$

Example 3.3. If $A \in \mathbb{R}^{m \times n}$ is a matrix and $Z \in \mathbb{R}^n$ is a random vector, then we find the expectation of $AZ \in \mathbb{R}^m$ as

$$\mathbb{E}(AZ)_i = \mathbb{E}(AZ_i) = \mathbb{E}(\sum_{j=1}^n a_{ij} Z_j) = \sum_{j=1}^n \mathbb{E}(a_{ij} Z_j) = \sum_{j=1}^n a_{ij} \mathbb{E}(Z_j) = \sum_{j=1}^n a_{ij} \mathbb{E}(Z)_j$$

for all $i \in \{1, \dots, m\}$ and thus we have $\mathbb{E}(AZ) = A\mathbb{E}(Z)$.

3.2 Covariance Matrix

The variance of random variables is replaced with the concept of a “covariance matrix” for random vectors.

Definition 3.2. Let $Z = (Z_1, \dots, Z_n) \in \mathbb{R}^n$ be a random vector. Then the covariance matrix of Z is the matrix $\mathrm{Cov}(Z) \in \mathbb{R}^{n \times n}$ given by

$$\mathrm{Cov}(Z)_{ij} = \mathrm{Cov}(Z_i, Z_j),$$

for all $i, j \in \{1, \dots, n\}$, where $\mathrm{Cov}(Z_i, Z_j)$ denotes the usual covariance between random variables.

We collect some basic properties of covariance matrices here. Most of these arguments use concepts and rules from linear algebra, as summarised in section A in the appendix.

- Since $\mathrm{Cov}(Z_i, Z_j) = \mathrm{Cov}(Z_j, Z_i)$, covariance matrices are symmetric.
- The diagonal elements of $\mathrm{Cov}(Z)$ are
$$\mathrm{Cov}(Z)_{ii} = \mathrm{Cov}(Z_i, Z_i) = \mathrm{Var}(Z_i). \tag{14}$$
- If the elements Z_i of Z are (statistically) independent, we have $\mathrm{Cov}(Z_i, Z_j) = 0$ and thus $\mathrm{Cov}(Z)_{ij} = 0$ for $i \neq j$. If Z is a vector of independent random variables, the covariance matrix of Z is diagonal.
- Let $\mu = \mathbb{E}(Z) \in \mathbb{R}^n$. If we interpret μ as a column vector, then $M = (Z-\mu)(Z-\mu)^\top$ is an $n \times n$ matrix and we have
$$M_{ij} = \bigl((Z-\mu)(Z-\mu)^\top\bigr)_{ij} = (Z-\mu)_i (Z-\mu)_j.$$
Taking expectations gives $\mathbb{E}(M_{ij}) = E\bigl((Z-\mu)_i(Z-\mu)_j\bigr) = \mathrm{Cov}(Z_i, Z_j)$ and thus we can write
$$\mathrm{Cov}(Z) = \mathbb{E}\bigl((Z-\mu)(Z-\mu)^\top\bigr). \tag{15}$$
- Covariance matrices are positive semi-definite. To see this, let $C = \mathrm{Cov}(Z)$ and $u \in \mathbb{R}^n$ be a vector. We have to show that $u^\top C u \geq 0$. Writing $\bar Z := Z - \mathbb{E}(Z)$ as an abbreviation, we get
$$\begin{aligned} u^\top C u &= u^\top \mathbb{E}(\bar Z \bar Z^\top) u \\ &= \mathbb{E}(u^\top \bar Z \bar Z^\top u) \\ &= \mathbb{E}\bigl((\bar Z^\top u)^\top \bar Z^\top u\bigr) \\ &= \mathbb{E}(\|\bar Z^\top u\|^2), \end{aligned}$$
where $\|\bar Z^\top u\|$ denotes the Euclidean length of the vector $\bar Z^\top u$. Since $\|\bar Z^\top u\|^2 \geq 0$ we find $u^\top C u \geq 0$. This shows that the covariance matrix C is positive semi-definite. (Note that, nevertheless, individual *elements* of the matrix C can be negative numbers.)

Example 3.4. The random vector ε in equation (13) has $\mathbb{E}(\varepsilon) = 0$. We have $\mathrm{Cov}(\varepsilon)_{ii} = \mathrm{Var}(\varepsilon_i) = \sigma^2$ for all $i \in \{1, \ldots, n\}$. Since we assumed the ε_i to be independent, the covariance matrix is diagonal and we find
$$\mathrm{Cov}(\varepsilon) = \sigma^2 I,$$
where I is the $n \times n$ identity matrix.

An important results about covariance matrices is given in the following lemma, which describes how the covariance matrix changes under affine transformations.

Lemma 3.1. *Let $Z \in \mathbb{R}^n$ be a random vector, $A \in \mathbb{R}^{m \times n}$ a matrix and $b \in \mathbb{R}^m$ a vector. Then*
$$\mathrm{Cov}(AZ + b) = A\,\mathrm{Cov}(Z) A^\top.$$

Proof. As in equation (15), we can write $\mathrm{Cov}(AZ+b)$ as
$$\mathrm{Cov}(AZ+b) = \mathbb{E}\bigl((AZ+b-\mu)(AZ+b-\mu)^\top\bigr),$$
where $\mu = \mathbb{E}(AZ+b) = \mathbb{E}(AZ) + b$. Thus, $AZ + b - \mu = AZ - \mathbb{E}(AZ)$ and we find
$$\begin{aligned} \mathrm{Cov}(AZ+b) &= \mathbb{E}\bigl((AZ - \mathbb{E}(AZ))(AZ - \mathbb{E}(AZ))^\top\bigr) \\ &= \mathrm{Cov}(AZ). \end{aligned}$$
This shows that the covariance matrix ignores non-random shifts.

Furthermore, we have $AZ - \mathbb{E}(AZ) = AZ - A\mathbb{E}(Z) = A(Z - \mathbb{E}(Z))$. Using equation (15) again, we find
$$\begin{aligned} \mathrm{Cov}(AZ) &= \mathbb{E}\Bigl((AZ - \mathbb{E}(AZ))(AZ - \mathbb{E}(AZ))^\top\Bigr) \\ &= \mathbb{E}\Bigl(A(Z - \mathbb{E}(Z))(Z - \mathbb{E}(Z))^\top A^\top\Bigr) \\ &= A\mathbb{E}\Bigl((Z - \mathbb{E}(Z))(Z - \mathbb{E}(Z))^\top\Bigr)A^\top \\ &= A\,\mathrm{Cov}(Z) A^\top. \end{aligned}$$
This completes the proof. □

3.3 The Multivariate Normal Distribution

Since we assume that the random errors ε_i are normally distributed, we will need to understand how vectors of normal distributed random variables behave.

Definition 3.3. A random vector $Z \in \mathbb{R}^n$ follows a **multivariate normal distribution**, if $u^\top Z$ is normally distributed or constant for every vector $u \in \mathbb{R}^n$.

This definition is takes its slightly surprising form to avoid some boundary cases which I will discuss in an example, below. To understand the definition, a good start is to consider the cases where u is one of the standard basis vectors, say $u_i = 1$ and $u_j = 0$ for all $j \neq i$. In this case we have

$$u^\top Z = \sum_{k=1}^{n} u_k Z_k = Z_i.$$

Thus, if Z follows a multivariate normal distribution, each of the components Z_i is normally distributed. Example 3.8, below, shows that the converse is not true.

