Introduction to Statistical Modelling

Joris Vankerschaver

2023-02-25

Table of contents

Preface

Most of the figures in this book are generated with R. If you are reading this book online, you can view the source code for each figure by clicking the little triangle next to the word “Code” above the figure. In the PDF version of this book the code is listed inline with the figure.

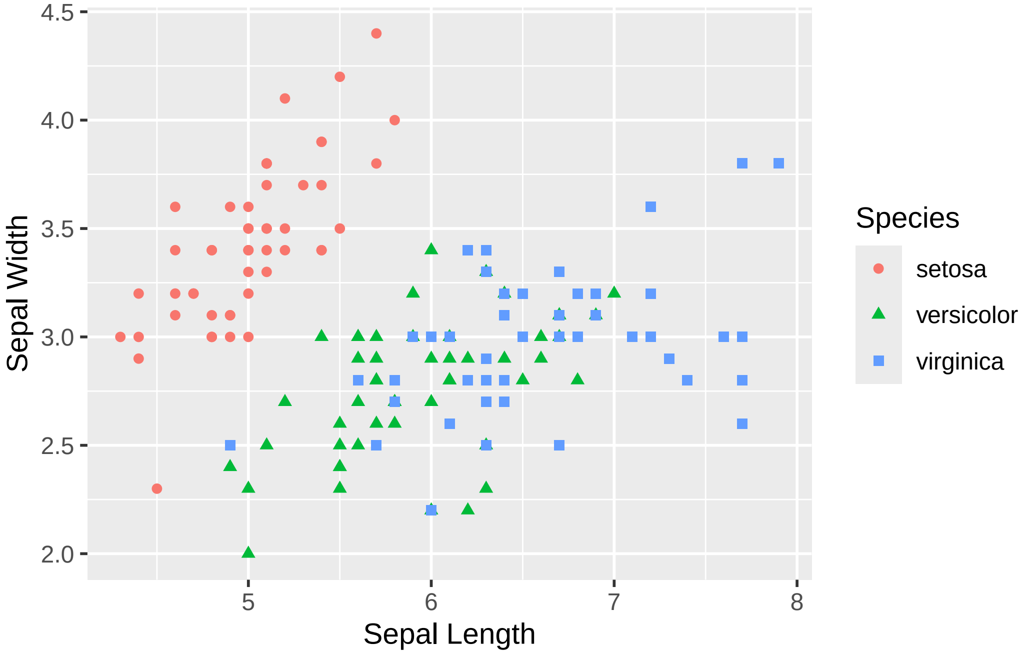


Figure 1: Length and width of the sepal petal of 150 Iris plants.

1 Introduction

This book is a work-in-progress collection of course notes on various aspects of statistical modeling. I expect that it will grow more complete as the selection of topics that are routinely covered becomes more standardized, and I plan to update the course notes on a regular basis.

The chapter on linear regression was taken from a previous set of course notes on statistical inference by Prof. Stijn Vansteelandt (with translations and amendments by Prof. Arnout Van Messem). The version in this book has been lightly edited to incorporate it in a Quarto-based environment (among other things, the code to produce the figures has been streamlined and made available along with the text).

2 Regression analysis

Often a lot of measurements for each subject (e.g., each animal or each plant) are collected in scientific experiments in biosciences. An ecologist could, for example, measure at the same time the number of a certain shrub on different plots of land, as well as the acidity of each plot in order to hopefully be able to describe a relationship between both. A biotechnologist might be interested in seeing which genes are important during which phase of the growth of a plant. He or she could therefore collect at appropriate times measurements for the expression of different genes and subsequently investigate the association between gene expression and time. The purpose of this chapter is to provide techniques to detect patterns and relations in complex datasets and then to use these relations to predict future outcomes. We will in particular focus on situations where we are interested in one specific continuous outcome and hope to understand its relationship with one or more continuous or qualitative variables.

2.1 The linear regression model

Although the correlation coefficient is frequently used in explorative and descriptive statistics to describe an association between 2 continuous measurements, it has a number of limitations:

1. Its numerical value is difficult to interpret;
2. It cannot be used to predict the value of the outcome Y (e.g., the number of nests of red land crabs in a certain area) based on some predictor value X (e.g., the biomass of crabs in that area);
3. It does not allow for an easy correction of the association between the variables X and Y for the disturbing influence of measured confounders;
4. It does not allow for an easy verification of whether the strength of the association between the variables X and Y depends on the value of a third variable Z (e.g., to verify if there exist gene-neighbourhood interactions where the influence that certain genes exert on the development of Chronic Obstructive Pulmonary Disease depends on smoking history);
5. It does not allow to describe nonlinear associations or associations between a continuous and a qualitative variable.

To handle these problems in a flexible way, we will use regression techniques.

ℹ Example: Woody debris and tree density

The human impact on freshwater environments concerns scientists a lot. Coarse woody debris (CWD) is fallen wood that provides a habitat for aquatic organisms and furthermore influences hydrological processes and the transport of organic materials within aquatic ecosystems. The presence of humans has altered the CWD input to aquatic systems. Chistensen et al. (1996) (TODO fix up reference) studied the connection between coarse woody debris and riparian vegetation in a sample of 16 North American lakes. They defined CWD as woody debris with a diameter larger than 5 cm and registered for a number of locations along the shoreline the CWD basal area (in m^2 per km of shoreline) and the tree density (in number per km of shoreline). To obtain a single measurement per lake, weighted averages were used.

The goal of this study is to describe the association between the tree density along the shoreline of the lake and the relative basal area of CWD. Since we want to explain the effect of the tree density on CWD, we call tree density the *explanatory*, *predictor* or *independent variable* and the CWD basal area the *outcome* or *dependent variable*, i.e., the variable in which we are primarily interested. For the rest of this section, we will always use X for the independent variables and Y for the dependent variables.

Figure 2.1 plots the CWD basal area (in m^2 per km) in function of the riparian tree density, together with a loess scatterplot smoother (dotted line). The plot gives no indication that the relation between both variables would not be linear. Hence the Pearson correlation coefficient is an appropriate measure. It is equal to 0.797, which suggests a strong increase in CWD basal area with an increased riparian tree density. We gain more insight in the strength of the association by studying the loess scatterplot smoother, since this function gives for every tree density value the expected outcome for the CWD basal area. Because this curve can be well approximated by a much simpler, linear relation, we also added the 'best fitting' straight line (full line; i.e., the least squares regression line) to the plot. This gives an even clearer image of the relationship between both variables than the correlation coefficient, and also uses only 1 parameter (namely the slope) to do so. In this section we will see how to construct and interpret this so-called regression line.

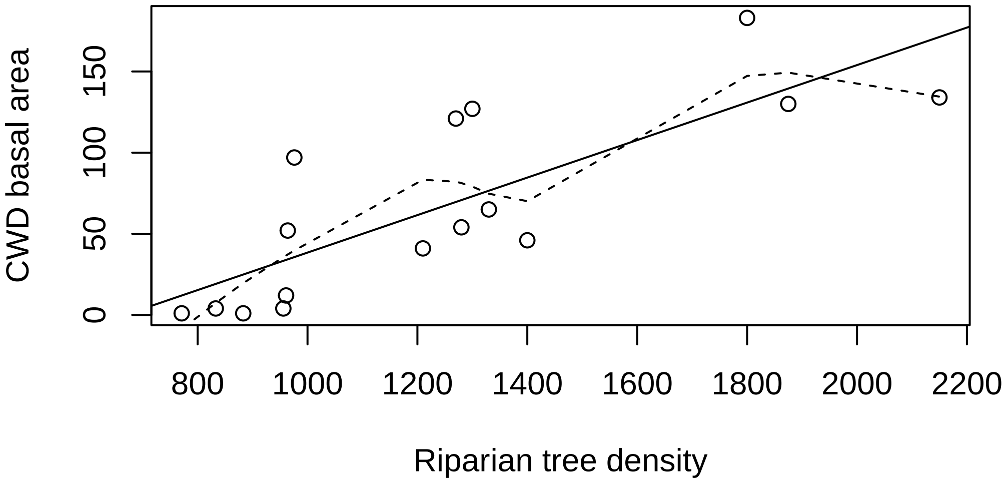


Figure 2.1: CWD basal area in function of tree density, with linear regression line (solid line) and loess scatterplot smoother (dashed line).

We denote with $E(Y|X = x)$ the mean outcome for the subgroup of the study population consisting of subjects for which the explanatory variable X takes on the value x. For example, for the CWD example above, $E(Y|X = 1200)$ is the mean CWD basal area per km of shoreline for lakes that have 1,200 trees per km along their shoreline. We could in principle calculate this mean by registering, for all lakes in the study population with 1,200 trees per km of shoreline, the CWD basal area and then taking the mean of these values. The mean $E(Y|X = x)$ is called a *conditional mean* because it describes a mean outcome, conditional on the fact that $X = x$.

Now suppose that the mean outcome can be described linearly in function of the explanatory variable X, which means that

$$E(Y|X = x) = \alpha + \beta x, \tag{2.1}$$

where α and β are unknown numbers. In this expression, $E(Y|X = x)$ represents the value on the Y-axis, x the value on the X-axis, the *intercept* α indicates the intersection with the Y-axis, and β is the *slope* of the line. This expression is called a *statistical model*. This naming suggests that certain assumptions will be placed on the distribution of the observations. In particular it assumes that the mean outcome varies linearly in function of the predictor X. For this reason, this is also called a *simple linear regression model*. According to this model, every measurement Y can be described, modulo an error term ϵ, as a linear function of the explanatory variable X:

$$Y = E(Y|X = x) + \epsilon = \alpha + \beta x + \epsilon,$$

where ϵ represents the deviation between the observed outcome and its (conditional) mean value, i.e., the uncertainty in the response variable.

The parameters α and β are unknowns. If we could observe the entire study population, we could determine both parameters exactly (by calculating for two x-values the mean outcome and then solve the resulting system of linear equations as given by Equation 2.1). In reality we only observe a limited sample from the study population and hence we need to estimate both parameters based on the available information. The parameters are estimated by searching for the line that "best fits" the data.

In order to obtain a best-fitting line, we want that for each subject i, the difference between the corresponding point on the regression line, $(x_i, \alpha + \beta x_i)$, and the observation itself, (x_i, y_i), is as small as possible. This can be realised by choosing values for α and β that minimise the sum of the squared distances between the predicted and the observed points:

$$\sum_{i=1}^{n} (y_i - \alpha - \beta x_i)^2.$$

The obtained line is called the *least squares (regression) line.* The corresponding values or estimations $\hat{\alpha}$ for α and $\hat{\beta}$ for β are called the *least squares estimates.* It can be shown that

$$\hat{\beta} = \frac{\text{Cor}(x, y) s_y}{s_x}$$

and that

$$\hat{\alpha} = \bar{y} - \hat{\beta}\bar{x}.$$

Note that the slope of the least squares line is proportional to the correlation between the outcome and the explanatory variable.

Given the estimates $\hat{\alpha}$ and $\hat{\beta}$, the linear regression model 2.1 allows us to do two things:

1. To predict the expected outcome for subjects with a given x-value for the explanatory variable. If these subjects have the predictor equal to $X = x$, then the expected outcome, Y, is on average
$$E(Y \mid X = x) = \hat{\alpha} + \hat{\beta} x.$$
2. To verify how much the outcome differs on average between two groups of subjects with a difference of δ units in the explanatory variable. This is:
$$E(Y \mid X = x + \delta) - E(Y \mid X = x) = \alpha + \beta(x + \delta) - \alpha - \beta x = \beta\delta.$$
In particular, β can be interpreted as the difference in mean outcome between two subjects that differ by one unit in X-value. This difference can be estimated by $\hat{\beta}$.

Woody debris and tree density, continued

We can build a linear model in R by means of the `lm` command:

```
Call:
lm(formula = CWD.BASA ~ RIP.DENS, data = trees)

Residuals:
   Min     1Q Median     3Q    Max
-38.62 -22.41 -13.33  26.16  61.35

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -77.09908   30.60801  -2.519 0.024552 *
RIP.DENS      0.11552    0.02343   4.930 0.000222 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 36.32 on 14 degrees of freedom
Multiple R-squared:  0.6345,    Adjusted R-squared:  0.6084
F-statistic:  24.3 on 1 and 14 DF,  p-value: 0.0002216
```

The software reports $\hat{\alpha} = -77.09908$ and $\hat{\beta} = 0.11552$. We conclude that, per km of shoreline, the CWD basal area increases on average with 1.2 m^2 per increase of 10 trees in tree density. Furthermore, we can predict what CWD basal area can be expected for any given number of trees per km of shoreline. For example, if the tree density is 1,600 trees per km shoreline, we expect a mean CWD basal area of $-77.09908 + 0.11552 \times 1600 = 108$ m^2 per km shoreline.

From Figure 2.1 you can see that the dataset does not contain any lakes with a tree density of approximately 1,600 trees per km of shoreline. Based on the dataset it would thus not be possible, without using a statistical model, to obtain an estimate for the mean CWD basal area for that given tree density. However, assuming that the mean CWD basal area varies linearly with the riparian tree density, we can use all observations to estimate this mean. Hence we obtain a meaningful and precise result, if the condition of linearity is met of course.

For the results obtained from the linear regression model to be valid, it is important to verify that all conditions that are imposed by the model are met. So far, the only assumption we made was that the mean outcome varies linearly in function of the explanatory variable (later on, we will add more conditions to also determine the variability of the data around the regression line). This assumption can easily be verified graphically using a scatterplot where we plot the outcome in function of the explanatory variable and then check if the relation seems to follow a linear pattern. Deviations from linearity can usually be discovered more easily by means of a *residual plot*. This is a scatterplot with the explanatory variable on the X-axis and the *residuals* on the Y-axis. The residuals are the prediction errors that can be calculated as

$$e_i = y_i - \hat{y}_i = y_i - \hat{\alpha} - \hat{\beta}x_i.$$

They represent the vertical distance between the observation for subject i and its prediction on the regression line.

In practice, it is often more convenient to put the fitted values on the x-axis, rather than the values of the predictor. This is especially useful for multiple linear regression, where there are several predictors. We will follow this convention for the residual plot from now on.

If the assumption of linearity holds, then there should be no pattern visible in the residual plot. This is the case in Figure 2.2 that shows a residual plot for the regression analysis of the CWD example. However, when the residuals reveal a nonlinear pattern, this means that extra terms should be added to the model to correctly predict the mean outcome. For example, if the residuals show a quadratic pattern, then we could write that approximately $e_i \approx \delta_0 + \delta_1 x_i + \delta_2 x_i^2$ for some numbers δ_0, δ_1, and δ_2, and hence that the outcome $y_i = \hat{\alpha} + \hat{\beta}x_i + e_i \approx (\hat{\alpha} + \delta_0) + (\hat{\beta} + \delta_1)x_i + \delta_2 x_i^2$ (modulo an error term) is a quadratic function of x_i. In that case it is best to switch to a quadratic regression model.

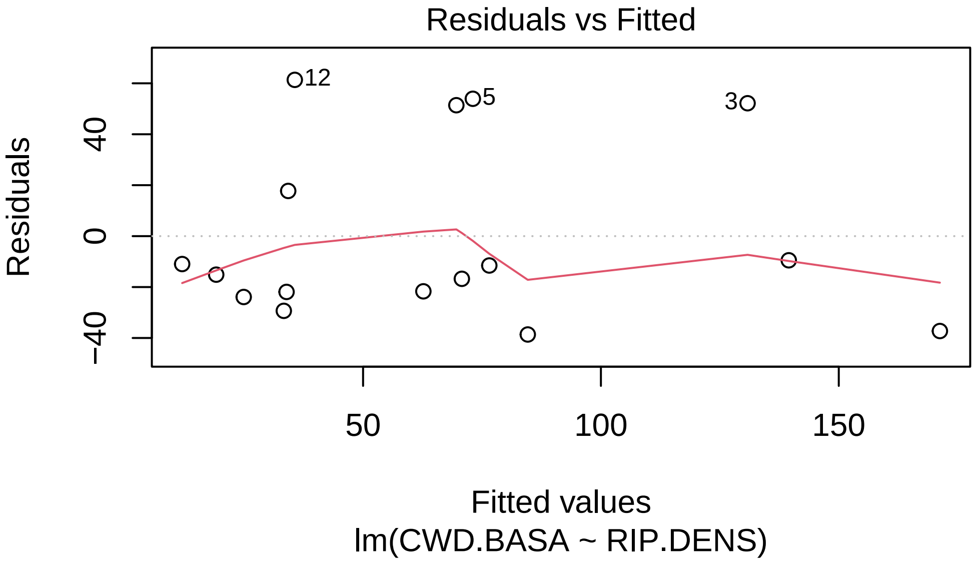


Figure 2.2: Residual plot for the CWD data.

Since the linearity of the model can only be verified over the observed range of the explanatory variable (for example, over the interval [771; 2,150] for the CWD data), it is important to understand that the results of a linear model cannot just be extrapolated past the largest or smallest observed X-value. In the CWD example we can estimate that the mean CWD basal area for lakes with a riparian tree density of 750 per km of shoreline will be $-77.09908 + 0.11552 \times 750 = 9.5$ m^2, but the observed data do not allow us verify the reliability of this estimation. After all, it could be that the regression line for low values of the predictor variable increases or decreases, causing the linear extrapolation to be misleading. Note that, for example, the prediction for a tree density of 500 per km of shoreline is very misleading, since it gives a negative result ($-77.09908 + 0.11552 \times 500 = -19$ m^2).

ℹ Frequency of Lap94 and distance from Southport

For the next example, we consider the blue or common mussel (*Mytilus edulis*). We are especially interested in the frequency of the allele Lap94 with respect to the eastern distance from Southport, Connecticut, U.S.A.

Figure 2.3: Mytilus edulis, the common mussel. Figure courtesy of Wikipedia, CC BY-SA 3.0.

We have obtained a dataset of 17 mussels, where for each mussel we record the frequency of the Lap94 allele, and the distance to the Southport seabord. This dataset is shown in Figure 2.4.

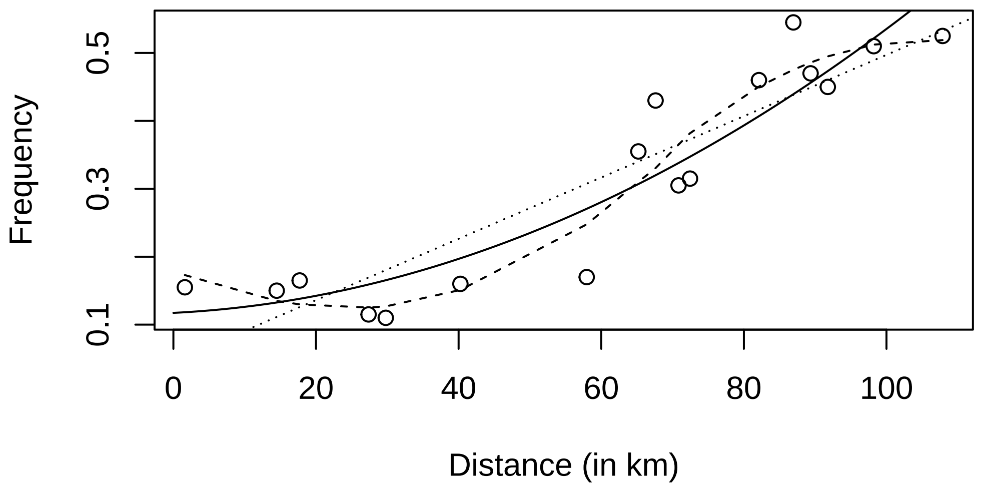


Figure 2.4: Frequency of Lap94 in function of the eastern distance from Southport, with linear regression line (dotted line), quadratic regression line (solid line), and loess scatterplot smoother (dashed line)

A linear regression model shows that the expected gene frequency differs by 2.2% between mussels that are located at a 10 km eastern distance from each other.

```
Call:
lm(formula = freq ~ km, data = southport)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.13744 -0.05784  0.01486  0.03923  0.10675 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 0.045830   0.036361    1.26    0.227    
km          0.004514   0.000536    8.42 4.56e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.06955 on 15 degrees of freedom
Multiple R-squared:  0.8254,    Adjusted R-squared:  0.8137 
F-statistic:  70.9 on 1 and 15 DF,  p-value: 4.557e-07
```

Figure 2.5 shows a residual plot based on linear regression. We see that that residuals display a systematic pattern, which is approximately parabolic. This suggests that adding a quadratic term to the model will improve the reliability of the regression model.

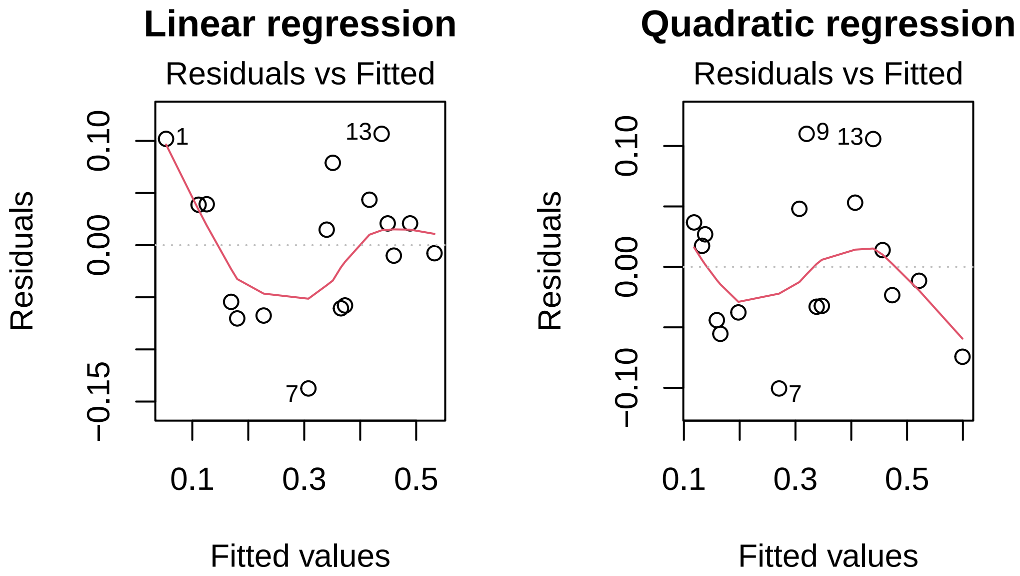


Figure 2.5: Residual plots for linear regression (left) and quadratic regression (right), with loess scatterplot smoother.

Adding a quadratic term gives the following model:

```
Call:
lm(formula = freq ~ km + I(km^2), data = southport)

Residuals:
     Min       1Q   Median       3Q      Max
-0.10054 -0.03766 -0.01147  0.03678  0.11003

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 1.173e-01  4.805e-02   2.441   0.0285 *
km          5.257e-04  2.008e-03   0.262   0.7973
I(km^2)     3.655e-05  1.786e-05   2.047   0.0599 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.06316 on 14 degrees of freedom
Multiple R-squared:  0.8656,    Adjusted R-squared:  0.8464
F-statistic: 45.09 on 2 and 14 DF,  p-value: 7.919e-07
```

Figure 2.5 (right) shows a residual plot based on this quadratic regression and indicates that the previously found pattern has mostly disappeared. Hence this model better describes the data. In subsequent sections we will investigate whether the model can be improved further.

2.2 The residual standard deviation

The CWD linear regression model from the previous section indicates how much (in particular, what basal area of) woody debris we can expect along North American lakes with a certain tree density. However, it does not inform us how much this area can vary between lakes with the same tree density. Nevertheless, it is of the utmost importance to know this when we want to make predictions based on the regression model, since outcomes that vary a lot around the regression line can of course not be predicted accurately by using the regression line, as opposed to outcomes with little variation around the line which can be predicted relatively accurately.

If the outcomes for lakes with the same predictor value x (i.e., tree density) are normally distributed, then it makes sense to express the variation of the outcomes around their mean by means of a *conditional variance* $\mathrm{Var}(Y \mid X = x)$. Similarly to the conditional mean $E(Y \mid X = x)$, this indicates the variance on the outcomes for the subgroup from the study population consisting of lakes for which the tree density X takes on the value x. For example, $\mathrm{Var}(Y \mid X = 1,300)$ in the CWD example is the variance on the CWD basal area per km of shoreline for lakes with a riparian tree density of 1,300 per km. These variances cannot just be estimated, since there is only 1 observation in the dataset with a tree density of 1,300. When we examine Figure 2.2, we see that the points are equally spread around the regression line, irrespective of the tree density, and that the variability on the basal area of coarse woody debris does not seem to depend on the tree density. In this case, we call the outcomes *homoscedastic* or the variance is said to be *homogeneous*. It then makes sense to assume that the conditional variance $\mathrm{Var}(Y \mid X = x)$ is constant:

$$\mathrm{Var}(Y \mid X = x) = \sigma^2. \tag{2.2}$$

The constant σ is called the *residual standard deviation*. If we assume Equation 2.2, we are able to determine the conditional variance $\mathrm{Var}(Y \mid X = 1300) = \sigma^2$, because then it can be estimated based on the data for all lakes, as will be illustrated in the next paragraph.

As we have learned in descriptive statistics, the variation of the outcomes around their conditional mean can be described by means of the differences between the observations y_i and their (estimated) mean $\hat{\alpha} + \hat{\beta} x_i$, or in other words, through the residuals. However, the mean of the residuals is always 0 because the positive and negative deviations cancel each other out. Hence the mean residual is not a good measure for the variation, and it is more sensible to use the squared deviations e_i^2. The mean of these *squared residuals* therefore will give a good measure. In particular it can be shown that the so-called *residual mean squared error*, given by

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^{n} e_i^2}{n-2} = s_y^2\{1 - \mathrm{Cor}(x,y)^2\} \tag{2.3}$$

is a good (i.e., unbiased) estimate of σ^2. Remark that the size of the residual mean squared error is closely connected with the correlation. If the outcome is independent from the predictor variable, the variability of the outcomes around the regression line is the same as the total variability, as denoted by the standard deviation s_y. If the variables X and Y are (perfectly) linearly dependent on each other, the correlation is 1, and hence there is no variation around the regression line. This is logical, since in that case the data points form a straight line and therefore do not vary around that line.

Woody debris and tree density, continued

Previously, the R model summary for the CWD model gave the following estimate for the residual standard deviation:

```
Residual standard error: 36.32 on 14 degrees of freedom
```

Assuming the CWD basal area is normally distributed for a given tree density, we can conclude that respectively 68% and 95% of those basal areas given a tree density of 1,300 per km can be expected to fall in the intervals $[73-36, 73+36] = [37, 109]$ and $[73-2\times 36, 73+2\times 36] = [1, 145]$. Thus we obtain a fairly wide 95% reference interval for the CWD basal area when the tree density is 1,300 trees per km of shoreline. These intervals are not completely accurate since they do not take the imprecision of the estimates of the mean outcome and residual standard deviation into account. in the literature there exist so-called *prediction intervals* which do take this imprecision into account.

For the previous results to be valid, it is of course again important that all conditions imposed by the model are met. This time we do not only have the assumption of linearity, but more importantly also the assumption of homoscedasticity of the outcomes. Since, according to Equation 2.3, the squared residuals are indicative for the variability that is present in the data, we can investigate this assumption by making a scatterplot of the squared residuals (on the Y-axis) in function of the explanatory variable (on the X-axis).This is illustrated for the CWD data in Figure 2.6 (left), which seems to suggest that the variability increases for increasing tree densities. Consequently, the reference intervals that have been calculated assuming homoscedasticity cannot be fully trusted. In particular they might be too narrow for high tree densities and too wide for low tree densities.

Even if the variance would be homogeneous, it is still important to verify that the outcomes are normally distributed for subjects with the same predictor value in order for the residual standard deviation to be a meaningful measure to describe the variability on the data and for the calculated reference intervals to be correct. A QQ-plot of the outcomes would be

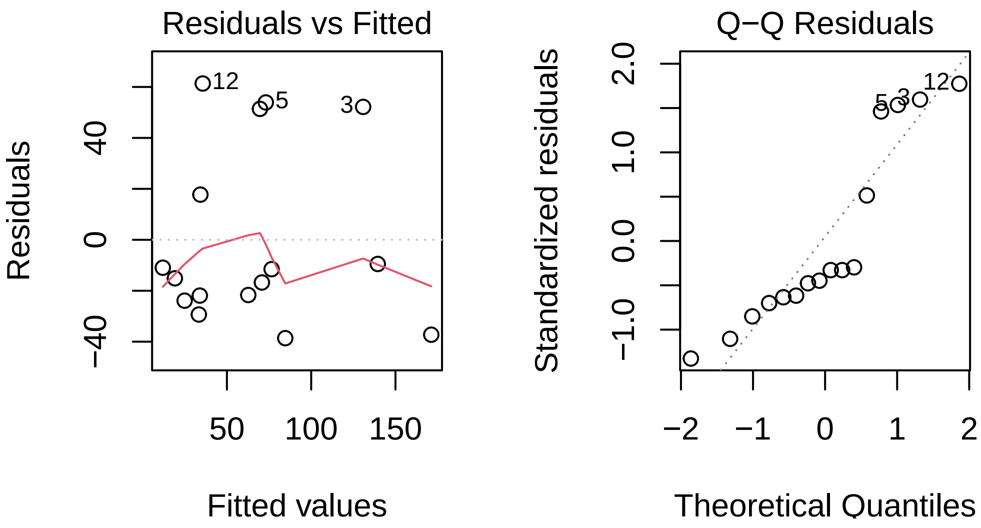


Figure 2.6: Analysis of the CWD data. Left: scatterplot of the squared residuals. Right: QQ-plot of the residuals.

misleading, since this checks the normality of all measurements as a whole, not the normality of the measurements for subjects with the same predictor value. It can be shown that normally distributed outcomes for a given x-value implies that the residuals are also approximately normally distributed. Hence deviations from normality in a QQ-plot for the residuals indicate that the outcomes are not normally distributed for a fixed x. Figure 2.6 (right) illustrates this for the CWD data and shows deviations from normality. This is not surprising, since heterogeneity of the variance often goes together with non-normality, in particular skewness, of the data.

Finally it is also necessary that all outcomes are independent to obtain correct estimations of the residual standard deviation. This would not be the case in so-called longitudinal studies where the outcome is measured repeatedly over time for the same subject.

In the next section we will describe how to handle deviations from the previously mentioned assumptions.

2.3 Deviations from the assumptions in linear regression analysis

The primary assumption in linear regression analysis is the assumption that the outcome varies linearly in the predictor. Whenever residual plots suggest that this is not the case, we could consider transforming the explanatory variable. In dose-response studies where, for example, the impact of increasing doses of a toxic substance on a phenotype in test animals is studied, the (mean) outcome will often not vary linearly in function of the administered dose, but will vary linearly in function of the logarithm of the administered dose. In that case, we could opt to include the log-transformed explanatory variable as a predictor in the model. For other examples it might happen that other transformations than the log-transformation are better suited, such as the square root ($\sqrt{x}$) or the inverse ($1/x$) transformation.

An advantage of transforming the explanatory variable is that it is easy to accomplish, a disadvantage is that this often complicates the interpretation of the coefficient in the model. However, this will not happen when applying a log-transformation, because an increase in log-dose with, for example, 1 unit is equivalent with a change in dose with a factor $\exp(1) = 2.78$. Transforming the explanatory variable does not have a direct influence on the homogeneity of the variance or on the normality of the outcomes (for fixed values of the predictor variable), except by improving the linearity of the model. Therefore this option is often less suitable when there are strong deviations from normality.

An alternative option to improve the linearity of the model, is *higher-order regression.* Here nonlinear relations are directly modeled by including higher-order terms in the model. We could, for example, consider a second-order model

$$E(Y|X) = \alpha + \beta_1 X + \beta_2 X^2,$$

in which case the regression curve will be parabolic, or a third-order model:

$$E(Y|X) = \alpha + \beta_1 X + \beta_2 X^2 + \beta_3 X^3,$$

where the regression curve will be a polynomial of degree 3. This method can be seen as some sort of transformation of the explanatory variable and essentially has the same properties, advantages, and disadvantages. However, an additional advantage is that in this case there is no need to decide yourself on a transformation, because the method itself will implicitly estimate a good polynomial.

Finally we could also consider transforming the outcome instead of the explanatory variable. For example, when the outcomes are right skewed, it is often appropriate to perform a log-transformation of the outcomes and include this new variable as the outcome variable in the model. Usually this not only improves the linearity of the model, but it will also improve the normality of the residuals with a more constant variability. This method has the same advantages and disadvantages as a transformation of the explanatory variable. A big difference between both options that greatly influences the choice between both methods is that, contrary to transformations of the outcome, transformations of the independent variable have little to no influence on the distribution of the residuals (unless via changes in their mean). Normally distributed residuals in particular will remain rather normally distributed after transforming the explanatory variable, whereas they might no longer be normally distributed after a transformation of the outcome variable, and vice versa.

Woody debris and tree density, continued

In the analysis of the CWD model we determined that, although the linearity assumption is well met for the chosen model, the residuals are not normally distributed with a constant variance. Therefore a transformation of the outcome is the only sensible choice out of the previously mentioned options. The fact that the outcomes can only take on nonnegative values compels us to consider the log-transformation because that transformation extends the range of the outcomes to all real values. This is indeed desirable since a linear regression model basically allows the mean outcome to vary from $-\infty$ to $+\infty$, as long as we vary the predictor x enough. Remark that we, as a result of this, indeed obtained negative predictions for the CWD basal area for lakes with relatively few trees along the shoreline.

Log-transforming the outcome gives us the following model:

```
Call:
lm(formula = log(CWD.BASA) ~ RIP.DENS, data = trees)

Residuals:
     Min       1Q   Median       3Q      Max
-2.23086 -0.78379  0.04559  0.72335  2.05022

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.5570100  1.0739690  -0.519   0.6121
RIP.DENS     0.0031573  0.0008222   3.840   0.0018 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.274 on 14 degrees of freedom
Multiple R-squared:  0.513, Adjusted R-squared:  0.4782
F-statistic: 14.75 on 1 and 14 DF,  p-value: 0.001802
```

The residual plots for this model are shown in Figure 2.7 below.

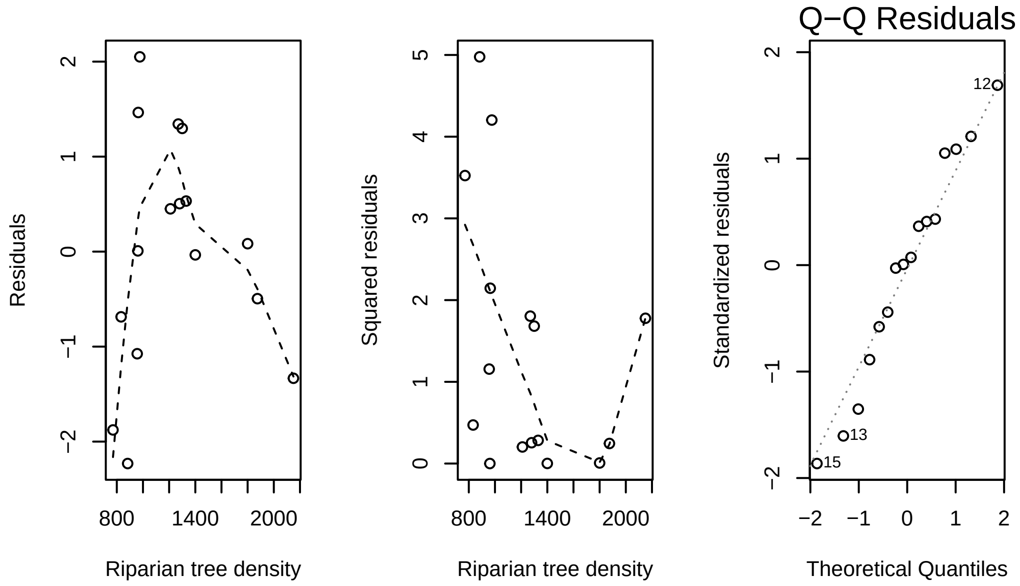


Figure 2.7: Analysis of the CWD data (linear trend on logarithmic scale). Left: scatterplot of the residuals. Middle: scatterplot of the squared residuals. Right: QQ-plot of the residuals.

Although the residuals follow relatively well the normal distribution, deviations from linearity and homoscedasticity emerge. The pattern in Figure 2.7 seems to be parabolic and makes it necessary to include a second order term in the model, so that we obtain the following model:

```
Call:
lm(formula = log(CWD.BASA) ~ RIP.DENS + I(RIP.DENS^2), data = trees)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.6872 -0.4462 -0.1621  0.4214  2.1399 

Coefficients:
                Estimate Std. Error t value Pr(>|t|)   
(Intercept)   -9.686e+00  3.114e+00  -3.110  0.00828 **
RIP.DENS       1.726e-02  4.673e-03   3.693  0.00270 **
I(RIP.DENS^2) -4.960e-06  1.628e-06  -3.047  0.00935 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 1.01 on 13 degrees of freedom
Multiple R-squared:  0.7159,    Adjusted R-squared:  0.6722
F-statistic: 16.38 on 2 and 13 DF,  p-value: 0.0002801
```

Compared to Figure 2.7, the residual plots for this model show somewhat less of a pattern.

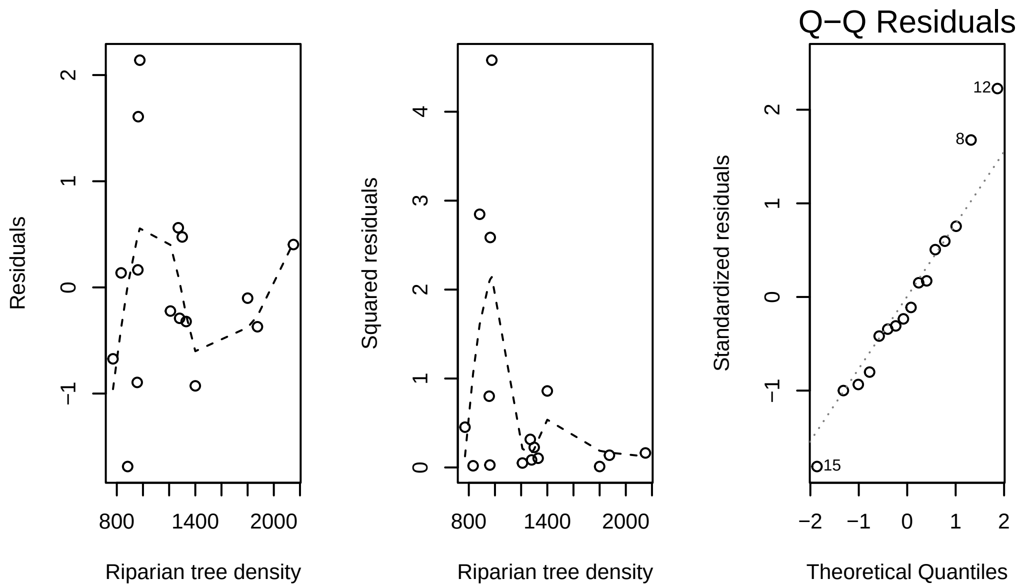


Figure 2.8: Analysis of the CWD data (quadratic trend on logarithmic scale). Left: scatterplot of the residuals. Middle: scatterplot of the squared residuals. Right: QQ-plot of the residuals.

```
Call:
lm(formula = log(CWD.BASA) ~ RIP.DENS + I(RIP.DENS^2), data = trees)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.6872 -0.4462 -0.1621  0.4214  2.1399 

Coefficients:
                Estimate Std. Error t value Pr(>|t|)   
(Intercept)   -9.686e+00  3.114e+00  -3.110  0.00828 **
RIP.DENS       1.726e-02  4.673e-03   3.693  0.00270 **
I(RIP.DENS^2) -4.960e-06  1.628e-06  -3.047  0.00935 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 1.01 on 13 degrees of freedom
Multiple R-squared:  0.7159,    Adjusted R-squared:  0.6722
F-statistic: 16.38 on 2 and 13 DF,  p-value: 0.0002801
```

We conclude that

$$E\{\ln(Y)|X\} = -9.7 + 0.017X - 5.0\ 10^{-6}X^2$$

or, equivalently, that the geometric mean CWD basal area varies in function of the tree density X as $\exp(-9.7+0.017X-5.0\ 10^{-6}X^2)$. Although the estimated geometric mean for high tree densities suggests a stabilising or even declining trend in the amount of CWD with increasing tree density, the accompanying 95% confidence intervals indicate that this suggestion is very imprecise and that even strong increases are compatible with the observed data. The regression line and associated confidence intervals are shown in Figure 2.9 below.

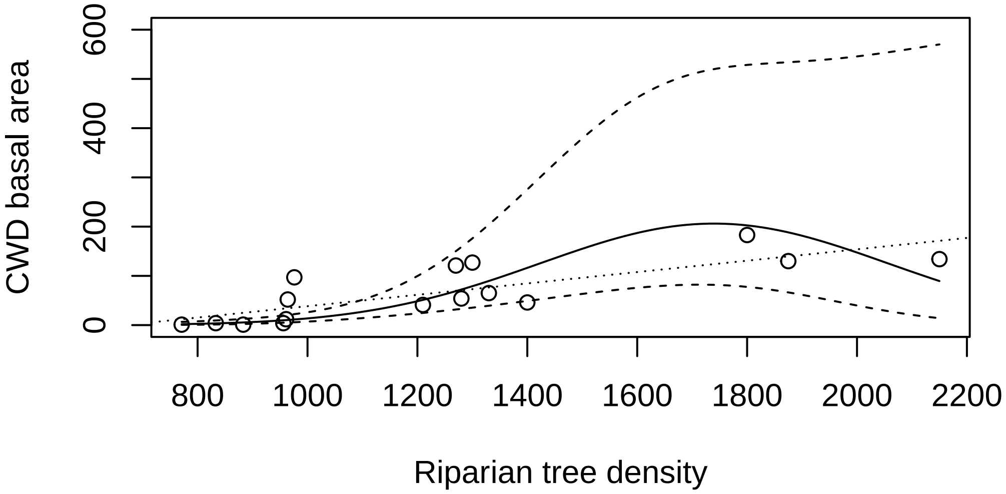


Figure 2.9: Scatterplot of the CWD basal area versus tree density X with linear regression line (dotted line), estimated geometric mean $\exp(-9.7+0.017X-5.0\ 10^{-6}X^2)$ (solid line), and accompanying 95% confidence intervals (dashed lines).

For certain types of outcomes there exist *variance stabilising transformations* for the outcome that are aimed at fulfilling the assumption of homoscedasticity. For proportions or percentages we often use the arcsin-transformation which transforms the outcome Y in $\arcsin\sqrt{Y}$, because it can be shown that percentages (given certain conditions) have a constant variance after such a transformation. If the transformation of the outcome is not helping or is not appropriate (e.g., because it is harmful for the interpretation of the model) and there is a consistent pattern of unequal variance (e.g., increasing variance in the outcome for increasing predictor values), we could also determine *weighted least squares* estimates. Another alternative would be to estimate *generalized linear models*, which also allow other distributions than the normal one. Both types of solutions, i.e., weighted least squares estimates and generalized linear models, are beyond the scope of this course.

2.4 Inference in regression models

In linear regression analysis we often want to test whether or not the slope β is equal to 0, i.e., whether or not a linear relation between Y and X exists. We wish, for example, to test if the CWD basal area per km is linearly related to the tree density, or if there exists a linear relation between the gene frequency of the allele Lap94 in the mussel Mytilus edulis and the eastern distance from Southport. We can show that the least squares estimate $\hat{\beta}$ is a sensible measure to perform tests on β since it is an unbiased estimator of β (and hence not systematically too high or too low), on the condition that the model is correct. If additionally the outcomes are normally distributed for a given predictor value X, and have a homogeneous variance, then its standard error can be estimated as

$$SE(\hat{\beta}) = \sqrt{\frac{MSE}{\sum_i (X_i - \bar{X})^2}},$$

where the *mean squared error* (MSE) is defined as $\frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$, and we can obtain tests and confidence intervals for β based on

$$\frac{\hat{\beta} - \beta}{SE(\hat{\beta})} \sim t_{n-2}.$$

Frequency of Lap94 and distance from Southport

Previously we established, admittedly under the wrong assumption that the gene frequency varies linearly in function of the eastern distance from Southport, the following model:

```

Call:
lm(formula = freq ~ km, data = southport)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.13744 -0.05784  0.01486  0.03923  0.10675 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
```

```
(Intercept) 0.045830   0.036361    1.26    0.227
km          0.004514   0.000536    8.42 4.56e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.06955 on 15 degrees of freedom
Multiple R-squared:  0.8254,    Adjusted R-squared:  0.8137
F-statistic:  70.9 on 1 and 15 DF,  p-value: 4.557e-07
```

Based on this output we can construct a 95% confidence interval for β as

$$[0.004514 - 2.13 \times 0.000536, 0.004514 + 2.13 \times 0.000536] = [0.003372, 0.005656],$$

where we use the fact that we posses $n = 17$ observations and that $t_{15,0.975} = 2.13$. We conclude that the mean increase in the frequency of the allele Lap94 as we move 10 km east of Southport can be expected between 3.372% and 5.656% with 95% confidence. The p-value for the test of the null hypothesis $\beta = 0$ (versus the alternative $\beta \neq 0$) is the probability that a t_{15}-distributed random variable in absolute value is larger than $0.004515/0.000536 = 8.42$. This probability is 4.56×10^{-7} and gives a very strong indication that there is a change in mean allele frequency in the eastern direction from Southport.

We could also test if the intercept takes on a certain value (e.g., 0). Again the least squares estimate $\hat{\alpha}$ is meaningful since it is an unbiased estimator of α, provided that the model is correct. If additionally the outcomes are normally distributed for a given predictor value X, and have a homogeneous variance, then its standard error can be estimated as

$$SE(\hat{\alpha}) = \sqrt{MSE\left\{\frac{1}{n} + \frac{\bar{X}^2}{\sum_i (X_i - \bar{X})^2}\right\}}.$$

Tests and confidence intervals for α can be obtained based on

$$\frac{\hat{\alpha} - \alpha}{SE(\hat{\alpha})} \sim t_{n-2}.$$

Similarly, $\hat{Y}_h = \hat{\alpha} + \hat{\beta} X_h$ will be an unbiased estimator of $E(Y|X_h) = \alpha + \beta X_h$. Its standard error is

$$SE(\hat{Y}_h) = \sqrt{MSE\left\{\frac{1}{n} + \frac{(X_h - \bar{X})^2}{\sum_i (X_i - \bar{X})^2}\right\}}.$$

and tests and confidence intervals for $E(Y|X_h)$ depend on

$$\frac{\hat{Y}_h - E(Y|X_h)}{SE(\hat{Y}_h)} \sim t_{n-2}.$$

ℹ Frequency of Lap94 and distance from Southport

The p-value for the test that the allele frequency α in Southport is 1%, versus the alternative that it would be less, is the probability that a t_{15}-distributed random variable is smaller than $(0.045830 - 0.01)/0.036361 = 0.9853964$. This probability is 0.17 and suggests that, at the 5% significance level, there is not enough evidence to support the claim that the allele frequency in Southport is less than 1%. A 95% confidence interval for α is obtained as

$$[0.045830 - 2.13 \times 0.036361, 0.045830 + 2.13 \times 0.036361] = [-0.031619, 0.123279].$$

The fact that this interval contains negative values suggests that the allele frequency α in Southport was estimated very inaccurately and that the linear model probably does not describe these data very well (since theoretically allele frequencies cannot be negative).

2.5 The multiple correlation coefficient

For the data in the CWD example, the standard deviation on the CWD basal area is 58.03 per km and the residual standard deviation is 36.32 per km. This shows that the measurements of the outcome vary less for lakes with the same tree density than in the entire population of lakes. This is not surprising, because part of the variability on the outcome measurements is explained by the fact that different lakes have different tree densities. In this section we will further exploit this idea to gain insight in the quality of the regression line.