One can show that a multivariate normal distribution is completely determined by the mean $\mu = \mathbb{E}(Z)$ and the covariance $\Sigma = \mathrm{Cov}(Z)$. The distribution of such a Z is denoted by $\mathcal{N}(\mu, \Sigma)$. Also, for every $\mu \in \mathbb{R}^n$ and every positive semi-definite matrix $\Sigma \in \mathbb{R}^{n\times n}$ there is a random vector Z which follows a multivariate normal distribution with this mean and covariance.

Example 3.5. Consider the vector ε from the model (13). This vector has components $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$ and by assumption, the componens ε_i are independent. For $u \in \mathbb{R}^n$ we have

$$u^\top \varepsilon = \sum_{i=1}^{n} u_i \varepsilon_i.$$

Since this is a sum of independent, one-dimensional, normally distributed random variables, $u^\top \varepsilon$ is also normally distribution, for every u. (The independence of the ε_i is important in this argument.) Thus, ε is a normally distributed random vector. We have already seen $\mathbb{E}(\varepsilon) = 0$ and $\mathrm{Cov}(\varepsilon) = \sigma^2 I$, and thus $\varepsilon \sim \mathcal{N}(0, \sigma^2 I)$.

Without proof we state here some properties of the multivariate normal distribution:

- If $Z \sim \mathcal{N}(\mu, \Sigma)$ and $a \in \mathbb{R}^n$, then $Z + a \sim \mathcal{N}(\mu + a, \Sigma)$.
- If $Z \sim \mathcal{N}(\mu, \Sigma)$ and $A \in \mathbb{R}^{m\times n}$, then $AZ \sim \mathcal{N}(A\mu, A\Sigma A^\top)$.
- If $Z_1 \sim \mathcal{N}(\mu_1, \Sigma_1)$ and $Z_2 \sim \mathcal{N}(\mu_2, \Sigma_2)$ are independent, then $Z_1 + Z_2 \sim \mathcal{N}(\mu_1 + \mu_2, \Sigma_1 + \Sigma_2)$.

Example 3.6. Let $Z = (Z_1, Z_2)$ where Z_1 and Z_2 are independently standard normal distributed. Let

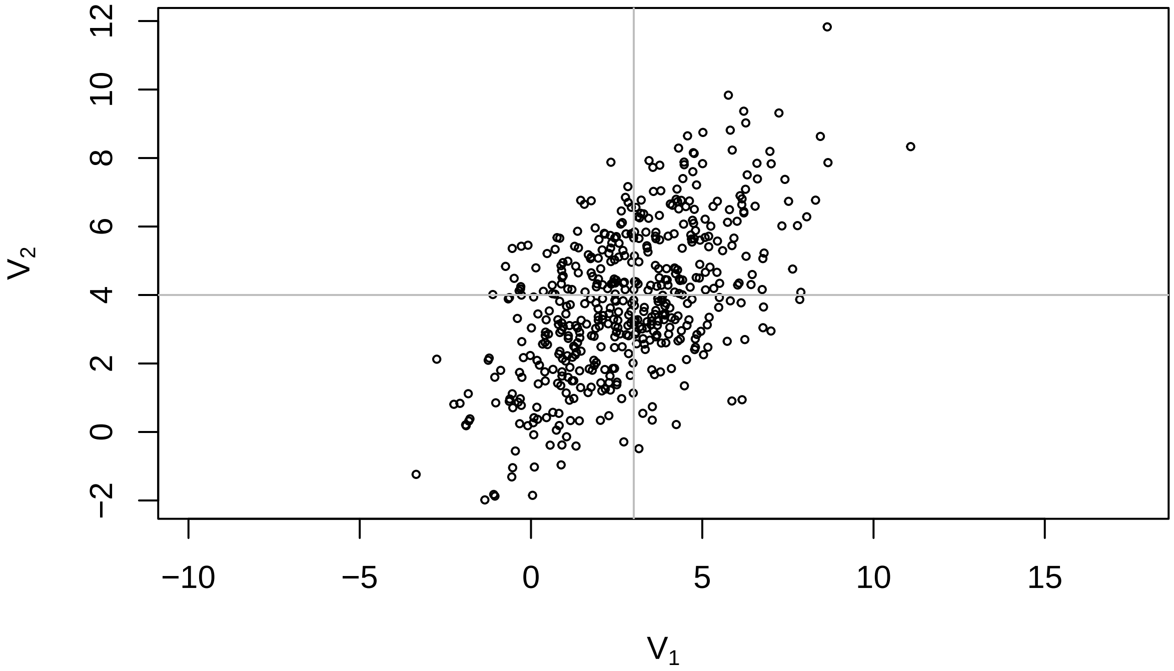
$$A := \begin{pmatrix} 2 & -1 \\ 2 & 1 \end{pmatrix} \qquad \text{and} \qquad b := \begin{pmatrix} 3 \\ 4 \end{pmatrix}.$$

Then $AZ + b \sim \mathcal{N}(b, \Sigma)$ where

$$\Sigma = A\,\mathrm{Cov}(Z) A^\top = \begin{pmatrix} 2 & -1 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 2 \\ -1 & 1 \end{pmatrix} = \begin{pmatrix} 5 & 3 \\ 3 & 5 \end{pmatrix}$$

We can use R to plot a sample of this two-dimensional normal distribution. (The grey cross indicates the mean.)

```r
N <- 500
Z <- rbind(rnorm(N), rnorm(N))
A <- matrix(c(2, 2, -1, 1), 2, 2)
b <- c(3, 4)
V <- A %*% Z + b
plot(V[1,], V[2,], asp = 1, cex = .5,
     xlab = expression(V[1]),
     ylab = expression(V[2]))
abline(v = 3, col = "grey")
abline(h = 4, col = "grey")
```



Example 3.7. The random vector Y in (13) satisfies $Y = X\beta + \varepsilon$ and since $\varepsilon \sim \mathcal{N}(0, \sigma^2 I)$ we have $Y \sim \mathcal{N}(X\beta, \sigma^2 I)$.

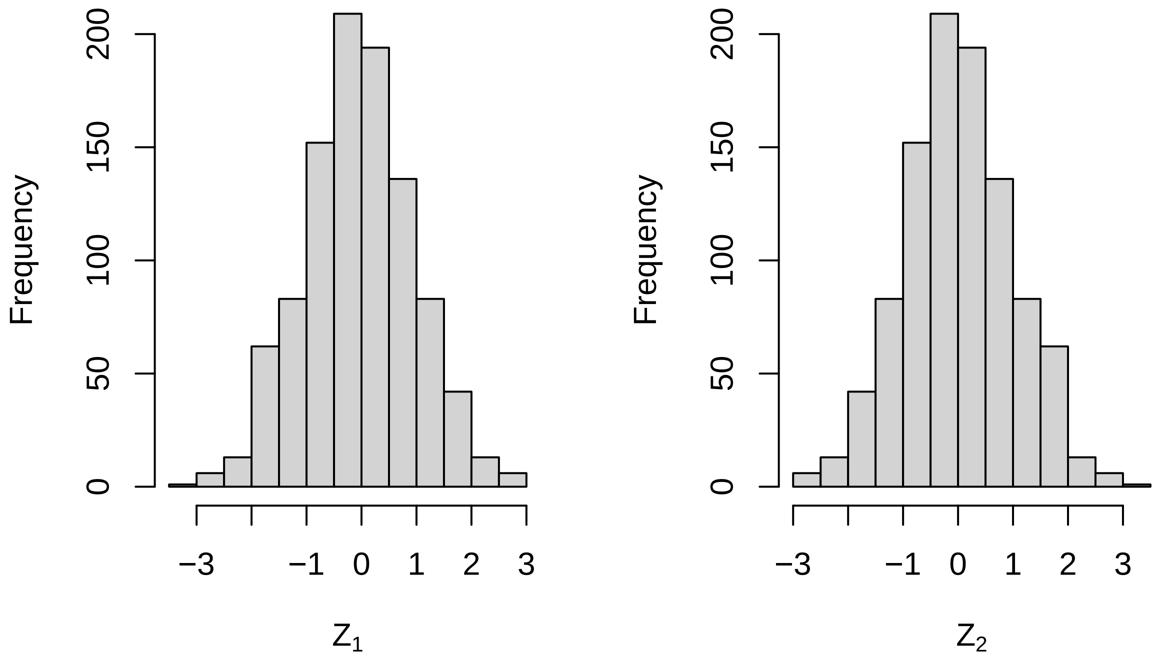
Example 3.8. Let $Y \sim \mathcal{N}(0, 1)$ be a random variable. Define a random vector $Z = (Z_1, Z_2)$ as $Z_1 = Y$ and

$$Z_2 = \begin{cases} Y & \text{if } |Y| < 1, \text{ and} \\ -Y & \text{otherwise.} \end{cases}$$

Clearly Z_1 is standard normally distributed. Since $\mathcal{N}(0, 1)$ is symmetric, both Y and $-Y$ are standard normally distributed and it follows that Z_2 is also standard normally distributed. Nevertheless, the random vector Z does not follow a multivariate normal distribution. Instead of giving a proof of this fact, we illustrate this here using an R experiment. We start by verifying that Z_1 and Z_2 are normally distributed.

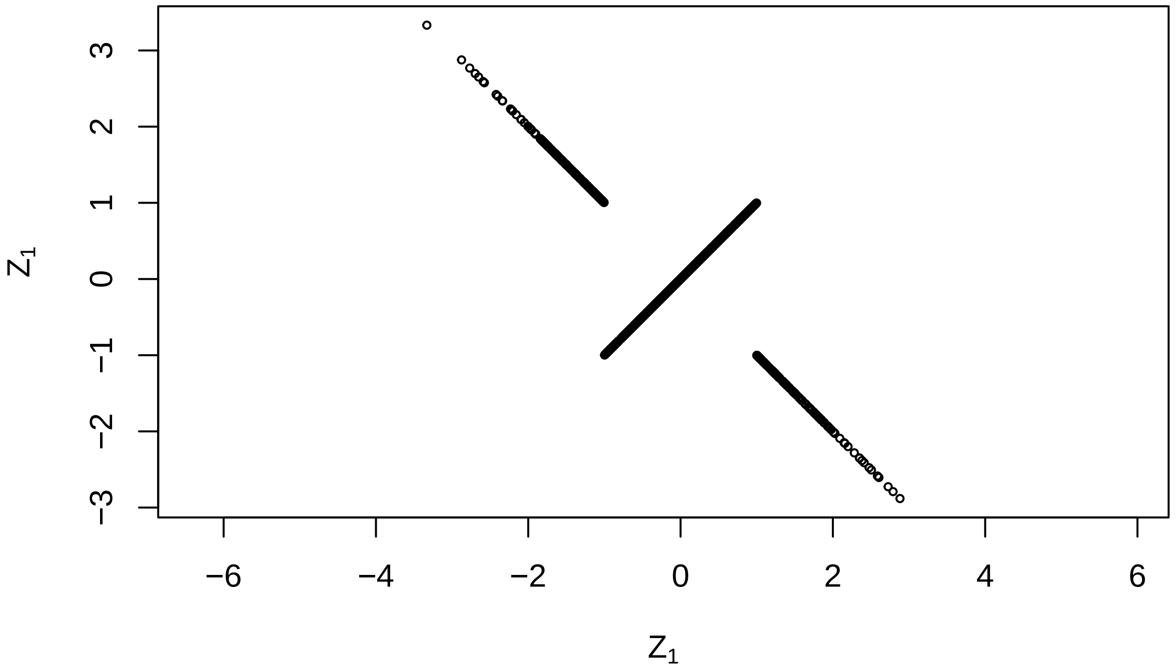
```
N <- 1000
Y <- rnorm(N)
Z1 <- Y
Z2 <- ifelse(abs(Y)<1, Y, -Y)

par(mfrow=c(1,2))
hist(Z1, main=NULL, xlab=expression(Z[1]))
hist(Z2, main=NULL, xlab=expression(Z[2]))
```



The histograms make it plausible that the components are indeed normally distributed. Now we use a scatter plot to show the joint distribution of Z_1 and Z_2:

```
plot(Z1, Z2, cex=0.5, asp=1,
     xlab=expression(Z[1]),
     ylab=expression(Z[1]))
```



This plot looks peculiar! Most people would not call this a normal distribution and the formal definition of a multivariate normal distribution is made to exclude cases like this.

Summary

- We learned the rules for computing the expectation of a random vector.
- The covariance matrix of random vectors plays the role of the variance for numeric random variables.
- We learned about the definition of the multivariate normal distribution.

4 Properties of the Least Squares Estimate

Like in the one-dimensional case, we can build a **statistical model** for the data. Here we assume that the residuals are random. More precisely we have

$$Y = X\beta + \varepsilon. \tag{16}$$

for all $i \in \{1, 2, \dots, n\}$, where $\varepsilon_1, \dots, \varepsilon_n$ are now assumed to be i.i.d. random variables with $\mathbb{E}(\varepsilon_i) = 0$ and $\mathrm{Var}(\varepsilon_i) = \sigma^2$.

- Again, we assume that the x-values are fixed and known. The only random quantities in the model are ε_i and Y_i.
- The parameters in this model are now $\beta = (\beta_0, \beta_1, \cdots, \beta_p) \in \mathbb{R}^{p+1}$ and σ^2. The parameters β_j are often called the regression coefficients.

The usual approach in statistics to quantify how well an estimator works is to apply it to random samples from the statistical model, where we can assume that we know the parameters, and then to study how well the parameters are reconstructed by the estimators. Since this approach uses random samples as input the the estimator, we obtain random estimates and we need to use statistical methods to quantify how close the estimate is to the truth.

4.1 Mean and Covariance

The bias of an estimator is the difference between the expected value of the estimate and the truth. For the least squares estimator we have

$$\mathrm{bias}(\hat\beta) = \mathbb{E}(\hat\beta) - \beta,$$

where

$$\hat\beta = (X^\top X)^{-1} X^\top Y$$

and Y is the random vector from (16).