The total squared deviation of the data around their mean can be split as follows:

$$\begin{aligned} SS_{Total} = \sum_{i=1}^{n}(y_i - \bar{y})^2 &= \sum_{i=1}^{n}(\hat{\alpha} + \hat{\beta}x_i - \bar{y})^2 + \sum_{i=1}^{n}(y_i - \hat{\alpha} - \hat{\beta}x_i)^2 \\ &= \sum_{i=1}^{n}(\hat{\alpha} + \hat{\beta}x_i - \bar{y})^2 + \sum_{i=1}^{n} e_i^2 \\ &= SS_{Regression} + SS_{Residual}\,. \end{aligned}$$

For the second equality, we've used the defining relations for the coefficients of a linear regression, i.e.

$$\sum_{i=1}^{n}(\hat{\alpha} + \hat{\beta}x_i - y_i) = 0 \quad \text{and} \quad \sum_{i=1}^{n}(\hat{\alpha} + \hat{\beta}x_i - y_i)x_i = 0$$

to show that the cross-product vanishes.

In this sum, $SS_{Regression} = \sum_{i=1}^{n}(\hat{\alpha}+\hat{\beta}x_i-\bar{y})^2$ indicates how much the points on the regression line vary around the group mean $\bar{y}$, and the $SS_{Residual} = \sum_{i=1}^{n} e_i^2$ reflects the residual variation of the observations around the regression line. The latter is called the *residual sum of squares* and indicates how much of the variation on the measurements is not explained by the regression model. The former is the *regression sum of squares* and indicates the amount of variability on the measurements that is explained by the regression model. The ratio of the regression sum of squares and the total sum of squares SS_{Total},

$$R^2 = \frac{SS_{Regression}}{SS_{Total}}$$

expresses the percentage of the variation on the data that is captured by their association with the explanatory variable, and is called the *coefficient of determination* or *multiple correlation coefficient*. It is generally denoted by R^2 and is a measure for the *predictive value* of the explanatory variable. In other words, it expresses how well the explanatory variable(s) predict(s) the outcome. This coefficient lies always between 0 and 1, where a value equal to 1 indicates that there is no residual variation around the regression line en hence the outcome shows a perfect (linear) relationship with the predictor. Analogously, a value 0 of R^2 implies that there is no association between the outcome and the predictor.

Often it is incorrectly claimed that a linear regression model is bad if the multiple correlation coefficient is low (e.g., 0.2). If the goal of the study is to predict the outcome based on the explanatory variables, a large value of R^2 is indeed needed because in the case of a low value there remains a lot of variability on the outcomes that is not explained by the explanatory variables. However, if the goal of the study is to determine the effect of an exposure on the outcome, then a linear regression model is good as soon as it correctly describes the association between on the one hand the outcome and on the other hand the exposure and possible confounders. Whenever exposure and confounders are weakly related to the outcome, then a small R^2-value is expected, even if a correct regression model is used.

Woody debris and tree density

Recall that we found the following model for the logarithm of the CWD basal area and the riparian tree density:

```

Call:
lm(formula = log(CWD.BASA) ~ RIP.DENS + I(RIP.DENS^2), data = trees)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.6872 -0.4462 -0.1621  0.4214  2.1399 

Coefficients:
                Estimate Std. Error t value Pr(>|t|)   
(Intercept)   -9.686e+00  3.114e+00  -3.110  0.00828 **
RIP.DENS       1.726e-02  4.673e-03   3.693  0.00270 **
```

```
I(RIP.DENS^2) -4.960e-06  1.628e-06  -3.047  0.00935 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.01 on 13 degrees of freedom
Multiple R-squared:  0.7159,    Adjusted R-squared:  0.6722
F-statistic: 16.38 on 2 and 13 DF,  p-value: 0.0002801
```

We conclude that 71.6% of the variability on the log-transformed CWD basal area is explained by its association with the tree density. The riparian tree density is thus strongly predictive for the CWD basal area.

2.6 Multiple linear regression

So far we focused on describing the association between a certain outcome Y and a single predictor X. However, it is often more useful to describe the mean outcome not in terms of only one, but in terms of multiple predictors at the same time. This is illustrated in the following examples:

1. Often the association between a predicting variable X and an outcome Y will be disturbed due to a confounder C. For example, when determining the effect of asbestos (X) on the respiratory function (Y), age (C) is a confounder because it influences both the duration of the exposure and the respiratory function. To correct for this confounding, it is necessary to describe the association between X and Y separately for people of the same age (in other words, individuals with the same value for the confounder). Performing a separate linear regression for each observed age c amongst those people of that age c, doesn't make sense since usually there are only very little people with exactly the same age in a study. Especially when there are multiple confounders, this becomes problematic. In this section we will solve this problem by including the confounder C in the linear model.
2. In many studies we are interested in knowing which group of variables influences the outcome most. For example, understanding which aspects of habitat and human activity have a major impact on the biodiversity of the rain forest is an important objective of conservation biology. To that end, not only the size of the forest has to be taken into account, but also other factors, such as age and altitude of the forest, proximity of other forests, and so on. A study of the simultaneous effect of the different variables will allow us to get a deeper understanding of the variation in biodiversity between different forests. By inspecting in particular forests with a low or high biodiversity, new predictive factors for biodiversity might be discovered.
3. Whenever we want to predict an outcome for individuals, it is crucial that a lot of predictive information is available and that this information is used simultaneously in

a regression model. For example, the prognosis after treatment is highly uncertain for patients with an advanced stage of breast cancer. However, based on measured predictors before and after the operation, it would be possible to construct regression models that allow to predict a prognosis for each patient, using his or her own characteristics. Related predictions (but then for mortality risk) are used daily in intensive care units to express the severity of a patient's health. It goes without saying that better predictions can be made when a large number of predictors is taken into account at the same time.

Mineral composition vs growth

In this example we study the relation between the growth and mineral composition of the needles of the Japanese larch. The height Y of 26 trees was measured in cm, and for each tree the proportions of nitrogen X_n, phosphorus X_f, potassium X_p and residual ash X_r in dried needles were registered. Univariate regression models such as

$$E(Y|X_f) = \alpha + \beta_f X_f$$

only allow us to predict the height of a tree based on a single mineral. Obviously, we could obtain more accurate predictions if multiple minerals are taken into account at the same time.

Figure 2.10: Japanese larch, or *Larix kaempferi*. Image courtesy of Wikipedia, CC BY-SA 3.0

Note that for example the coefficient β_f in such a model might not show the pure effect of phosphorus. It is true that β_f represents the mean difference in length between trees that differ by 1 unit in phosphorus, but even if phosphorus would not have an influence on the growth of the Japanese larch, it could still be possible that trees with a higher proportion of phosphorus are larger (and thus $\beta_f > 0$) because, for example, they also contain more potassium. This is a problem of confounding (the effect of phosphorus is

> confounded with the effect of potassium) that can be resolved by comparing trees with a different level of phosphorus, but with an identical proportion of potassium. We will see in this section that multiple linear regression models make this possible in a natural way.

The technique that we will use to this end is called *multiple linear regression*, as opposed to simple linear regression used before. Suppose we observed a number of explanatory variables $X_1, ..., X_p$ and an outcome Y for n subjects. Suppose furthermore that the mean outcome can be described linearly with respect to these variables, i.e.,

$$E(Y|X_1 = x_1, ..., X_p = x_p) = \alpha + \beta_1 x_1 + ... + \beta_p x_p, \tag{2.4}$$

where $\alpha, \beta_1, ..., \beta_p$ are unknown. The principle of the *least squares method*, which we applied before, can also be used on this model to obtain estimates of these unknown numbers. The formulas for these estimates are of course more complex than before, but we will rely on computer software to do the calculations for us. For any given estimates $\hat{\alpha}, \hat{\beta}_1, ..., \hat{\beta}_p$, the linear regression model 2.4 will allow us to

1. Predict the expected outcome for subjects with given values $x_1, ..., x_p$ of the explanatory variables. This outcome is estimated as $\hat{\alpha} + \hat{\beta}_1 x_1 + ... + \hat{\beta}_p x_p$.
2. Verify to what extent the mean outcome differs between 2 groups of subjects that differ δ units for one of the explanatory variables $X_j, j = 1, ..., p$, but have the same values for all other variables $\{X_k, k = 1, ..., p, k \neq j\}$. After all:

   $$\begin{aligned} & E(Y|X_1 = x_1, ..., X_j = x_j + \delta, ..., X_p = x_p) - E(Y|X_1 = x_1, ..., X_j = x_j, ..., X_p = x_p) \\ & = \alpha + \beta_1 x_1 + ... + \beta_j (x_j + \delta) + ... + \beta_p x_p - \alpha - \beta_1 x_1 - ... - \beta_j x_j - ... - \beta_p x_p \\ & = \beta_j \delta\,. \end{aligned}$$

   In particular, we can interpret β_j as the difference in mean outcome between subjects that differ 1 unit in the value of X_j, but have the same value for all other explanatory variables in the model. This difference is estimated by $\hat{\beta}_j$.

> **Mineral composition vs growth**
>
> An analysis of the simple linear regression model $E(Y|X_f) = \alpha + \beta_f X_f$ in R gives the following output:
>
> ```
> Call:
> lm(formula = length ~ phosphor, data = needles)
>
> Residuals:
>      Min       1Q   Median       3Q      Max 
> -103.398  -42.582    2.331   40.845  120.220 
> ```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   -69.11      45.99  -1.503    0.146
phosphor     1060.29     177.08   5.988 3.51e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 61.71 on 24 degrees of freedom
Multiple R-squared:  0.599, Adjusted R-squared:  0.5823
F-statistic: 35.85 on 1 and 24 DF,  p-value: 3.511e-06
```

Based on these data, we conclude that trees with a phosphorus level that is 0.1% higher, are on average 1.06m taller. An analysis of the multiple linear regression model

$$E(Y|X_n, X_f, X_p, X_r) = \alpha + \beta_n X_n + \beta_f X_f + \beta_p X_p + \beta_r X_r$$

drastically changes this result, as indicated in the output:

```
Call:
lm(formula = length ~ nitrogen + phosphor + potassium + residu,
    data = needles)

Residuals:
   Min     1Q Median     3Q    Max
-61.56 -29.11  10.28  24.72  80.29

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -185.33      36.30  -5.106 4.67e-05 ***
nitrogen       97.76      24.57   3.979 0.000684 ***
phosphor      256.97     169.91   1.512 0.145321
potassium     126.57      46.43   2.726 0.012653 *
residu         40.28      36.61   1.100 0.283773
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 37.87 on 21 degrees of freedom
Multiple R-squared:  0.8679,    Adjusted R-squared:  0.8427
F-statistic: 34.48 on 4 and 21 DF,  p-value: 5.967e-09
```

The coefficient for phosphorus now implies that trees with an increase of 0.1% in their phosphorus level, but with the same levels of nitrogen, potassium, and residual ash, are

only 25.7 cm taller. The reason why we find such a difference of more than 1 m, can be found in the fact that trees that differ 0.1% in phosphorus level, also often differ in their levels of nitrogen, potassium, and residual ash.
The R^2-value in the output is 86.8%, which tells us that the majority of the variability on the length of Japanese larches can be explained by the mineral composition of the needles. The 4 chosen minerals are therefore strongly predictive for the outcome.

The previous example illustrates that multiple linear regression can be usefully applied to control for confounding. Assume, for example, that the association between variables X (e.g., exposure to asbestos) and Y (e.g., respiratory function) is perturbed by a third variable C (e.g., age). We can then control for this by fitting the following multiple regression model:

$$E(Y|X,C) = \alpha + \beta_1 X + \beta_2 C.$$

Assuming the linearity conditions of the model are fulfilled, the association between X and Y is now indeed controlled for the confounder C. After all,

$$\begin{aligned} \beta_1 &= \alpha + \beta_1(x+1) + \beta_2 c - \alpha - \beta_1 x - \beta_2 c \\ &= E(Y|X = x+1, C = c) - E(Y|X = x, C = c)\,. \end{aligned}$$

In other words, β_1 represents the mean difference in outcome between individuals that differ 1 unit in X, but all have the same value for the confounder C (e.g., people of the same age). This way we compare comparable groups of individuals, effectively correcting for the perturbing effect of the confounder. In the literature, the estimate for β_1 is therefore called the *adjusted effect* of X on Y, to indicate that we controlled for confounders. If C is the only confounder[1] for the association between X and Y, then this can indeed be interpreted as the causal effect of an increase of 1 unit of X on the mean outcome. The estimate for the association between X and Y in the model without confounders (i.e., the estimate β_1^* in the model $E(Y|X) = \alpha^* + \beta_1^* X$) is then called the *unadjusted effect* of X on Y, even though it has, due to confounding, no causal meaning (in other words, it does not represent an effect of a unit change in X on the mean outcome).

In some cases we would also like to know if the effect of a variable X on another variable Y depends on a third variable C. This could, for example, happen when we wish to investigate if the effect of asbestos on the respiratory function changes with increasing age. Such issues are very relevant in the context of scientific studies on gene-environment interactions. For asthma or COPD, for example, there are strong indications that gene-environment interactions with a history of smoking are important factors in determining the severity of the disease (in particular that the role of certain genes is amplified by a history of smoking). In pharmacogenetics there exists a huge interest in gene-medicine interactions in order to ascertain if certain drugs are

[1] In practice we seldom know for sure (unless in randomised studies) if a certain variable C is the only confounder for the association between X and Y; often there can be unmeasured confounders that are not included in the dataset and for which it thus is also impossible to correct.

especially effective in the presence of certain genes. For example, gene-medicine interactions were discovered for steroids with regard to their effect on the respiratory function of asthma patients.

To statistically model such *interaction* or *effect modification* between 2 variables X (e.g., use of steroids or not) and C (e.g., presence/absence of a certain gene), we could add the product of those variables to the model:

$$E(Y|X,C) = \alpha + \beta_1 X + \beta_2 C + \beta_3 XC.$$

The effect of a change in X on the mean outcome now is

$$\begin{aligned} E(Y|X=x+1, C=c) - E(Y|X=x, C=c) &= \alpha + \beta_1(x+1) + \beta_2 c + \beta_3(x+1)c \\ &\quad -\alpha - \beta_1 x - \beta_2 c - \beta_3 xc \\ &= \beta_1 + \beta_3 c \end{aligned}$$

when C remains unchanged. Remark that the effect of a change in X for a constant value of C now indeed depends on the chosen value of C.

Mineral composition vs growth

The regression model

$$E(Y|X_n, X_f) = \alpha + \beta_n X_n + \beta_f X_f \tag{2.5}$$

assumes that nitrogen and phosphorus are linearly associated with the length of Japanese larches, but that the strength of the association between nitrogen and length does not depend on the phosphorus level in the needles. This model is described in R as

```
Call:
lm(formula = length ~ nitrogen + phosphor, data = needles)

Residuals:
    Min      1Q  Median      3Q     Max 
-57.834 -34.950  -0.539  20.364 127.287 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -189.64      42.53  -4.460 0.000179 ***
nitrogen      123.83      26.62   4.652 0.000111 ***
phosphor      604.44     162.65   3.716 0.001135 ** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 45.25 on 23 degrees of freedom
```

```
Multiple R-squared:  0.7934,    Adjusted R-squared:  0.7755
F-statistic: 44.17 on 2 and 23 DF,  p-value: 1.329e-08
```

In particular, the output suggests that trees that differ 0.1% in their nitrogen level, but have the same level of phosphorus, differ on average 0.12 m in length, regardless from the actual level of phosphorus. Figure 2.11 (left) illustrates indeed that the mean tree length increases at the same rate with an increase of the level of nitrogen, irrespective of the level of phosphorus in the needles. Similarly, Figure 2.11 (right) shows that the association between tree length and nitrogen is 123.8, no matter what the level of phosphorus is.

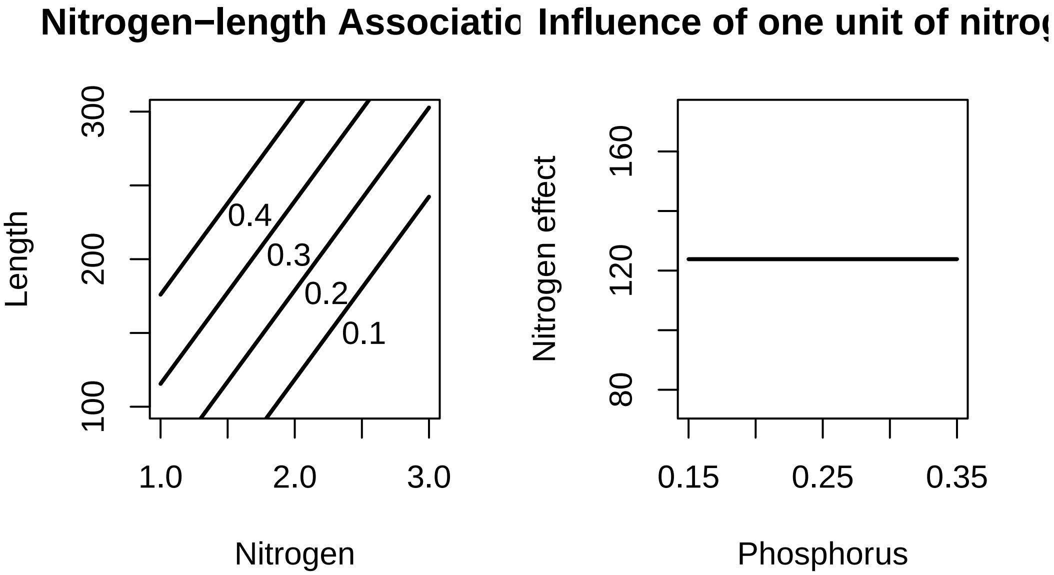


Figure 2.11: Association between tree length and levels of nitrogen and phosphorus in model 2.5. Left: mean tree length in function of nitrogen level for different levels of phosphorus. Right: size of the association between nitrogen and tree length for different levels of phosphorus.

The linear regression model

$$E(Y|X_n, X_f) = \alpha + \beta_n X_n + \beta_f X_f + \beta_{nf} X_n X_f \tag{2.6}$$

also assumes that nitrogen and phosphorus are linearly associated with the length of Japanese larches, but allows the association between nitrogen and length to vary with the level of phosphorus in the needles. This model is described in R as

```

Call:
lm(formula = length ~ nitrogen * phosphor, data = needles)
```

```
Residuals:
    Min      1Q  Median      3Q     Max 
-57.533 -32.025   0.205  23.121 107.795 

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)  
(Intercept)         198.42     211.94   0.936   0.3593  
nitrogen            -79.04     111.67  -0.708   0.4865  
phosphor           -971.01     858.65  -1.131   0.2703  
nitrogen:phosphor   794.97     426.20   1.865   0.0755 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 42.99 on 22 degrees of freedom
Multiple R-squared:  0.8216,    Adjusted R-squared:  0.7973 
F-statistic: 33.78 on 3 and 22 DF,  p-value: 2.057e-08
```

In particular, the output suggests that trees that differ by 0.1% in nitrogen level, but with the same level of phosphorus x_f, differ on average by $-7.9 + 79.5x_f$ centimeter in length. Trees with a phosphorus level of 0.1% will therefore have approximately the same length, irrespective of their level of nitrogen (because $-0.79+7.95\times 0.1 \approx 0$). Trees that differ 0.1% in nitrogen level and have a phosphorus level of 0.2%, will on average differ $-7.9 + 79.5 \times 0.2 = 8$ cm in length. Figure 2.12 (left) shows indeed that the mean tree length grows at different rates with increasing levels of nitrogen, depending on the phosphorus level. Similarly, Figure 2.12 (right) illustrates that the association between tree length and nitrogen changes linearly according to the expression $-7.9 + 79.5x_f$ in function of the level of phosphorus x_f. In particular, the influence of nitrogen increases when the needles contain more phosphorus.

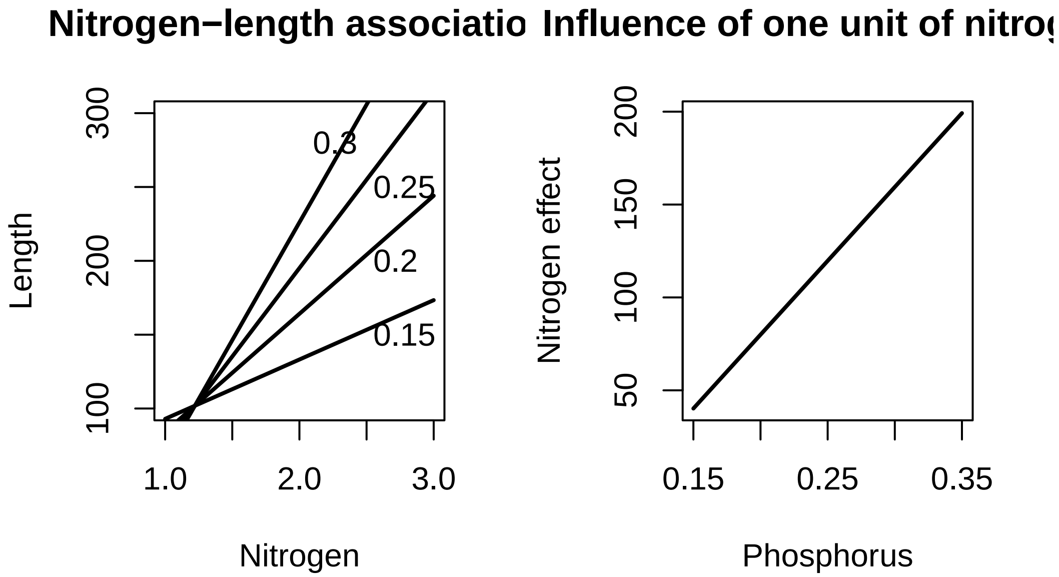


Figure 2.12: Association between tree length and levels of nitrogen and phosphorus in model (ef{eq:regrint2}). Left: mean tree length in function of nitrogen level for different levels of phosphorus. Right: size of the association between nitrogen and tree length for different levels of phosphorus.

Finally, we wish to remark that residual plots similar to the ones used for simple linear regression models can be used to verify the linearity assumption for multiple linear regression models. However, since the model now contains multiple predictors, it makes sense to plot the residuals and squared residuals in function of each of the predictors separately. In that way we can verify with respect to which variable the linearity of the model might fail. To avoid too much work in models with a lot of predictors, we sometimes also plot the residuals in function of the predictions $\hat{\alpha} + \hat{\beta}_1 x_1 + ... + \hat{\beta}_p x_p$. After all, if we note a deviation with respect to the predictions, then that deviation also holds with respect to at least 1 of the predictors (since the predictions are function of the predictors).

2.7 Inference in regression models and model construction

In practice we often possess a large number of predictors and it is therefore not always obvious what regression model to consider. After all, such a model does not necessarily contain just the predictors separately, but possibly also higher order terms when one of the predictors is not linearly associated with the outcome, or interactions between 2 predictors when there is an indication that the effect of a certain predictor depends on the value of another predictor. All in all, there are in practice easily thousands of possible models that could have generated the data.

Hoping to obtain a model that is as accurate as possible, we could opt to include as many predictors as possible in the model, together with their interactions and higher order terms. However, such a strategy comes with big disadvantages. First of all, the final model will contain an enormous amount of predictors. This means that based on a limited number of observations, a lot of parameters need to be estimated, leading to imprecise estimates. With this we mean that, if we would repeat the study in a similar way, the results for the regression parameters can vary strongly between samples and that there is thus a big risk that they deviate greatly from the true population values. Secondly, coefficients in models with higher order terms and interactions are more difficult to interpret. This makes complex models less interesting for scientific purposes since in that case we pursue as much simplicity as possible (unless this would give incorrect models). After all, models that contain superfluous terms have the tendency, even if they are correct, to *overfit* the data. This means that predictions obtained by those models will give good approximations for those outcomes that have been observed, but bad approximations for outcomes that are observed in a similar sample that was not used to construct the regression model. Keeping these disadvantages in mind, we will strive to construct a model that is as simple as possible by preventing unimportant predictors from entering the model.

Starting from a given regression model 2.4, we can decide for each predictor x_j whether or not it is essential in the model by testing the null hypothesis that the corresponding coefficient $\beta_j = 0$ (versus a two-sided alternative). If the outcome is normally distributed for given values of the predictors or if the sample is sufficiently large, and if furthermore the variance on the outcomes is homogeneous (i.e., does not depend on the predictors), then such tests can relatively easy be obtained by using the knowledge that for each coefficient β_j in the model

$$\frac{\hat{\beta}_j - \beta_j}{SE(\hat{\beta}_j)} \sim t_{n-p}$$

where p represents the number of unknown parameters in the model. We will not discuss the details on the calculation of the standard errors, but trust the software to estimate those for us. Confidence intervals for β_j can also be obtained using this result.

Mineral composition vs growth

Based on the linear regression model 2.6 we can decide if there exists an indication that the measure in which tree length is associated with the nitrogen level in the needles, is also influenced by the presence of phosphorus. To gain this kind of insight, we can test whether $\beta_{nf} = 0$ versus a two-sided alternative. Based on previous R-output, we find that the test statistic equals 794.9668/426.1981 = 1.8653. This value can also be directly found in the output under the output for t-value. For 26 available observations and 4 unknown parameters (intercept, 2 predictors and their interaction), we find that the corresponding p-value is the probability that a t_{22}-distributed random variable is, in absolute value, more extreme than 1.8653. This probability equals 7.55% as can be seen in the output under `Pr(>|t|)`. Hence, there is insufficient proof, at the 5% significance level, to conclude that the measure in which tree length is associated with the needles' nitrogen level, is phosphorus-dependent. A 95% confidence interval for β_{sf} is found as

$$794.9668 \pm t_{22,0.025} \times 426.1981 = [-88.91, 1678.85]$$

The interval allows for large effect modifications, because of which we cannot just conclude that the association between nitrogen and tree length is not phosphorus-dependent. The relatively small negative values suggest that a weak enfeeblement of the association between nitrogen and the tree length for increasing phosphorus values is compatible with the data. The positive values in the interval also suggest that a small to large strengthening of the association between nitrogen and the tree length for increasing levels of phosphorus is compatible with the data.

Whenever the aim of the regression model is to predict the outcomes based on the predictors or to describe associations between on the one hand the outcome and on the other hand the predictors, *automatic selection procedures* can often come in very handy. One of these procedures, the *stepwise selection procedure*, starts with a model that only contains the intercept. In the next step, the predictor that is most strongly associated with the outcome (in the sense that the corresponding regression parameter has the largest absolute t-value or the smallest p-value) is included in the model, provided that it is significantly associated with the outcome.

Starting from the resulting model, the predictor that is now most strongly associated with the outcome (in the sense that the corresponding regression parameter has the largest absolute t-value or the smallest p-value) is added. It is often advised to perform these tests at the 10% significance level to avoid that important predictors disappear from the model. Then the predictors that are no longer significantly associated with the outcome are, starting with the least significant, one by one removed from the model until we obtain a model with only significant predictors. Remark that predictors which were significantly associated with the outcome in previous steps can become insignificant due to confounding and similar reasons. Next, the predictor that is most strongly associated with the outcome is included in the new model. This algorithm is repeated until the model doesn't change any more.

Mineral composition vs growth

Simple regression models produced the following t-values: 6.97, 5.99, 6.35, and 5.87 for a model with respectively only nitrogen, only phosphorus, only potassium, and only residual ash. In a stepwise selection procedure, we will opt to first include nitrogen in the model. It is indeed retained, because it is significantly associated with the tree length at the 10% significance level. Adding respectively phosphorus, potassium or residual ash to the resulting model gives t-values of 3.72, 4.91, and 3.07. Based on these t-values, we now add potassium to the model. This gives the following output:

```
Call:
lm(formula = length ~ nitrogen + potassium, data = needles)

Residuals:
    Min      1Q  Median      3Q     Max 
-75.625 -30.298   5.557  27.527  61.897 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -180.87      36.94  -4.896 6.04e-05 ***
nitrogen      123.26      22.41   5.499 1.36e-05 ***
potassium     188.69      38.40   4.913 5.79e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 39.98 on 23 degrees of freedom
Multiple R-squared:  0.8387,    Adjusted R-squared:  0.8247 
F-statistic: 59.79 on 2 and 23 DF,  p-value: 7.727e-10
```

Both predictors nitrogen and potassium are significantly associated with the outcome and are thus retained in the model. This procedure continues until the model is stable. Once the first order structure is known (i.e., the algorithm has converged, but so far no higher order terms or interactions have been added), the same principle can be applied for higher order terms and interactions. Please note that this is applied hierarchically, meaning that lower order terms will never be removed from the model as long as the higher order terms are significantly associated with the outcome. Continuing in this way, we obtain the model:

```
Call:
lm(formula = length ~ nitrogen + phosphor + potassium + residu + 
    phosphor:residu + phosphor:nitrogen, data = needles)

Residuals:
    Min      1Q  Median      3Q     Max 
-57.559 -18.976   0.007  11.608  55.941 

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)         129.09     169.33   0.762   0.4552
nitrogen           -150.76      97.08  -1.553   0.1369
phosphor          -1000.02     682.98  -1.464   0.1595
```

```
potassium           137.97      41.24   3.346   0.0034 **
residu              193.80      89.10   2.175   0.0425 *
phosphor:residu    -598.08     290.02  -2.062   0.0531 .
nitrogen:phosphor   951.78     371.57   2.562   0.0191 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 33.43 on 19 degrees of freedom
Multiple R-squared:  0.9069,    Adjusted R-squared:  0.8774
F-statistic: 30.83 on 6 and 19 DF,  p-value: 8.159e-09
```

Remark that nitrogen and phosphorus, although not significantly associated with tree length at the 10% significance level, are retained in the model because these terms appear in higher order terms (i.e., `phosphorus:nitrogen`). As an exercise, think about how the parameters in this "final model" can be interpreted.

Although the previously described algorithm offers a relatively simple and natural way to construct models, this strategy also has its disadvantages. It relies upon a large number of hypothesis tests and is therefore extremely sensitive to the problem of multiple testing. The risk with this is that predictors that are insignificant at population level will be included in the model because by chance they seem to be associated with the outcome in the sample. Keeping this in mind, it is wise to avoid the problem of multiple testing in practice by not exhaustively considering all possible higher order terms and interactions, but first make a limited selection of higher order terms and interactions that might be plausible, based on biological judgment and insight gained from diagnostic plots.

In recent years, several alternative techniques for model construction have been developed in the domain of machine learning which are based on *cross validation*. In short, cross validation is a technique where the regression model is repeatedly estimated based on a (varying) subset of the observations and its performance is then evaluated based on how well it predicts the remaining observations.

So far we considered situations where the aim of the regression model consisted in predicting the outcome or describing associations. If the goal of the regression model is to estimate the causal effect of a certain exposure (e.g., exposure to asbestos) on a certain outcome (e.g., respiratory function), then of course this exposure certainly needs to be part of the model and what remains is to make sure that all confounders for the association between both are included in the model. Recall that confounders are measurements that are at the same time associated with the exposure and the outcome, but which are influenced by neither. In this case it is important to try to form an idea (based on biological insights) of what possible confounders there are and to certainly not include any consequences of the exposure or the outcome in the regression model. Because of this, it is often not advisable to apply automatic selection procedures in this situation. Expected confounders that turn out to be not significantly associated with the outcome can, of course, still be removed one by one from the model.

Finally we like to remark that the previously mentioned hypothesis tests are all based on the assumption of normally distributed outcomes for a given predictor value (unless the sample is sufficiently large) and that the variance is homogeneous in the predictors. Residual plots can be used to verify these assumptions. Since the model now contains multiple predictors, it makes sense to plot the squared residuals against each of these predictors separately or against the predictions $\hat{\alpha} + \hat{\beta}_1 x_1 + ... + \hat{\beta}_p x_p$ obtained by the model.

2.8 Regression diagnostics

2.8.1 Multicollinearity

An important problem in multiple linear regression, and one that is often overlooked, is the impact of correlated predictor variables on parameter estimates and on hypothesis tests concerning these parameters. When the predictors are correlated, as is often the case for biological data, we say that the data are subject to *multicollinearity.* Heavy multicollinearity can have a serious impact on the estimated regression parameters. After all, when 2 predictors are strongly correlated, they share for a large part the same information and it thus becomes difficult to estimate the separate effects of both on the outcome. This is expressed by the fact that the calculations of the least squares estimates become numerically unstable in the sense that small modifications to the data or even adding or removing a predictor variable will have a huge impact on the size, and possibly even the sign, of the estimated regression coefficients. A second effect of multicollinearity is that the standard errors of the estimated regression coefficients can be largely inflated and the corresponding confidence intervals thus become very wide. However, as long as we only try to make predictions based on the regression model without extrapolating outside the range of the predictors, multicollinearity does not pose a problem.

Problems caused by multicollinearity can be recognised by the fact that results become numerically unstable. Large changes can occur in the coefficients after inclusion of a predictor, very wide confidence intervals can be obtained for some coefficients, or unexpected results can be found. Formally, we can get an idea of the measure of multicollinearity by inspecting the correlations between each pair of predictors in the regression model or through a *scatterplot matrix* that plots each pair of predictors on a scatterplot. However, such diagnostics for multicollinearity are not ideal. First of all, they give no information on how unstable the results become by the observed multicollinearity. Secondly, in models with 3 or more predictors, let's say X_1, X_2, X_3, it can happen that heavy multicollinearity exists, despite the fact that all pairwise correlations between predictors are low. This could happen, for example, when X_1 is strongly correlated with a linear combination of X_2 and X_3.

These mentioned disadvantages can be avoided by investigating the *variance inflation factor (VIF)*, which is defined for the k^{th} coefficient in the regression model as

$$\text{VIF}_k = \left(1 - R_k^2\right)^{-1}.$$

In this expression, R_k^2 represents the multiple correlation coefficient of a linear regression model for the k^{th} predictor based on all other predictors in the model. The VIF has the property that it equals 1 if the k^{th} predictor is not linearly associated with the other predictors in the model, and consequently when the k^{th} coefficient in the model is not subject to multicollinearity. The VIF is larger than 1 in all other cases. In particular, it expresses how much larger the observed variance on the estimate of the k^{th} coefficient would be compared to when all predictors would be independent. Therefore, the larger the VIF, the less stable the estimates will be. The average of the VIFs for the different predictors can, up to a proportionality factor, be interpreted as the mean squared distance between the estimated coefficients and the true coefficients in the model. The smaller the VIF, the closer the estimates are thus expected to be to their population values. In practice, multicollinearity for a regression coefficient is considered problematic when its VIF surpasses 10.

ℹ Prediction of body fat

It is relatively laborious and costly to determine the proportion of body fat in a person. For that reason, a number of studies have been set up in the past in order to unravel the pattern between the true percentage of body fat and several, more easily measurable, surrogates. One of these studies measured for 20 healthy women between the ages of 25 and 34 years the proportion of body fat Y, the thickness of the triceps skin fold X_1, the thigh circumference X_2, and the midarm circumference X_3. Pairwise scatter plots for this dataset are shown in Figure 2.13.

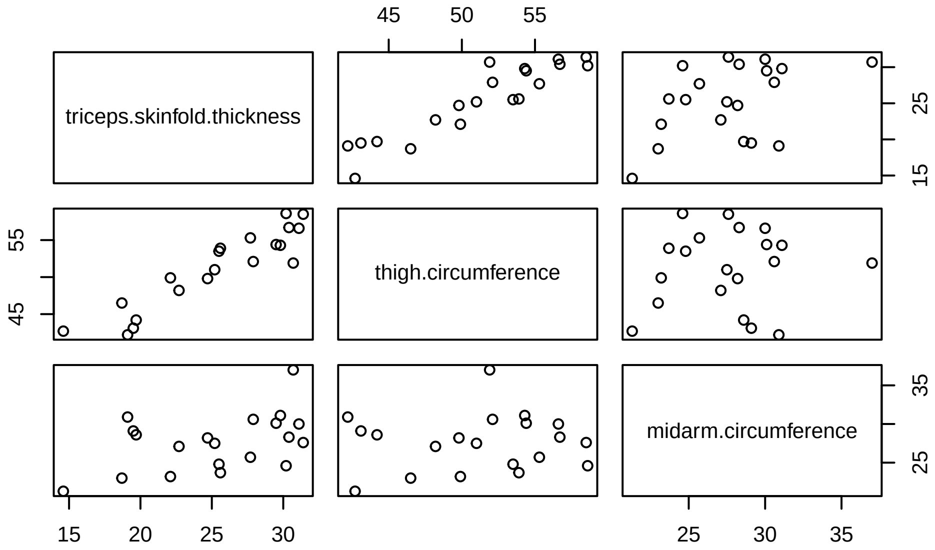


Figure 2.13: Scatterplot matrix for the bodyfat dataset.

If we are able to construct an accurate regression model based on these data, it will allow us in the future to make predictions of the proportion of body fat for healthy women between 25 and 34 years old, based on their triceps skin fold thickness, their thigh circumference, and their midarm circumference.
Including these 3 predictors simultaneously in the regression model gives the following model:

```
Call:
lm(formula = bodyfat ~ triceps.skinfold.thickness + thigh.circumference +
    midarm.circumference, data = bodyfat)

Residuals:
    Min      1Q  Median      3Q     Max
-3.7263 -1.6111  0.3923  1.4656  4.1277

Coefficients:
                           Estimate Std. Error t value Pr(>|t|)
(Intercept)                 117.085     99.782   1.173    0.258
triceps.skinfold.thickness    4.334      3.016   1.437    0.170
thigh.circumference          -2.857      2.582  -1.106    0.285
midarm.circumference         -2.186      1.595  -1.370    0.190

Residual standard error: 2.48 on 16 degrees of freedom
```

```
Multiple R-squared:  0.8014,    Adjusted R-squared:  0.7641
F-statistic: 21.52 on 3 and 16 DF,  p-value: 7.343e-06
```

The scatterplot matrix in Figure 2.13 indicates that there is multicollinearity in terms of the predictors X_1 and X_2, but not immediately for the midarm circumference X_3. Nevertheless we obtain substantial VIF values of respectively 708.84, 564.34, and 104.61 for the 3 predictors. This suggests that also the midarm circumference is susceptible to severe multicollinearity and hence shows that the scatterplot matrix indeed only gives a limited view on the problem of multicollinearity. On average, the VIF is 460, which shows that the mean (squared) distance between the estimates of the regression parameters and their true values is 460 times as large as when there would be no multicollinearity. In other words, there is a large problem of multicollinearity. This can also be observed in the regression output: not a single of the predictors is significantly associated with body fat, although the F-statistic[2] indicates, with a p-value of $7.34\ 10^{-6}$, that at for least 1 of the predictors there is strong evidence for an association with the outcome.

In the literature, numerous suggestions have been made for how to deal with the problem of multicollinearity. The most simple solution is to ban predictors that are strongly correlated with other predictors from the model. This makes sense when multiple predictors measure the same or a similar biological entity (e.g., a number of strongly correlated morphological traits). In other cases this might introduce a severe bias, namely when the predictor that is deleted from the model is an important confounder for the association between one of the remaining predictors and the outcome. Another approach, called *principal component regression*, roughly consists in transforming the predictors to a series of uncorrelated predictors, which solves the problem of multicollinearity. A third option, for example used in *ridge regression*, will allow that the estimates of the regression parameters are slightly biased in favour of a large increase in stability.

ℹ Mineral composition vs growth

The addition of higher order and interaction terms to the model typically introduces multicollinearity problems since normally a predictor X_1 is correlated with any function, e.g., X_1X_2 or X_1^2, of itself. In the previous section we constructed a model that included interactions between phosphorus and nitrogen as well as between phosphorus and residual ash. It therefore doesn't come as a surprise that Figure 2.14 (left) shows large VIFs, especially for the interaction terms. In such situations, namely when the model contains interactions and higher order terms, it typically helps to center the concerned variables. To this end, the concerned predictors are transformed by subtracting their respective sample mean. For example, X_n will be transformed to $X_n - \bar{X}_n$ and will be denoted by

[2] The F-statistic in the regression output always gives the result for a test of the null hypothesis that the regression parameters for all predictors are zero (in other words, that none of the predictors is associated with the outcome) versus the alternative that at least 1 predictor is associated with the outcome.

cnitrogen in the R code. Note that in Figure 2.14 (right) the problem of multicollinearity essentially has disappeared for the model that contains only these *centered variables*. Furthermore, the standard errors on the estimated regression coefficients have shrunk tremendously as can be seen in the output.

```

Call:
lm(formula = length ~ cnitrogen + cphosphorus + cpotassium +
    cresidu + cphosphorus:cresidu + cphosphorus:cnitrogen, data = needles_centered)

Residuals:
    Min      1Q  Median      3Q     Max
-57.559 -18.976   0.007  11.608  55.941

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)            188.933      9.013  20.963 1.35e-14 ***
cnitrogen               87.731     22.431   3.911 0.000939 ***
cphosphorus            273.664    152.171   1.798 0.088022 .
cpotassium             137.966     41.237   3.346 0.003397 **
cresidu                 43.938     34.908   1.259 0.223398
cphosphorus:cresidu   -598.078    290.020  -2.062 0.053134 .
cnitrogen:cphosphorus  951.782    371.568   2.562 0.019086 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 33.43 on 19 degrees of freedom
Multiple R-squared:  0.9069,    Adjusted R-squared:  0.8774
F-statistic: 30.83 on 6 and 19 DF,  p-value: 8.159e-09