Lemma 4.1. *We have*

1) $\hat\beta = \beta + (X^\top X)^{-1} X^\top \varepsilon$ *and*
2) $\hat\beta \sim \mathcal{N}(\beta, \sigma^2 (X^\top X)^{-1})$.

Proof. From lemma 2.1 we know

$$\hat\beta = (X^\top X)^{-1} X^\top Y,$$

Using the definition of Y we can write this as

$$\begin{aligned}
\hat\beta &= (X^\top X)^{-1} X^\top Y \\
&= (X^\top X)^{-1} X^\top (X\beta + \varepsilon) \\
&= (X^\top X)^{-1} X^\top X\beta + (X^\top X)^{-1} X^\top \varepsilon \\
&= \beta + (X^\top X)^{-1} X^\top \varepsilon.
\end{aligned}$$

This proves the first claim.

Since ε follows a multi-variate normal distribution, $\beta + (X^\top X)^{-1} X^\top \varepsilon$ is also normally distributed. Taking expectations we get

$$\begin{aligned}
\mathbb{E}(\hat\beta) &= \mathbb{E}(\beta + (X^\top X)^{-1} X^\top \varepsilon) \\
&= \beta + (X^\top X)^{-1} X^\top \mathbb{E}(\varepsilon) \\
&= \beta,
\end{aligned}$$

since $\mathbb{E}(\varepsilon) = 0$.

For the covariance we find

$$\begin{aligned}
\mathrm{Cov}(\hat\beta) &= \mathrm{Cov}(\beta + (X^\top X)^{-1} X^\top \varepsilon) \\
&= \mathrm{Cov}((X^\top X)^{-1} X^\top \varepsilon) \\
&= (X^\top X)^{-1} X^\top \,\mathrm{Cov}(\varepsilon) \big((X^\top X)^{-1} X^\top\big)^\top \\
&= (X^\top X)^{-1} X^\top \,\mathrm{Cov}(\varepsilon) X (X^\top X)^{-1}.
\end{aligned}$$

Since $\mathrm{Cov}(\varepsilon) = \sigma^2 I$, this simplifies to

$$\begin{aligned} \mathrm{Cov}(\hat\beta) &= (X^\top X)^{-1} X^\top \sigma^2 I X (X^\top X)^{-1} \\ &= \sigma^2 (X^\top X)^{-1} X^\top X (X^\top X)^{-1} \\ &= \sigma^2 (X^\top X)^{-1}. \end{aligned}$$

This completes the proof. □

The lemma implies that $\mathbb{E}(\hat\beta) = \beta$, *i.e.* the estimator $\hat\beta$ is unbiased. Note that for this statement we only used $\mathbb{E}(\varepsilon) = 0$ to compute the expectation of $\hat\beta$. Thus, the estimator will still be unbiases for correlated or for noise which is not normally distributed.

We have seen earlier that the diagonal elements of a covariance give the variances of the elements of the random vector. Setting $C := (X^\top X)^{-1}$ as a shorthand here, we find that the individual estimated coefficients $\hat\beta_i$ satisfy

$$\hat\beta_i \sim \mathcal{N}(\beta, \sigma^2 C_{ii}) \tag{17}$$

for all $i \in \{1, \dots, n\}$.

These results about the (co-)variances of the estimator are not very useful in practice, because the error variance σ^2 is unknown. To derive more useful results, we will consider how to estimate this variance.

4.2 Properties of the Hat Matrix

In this and the following sections we will use various properties of the hat matrix $H = X(X^\top X)^{-1} X^\top$.

Lemma 4.2. *The hat matrix H has the following properties:*

1) H is symmetric, i.e. $H^\top = H$.
2) H is idempotent, i.e. $H^2 = H$.

Proof. For the first statement we have

$$\begin{aligned} H^\top &= \left(X(X^\top X)^{-1} X^\top\right)^\top \\ &= (X^\top)^\top \left((X^\top X)^{-1}\right)^\top X^\top \\ &= X(X^\top X)^{-1} X^\top \\ &= H, \end{aligned}$$

where we used that the inverse of a symmetric matrix is symmetric. The second statement follow from

$$\begin{aligned} H^2 &= \left(X(X^\top X)^{-1} X^\top\right)\left(X(X^\top X)^{-1} X^\top\right) \\ &= X\left((X^\top X)^{-1} X^\top X\right)(X^\top X)^{-1} X^\top \\ &= X(X^\top X)^{-1} X^\top. \end{aligned}$$

This completes the proof. □

Both properties from the lemma carry over from H to $I - H$: we have $(I - H)^\top = I^\top - H^\top = I - H$ and

$$\begin{aligned} (I - H)^2 &= (I - H)(I - H) \\ &= I^2 - HI - IH + H^2 \\ &= I - H - H + H \\ &= I - H. \end{aligned}$$

For future reference we also state two simpler results: we have

$$HX = X(X^\top X)^{-1} X^\top X = X \tag{18}$$

and

$$(I - H)X = IX - HX = X - X = 0. \tag{19}$$

Finally, if we have a vector $v \in \mathbb{R}^n$ we can write v as

$$\begin{aligned} v &= (H + I - H)v \\ &= Hv + (I - H)v. \end{aligned}$$

The inner product between these two components is

$$\begin{aligned} (Hv)^\top (I - H)v &= v^\top H^\top (I - H)v \\ &= v^\top H(I - H)v \\ &= v^\top (H - H^2)v \\ &= v^\top (H - H)v \\ &= 0, \end{aligned}$$

so the two vectors are orthogonal. As a result we get

$$\begin{aligned} \|v\|^2 &= v^\top v \\ &= \bigl(Hv + (I - H)v\bigr)^\top \bigl(Hv + (I - H)v\bigr) \\ &= (Hv)^\top Hv + 2(Hv)^\top (I - H)v + \bigl((I - H)v\bigr)^\top (I - H)v \\ &= \|Hv\|^2 + \|(I - H)v\|^2 \end{aligned}$$

(This is Pythagoras' theorem in $\mathbb{R}^n$.) Since $\hat{y} = Hy$ and $\hat{\varepsilon} = (I - H)y$, we can apply this idea to the vector y of observations to get $\|y\|^2 = \|\hat{y}\|^2 + \|\hat{\varepsilon}\|^2$.

We note without proof that geometrically, H can be interpreted as the orthogonal projection onto the subspace of $\mathbb{R}^n$ which is spanned by the columns of X. This subspace contains the possible output vectors of the linear system and the least squares procedure finds the point $\hat{y}$ in this subspace which is closest to the observed data $y \in \mathbb{R}^n$.

Some authors define:

- $\mathrm{SS}_\mathrm{T} = \sum_{i=1}^n y_i^2$ (where "T" stands for "total")
- $\mathrm{SS}_\mathrm{R} = \sum_{i=1}^n \hat{y}_i^2$ (where "R" stands for "regression")
- $\mathrm{SS}_\mathrm{E} = \sum_{i=1}^n (y_i - \hat{y}_i)^2$ (where "E" stands for "error")

Using this notation, our equation $\|y\|^2 = \|\hat{y}\|^2 + \|\hat{\varepsilon}\|^2$ turns into

$$\mathrm{SS}_\mathrm{T} = \mathrm{SS}_\mathrm{R} + \mathrm{SS}_\mathrm{E}.$$

4.3 Cochran's theorem

Our main tool in this and the following section will be a simplified version of Cochran's theorem.

Theorem 4.1. *The following statements are true:*

1) $\frac{1}{\sigma^2}\varepsilon^\top H\varepsilon \sim \chi^2(p + 1)$

2) $\frac{1}{\sigma^2}\varepsilon^\top (I - H)\varepsilon \sim \chi^2(n - p - 1)$

3) $H\varepsilon$ *and* $(I - H)\varepsilon$ *are independent.*

Proof. Since H is symmetric, we can diagonalise H (see A.1 in the appendix): there is an orthogonal matrix U such that $D := UHU^\top$ is diagonal, and the diagonal elements of D are the eigenvalues of H. Since H is idempotent, these diagonal elements can only be 0 or 1. Also, since U is orthogonal, we have $U^\top U = I$ and thus

$$U^\top DU = U^\top UHU^\top U = H.$$

The same matrix U also diagonalises $I - H$, since $U(I - H)U^\top = UU^\top - UHU^\top = I - D$. Exactly one of the diagonal elements D_{ii} and $(I - D)_{ii}$ is 1 and the other one is 0 for every i.

Since $\varepsilon \sim \mathcal{N}(0, \sigma^2 I)$ we find that $\eta := U\varepsilon$ is normally distributed with mean $U0 = 0$ and covariance matrix $\sigma^2 UIU^\top = \sigma^2 UU^\top = \sigma^2 I$. Thus η has the same distribution as ε does: $\eta \sim \mathcal{N}(0, \sigma^2 I)$ and the components η_i are independent of each other. We have

$$H\varepsilon = U^\top DU\varepsilon = U^\top D\eta.$$

and

$$(I - H)\varepsilon = U^\top (I - D)U\varepsilon = U^\top (I - D)\eta.$$

Since $(D\eta)_i = 0$ if $D_{ii} = 0$ and $((I - D)\eta)_i = 0$ otherwise, each component of η contributes to exactly one of the two vectors $D\eta$ and $(I - D)\eta$. Thus, $D\eta$ and $(I - D)\eta$ are independent, and thus $H\varepsilon$ and $(I - H)\varepsilon$ are also independent. This proves the third statement of the theorem.

For the first statement, we note that

$$\begin{aligned}
\varepsilon^\top H\varepsilon &= \varepsilon^\top U^\top DU\varepsilon \\
&= \eta^\top D\eta \\
&= \sum_{\substack{i=1 \\ D_{ii}=1}}^{n} \eta_i^2.
\end{aligned}$$

Since $(X^\top X) \in \mathbb{R}^{(p+1)\times(p+1)}$ is invertible, one can show that $\mathrm{rank}(H) = p + 1$ and thus that there are $p + 1$ terms contributing to the sum (we skip the proof of this statement here). Thus,

$$\frac{1}{\sigma^2}\varepsilon^\top H\varepsilon = \sum_{\substack{i=1 \\ D_{ii}=1}}^{n} (\eta_i/\sigma)^2$$

is the sum of the squares of $p+1$ independent standard normals, and thus is $\chi^2(p+1)$ distributed. This completes the proof of the first statement.

Finally, the second statement follows in much of the same way as the first one, except that H is replaced with $I - H$ and the sum is over the $n - p - 1$ indices i where $D_{ii} = 0$. This completes the proof. □

Expressions of the form $x^\top Ax$ for $x \in \mathbb{R}^n$ and $A \in \mathbb{R}^{n\times n}$ are called **quadratic forms**.

While the theorem as written only states that $H\varepsilon$ and $(I - H)\varepsilon$ are independent of each other, we can replace one or both of these terms the corresponding quadratic forms as still keep the independence. Since $(H\varepsilon)^\top(H\varepsilon) = \varepsilon^\top H^\top H\varepsilon = \varepsilon^\top H\varepsilon$, the quadratic form $\varepsilon^\top H\varepsilon$ is a function of $H\varepsilon$ and a similar statement holds with $H - I$ instead of H.

4.4 Estimating the Error Variance

So far we have only considered how to estimate the parameter vector β and we have ignored the parameter σ^2. We will see that an unbiased estimator for σ^2 is given by

$$\hat\sigma^2 = \frac{1}{n-p-1}\sum_{i=1}^{n}(y_i - \hat y_i)^2, \tag{20}$$

where $\hat y_i$ are the fitted values from equation (10). As for the one-dimensional case in (4), the estimate does not have the prefactor $1/n$, which one might naively expect, but the denomintor is decreased by one for each component of the vector β. Using Cochran's theorem, we can now show that the estimator $\hat\sigma^2$ is unbiased.

We first note that

$$\begin{aligned}
(n - p - 1)\hat\sigma^2 &= (y - \hat y)^\top (y - \hat y) \\
&= (y - Hy)^\top (y - Hy) \\
&= y^\top (I - H)^\top (I - H)y \\
&= y^\top (I - H)y
\end{aligned}$$

To determine the bias, we need to use $Y = X\beta + \varepsilon$ in place of the data. This gives

$$\begin{aligned}
(n - p - 1)\hat\sigma^2 &= Y^\top (I - H)Y \\
&= (X\beta + \varepsilon)^\top (I - H)(X\beta + \varepsilon) \\
&= \beta^\top X^\top (I - H)X\beta + 2\varepsilon^\top (I - H)X\beta + \varepsilon^\top (I - H)\varepsilon \\
&= \varepsilon^\top (I - H)\varepsilon,
\end{aligned}$$

where we used equation (19) to see that the first two terms in the sum equal zero.

Now we can apply Cochran's theorem. This shows that

$$\frac{1}{\sigma^2}(n-p-1)\hat{\sigma}^2 = \frac{1}{\sigma^2}\varepsilon^\top(I-H)\varepsilon \sim \chi^2(n-p-1). \tag{21}$$

Since the expectation of a $\chi^2(\nu)$ distribution equals ν (see appendix B.2), we find

$$\frac{1}{\sigma^2}(n-p-1)\mathbb{E}(\hat{\sigma}^2) = n-p-1$$

and thus

$$\mathbb{E}(\hat{\sigma}^2) = \sigma^2.$$

This proves that $\hat{\sigma}^2$ is an unbiased estimator for σ^2.

Summary

- The least squares estimator for the regression coefficients is unbiased.
- The hat matrix is idempotent and symmetric.
- Cochran's theorem allows to understand the distribution of some quadratic forms involving the hat matrix.
- $\hat{\sigma}^2$ is an unbiased estimator for σ^2.

5 Uncertainty for Individual Regression Coefficients

In this section we will consider different ways to study the uncertainty in the estimates $\hat{\beta}_i$ for the regression coefficient β_i individually. In the following sections we will then consider the problem of simultaneously estimating several or all coefficients.