         : carData

there are higher-order terms (interactions) in this model
consider setting type = 'predictor'; see ?vif
there are higher-order terms (interactions) in this model
consider setting type = 'predictor'; see ?vif
```

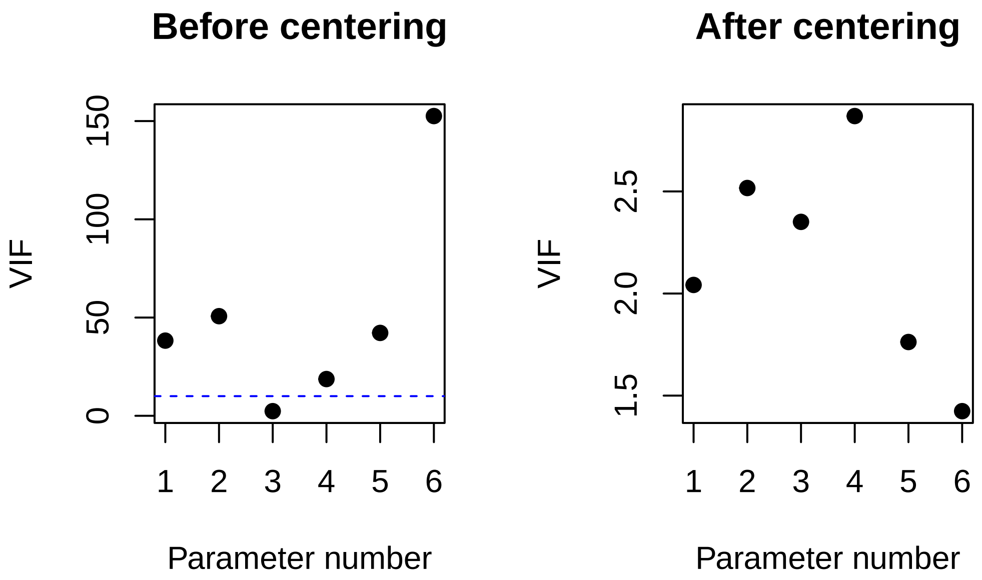


Figure 2.14: Variance inflation factors before (left) and after (right) centering. Parameter numbers 1 to 6 correspond respectively with $X_n, X_f, X_p, X_r, X_f X_r$ and $X_f X_n$.

2.8.2 Influential observations

Often, a dataset contains extreme observations, both for the outcome Y and the predictors X. These extreme observations can greatly influence the estimated regression parameters and regression line. This is no surprise, since the regression line represents the mean outcome in function of X and the mean is sensitive to outliers.

Figure 2.15 shows the possible influence of extreme observations, or outliers, on the regression line. Plot (a) shows a synthetic dataset consisting of 10 observations together with the regression line. The other plots show what happens to the regression line when one outlier, displayed in red, is added. In each case, we show the unmodified regression line (dashed line) and the regression line when the outlier is added (solid line).

The outlier in plot (b) follows the pattern of the unmodified regression line, and hence does not affect the regression line appreciably. We say that this point has high leverage (it has the potential of affecting the regression line), but low influence (its actual effect is small). We will make this more precise later on.Plot (c) shows an outlier with an extreme y-value, but whose x-value is close to the mean of the dataset. We will see that such outliers have low leverage: their potential to affect the regression line is low. Lastly, plot(d) shows an outlier with a large influence on the regression line. While the y-value of the outlier is not that extreme, the x-value is, and the regression line is noticeably affected.

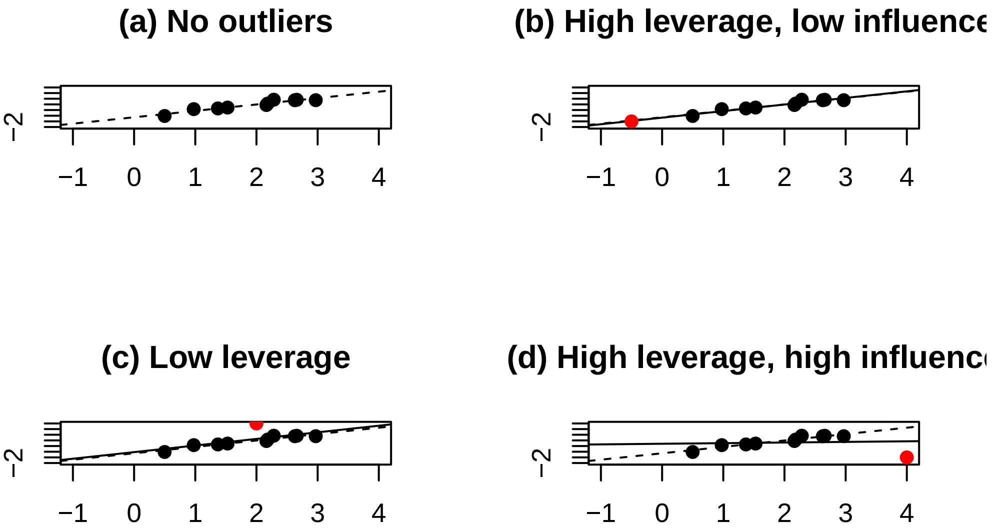


Figure 2.15: Influence of outliers on the regression line. Figure (a): original dataset and regression line. Figure (b): Outlier with high leverage but low influence on the regression line. Figure (c): Outlier with low leverage. Figure (d): Outlier with large leverage and large influence.

It is in general undesirable that a single observation has an outsized influence of the results of a linear regression analysis. Some diagnostics to track down extreme observations are therefore needed. *Residuals* indicate how much the outcome differs from the regression line and can thus be used to identify extreme outcomes. In particular, we already mentioned that residuals should be approximately normally distributed when the model is correct and the outcome is normally distributed (for fixed predictor values) with a homogeneous variance. Verifying whether or not a residual is extreme can then be done by comparing it to the normal distribution. Assume, for example, that we have a dataset with 100 observations. We then expect that approximately 95% of the residuals are, in absolute value, smaller than $1.96\hat{\sigma}$. Observing a lot more than 5% of extreme residuals will then give us an indication for outliers.

In the literature, a number of modifications of the residuals have been introduced to make them more suitable for outlier detection. After all, it is possible to show that, due to estimation errors, even when the model is correct and the outcomes are normally distributed (for fixed predictor values) with homogeneous variance, the residuals will not have a constant variance and are not perfectly normally distributed. *Studentized residuals* are a transformation of the previously defined residuals that do come with a constant variance and that are t-distributed with $n-1$ degrees of freedom under the assumptions of the model. Outliers can thus be detected more accurately by verifying if a lot more than 5% of the Studentized residuals are larger in absolute value than the 97.5% percentile of the t_{n-1}-distribution.

Extreme predictor values can in principle be detected using a scatterplot matrix of the outcome in function of the different predictors. However, when there are multiple predictors, these plots have serious shortcomings because it is possible that not the predictor values themselves, but a combination of the predictors is unusual, which may not be visible in these plots. It is therefore more sensible to investigate the so-called *leverage* (influence) of each observation. Leverage is a diagnostic measure for the influence of predictor observations (in contrast to residuals that give a diagnostic measure for the influence of the outcomes). In particular, the leverage of observation i is a measure for the distance of the predictor value of observation i to the mean predictor value in the sample. As a consequence, a large leverage for the i^{th} observation means that it has predictor values that strongly deviate from the mean. In this case, that observation might also have a large influence on the regression coefficients and the predictions. Leverage values normally vary between $1/n$ and 1 and are on average equal to p/n with p the number of unknown parameters. Typically, a leverage value is considered to be extreme if it is larger than $2p/n$.

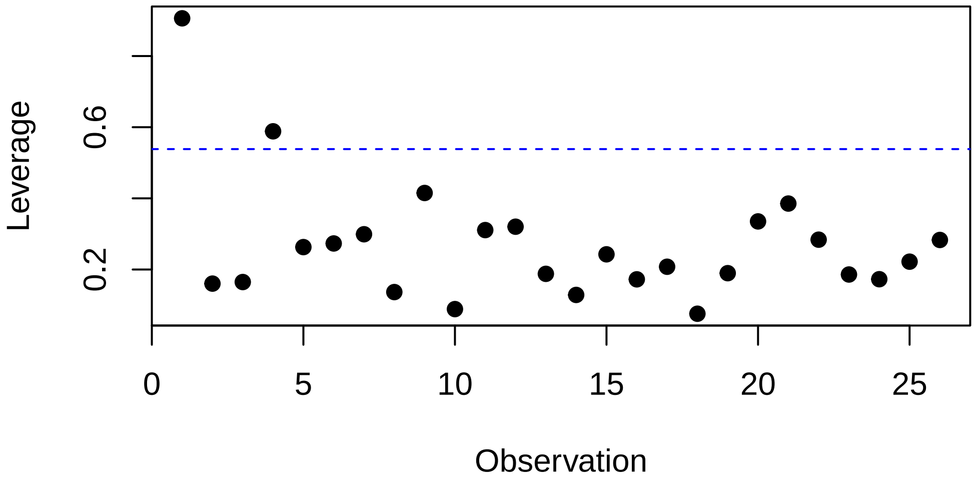


Figure 2.16: Leverage in function of observation number. The dashed line indicates the cut-off value of $2p/n$.

A more direct measure to express the influence of each observation on the regression analysis, is *Cook's distance*. The Cook's distance for the i^{th} observation is a diagnostic measure for the influence of that observation on all predictions, or equivalently for its influence on *all* estimated coefficients. It is obtained by comparing each prediction $\hat{Y}_j$, obtained based on the regression model for the j^{th} outcome, $j = 1, \dots, n$, with the corresponding prediction $\hat{Y}_{j(i)}$ that would have been obtained if observation i had not been used to fit the regression model:

$$D_i = \frac{\sum_{j=1}^{n} (\hat{Y}_j - \hat{Y}_{j(i)})^2}{p\text{MSE}}.$$

If the Cook's distance D_i is large, then observation i will have a large influence on the predictions and estimated coefficients. In particular, a Cook's distance is called extreme if it surpasses the 50% percentile of the $F_{p,n-p}$-distribution.

Mineral composition vs growth

The leverage of each observation in the regression analysis for this example is shown in Figure 2.16. It indicates that the first and fourth observation take on extreme predictor values and hence might have a large influence on the results of the analysis. The Cook's distance of the first observation is 1.5. Knowing that the model contains 7 parameters ($p = 7$) and 26 observations ($n = 26$), we conclude that this well exceeds the 50% percentile of the $F_{p,n-p}$-distribution, which is 0.94. Essentially, the value of 1.5 corresponds to the 77% percentile of that distribution. We thus conclude that the first observation has a large influence on the estimated regression coefficients. Figure 2.17 shows that the other observations have a far lesser and hardly influential impact.

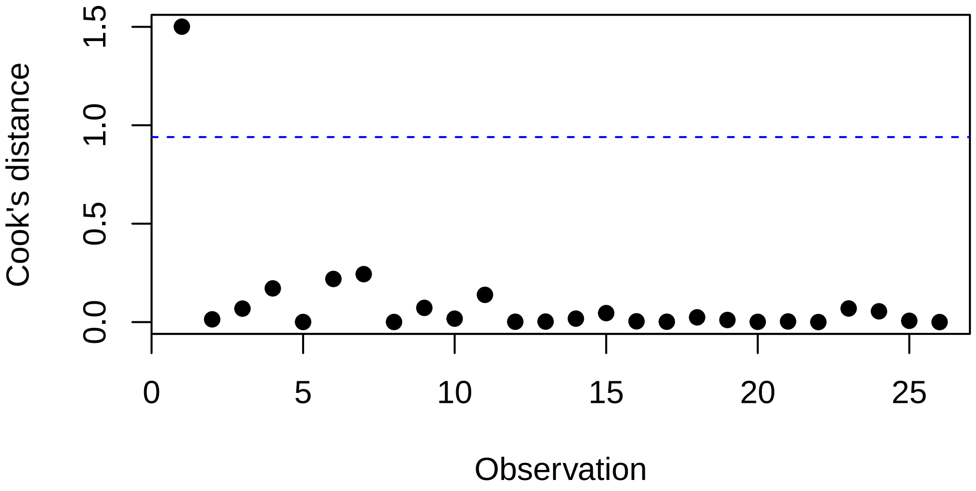


Figure 2.17: Cook's distance in function of observation number.

Once we observed that an observation is influential, the so-called *DFBETAs* can be used to determine on which regression coefficient(s) exactly it exercises its large influence. The DFBETAs of observation i are a diagnostic measure for the influence of that observation *on each regression coefficient separately*, contrary to Cook's distance which evaluates the influence on all coefficients simultaneously. In particular, the DFBETA for the i^{th} observation and j^{th} coefficient is obtained by comparing the j^{th} coefficient $\hat{\beta}_j$ with the coefficient $\hat{\beta}_{j(i)}$ from the regression model that is fitted without including the i^{th} observation in the analysis:

$$\mathrm{DFBETA}_{j(i)} = \frac{\hat{\beta}_j - \hat{\beta}_{j(i)}}{\mathrm{SD}(\hat{\beta}_j)}.$$

From this expression it follows that the sign of the DFBETA for observation i indicates whether omitting that observation from the analysis causes an increase (DFBETA$<$ 0) or decrease (DFBETA$>$ 0) in the corresponding coefficient. A DFBETA is called extreme if it exceeds 1 in small to medium-sized datasets or $2/\sqrt{n}$ in larger datasets.

Mineral composition vs growth

We concluded that the first observation is influential on the regression analysis for this example. The DFBETAs in Figure 2.18 show that it has a large influence on the interaction between phosphorus and residual ash. The corresponding coefficient observed in the regression analysis is -598 (SE 290). Using the expression of the DFBETA and the fact that its value is 2.16, we conclude that omitting the first observation will change the interaction between phosphorus and residual ash to approximately

$$-598 - 2.16 \times 290 = -1224.$$

Because of this, as well as the fact that the interaction between phosphorus and residual ash was only marginally significant, we decide to remove the interaction from the model. This leads to the output given below.

```
Call:
lm(formula = length ~ cnitrogen + cphosphorus + cpotassium +
    cresidu + cphosphorus:cnitrogen, data = needles_centered)

Residuals:
    Min      1Q  Median      3Q     Max
-48.540 -26.313   6.115  16.557  67.602

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)             185.20       9.52  19.454 1.83e-14 ***
cnitrogen                99.40      23.40   4.247 0.000395 ***
cphosphorus             229.46     162.44   1.413 0.173167
cpotassium              128.84      44.21   2.914 0.008574 **
cresidu                  23.51      36.09   0.651 0.522186
cnitrogen:cphosphorus   661.50     370.78   1.784 0.089595 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 36.05 on 20 degrees of freedom
Multiple R-squared:  0.886, Adjusted R-squared:  0.8575
F-statistic: 31.09 on 5 and 20 DF,  p-value: 8.924e-09
```

Figure 2.19 confirms that there is now no longer a single observation that has an outsize influence on the results of the obtained model.

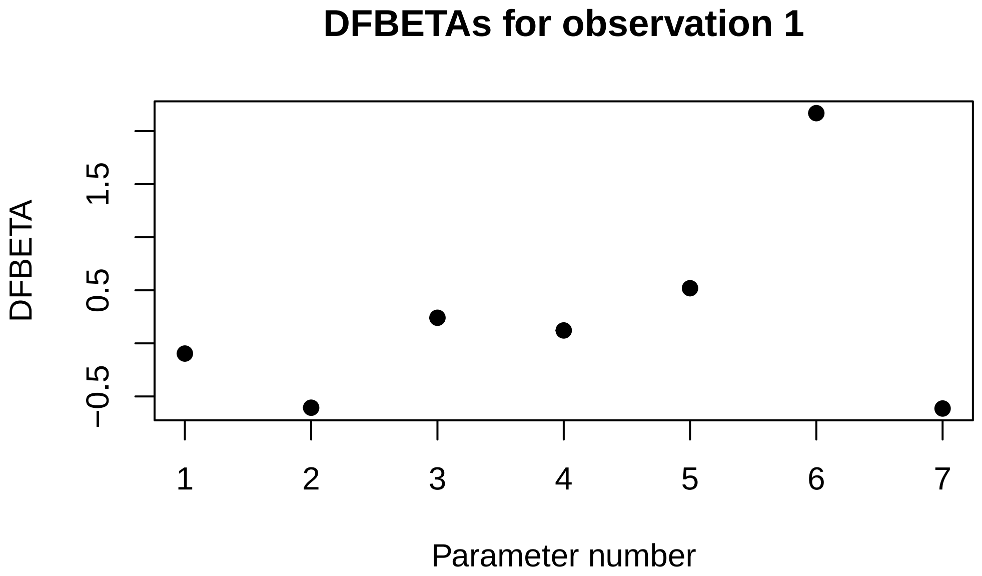


Figure 2.18: DFBETAs for the first observation in function of the coefficient number.

Please remark that now the influence of residual ash on tree length has become insignificant and that in a next step this parameter will have to be removed from the model.

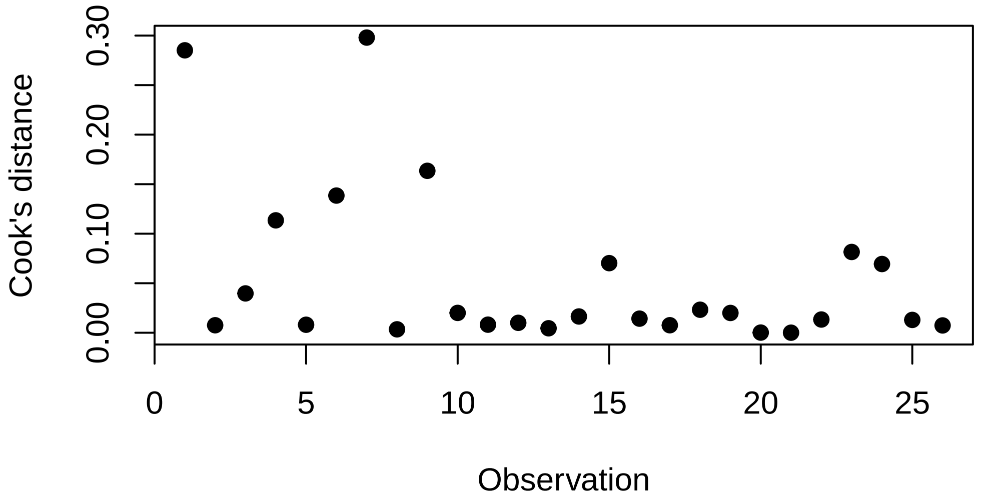


Figure 2.19: Cook's distance in function of observation number after removing the interaction between phosphorus and residual ash.

3 Principal component analysis

In the previous chapter we built a linear model to predict the amount of body fat, given measurements of the thickness of the triceps skin fold, the thigh circumference, and the midarm circumference. We saw that the dataset used for this model suffers from multicollinearity, meaning that some of the predictors (or linear combinations of predictors) are correlated with one another. Intuitively speaking, multicollinearity means that some variables don't contribute much to the expressivity of the data: we could omit them and end up with a dataset that is almost as informative as the original one.

To find out which combinations of variables contribute most to the variability of our data, we will turn to *principal component analysis*, one of the mainstays of statistical data analysis. Principal component analysis will allow us to identify the main sources of variability in our dataset, and will tell us which combinations of variables can be omitted with only little impact on the data itself. This allows us to reduce the number of features in the dataset, and makes principal component analysis into what is called a technique for *dimensionality reduction*. This is useful for a number of reasons:

- As a *pre-processing technique*: Many statistical techniques, such as multiple linear regression, do not perform well in the presence of highly correlated features. They either fail to converge outright, or they give unreliable results (for example, model coefficients and predictions that change drastically when the data is slightly perturbed). This is even more of an issue when there are more predictors than data points, a situation that often occurs when analysing gene expression data or spectroscopy data.
- To save on *computational processing time*: analyzing superfluous variables comes with a cost, which can often increase drastically with the number of features. We will see an example of this phenomenon in Section 4.2, where the data points are vectors with 4096 components. After principal component reduction, the dimensionality of the dataset can be reduced to 50-100 components, a reduction by more than 98%.
- To *visualize* the data: for datasets with a limited number of features, we can use a scatter matrix to view the distribution of the features and their relations with one another. Scatter matrices become uninformative, however, as soon as there are more than 4 or 5 features. Moreover, scatter matrices may hide correlations that occur between different linear combinations of variables, as we have seen in the chapter on linear modeling.

3.1 Intuition

Principal component analysis (PCA) finds a low-dimensional approximation to the dataset which retains as much as possible the variability of the original dataset. To understand what is meant by this, let's revisit the body fat dataset of chapter 1. In this dataset, the features are the measurements of the thickness of the triceps skin fold, the thigh circumference, and the midarm circumference. The total body fat is the outcome, but we will not consider it for the time being.

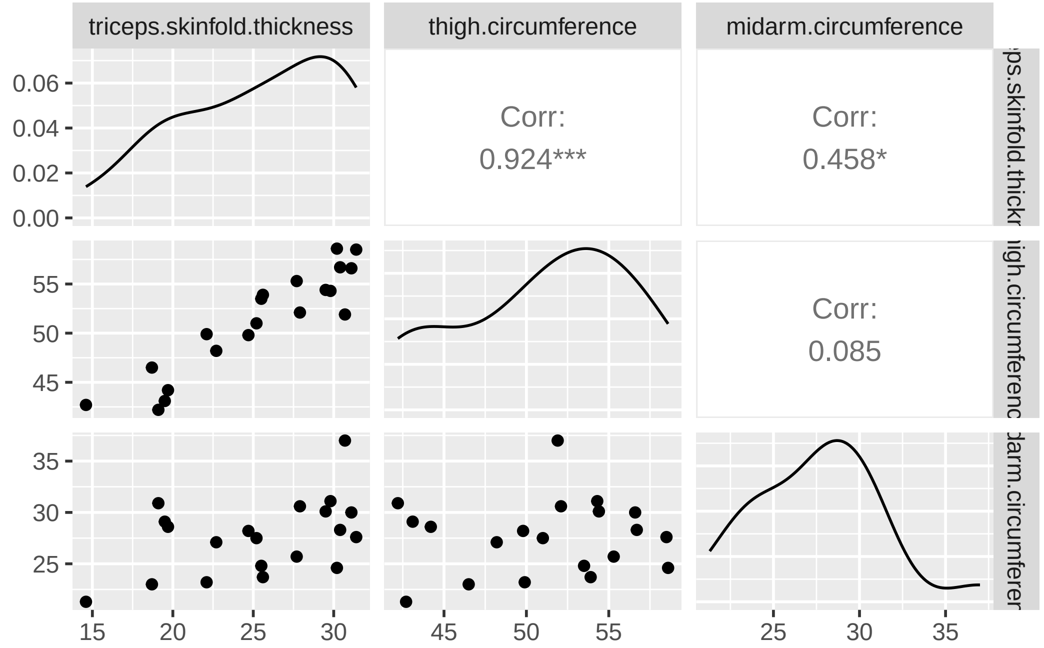
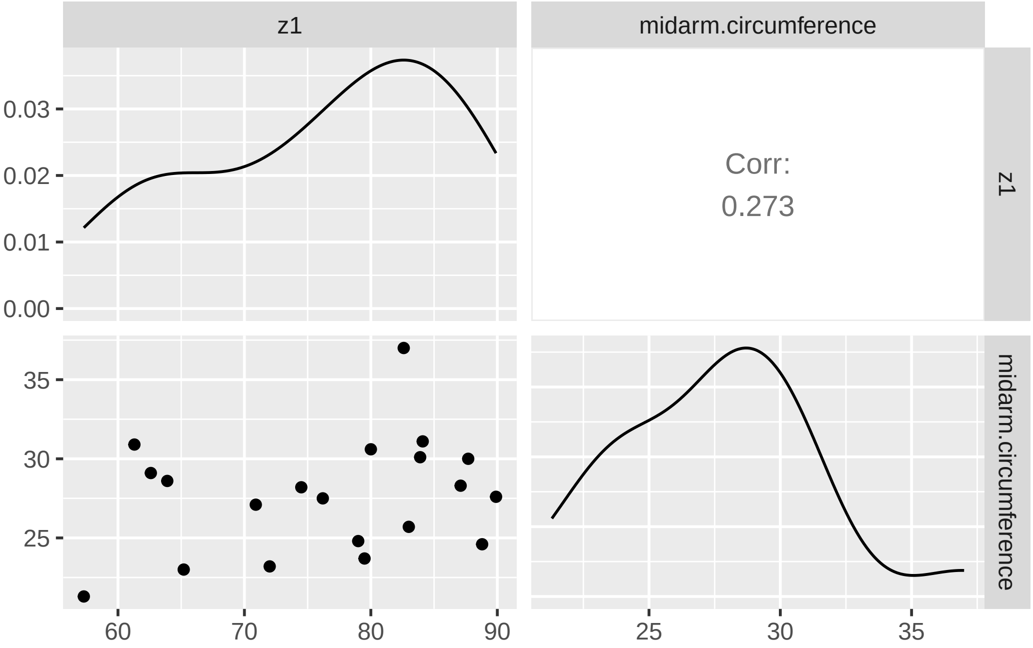


Figure 3.1: Correlations between the different variables in the body fat dataset. The variables `triceps.skinfold.thickness` and `thigh.circumference` are highly correlated.

From the scatter matrix, we see that `triceps.skinfold.thickness` and `thigh.circumference` are highly correlated: if you know one, you can predict the other one reasonably well. This makes it feel like a waste to analyze both: since they carry the same information, we can just as well throw one or the other away, or replace both by a linear combination. Let's do the latter, and introduce a new variable `z1` which is the sum of both. In terms of this variable and `midarm.circumference`, which we leave unchanged, the dataset has only two features that are mildly correlated, as shown on the scatter plot below.



We have succeeded in our aim to reduce the number of features in our dataset from 3 to 2, but a number of questions immediately pop up:

1. What is the meaning of the `z1` variable, and can we find it without looking at a scatter plot?
2. How much information do we lose by considering two variables instead of three? Will the conclusions from the reduced dataset still be valid for the full dataset?

It will turn out that the variable `z1`, which we constructed in an ad-hoc way, is remarkably close to the first principal component of the dataset, and we will discover a way to compute all principal components. We will also see that by discarding more or fewer principal components, as much variability of the original dataset can be retained as is needed.

3.2 Derivation of the PCA algorithm

PCA works by making linear combinations of the original variables so that the total amount of variation is maximal. There is another way of computing principal components, by minimizing the reconstruction error, and it can be shown that both approaches give the same principal components (Bishop (2006), section 12). In this course, we will develop the first method further.

We assume that we have N observations $\mathbf{x}_i$, where each $\mathbf{x}_i$ is a column vector in $\mathbb{R}^D$. We assemble these observations into an $N \times D$ *data matrix* $\mathbf{X}$, where each row of $\mathbf{X}$ is an observation

$\mathbf{x}_i$:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \cdots \\ \mathbf{x}_N^T \end{bmatrix}.$$

Keep in mind that the columns of $\mathbf{X}$ are the features (also referred to as independent variables or predictors) of the dataset. In the case of the body fat dataset, $\mathbf{X}$ is a 20×3 matrix, since there are 20 observations, with 3 features for each observation. We will refer to the jth column of $\mathbf{X}$ by the notation $\mathbf{X}_j$, where $j = 1, \dots, D$. Note that $\mathbf{X}_j$ is a column vector, with N entries, and there are D such columns. For the body fat dataset, the columns correspond to the following features:

- $\mathbf{X}_1$: `triceps.skinfold.thickness`
- $\mathbf{X}_2$: `thigh.circumference`
- $\mathbf{X}_3$: `midarm.circumference`

3.2.1 The first principal component

The first principal component, $\mathbf{Z}_1$ is a linear combination of the features, so that the amount of variation in $\mathbf{Z}_1$ is maximized. Let's unpack these ideas one at a time. The fact that $\mathbf{Z}_1$ is a linear combination means that it can be written as

$$\mathbf{Z}_1 = v_1 \mathbf{X}_1 + \cdots + v_D \mathbf{X}_D = \sum_{j=1}^{D} v_j \mathbf{X}_j.$$

where the v_j are coefficients that we have to determine. These coefficients are sometimes referred to as the principal component **loadings**, since v_j expresses how much of $\mathbf{X}_j$ is added ("loaded") to the principal component.

We can write this expression for $\mathbf{Z}_1$ in a compact way by assembling all the coefficients v_j into a vector $\mathbf{v}$, called the **loadings vector**:

$$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \cdots \\ v_D \end{bmatrix}.$$

With this expression, $\mathbf{Z}_1$ can be written as a matrix-vector product:

$$\mathbf{Z}_1 = \mathbf{X}\mathbf{v}. \tag{3.1}$$

We will often use this way of expressing $\mathbf{Z}_1$ as a matrix-vector product, since it makes subsequent calculations easier.

Before we go on to determine the loadings v_j, let's focus on the geometry behind Equation 3.1. Each component of $\mathbf{Z}_1$ can written as

$$(\mathbf{Z}_1)_i = \mathbf{x}_i^T \mathbf{v}.$$

This is the dot product of the ith observation $\mathbf{x}_i$ with the loadings vector $\mathbf{v}$. This dot product tells us how much of the vector $\mathbf{x}_i$ is parallel to $\mathbf{v}$, as shown in Figure 3.2. For example, a data point that is at right angles to $\mathbf{v}$ will have dot product 0 (no component at all along $\mathbf{v}$), while one that is parallel to $\mathbf{v}$ will have a dot product that is maximal in magnitude.

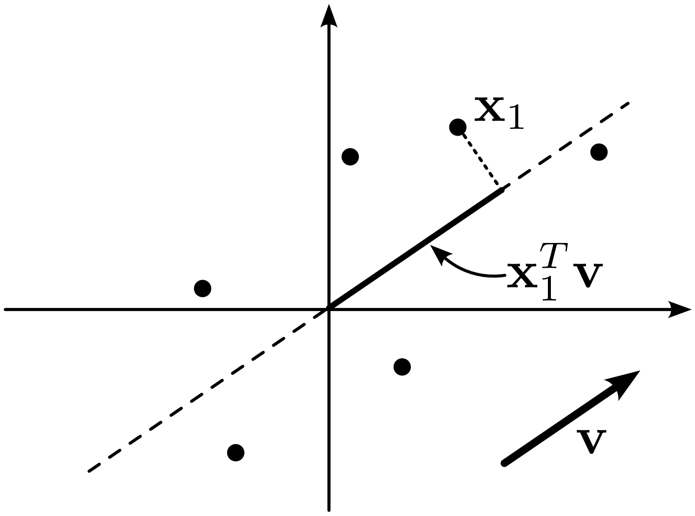


Figure 3.2: The ith component of the first principal component $\mathbf{Z}_1$ is the length of the orthogonal projection of $\mathbf{x}_i$ onto the line in the direction of $\mathbf{v}$.

In other words, we obtain $\mathbf{Z}_1$ by taking a fixed vector $\mathbf{v}$ and projecting all of our data points on the line through the origin in the direction of $\mathbf{v}$. If we choose another vector, $\mathbf{w}$, we obtain a different projection, as indicated on Figure 3.3.

Our goal is now to find the loadings vector $\mathbf{v}$ so that the variance of the projected dataset is maximal. To make this problem well-posed, we will assume that $\mathbf{v}$ has unit norm:

$$\mathbf{v}^T\mathbf{v} = 1. \tag{3.2}$$

If we did not impose this constraint, we could increase the amount of variance simply by making $\mathbf{v}$ longer.

The mean of the projected data is given by

$$\bar{\mathbf{Z}}_1 = \frac{1}{N}\sum_{j=1}^{N}(\mathbf{Z}_1)_j = \frac{1}{N}\sum_{j=1}^{N}\mathbf{x}_j^T\mathbf{v} = \bar{\mathbf{x}}^T\mathbf{v},$$

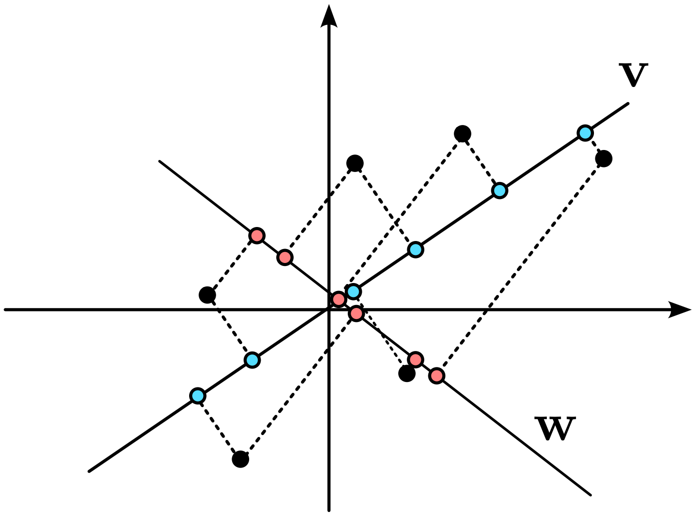


Figure 3.3: Two different projections, onto the loadings vector $\mathbf{v}$ (blue) and $\mathbf{w}$ (red).

where $\bar{\mathbf{x}}$ is the (sample) mean of the original data points. In other words, the mean $\bar{\mathbf{Z}}_1$ is just the mean $\bar{\mathbf{x}}$ of the data points, projected onto $\mathbf{v}$.

The variance of the projected data is given by

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}\left((\mathbf{Z}_1)_i - \bar{\mathbf{Z}}_1\right)^2 = \frac{1}{N}\sum_{i=1}^{N}\left(\mathbf{x}_i^T\mathbf{v} - \bar{\mathbf{x}}^T\mathbf{v}\right)^2.$$

This expression can be rewritten as a matrix product:

$$\sigma^2 = \mathbf{v}^T\mathbf{S}\mathbf{v}, \tag{3.3}$$

where $\mathbf{S}$ is the covariance matrix, given by

$$\mathbf{S} = \frac{1}{N}\sum_{i=1}^{N}\left(\mathbf{x}_i\mathbf{x}_i^T - \bar{\mathbf{x}}\bar{\mathbf{x}}^T\right). \tag{3.4}$$

We are now ready to translate our problem into a mathematical form, so that we can solve it. To find the first principal component $\mathbf{Z}_1$, we want to find a loadings vector $\mathbf{v}$ so that the projected variance σ^2, given in Equation 3.3, is maximized. In addition, we want $\mathbf{v}$ to have unit length, as in Equation 3.2. In mathematical terms:

$$\mathbf{v} = \operatorname{argmax} \mathbf{v}^T\mathbf{S}\mathbf{v}, \quad \text{such that } \mathbf{v}^T\mathbf{v} = 1.$$

We can solve this optimization problem using the theory of Lagrange multipliers. If we introduce a Lagrange multiplier λ for the unit-length constraint, then the desired vector $\mathbf{v}$ is given by

$$\mathbf{v} = \operatorname{argmax} L(\mathbf{v})$$

where L is given by

$$L(\mathbf{v}) = \mathbf{v}^T\mathbf{S}\mathbf{v} - \lambda(\mathbf{v}^T\mathbf{v} - 1).$$

A necessary condition for $\mathbf{v}$ to be a maximum of L is that the gradient vanishes at $\mathbf{v}$. Taking the gradient of L with respect to $\mathbf{v}$ and setting the resulting expression equal to zero gives

$$\mathbf{S}\mathbf{v} = \lambda\mathbf{v}. \tag{3.5}$$

This is a very important result: it tells us that the $\mathbf{v}$ we are looking for is an **eigenvector** of the matrix $\mathbf{S}$, with corresponding **eigenvalue** λ. This will hold true generally, not just for the first principal component: finding the principal components of a data set will involve solving an eigenvalue problem, and selecting the largest eigenvalues.

Last, we have to find the Lagrange multiplier λ. This can be done by multiplying Equation 3.5 from the left by $\mathbf{v}^T$ to get

$$\mathbf{v}^T\mathbf{S}\mathbf{v} = \lambda\mathbf{v}^T\mathbf{v} = \lambda,$$

where we have used the unit-length constraint Equation 3.2.

We see that the Lagrange multiplier λ is precisely the variance σ^2 of the first principal component $\mathbf{Z}_1$. For this reason, we will refer to the eigenvalue λ as the amount of *retained variance*, since it expresses how much variance is captured by projecting the entire dataset onto the direction $\mathbf{v}$.

To sum up, the first principal component $\mathbf{Z}_1$ is a linear combination of the original features (columns) of our dataset, chosen so that the variance of $\mathbf{Z}_1$ is maximal. We can find $\mathbf{Z}_1$ by looking for the largest eigenvalue λ of the covariance matrix, with unit length eigenvector $\mathbf{v}$, and projecting the data matrix $\mathbf{X}$ onto $\mathbf{v}$.

3.2.2 The remaining principal components

Now that we've computed the first principal component, how do we compute the others? It probably won't come as a surprise that the next principal components, $\mathbf{Z}_2$, $\mathbf{Z}_3$, and so on, involve the amount of variation that is left in the data after $\mathbf{Z}_1$ has been removed, and that they involve the second, third, ... largest eigenvalues.

Assuming that we have computed $\mathbf{Z}_1$ as in the previous section, and denote the loadings vector by $\mathbf{v}_1$. Recall that $\mathbf{v}_1$ points in the direction of the largest variance.

To find the next principal component, we consider the variability in the dataset that is not already accounted for by $\mathbf{Z}_1$. More precisely, we look for a loadings vector $\mathbf{v}_2$ which is orthogonal to $\mathbf{v}_1$, has unit length, and maximizes the amount of variability $\mathbf{v}_2^T\mathbf{S}\mathbf{v}_2$. By a similar reasoning as in the previous section, one can show that this $\mathbf{v}_2$ is an eigenvector associated with the second largest eigenvalue of $\mathbf{S}$. The projection of the dataset onto this loadings vector then gives us the second principal component:

$$\mathbf{Z}_2 = \mathbf{X}\mathbf{v}_2.$$

This procedure can be applied to find all D principal components and results in the following algorithm to compute the principal components:

1. Compute the sample covariance matrix $\mathbf{S}$ using Equation 3.4.
2. Compute the eigenvalues of $\mathbf{S}$ and order them from largest to smallest: $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_D$.
3. Find the corresponding eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_D$ and normalize them to unit length, if necessary. These vectors are the loading vectors and they point in the directions of highest variance.
4. Project the dataset onto the loading vectors to obtain the principal components $\mathbf{Z}_1, \mathbf{Z}_2, \ldots, \mathbf{Z}_D$.

Typically, we do not have to compute the principal components by hand: most data analysis packages will do this for us, either via a builtin command, such as in R (prcomp) or minitab, or via an extra package, such as scikit-learn (Python) or xlstat (Excel). It is instructive, however, to know the principles behind PCA, so that you can interpret and understand the results.

> ℹ Note
>
> Software packages that compute the principal components of a dataset typically do not compute the eigenvalues and eigenvectors of the covariance matrix, despite our derivation above. Instead, they rely on the so-called *singular value decomposition* (SVD) of the data matrix (after centering). The SVD is typically more accurate and easier to compute, and the principal components obtained in this way agree with the ones computed using the eigendecomposition (to within numerical roundoff).

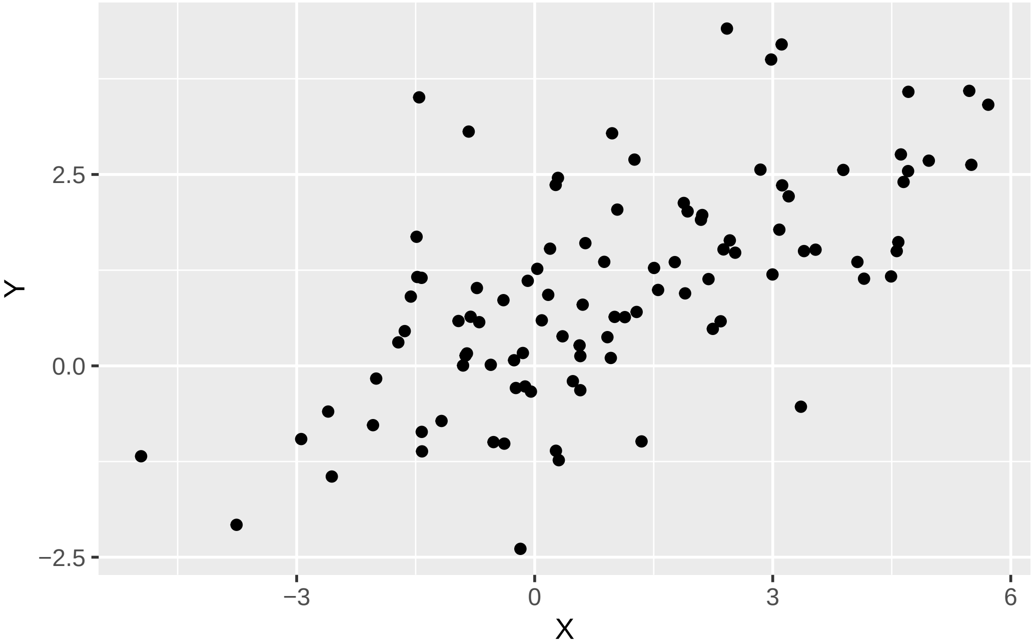
3.2.3 Worked-out example

For this example, we will calculate the principal components in two different ways. We will first compute the principal components by hand, by solving an eigenvalue problem. This is possible because the data are two-dimensional, and solving the characteristic equation for a two- or three-dimensional matrix can be done by hand. This is not practical for real-world datasets, which often contains dozens, thousands, or millions of features, and we will therefore also cover computing the principal components using R.

Our dataset consists of 100 observations, where each observation has two components. The dataset is shown below and has been carefully constructed so that the covariance matrix $\mathbf{S}$

and mean $\bar{\mathbf{x}}$ are exactly equal to

$$\mathbf{S} = \begin{bmatrix} 5 & 2 \\ 2 & 2 \end{bmatrix}, \quad \text{and} \quad \bar{\mathbf{x}} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}. \tag{3.6}$$



To find the loading vectors, we must find the eigenvalues of **S**, which we can do via the characteristic equation:

$$\det(\mathbf{S} - \lambda\mathbf{I}) = 0.$$

Substituting the expression given in Equation 3.6 for the covariance matrix and expanding the determinant gives

$$\det \begin{bmatrix} 5-\lambda & 2 \\ 2 & 2-\lambda \end{bmatrix} = (5-\lambda)(2-\lambda) - 4 = 0.$$

The roots of this equation are $\lambda_1 = 6$ and $\lambda_2 = 1$. The corresponding eigenvectors are

$$\mathbf{v}_1 = \frac{1}{\sqrt{5}} \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \quad \mathbf{v}_2 = \frac{1}{\sqrt{5}} \begin{bmatrix} -1 \\ 2 \end{bmatrix}.$$

These are our loading vectors, and they indicate the direction in which the data varies the most (for $\mathbf{v}_1$) and the "second-most" (for $\mathbf{v}_2$). Figure Figure 3.4 shows the dataset again, now with the two loading vectors superimposed. Each loading vector has been rescaled by multiplying it with the square root of the corresponding eigenvalue. Why the square root? The eigenvalue itself represents the *variance* in that direction, the square root the standard deviation.

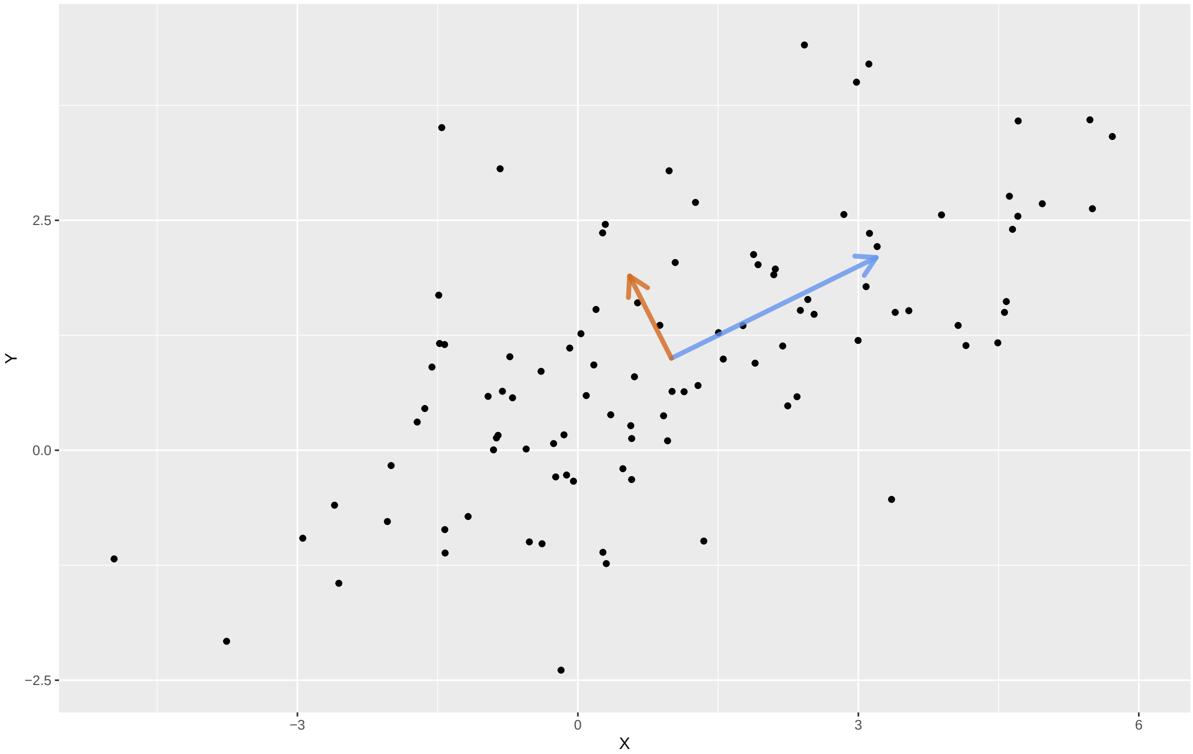


Figure 3.4: The dataset with the first loading vector (blue) and the second loading vector (orange) superimposed. Each loading vector has been rescaled by the square root of the corresponding eigenvalue, to give an indication of the variability in that direction.

To compute the principal components directly via R, we can use the prcomp command, as shown below. This is generally the preferred option over computing the eigenvalue decomposition by hand: prcomp is more flexible (allowing one, for example, to scale the data before computing the principal components) and is also numerically more stable.[1] For our simple dataset, the end result is the same:

```r
pca <- prcomp(df)
pca
```

```
Standard deviations (1, .., p=2):
[1] 2.44949 1.00000

Rotation (n x k) = (2 x 2):
        PC1        PC2
X 0.8944272 -0.4472136
Y 0.4472136  0.8944272
```

> ⚠ Warning
>
> Note that prcomp returns (among other things) the standard deviations, which are the square roots of the variances (eigenvalues). To compare the output of prcomp with the results of the eigenvalue analysis, **make sure to take the square of the standard deviations**, and you will see the eigenvalues appear:
>
> ```r
> pca$sdev^2
> ```
>
> ```
> [1] 6 1
> ```

3.2.4 Standardizing the predictors

Prior to doing principal component analysis, the data are often standardized by subtracting the mean for each feature and dividing by the standard deviation. If the original predictors in our dataset are given by $\mathbf{X}_i$, $i = 1, \ldots, D$, this means that we introduce standardized variables $\mathbf{Y}_i$, given by

$$\mathbf{Y}_i = \frac{\mathbf{X}_i - \bar{\mathbf{X}}_i}{\sqrt{S_i^2}},$$

[1] R has two commands to compute the principal components: prcomp and princomp. The former computes principal components using the so-called singular value decomposition (SVD) and is preferred for numerical stability. The latter, princomp, uses the eigenvalue decomposition as is provided for backwards compatibility with SAS.

where S_i^2 is the variance of $\mathbf{X}_i$. The resulting variables $\mathbf{Y}_i$ will have mean 0 and variance 1.

Standardizing the predictors means that they will be comparable in magnitude: variables whose variance is small will gain in importance and large variables will decrease in importance, roughly speaking. This may change the PCA output significantly!

As a rule of thumb, you should standardize variables that are measured in different units (e.g. seconds, meters, Watt, …), since the unit can be rescaled without affecting the physical meaning of the variable, or its relation to other variables (e.g., rescaling a variable expressed in meters by a factor of 1000 is the same as expressing that variable in kilometers). By contrast, variables that are measured in the same units or that are unitless should not be rescaled (or should be rescaled collectively).

For an example of the latter, think about pixel intensities in an image dataset (such a dataset is in fact analyzed in Section 4.2). Pixels near the edge of the image presumably are part of the background, don't vary that much, and are relatively unimportant, whereas pixels in the center are likely to be part of the image subject, vary a lot, and carry a lot of information. Standardizing the pixels would make the pixels near the edge just as variable as the pixels in the center, which would greatly amplify the noise in the image at the expense of the useful information in it!

Standardizing the predictors is also referred to as *scaling*.

3.3 Interpreting the PCA results

In this section we will discuss a number of useful results that follow from PCA. We will use the bodyfat dataset as an illustration throughout.

```r
pc <- prcomp(bodyfat_predictors)
pc
```

```
Standard deviations (1, .., p=3):
[1] 7.2046011 3.7432587 0.1330841

Rotation (n x k) = (3 x 3):
                                PC1        PC2        PC3
triceps.skinfold.thickness 0.6926671  0.1511979  0.7052315
thigh.circumference        0.6985058 -0.3842734 -0.6036751
midarm.circumference       0.1797272  0.9107542 -0.3717862
```

We can also ask R to print a summary of the principal component analysis for us. This will give us a table with the proportion of variance explained by each principal component, as well as the cumulative proportion (the amount of variance retained by that principal component and all previous ones).

```
summary(pc)
```

```
Importance of components:
                          PC1    PC2     PC3
Standard deviation     7.2046 3.7433 0.13308
Proportion of Variance 0.7872 0.2125 0.00027
Cumulative Proportion  0.7872 0.9997 1.00000
```

3.3.1 The score plot

Perhaps the most useful visualization of the PCA is the so-called **score plot**, which is nothing but a scatter plot of the first two principal components. It often happens that the score plot is sufficient to discern patterns in the data, such as clusters.

Figure 3.5 shows a score plot for the bodyfat dataset. While no obvious patterns in this dataset stand out, the plot does show that the principal components are uncorrelated, and this is a good confirmation of what we already know on theoretical grounds. It is customary to put the percentages of variance explained on the axis labels of the score plot, so that the person interpreting it can have an idea of how well the first two principal components describe the data.

As an aside, above we noted that the principal components are uncorrelated with one another. We can also verify that this is the case numerically. The result is not exactly zero, but it is very small:

```
cov(pc$x[,1], pc$x[,2])
```

```
[1] 3.238883e-15
```

3.3.2 The loadings plot

A **loadings plot** shows the relations of the original variables and the principal components. This is often a useful visualization to see how each variable contributes to the principal components. Moreover, one can show that the loadings are proportional to the Pearson correlations between the principal components and the variables. Hence, if a loading is positive (negative), that variable will be positively (negatively) correlated with that principal component.[2]

[2] There is a closely related plot type, the profile plot, which differs from the loadings plot in that it has the Pearson correlations on the y-axis. Otherwise the two plot types are identical.

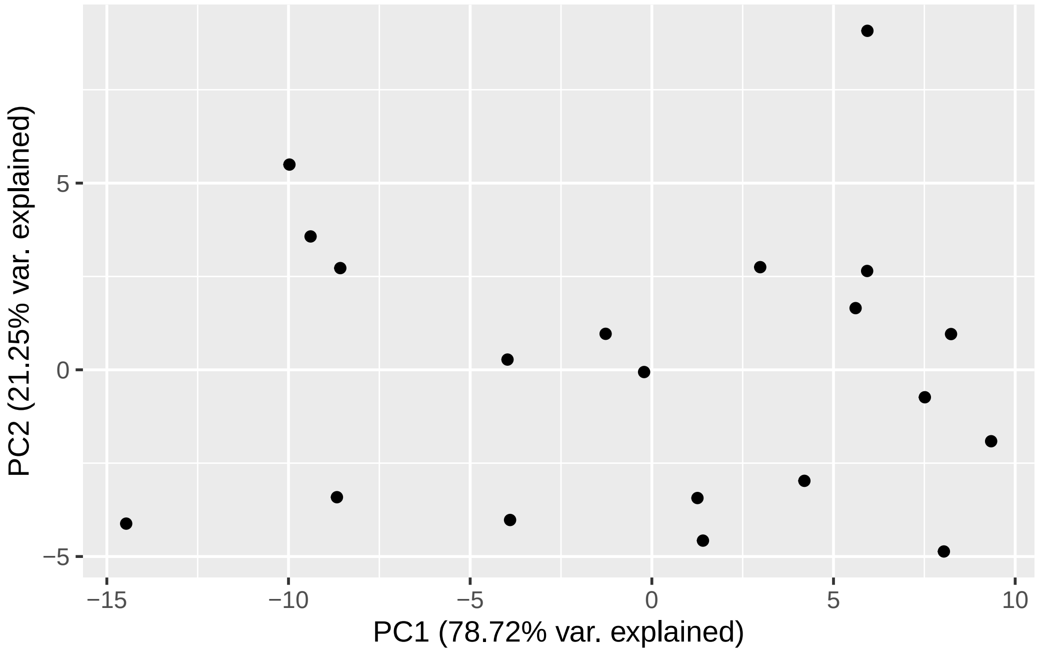


Figure 3.5: Score plot for the body fat dataset.

At the beginning of this chapter we hypothesized that the variables `thigh.circumference` and `triceps.skinfold.thickness` would contribute about equally to the first principal component. From Figure 3.6 we see that this is indeed the case: both variables have loadings approximately equal to 0.7, when we consider the first principal component. We also see that the second principal component is mostly made up of the variable `midarm.circumference`.

Unfortunately there is no command in base R or ggplot to create a loadings plot – you have to make one yourself.

3.3.3 The number of principal components

We now know how to calculate the D principal components for a given D-dimensional dataset, and we've seen that the principal components correspond to the directions in which the dataset varies the most. The real power of PCA, and the reason why it is so ubiquitous in data analysis, is that we can now selectively discard principal components that are not informative. By doing this, we obtain a dataset with fewer features, which is hence easier to analyze, and where the discarded features do not contribute too much to the expressivity of the data. This is what makes PCA into a *dimensionality reduction* method.

The question remains what principal components to discard. There are no universally accepted rules for this, but there are a couple of rules of thumb that can help us make an informed

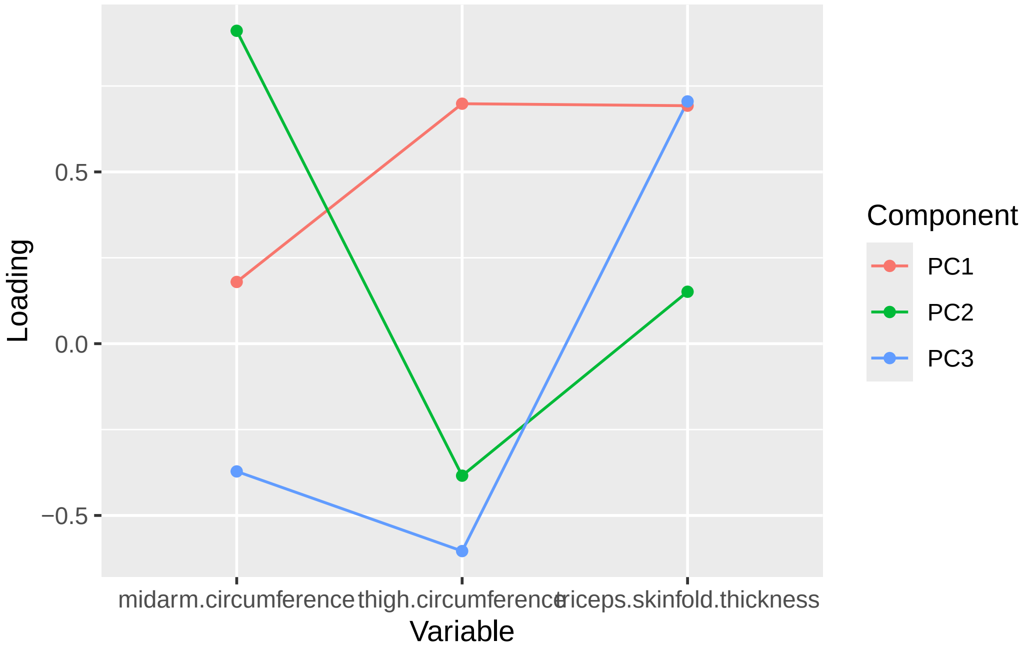


Figure 3.6: The loadings plot shows how the original variables are related to the principal components.

choice. Most of these rules take into account the total amount of variance retained by the first K principal components, defined as

$$S_K = \frac{\sum_{i=1}^{K} \lambda_i}{\sum_{j=1}^{D} \lambda_j},$$

Recall that the total amount of variance can be computed directly within R by using the `summary` command (and look for the "Cumulative Proportion" row):