5.1 Measuring the Estimation Error

We have seen that $\hat{\beta} \sim \mathcal{N}(0, \sigma^2 X)$, where $(X^\top X)^{-1}$. Restricting this to a single coefficient, we find

$$\hat{\beta}_i \sim \mathcal{N}(\beta, \sigma^2 C_{ii}),$$

since the diagonal elements of the covariance matrix contains the variances of the elements of a random vector. In practice we will not know the value of σ^2, so we have to estimate this from data, using the estimator

$$\hat{\sigma}^2 = \frac{1}{n-p-1} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2,$$

from equation (20). As a first application of Cochran's theorem we showed in equation (21) that

$$\frac{1}{\sigma^2}(n-p-1)\hat{\sigma}^2 \sim \chi^2(n-p-1).$$

Note that in the equations above, we index the rows and columns of C using $i, j \in \{0, 1, \dots, p\}$, *i.e.* the first row and column are using the index 0 each. This is to match the convention for the components of $\beta = (\beta_0, \beta_1, \dots, \beta_p)$.

Lemma 5.1. *The random vector $\hat{\beta}$ and the random number $\hat{\sigma}^2$ are independent of each other.*

Proof. We will show that $\hat{\beta}$ can be written as a function of $H\varepsilon$ and that $\hat{\sigma}^2$ can be written as a function of $(I - H)\varepsilon$. The result then follows from Cochran's theorem.

From lemma 4.1 we know that the least squares estimate $\hat{\beta}$ can be written as

$$\hat{\beta} = \beta + (X^\top X)^{-1} X^\top \varepsilon$$

and that $H = X(X^\top X)^{-1} X^\top$. Thus we can write $\hat{\beta}$ as

$$\begin{aligned} \hat{\beta} &= \beta + (X^\top X)^{-1} X^\top X \, (X^\top X)^{-1} X^\top \varepsilon \\ &= \beta + (X^\top X)^{-1} X^\top H \varepsilon, \end{aligned}$$

which is a function of $H\varepsilon$.

Similar to the argument at the end of the previous section, we can write $\hat{\sigma}^2$ as

$$\begin{aligned} (n-p-1)\hat{\sigma}^2 &= (Y - \hat{Y})^\top (Y - \hat{Y}) \\ &= (Y - HY)^\top (Y - HY) \\ &= Y^\top (I - H)^\top (I - H) Y \\ &= \|(I - H)Y\|. \end{aligned}$$

Since $Y = X\beta + \varepsilon$ and since we know $(I - H)X = 0$ from equation (19), we find

$$\begin{aligned} \hat{\sigma}^2 &= \frac{1}{n-p-1} \|(I - H)(X\beta + \varepsilon)\| \\ &= \frac{1}{n-p-1} \|(I - H)\varepsilon\|, \end{aligned}$$

which is a function of $(I - H)\varepsilon$.

From Cochran's theorem we know that $H\varepsilon$ and $(I - H)\varepsilon$ are independent and thus we can conclude that $\hat{\beta}$ and $\hat{\sigma}^2$ are also independent of each other. This completes the proof. □

We now construct a quantity T which measures the distance between the estimated value $\hat{\beta}_i$ and the unknown true value β_i:

$$T := \frac{\hat{\beta}_i - \beta_i}{\sqrt{\hat{\sigma}^2 C_{ii}}}. \tag{22}$$

While there are many ways to measure this distance, the T constructed here has two main advantages:

- The value of T can be computed from the given data, without any reference to unknown quantities.
- Below, we will be able to find the distribution of T. This will allow us to use T to construct confidence intervals and statistical tests.

Lemma 5.2. *Assume that the data follows the model* (16). *Then* $T \sim t_{n-p-1}$, i.e. *T follows a t-distribution with $n - p - 1$ degrees of freedom (see appendix B.3).*

Proof. We have

$$\begin{aligned} T &= \frac{(\hat{\beta}_i - \beta_i)/\sqrt{C_{ii}}}{\sqrt{\hat{\sigma}^2}} \\ &= \frac{(\hat{\beta}_i - \beta_i)/\sqrt{\sigma^2 C_{ii}}}{\sqrt{\hat{\sigma}^2/\sigma^2}} \\ &= \frac{(\hat{\beta}_i - \beta_i)/\sqrt{\sigma^2 C_{ii}}}{\sqrt{(n-p-1)\hat{\sigma}^2/\sigma^2/(n-p-1)}} \\ &=: \frac{Z}{\sqrt{Y/(n-p-1)}}, \end{aligned}$$

where $Z = (\hat{\beta}_i - \beta_i)/\sqrt{\sigma^2 C_{ii}} \sim \mathcal{N}(0,1)$ and $Y = (n-p-1)\hat{\sigma}^2/\sigma^2 \sim \chi^2(n-p-1)$ are independent, by lemma 5.1. Thus, $T \sim t_{n-p-1}$ as required. □

The quantity $\sqrt{\sigma^2 C_{ii}}$ is sometimes called the **standard error** of the estimator $\hat{\beta}_i$, denoted by $\mathrm{se}(\hat{\beta}_i)$.

5.2 Confidence Intervals

Using the scaled distance T, it is easy to construct a confidence interval for $\hat{\beta}_i$: For $\alpha \in (0,1)$, say $\alpha = 5\%$, lemma 5.2 shows that

$$P\Big(T \in [-t_{n-p-1}(\alpha/2), +t_{n-p-1}(\alpha/2)]\Big) = 1 - \alpha,$$

where $t_{n-p-1}(\alpha/2)$ is the $(1-\alpha/2)$-quantile of the $t(n-p-1)$-distribution. Rewriting this expression as a condition on $\hat{\beta}_i$ instead of on T gives a confidence interval for β_i.

Lemma 5.3. *The interval*

$$[U, V] := \Big[\hat{\beta}_i - \sqrt{\hat{\sigma}^2 C_{ii}} t_{n-p-1}(\alpha/2), \hat{\beta}_i + \sqrt{\hat{\sigma}^2 C_{ii}} t_{n-p-1}(\alpha/2)\Big]$$

is a $(1-\alpha)$-confidence interval for β_i.

Proof. We have to show that $P(\beta_i \in [U,V]) \geq 1 - \alpha$. We have

$$\begin{aligned} &\beta_i \in [U, V] \\ &\quad\Longleftrightarrow |\hat{\beta}_i - \beta_i| \leq \sqrt{\hat{\sigma}^2 C_{ii}} t_{n-p-1}(\alpha/2) \\ &\quad\Longleftrightarrow \Big|\frac{\hat{\beta}_i - \beta_i}{\sqrt{\hat{\sigma}^2 C_{ii}}}\Big| \leq t_{n-p-1}(\alpha/2) \\ &\quad\Longleftrightarrow T \in [-t_{n-p-1}(\alpha/2), +t_{n-p-1}(\alpha/2)] \end{aligned}$$

and thus $P(\beta_i \in [U,V]) = 1 - \alpha$. This completes the proof. □

5.3 Hypthesis Tests

Very similar to the argument for confidence intervals, we can derive a hypothesis test to test the hypothesis

$$H_0 \colon \beta_i = b$$

against the alternative

$$H_0 \colon \beta_i \neq b.$$

Here we redefine T as

$$T := \frac{\hat\beta_i - b}{\sqrt{\hat\sigma^2 C_{ii}}},$$

using b in place of the β_i above. Then the new defintion of T is the same as (22) if H_0 is true.

Lemma 5.4. *The test which rejects H_0 if and only if $|T| > t_{n-p-1}(\alpha/2)$ has confidence level α.*

Proof. We have to show that the probability of type I errors (*i.e.* of wrongly rejecting H_0 when it is true) is less than or equal to α. Assume that H_0 is true. Then we have $\beta_i = b$ and thus the T defined in this section coincides with the expression from equation (22). From lemma 5.2 we know that $T \sim t(n-p-1)$. Thus we have

$$\begin{aligned}
P(\text{type I error}) &= P(|T| > t_{n-p-1}(\alpha/2)) \\
&= P(T < -t_{n-p-1}(\alpha/2)) + P(T > t_{n-p-1}(\alpha/2)) \\
&= 2P(T > t_{n-p-1}(\alpha/2)) \\
&= 2P(T > t_{n-p-1}(\alpha/2)) \\
&= 2\frac{\alpha}{2} \\
&= \alpha.
\end{aligned}$$

This completes the proof. □

If we use $b = 0$ in the test, we can test whether $\beta_i = 0$. If $\beta_i = 0$ is true, the corresponding input x_i has no influence on the output.

As usual with statistical tests, one needs to be extremely careful when performing several tests on the same data. In particular, it would be unwise to test more than one component of β using this procedure for the same data. Instead, in the next section we will consider how to perform tests for several components of β simultaneously. Before we do this, we will perform some experiments with R.

5.4 R Experiments

5.4.1 Fitting the model

```
m <- lm(stack.loss ~ ., data = stackloss)
summary(m)
```

```
##
## Call:
## lm(formula = stack.loss ~ ., data = stackloss)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.2377 -1.7117 -0.4551  2.3614  5.6978
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -39.9197    11.8960  -3.356  0.00375 **
## Air.Flow      0.7156     0.1349   5.307  5.8e-05 ***
## Water.Temp    1.2953     0.3680   3.520  0.00263 **
## Acid.Conc.   -0.1521     0.1563  -0.973  0.34405
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.243 on 17 degrees of freedom
## Multiple R-squared:  0.9136, Adjusted R-squared:  0.8983
## F-statistic:  59.9 on 3 and 17 DF,  p-value: 3.016e-09
```

5.4.2 Estimating the Variance of the Error

We can get the design matrix X and the covariance matrix C as follows:

```
X <- model.matrix(m)
C <- solve(t(X) %*% X)
round(C, 4)
```

```
##             (Intercept) Air.Flow Water.Temp Acid.Conc.
## (Intercept)     13.4527   0.0273    -0.0620    -0.1594
## Air.Flow         0.0273   0.0017    -0.0035    -0.0007
## Water.Temp      -0.0620  -0.0035     0.0129     0.0000
## Acid.Conc.      -0.1594  -0.0007     0.0000     0.0023
```

Next we need to estimate the variance σ^2:

```
y <- stackloss$stack.loss
n <- nrow(stackloss)
p <- ncol(stackloss) - 1
hat.sigma2 <- sum((y - fitted(m))^2) / (n - p - 1)
hat.sigma2
```

```
## [1] 10.51941
```

The square root of this number, so the estimated standard deviation of the ε_i is shown as `Residual standard error` in the summary output above. We check that we get the same result:

```
sqrt(hat.sigma2)
```

```
## [1] 3.243364
```

This result is also listed as the `Residual standard error` near the bottom of the `summary(m)` output, above.

5.4.3 Estimating the Standard Errors

We can the standard errors, *i.e.* the standard deviations $\mathrm{stdev}(\hat\beta)_i$ as $\sqrt{\hat\sigma^2 C_i i}$:

```
se <- sqrt(hat.sigma2 * diag(C))
se
```

```
## (Intercept)    Air.Flow  Water.Temp  Acid.Conc.
##  11.8959969   0.1348582   0.3680243   0.1562940
```

These values are also listed in the `Std. Error` column of the `summary(m)` output.

5.4.4 Hypothesis tests

Let us now test the hypothesis $H_0 \colon \beta_i = 0$. The test statistic for this case is the following:

```
T <- coef(m) / se
T
```

```
## (Intercept)    Air.Flow  Water.Temp  Acid.Conc.
##  -3.3557234   5.3066130   3.5195672  -0.9733098
```

These values are also listed in the `t value` column of the `summary(m)` output.

Before we can perform the test, we need to choose α and to find the corresponding critical value:

```
alpha <- 0.05
t <- qt(1 - alpha/2 , n - p - 1)
t
```

```
## [1] 2.109816
```

Using the critical value t we can decided whether H_0 should be accepted or rejected. For example, looking at the intercept β_0, we find $|T_0| = |-3.3557234| > 2.109816 = t_{n-p-1}(1-\alpha/2)$ and thus we can reject the hypothesis $H_0 \colon \beta_0 = 0$. This means that the intercept is significantly different from 0.

5.4.5 Confidence Intervals

Using the quantile `t` we can also get confidence intervals. Here we only show the confidence interval for the intercept β_0:

```
c(coef(m)[1] - se[1] * t, coef(m)[1] + se[1] * t)
```

```
## (Intercept) (Intercept)
##   -65.01803   -14.82131
```

Confidence intervals for the remaining coefficients can be obtained simimlarly.

Summary

- We know how to scale the distance between individual parameter estimates and the truth.
- We have seen how to construct confidence intervals for β_i.
- We have seen how to construct statistical tests for β_i.
- We have understood some more of the summary output for the `lm()` function in R.

A Linear Algebra Reminders

A.1 Vectors

We write $v \in \mathbb{R}^d$ if $v = (v_1, \dots, v_d)$ for numbers $v_1, \dots, v_d \in \mathbb{R}$. We say that v is a d-dimensional vector, and $\mathbb{R}^d$ is the d-dimensional Euclidean space. Vectors are often graphically represented as "column vectors":

$$v = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_d \end{pmatrix}.$$

If $u, v \in \mathbb{R}^d$ are two vectors, the **inner product** of u and v is given by

$$u^\top v = \sum_{i=1}^{d} u_i v_i. \tag{23}$$

Note that the two vectors must have the same length for the inner product to exist. The vectors u and v are said to be **orthogonal**, if $u^\top v = 0$.

Using this notation, the **Euclidean length** of a vector v can be written as

$$\|v\| = \sqrt{\sum_{i=1}^{d} v_i^2} = \sqrt{v^\top v}.$$

A.2 Matrices

We write $A \in \mathbb{R}^{m \times n}$ if

$$A = \begin{pmatrix} a_{1,1} & \cdots & a_{1,n} \\ a_{2,1} & \cdots & a_{2,n} \\ \vdots & \ddots & \vdots \\ a_{m,1} & \cdots & a_{m,n} \end{pmatrix},$$

where $a_{i,j}$, sometimes also written as a_{ij} are numbers for $i \in \{1, \dots, m\}$ and $j \in \{1, \dots, n\}$.