```
summary(pc)
```

```
Importance of components:
                          PC1    PC2     PC3
Standard deviation     7.2046 3.7433 0.13308
Proportion of Variance 0.7872 0.2125 0.00027
Cumulative Proportion  0.7872 0.9997 1.00000
```

The number K of principal components can be chosen so that a fixed amount of variance (for example, 95%) is retained. To see this idea in action, let's apply it to the body fat dataset. The relative amount of variance explained by each principal component can then be calculated as follows:

```r
var_explained <- pc$sdev^2 / sum(pc$sdev^2)
var_explained
```

```
[1] 0.787222422 0.212508963 0.000268615
```

and the total amount of variance explained cumulatively by

```r
total_var <- cumsum(var_explained)
total_var
```

```
[1] 0.7872224 0.9997314 1.0000000
```

Note that these numbers agree with the output of the `summary` command. For what follows, it will be easier to have access to these quantities as straightforward R vectors.

We see that the first two principal components explain 78.7% and 21.3% of the total variance, respectively, and together they explain more than 99.97% of variance in the dataset. It therefore seems reasonable to discard the last principal component, which contributes less than 0.03% of variance.

For datasets with many features, a **scree plot** or **elbow plot** can be helpful to identify high-variance principal components. In a scree plot, the amounts of variance are plotted in descending order, so that one can identify at a glance the amount of variability contributed by each principal component. Some scree plots also include the total amount of variability, S_K, as a function of K.

R can make a pretty basic scree plot for you, via the screeplot command.

```r
screeplot(pc, main = "Principal components", type = "lines")
```

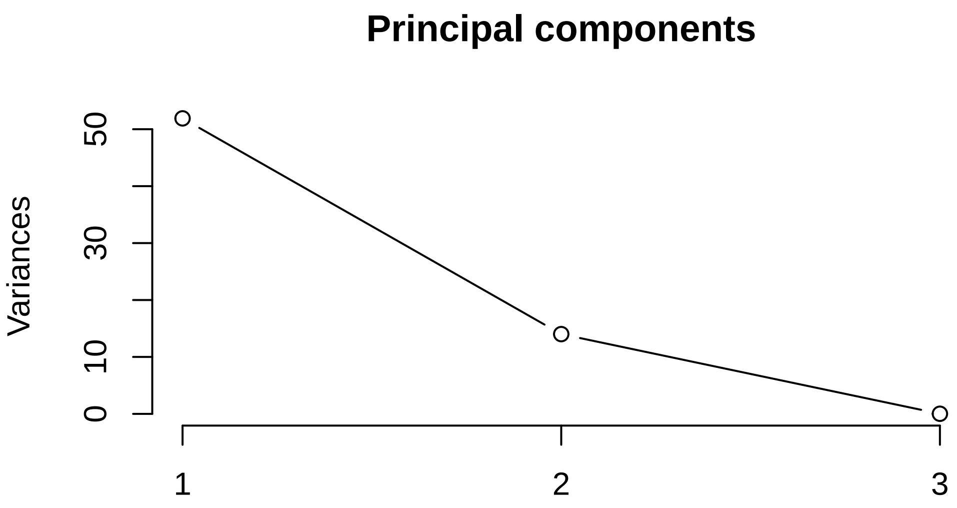


Figure 3.7: A scree plot made with the base R `screeplot` command.

With a bit more work, you can build your own scree plot, which is often preferable if you want to customize the plot, to include for example the cumulative variance, as in figure Figure 3.8.

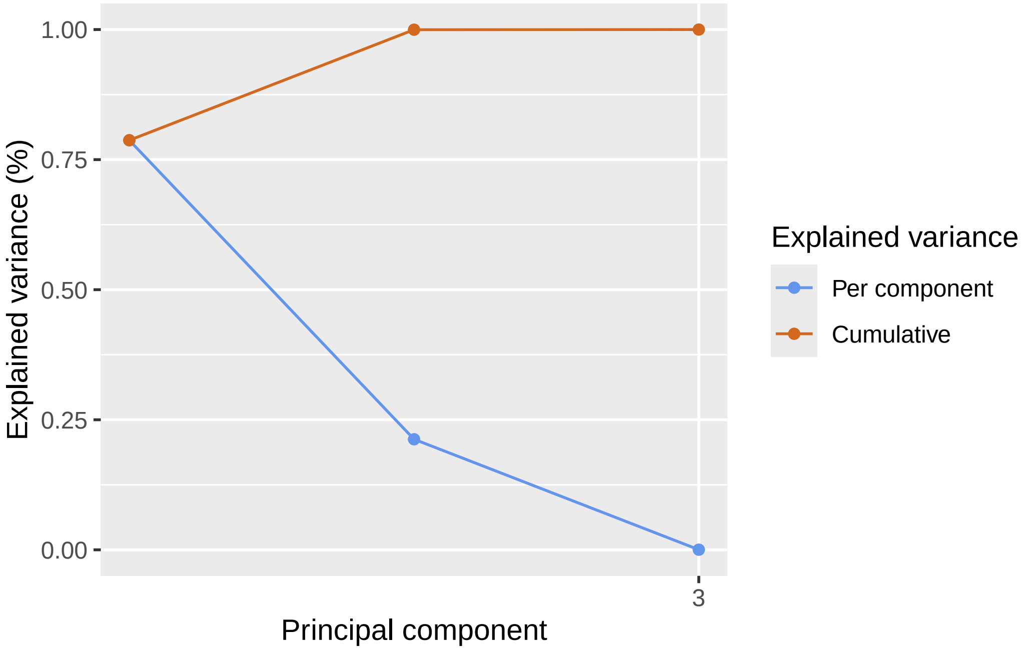


Figure 3.8: A scree plot for the body fat dataset confirms that the first two principal components explain almost all of the variance in the data.

Scree plots are often also referred to as “elbow plots”, since the plot typically (but not always) shows an “elbow” where the explained variance levels off. However, spotting the exact location of the elbow is very subjective, and it is typically more appropriate to take the point where the remaining principal components only contribute some small percentage of variation, for example 5%.

3.3.4 Biplots

The **biplot** consists of a loadings plot overlaid on a score plot. By default, it shows the first two principal components as a scatter plot, together with a set of vectors, one for each original variable, showing the contribution of that variable to the first two principal components. Biplots are relatively complex, but it is worth understanding what they encode.

```
biplot(pc)
```

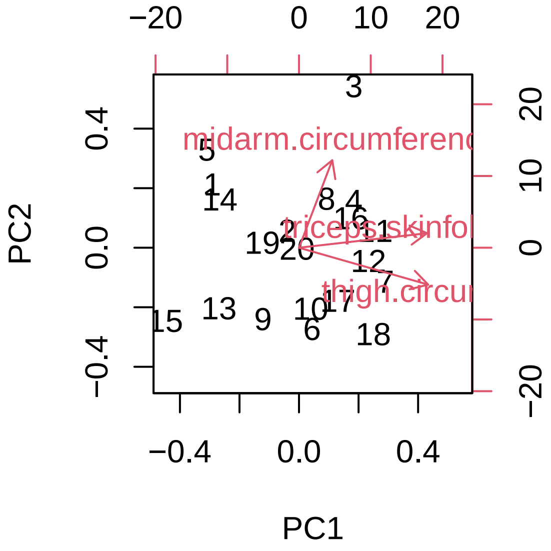


Figure 3.9: The biplot provides information about the transformed data and the loadings.

The numbers on the plot represent the data points from the original dataset, expressed relative to the first two principal components. This is the part of the biplot that is like a score plot. The red arrows, on the other hand, represent the original variables in the dataset (as shown by the labels attached to them) and are expressed on the top and right-most axis, which show how much each variable contributes to the first two principal components. The red arrows carry the same information as a loadings plot (in a different form), when you consider only the first two principal component.

At one glance, we see that `triceps.skinfold` and `thigh.circumference` are the most important contributors to the first principal component, and that the second principal component is almost entirely made up by `midarm.circumference`. This confirms our intuition from the beginning of this chapter, as well as the conclusions that we drew from the loadings plot.

3.4 Principal component regression

Once we have performed PCA, we can build a linear model using the principal components that we have chosen to retain. This is known as *principal component regression*. Let's see this in action for the bodyfat dataset. We start from a model where we include the first principal component only:

```r
pc1 <- pc$x[, "PC1"]
outcome <- bodyfat$bodyfat

model_1 <- lm(outcome ~ pc1)
summary(model_1)
```

```

Call:
lm(formula = outcome ~ pc1)

Residuals:
    Min      1Q  Median      3Q     Max 
-5.1357 -1.8821  0.2682  1.7107  3.4992 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 20.19500    0.58688  34.411  < 2e-16 ***
pc1          0.61366    0.08358   7.343 8.13e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.625 on 18 degrees of freedom
Multiple R-squared:  0.7497,	Adjusted R-squared:  0.7358 
F-statistic: 53.91 on 1 and 18 DF,  p-value: 8.128e-07
```

Adding the second principal component results in a bigger model, but the coefficient of the second principal component is not significant. We therefore do not include it in our final model and we stay with `model_1`.

```
pc2 <- pc$x[, "PC2"]
model_12 <- lm(outcome ~ pc1 + pc2)
summary(model_12)
```

```

Call:
lm(formula = outcome ~ pc1 + pc2)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.9876 -1.8822  0.2562  1.3209  4.0285 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 20.19500    0.56604  35.678  < 2e-16 ***
pc1          0.61366    0.08061   7.613 7.12e-07 ***
pc2         -0.23785    0.15514  -1.533    0.144    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.531 on 17 degrees of freedom
Multiple R-squared:  0.7801,	Adjusted R-squared:  0.7542 
F-statistic: 30.15 on 2 and 17 DF,  p-value: 2.564e-06
```

In this way, we obtain a simple least squares model, with one predictor and one outcome. Quite a simplification from our original linear model, which had three predictors and suffered from multi-collinearity. This reduction from 3 to 1 predictors may not seem like a big deal, but in the next chapter we will see examples of how PCA can be used to reduce datasets with many hundreds of variables to just 2-3 components, while maintaining the essential information in the dataset.

In this section we have built the linear model "by hand", but in reality you would probably not want to do this, for two reasons:

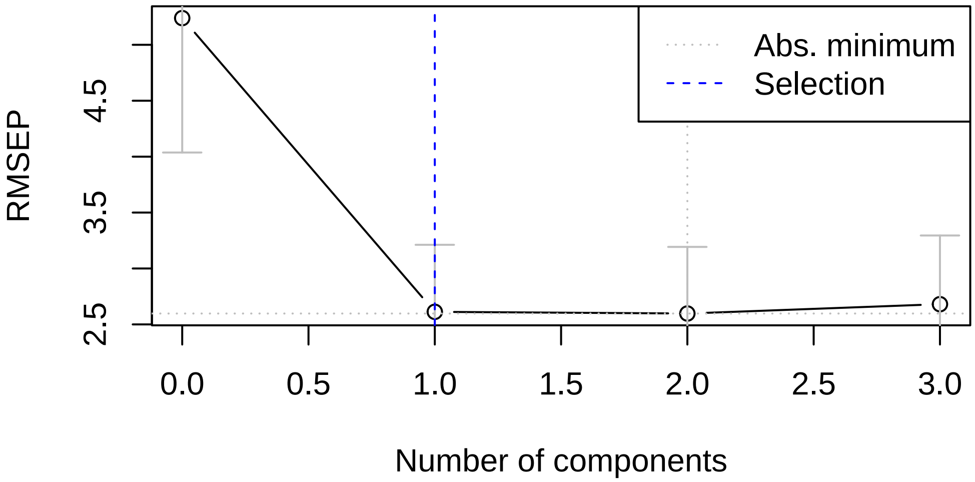
1. Testing each predictor at a time when there are many hundreds of candidates to choose from is impractical.
2. Using the forward model building procedure, as we did above, comes with significant multiple-testing issues.

Luckily, there are R packages that will do the PCA, select the optimal number of components, and build the regression model all for us. Here we use the `pls` package. We tell it to consider all principal components, and to assess the error through cross-validation. The details of this will be discussed in class, but the important line in the output is labeled RMSEP. This shows a measure of error in function of the number of components taken into account. We see that this error is the lowest when only the first principal component is considered.

```r
library(pls)

pcr_model <- pcr(bodyfat ~ ., data = bodyfat, validation = "CV")
summary(pcr_model)
```

```
Data:   X dimension: 20 3
    Y dimension: 20 1
Fit method: svdpc
Number of components considered: 3

VALIDATION: RMSEP
Cross-validated using 10 random segments.
       (Intercept)  1 comps  2 comps  3 comps
CV           5.239    2.612    2.597    2.681
adjCV        5.239    2.602    2.583    2.656

TRAINING: % variance explained
         1 comps  2 comps  3 comps
X          78.72    99.97   100.00
bodyfat    74.97    78.01    80.14
```

There are various rules to select the optimal number of components to consider. One such rule is the “1-sigma rule” (Hastie, Tibshirani, and Friedman 2009) which considers the model with the least number of components whose cross-validation error is at most one standard deviation away from the optimal model. In our case, this again results in a model with only one predictor.


RMSEP
2.5
3.5
4.5
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Number of components
Abs. minimum
Selection

```
[1] 1
```

4 Applications of principal component analysis

Principal component analysis is often a necessary first step when there are a large number of independent variables that need to be analyze simultaneously. Many devices in a modern lab produce this kind of high-dimensional data: for example, a reading for a single sample obtained via gas chromatography-mass spectrometry (GC-MS) or hyperspectral imagining (HSI) is a vector with 100s of entries, and with the number of samples often running in the 100s as well, we need a technique like PCA to find the needle in the haystack.

In this chapter we consider a number of real-world examples, from the life sciences and beyond, where PCA and dimensionality reduction prove to be essential.

> ⚠ Warning
>
> Given that the examples in this chapter deal with real-world data, which is often large and messy, the R code is at times a bit more complex than in the previous chapter. The underlying principles remain the same, however.

4.1 Adulteration of olive oil

This case study is a simplified version of an analysis done by the group of Prof. Van Haute of the Centre for Food Chemistry and Technology at GUGC to determine the adulteration of extra-virgin olive oil through the use of hyperspectral imaging data. The full analysis can be found in (Malavi et al. 2023). My thanks go to Prof. Van Haute for making the data available and for in-depth discussions regarding the analysis.

4.1.1 Problem statement

Extra-virgin olive oil (EVOO) is a type of oil that is made by cold-pressing the fruit of the olive tree without the use of chemicals or heat. It is considered the highest quality and most flavorful type of olive oil and is widely used in cooking and as a dressing for salads and other dishes. Extra-virgin olive is packed with antioxidants and healthy fats, making it a popular choice among health-conscious consumers. Due to its high quality and health benefits, extra-virgin olive oil is often more expensive than other types of olive oil.

Extra-virgin olive oil is sometimes adulterated, either accidentally, or deliberately, by the addition of other, cheaper vegetable oils. This is misleading to the customer and can pose health risks, for example through the introduction of allergens. As a result, manufacturers and food safety agencies have an interest in determining whether a given EVOO sample has been adulterated, and if so, to what degree.

One way to determine the chemical composition of an oil sample is through hyperspectral imaging (HSI). A hyperspectral imaging system will shine infrared light onto the sample and measure the reflection off the sample at different wavelengths. We will not go into the details of how this signal is acquired, but what is important is that the system outputs for each sample a so-called *spectral curve* describing the average reflectance at different wavelengths. By inspecting the spectral curve, we can establish whether the sample absorbs light of a specific wavelength, and this can point towards the presence of particular chemical structures that characterize the sample. An example of a spectral curve is shown in figure Figure 4.1.

In our case study we want to determine whether hyperspectral imaging can be used to detect the adulteration of extra-virgin olive oil. More precisely, we have the following research questions:

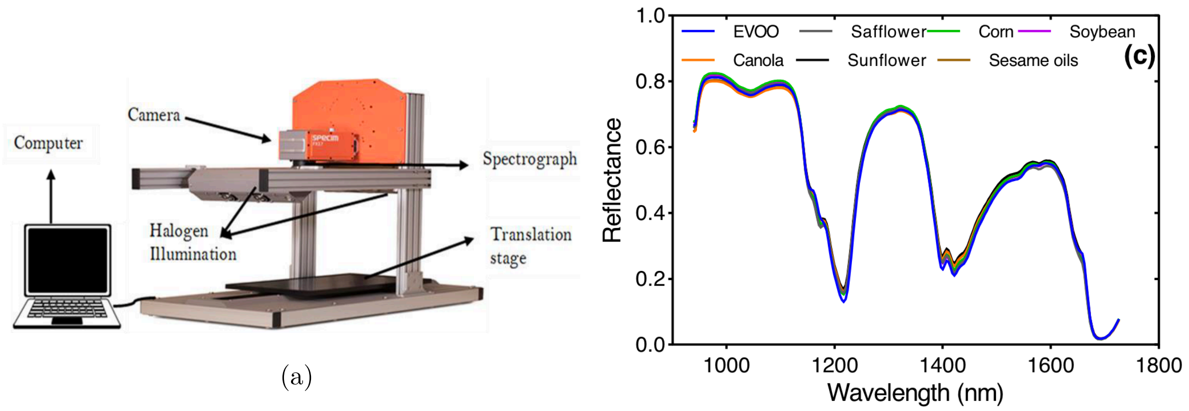
1. *Can hyperspectral imaging be used to detect whether olive oil has been adulterated with other kinds of oils?*
2. *If so, can the amount of adulteration be quantified (e.g. as a percentage)?*

To investigate these research questions, Malavi et al. (2023) acquired readings from 13 different kinds of unadulterated EVOO, together with readings from 42 adulterated mixtures. Each adulterated mixture was prepared by taking olive oil and adding one of 6 different vegetable oils at 7 different percentages (ranging from 1% to 20% adulterant). Each sample was prepared and imagined in triplicate, resulting in 183 spectra. Each spectrum is a vector of length 224, describing the reflectance at 224 equally distributed wavelengths from 700 to 1800 nm.

Below, we read in the dataset and we print out the first 6 columns of a random sample of 10 rows (the full dataset is too large to display in its entirety). Note that the dataset has 228 columns: 4 of these are metadata variables described below and the remaining 224 columns describe the spectrum for each sample. The metadata variables are:

- `Sample ID/Wavelength`: The name of the oil or mixture, and the mixture ratio (if applicable)
- `Sample`: A unique integer identifying each sample. Not used in the subsequent analysis.
- `Classification`: Whether the sample is primarily olive oil or not.
- `% Adulteration`: The percentage of food oil added to the mixture. For pure EVOO this is 0, while for pure food oil it is 100%. For the other mixtures, it is one of 1%, 2%, 4%, 8%, 12%, 16%, or 20%.

```
# A tibble: 10 x 6
   `Sample ID/Wavelength`       Sample Classification `% Adulteration`
```



(a)

(b)

Figure 4.1: Hyperspectral imaging system (left) and typical output spectra (right). Figure source: Malavi et al. (2023).

```
   <chr>                         <dbl> <chr>                     <dbl>
 1 EVOO flavored with Chilli        28 Olive                         0
 2 Olive Oil/Corn Oil 99/1          80 Olive                         1
 3 Olive Oil/Sunflower Oil 96/4    150 Olive                         4
 4 Olive Oil/Soybean Oil 99/1      101 Olive                         1
 5 Olive Oil/Soybean Oil 92/8      111 Olive                         8
 6 Olive Oil/Canola Oil 84/16      137 Olive                        16
 7 Olive Oil/Canola Oil 88/12      133 Olive                        12
 8 Olive Oil/Sesame Oil 98/2       166 Olive                         2
 9 Olive Oil/Sunflower Oil 99/1    144 Olive                         1
10 Olive Oil/Canola Oil 92/8       132 Olive                         8
# i 2 more variables: `938.94000200000005` <dbl>, `942.45001200000002` <dbl>
```

4.1.2 Inspecting the spectral plots

As a first step in our analysis, let’s compare the typical spectrum of EVOO with that of the other 6 vegetable oils. We see that overall the spectra are quite similar (they are all oils, after all) but that there are small differences between the different kinds of oil. On the other hand, if we are given a new, unlabeled spectrum, it would be quite difficult to “guess” just by looking what type of oil it is. This is where dimensionality reduction will help us!

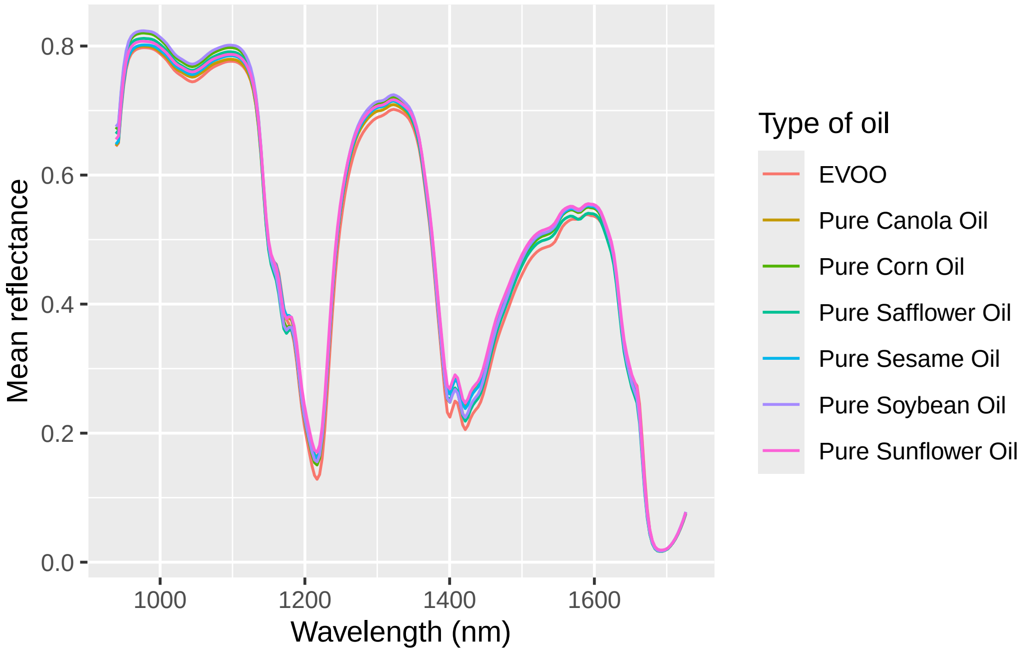


Figure 4.2: Typical spectra of EVOO and different types of vegetable oils. Note the similarity with figure Figure 4.1b.

4.1.3 Computing the principal components

We are now in a position to compute the principal components, and to figure out what they tell us about olive and other oils. Two issues to keep in mind:

1. Our dataset contains both metadata and spectral data: we want to make sure to compute the principal components only for the columns containing spectral data!
2. For spectral data, we definitely do **not** scale the different variables!