A.2.1 Transpose

If $A \in \mathbb{R}^{m \times n}$, then the **transpose** of A is the matrix $A^\top \in \mathbb{R}^{n \times m}$, with $(A^\top)_{ij} = a_{ji}$ for all $i \in \{1, \dots, n\}$ and $j \in \{1, \dots, m\}$. Graphically, this can be written as

$$A^\top = \begin{pmatrix} a_{1,1} & a_{2,1} \cdots & a_{m,1} & \\ \vdots & \vdots & \ddots & \vdots \\ a_{1,n} & a_{2,n} \cdots & a_{m,n} & \end{pmatrix},$$

Definition A.1. A matrix A is called **symmetric**, if $A^\top = A$.

A.2.2 Matrix-vector Product

If $A \in \mathbb{R}^{m \times n}$ and $v \in \mathbb{R}^n$, then $Av \in \mathbb{R}^m$ is the vector with

$$(Av)_i = \sum_{j=1}^{n} a_{ij} v_j$$

for all $i \in \{1, \dots, m\}$.

If we consider v to be a $(n \times 1)$-matrix instead of a vector, Av can also be interpreted as a matrix-matrix product between an $m \times n$ and an $n \times 1$ matrix. Using this convention, $v^\top$ is then interpreted as an $1 \times n$ matrix and if $u \in \mathbb{R}^m$ we have $u^\top A \in \mathbb{R}^{1 \times n} \cong \mathbb{R}^n$ with

$$(u^\top A)_j = \sum_{i=1}^{m} u_i a_{ij}$$

for all $j \in \{1, \dots, n\}$. Going one step further, this notation also motivates the expression $u^\top v$ in equation (23).

A.2.3 Matrix-matrix Product

If $A \in \mathbb{R}^{\ell \times m}$ and $B \in \mathbb{R}^{m \times n}$, then $AB \in \mathbb{R}^{\ell \times n}$ is the matrix with

$$(AB)_{ik} = \sum_{j=1}^{m} a_{ij} b_{jk}$$

for all $i \in \{1, \dots, \ell\}$ and $j \in \{1, \dots, n\}$. This is called the **matrix product** of A and B. Note that A and B must have compatible shapes for the product to exist.

Properties:

- The matrix product is associative: if A, B and C are matrices with shapes such that AB and BC exist, then we have $A(BC) = (AB)C$. It does not matter in which order we perform the matrix products here.
- The matrix product is transitive: if A, B and C have the correct shapes, we have $A(B + C) = AB + AC$.
- The matrix product is *not* commutative: if AB exists, in general A and B don't have the correct shapes for BA to also exist, and even if BA exists, in general we have $AB \neq BA$.
- Taking the transpose swaps the order in a matrix product: we have

$$(AB)^\top = B^\top A^\top \tag{24}$$

A.2.4 Matrix Inverse

If A is a square matrix and if there is a matrix B such that $AB = I$, then A is called **invertible** and the matrix B is called the **inverse** of A, denoted by $A^{-1} := B$. Some important properties of the inverse:

- The inverse, if it exists, is unique.
- Left-inverse and right-inverse for matrices are the same: $A^{-1}A = I$ holds if and only if $AA^{-1} = I$.
- If A is symmetric and invertible, then A^{-1} is also symmetric. (Proof: $A(A^{-1})^\top = (A^{-1}A)^\top = I^\top = I$ and thus $(A^{-1})^\top$ is an inverse of A. Since the inverse is unique, $(A^{-1})^\top = A^{-1}$.)

A.2.5 Orthogonal Matrices

Definition A.2. A matrix U is called **orthogonal**, if $U^\top U = I = UU^\top$.

If U is orthogonal, the inverse and the transpose are the same: $U^\top = U^{-1}$.

A.2.6 Positive Definite Matrices

Definition A.3. A symmetric matrix $A \in \mathbb{R}^{n \times n}$ is called **positive definite**, if

$$x^\top A x > 0$$

for all $x \in \mathbb{R}^n$ with $x \neq 0$. The matrix is called **positive semi-definite**, if

$$x^\top A x \geq 0$$

for all $x \in \mathbb{R}^n$.

A.2.7 Idempotent Matrices

Definition A.4. The matrix A is **idempotent**, if $A^2 = A$.

A.3 Eigenvalues

Definition A.5. Let $A \in \mathbb{R}^{n \times n}$ be a square matrix and $\lambda \in R$. The number λ is called an **eigenvalue** of A, if there exists a vector $v \neq 0$ such that $Ax = \lambda x$. Any such vector x is called an **eigenvector** of A with eigenvalue λ.

While there are very many results about eigenvectors and eigenvalues in Linear Algebra, here we will only use a small number of these results. We summarise what we need for this module:

- If A is idempotent and x is an eigenvector with eigenvalue λ, then we have $\lambda x = Ax = A^2x = \lambda Ax = \lambda^2 x$. Thus we have $\lambda^2 = \lambda$. This shows that the only eigenvalues possible for idempotent matrices are 0 and 1.

Theorem A.1. *Let $A \in \mathbb{R}^{n\times n}$ be symmetric. Then there is an orthogonal matrix U such that $D := UAU^\top$ is diagonal. The diagonal elements of D are the eigenvalues of A and the rows of U are corresponding eigenvectors.*

B Probability Reminders

B.1 Independence

Definition B.1. Two random variables X and Y are (statistically) **independent**, if $P(X \in A, Y \in B) = P(X \in A)P(Y \in B)$ for all sets A and B.

We list some properties of independent random variables:

- If X and Y are independent, and if f and g are functions, then $f(X)$ and $g(Y)$ are also independent.

B.2 The Chi-Squared Distribution

Definition B.2. Let $X_1, \dots, X_\nu \sim \mathcal{N}(0,1)$ be i.i.d. Then the distribution of $\sum_{i=1}^{\nu} X_i^2$ is called the χ^2-distribution with ν degrees of freedom. The distribution is denoted by $\chi^2(\nu)$.

Some important results about the χ^2-distribution are:

- χ^2-distributed random variables are always positive.
- If $Y \sim \chi^2(\nu)$, then $\mathbb{E}(Y) = \nu$ and $\mathrm{Var}(Y) = 2\nu$.
- The R command `pchisq(|`x`,`ν`)` gives the value $\Phi_\nu(x)$ of the CDF of the $\chi^2(\nu)$-distribution.
- The R command `qchisq(`α`,`ν`)` can be used to obtain the α-quantile of the $\chi^2(\nu)$-distribution.
- More properties can be found on Wikipedia.

B.3 The t-distribution

Definition B.3. Let $Z \sim \mathcal{N}(0,1)$ and $Y \sim \chi^2(\nu)$ be independent. Then the distribution of

$$T = \frac{Z}{\sqrt{Y/\nu}} \tag{25}$$

is called the t-distribution with ν degrees of freedom. This distribution is denoted by $t(\nu)$.

Some important results about the t-distribution are:

- The t-distribution is symmetric: if $T \sim t(\nu)$, then $-T \sim t(\nu)$
- If $T \sim t(\nu)$, then $\mathbb{E}(T) = 0$.
- The R command `pt(|`x`,`ν`)` gives the value $\Phi_\nu(x)$ of the CDF of the $t(\nu)$-distribution.
- The R command `qt(`α`,`ν`)` can be used to obtain the α-quantile of the $t(\nu)$-distribution.
- More properties can be found on Wikipedia.