```
pca_oils <- oils %>%
  select(all_of(spectra)) %>%
  prcomp(scale = FALSE)
```

The first two principal components explain more than 90% of the variance in the data, as shown below.

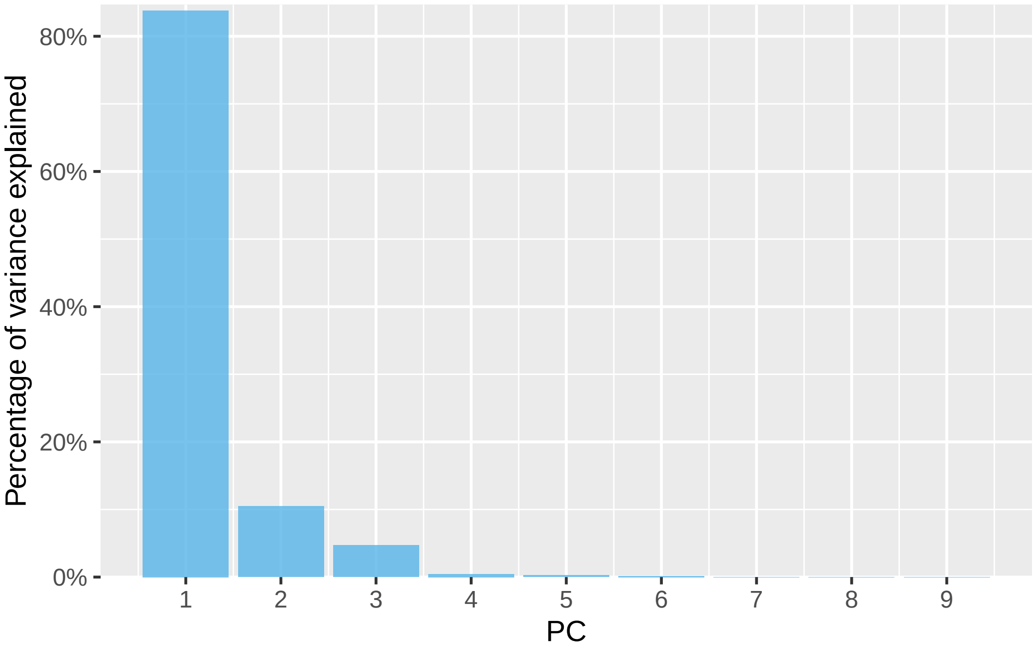


Figure 4.3: Percentage of variance explained by the first 10 principal components. The first two principal components explain 94% of the variance in the data.

The loadings for the first two principal components tell us something about what the spectra look like overall, and what variations between the individual spectra look like. This is of course an extremely coarse perspective, given that we only consider two principal components.

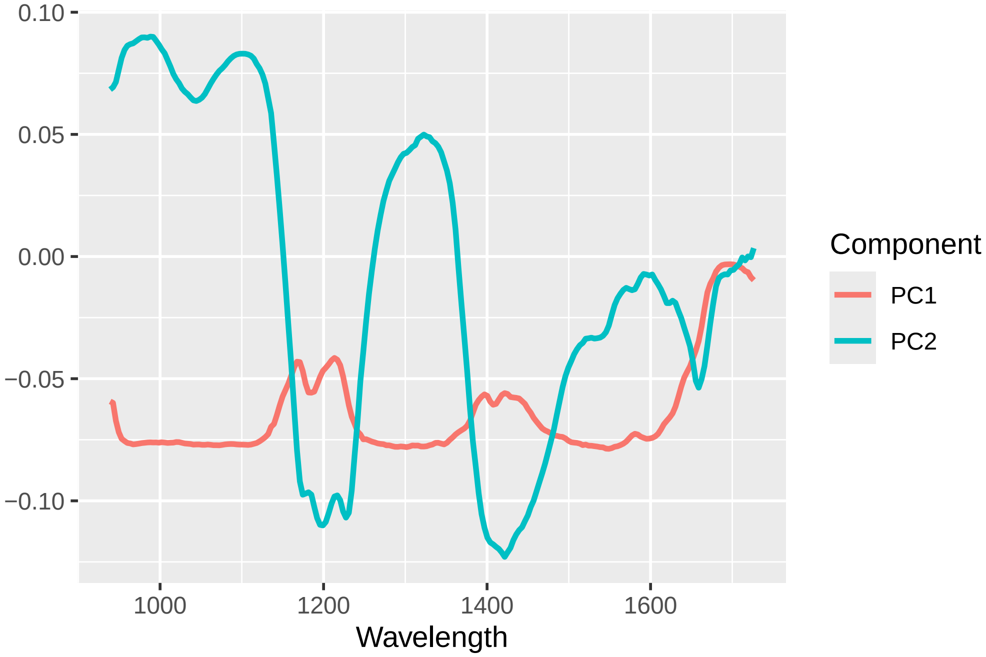


Figure 4.4: Loadings for the first two principal components.

We started this section with the observation that there are minute differences between the spectra of different oils, but we admitted that it would be quite difficult to tell two spectra apart just by eye. We are now in a position where we can solve this issue: by considering only the first few principal components we get a reduced number of variables that (hopefully) can be used to tell the spectra apart.

Below we have a scatterplot of the first two principal components. The EVOO samples are round dots, and the adulterated oils are triangles colored by the percentage of adulteration. The EVOO samples are clearly distinct from the adulterated oils (they are separated by the second principal component). This answers our first research question with a “yes”: we can clearly tell pure and adulterated oils apart.

4.1.4 Predicting the percentage of adulteration

Now that we are able to tell EVOO and adulterated samples apart, it is time to consider our second research question: given an adulterated oil sample, can we predict the percentage of vegetable oil that was added? Looking at Figure 4.5, it looks like we'll need more than two principal components: we see no clear pattern in the percentage adulteration that we or a linear model could exploit.

Before building our model, we need to do some data preparation. We select from our dataset only those rows that contain adulterated oils, and we split the resulting dataset into a *training*

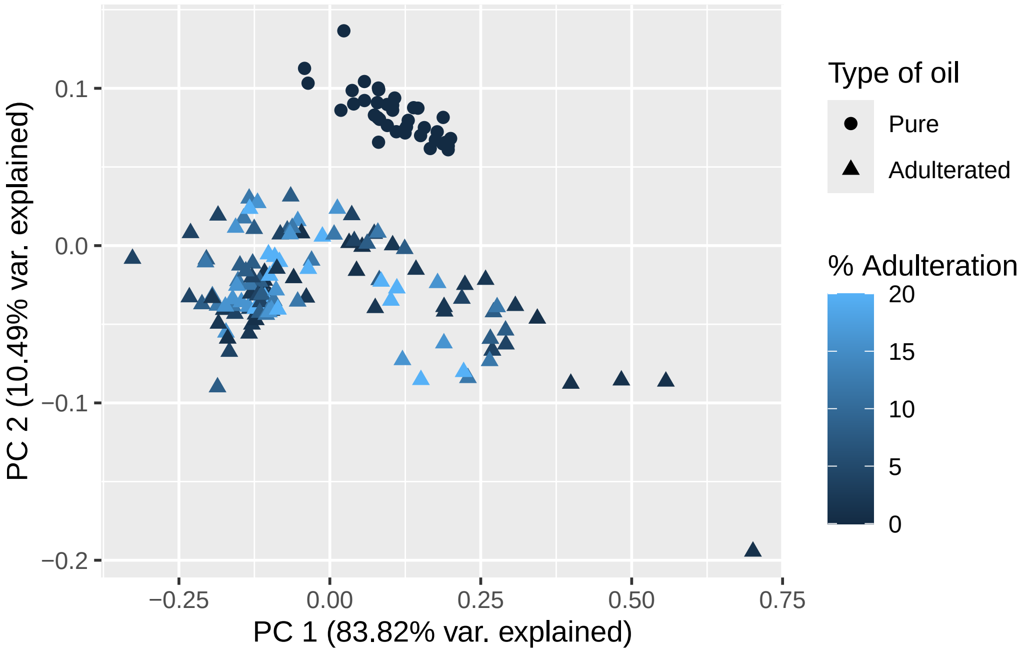


Figure 4.5: Principal component plot for the EVOO and adulterated oil samples. Pure vegetable oil samples are not shown.

dataset (containing 80% of the data) and a *test dataset* (containing the remaining 20%). The idea is that we set apart the test dataset and build our model using only the training dataset. The test dataset is used to evaluate how well the model performs on data that it has never seen before. Using the full dataset to build the model and evaluate it would result in an overly optimistic estimate of the predictive capabilities of our model.

```
Number of rows in the training dataset: 101
```

```
Number of rows in the test dataset: 25
```

We could build a linear model with more than two components by hand, just like we did for the bodyfat dataset in the previous chapter. However, it is easier to let R do the work for us, and we will use the the `pls` package for this. This package has a number of advantages: you can use the same formula syntax that you know from the `lm` command, and the package comes with a number of different regression models. Here we build a principal component regression model and a partial least squares regression model.

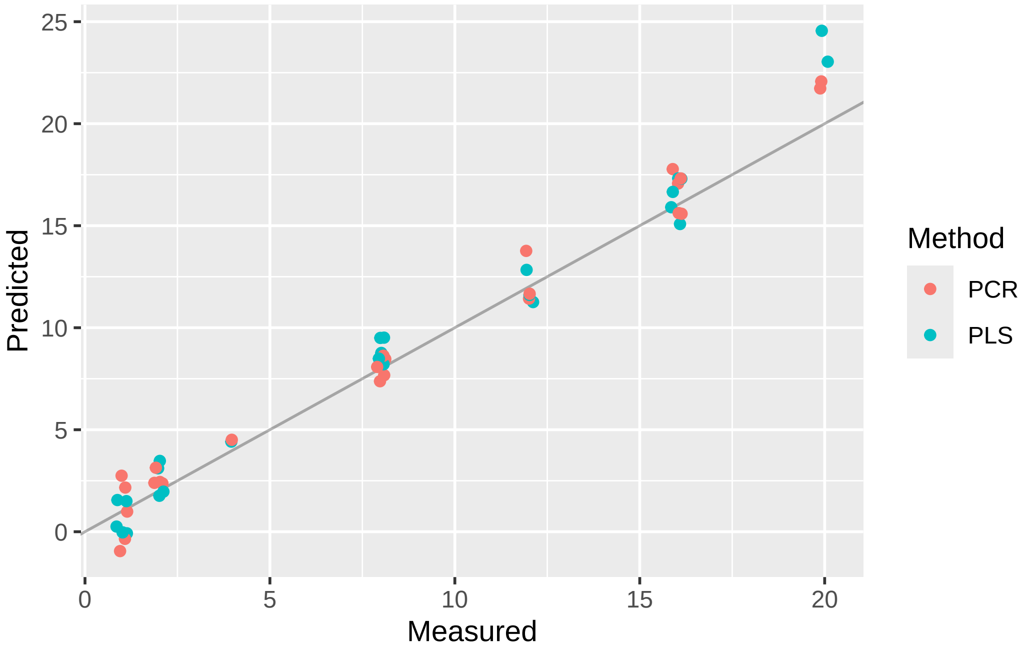
```r
library(pls)
```

```
# Principal component regression
pcr_model <- pcr(
  `% Adulteration` ~ ., data = adulterated_train, scale = FALSE, validation = "CV", ncomp =
)

# Partial least squares regression
pls_model <- plsr(
  `% Adulteration` ~ ., data = adulterated_train, scale = FALSE, validation = "CV", ncomp =
)
```

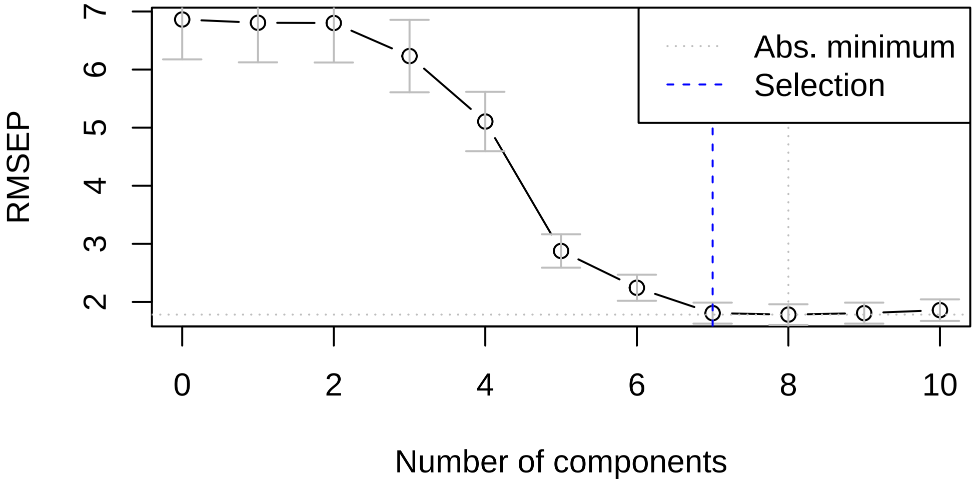
The argument `validation = "CV"` will cause the model to select by itself the appropriate number of components to use, up to a maximum of 10 components (this is controlled by the `ncomp = 10` argument). To do this it will use the so-called cross-validation (CV) strategy: it will repeatedly split the data into a training and a validation set (not to be confused with the training and test set that we created ourselves above) and use the performance on the validation set to select the appropriate number of components.

Now that we have two models, we can ask them to make predictions on the held-out test data, and we can compare the predictions made between both models. In the prediction plot below we see in general good agreement between the actual percentage of adulteration (on the x-axis) and the predicted percentage (on the y-axis). Both models perform comparably well, at least as far as can be judged from the plot.

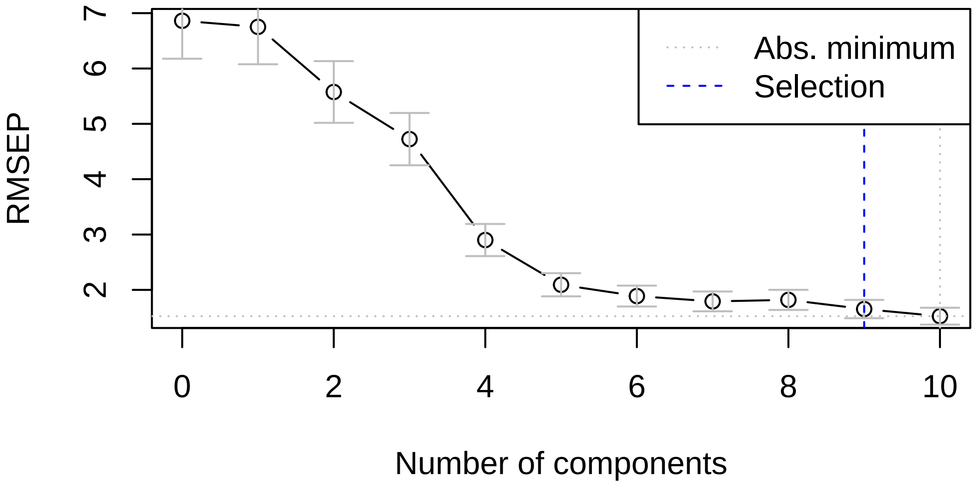
```
pcr_pred <- predict(pcr_model, adulterated_test, ncomp = 10)
pls_pred <- predict(pls_model, adulterated_test, ncomp = 10)
```



4.1.5 Optimal number of components



```
[1] 7
```



```
[1] 9
```

4.1.6 A cautionary note

In our model building procedure we have glossed over a few critical steps that may cause doubt on the validity of the model or the conclusions that we can draw from our analysis.

- Chiefly among these is the splitting into training and test data. We did that correctly to train our model, but everything that came before it (including the exploratory data analysis) used the entire data set. To build a model that is not overly optimistic, you should set aside some data to serve as test data at the beginning of your analysis, and then use only the training data to determine the appropriate number principal components, to classify oils into adulterated and pure, and so on. *Your model should never "see" the test data, until the very end.*
- We have trained and evaluated our model only on adulterated oils. In reality, however, we will present the model with unknown oil samples, which may or may not be adulterated, and the performance on such samples is an open question. A better model could be built by including EVOO samples in our dataset, or by first classifying oils into adulterated/pure. Each approach comes with a number of complications that would lead us too far.
- Our model willfully ignores a number of important variables in the dataset. For example, we don't take into account the kind of vegetable oil that was used to adulterate a sample. By distinguishing the type of adulterant, we could presumably build a better model, but this too would make the analysis significantly more complex.

A more complete model that addresses all of these concerns (and more) can be found in (Malavi et al. 2023).

4.2 Eigenfaces

Our last example is not life sciences based, but serves as an illustration to show that PCA is a powerful technique in data analysis, which can be used to reduce the number of degrees of freedom in a large dataset.

We use the Olivetti dataset of human faces, which contains 400 frontal photographs of human faces. Each face is a grayscale image of 64 by 64 pixels, where the intensity of each pixel is a value between 0 (completely black) to 255 (completely white). Each image can be represented as a 64×64 matrix, but it will be more convenient to take the columns of this matrix and lay them out one after the other to obtain a vector with $64 \times 64 = 4096$ entries, as in Figure 4.6.

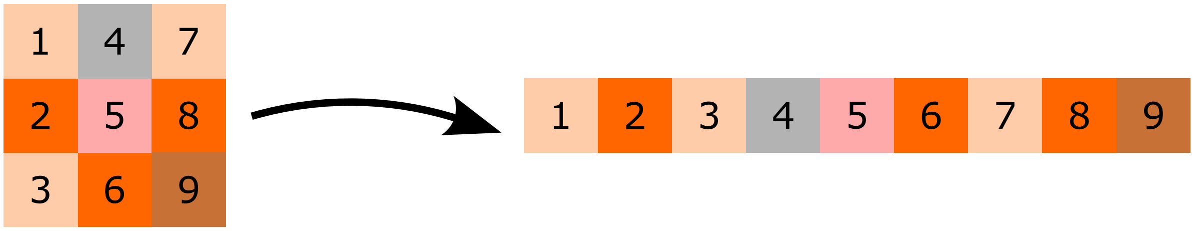


Figure 4.6: An image that is N pixels high and M pixels wide can be viewed as a matrix with N rows and M columns, or as a vector with $N \times M$ elements. Here, N and M are both equal to 3.

First, we load the dataset. Note that the dataset comes as a data matrix with 4096 rows and 400 columns.

```r
library(loon.data)
data(faces)
dim(faces)
```

```
[1] 4096  400
```

Each column in the data matrix represents a face, laid out as a column vector with 4096 as in Figure 4.6. We can assemble these vectors back into images and visualize them. This requires some R commands that we haven't covered; you don't have to understand what this code does.

Doing a principal component analysis is a simple matter of running `prcomp`. Despite the size of the dataset, this component should not take more than a second to run.

```r
pc_olivetti <- prcomp(faces)
```

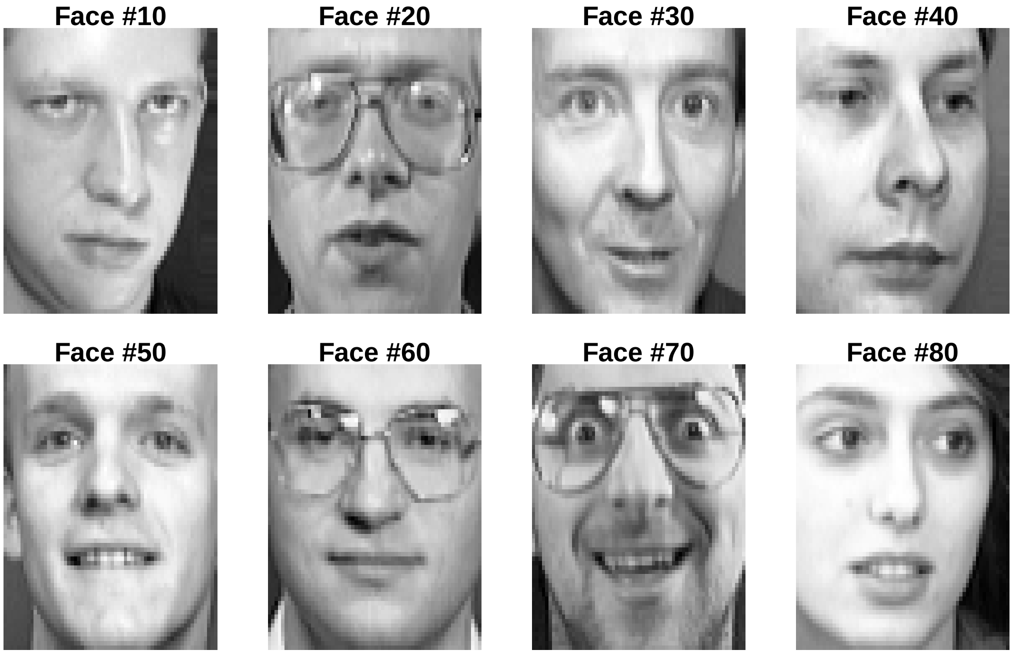
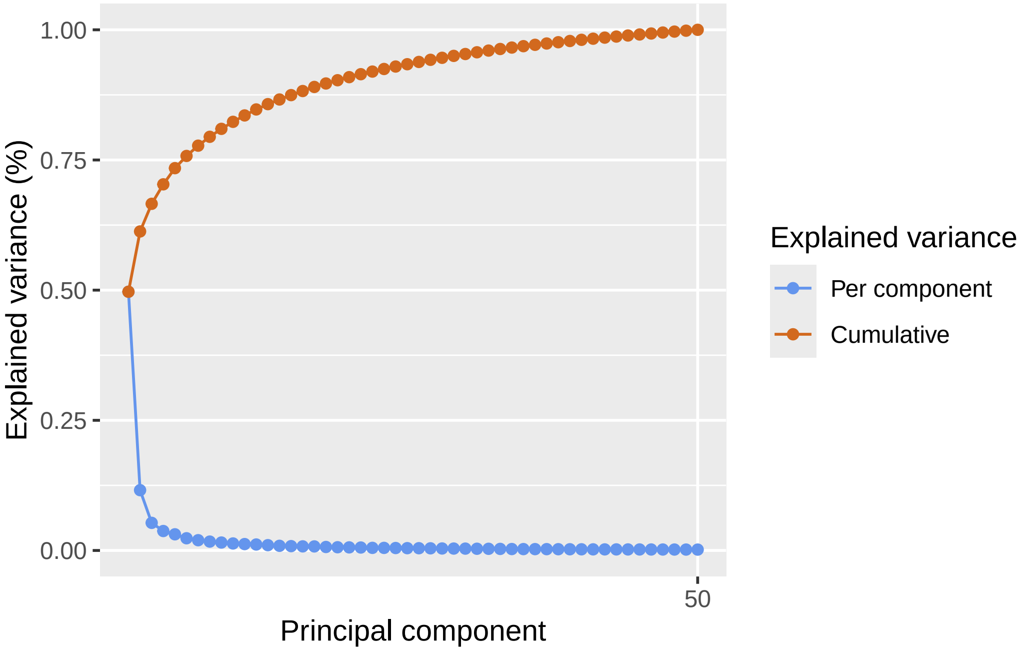


Figure 4.7: Six faces from the Olivetti dataset.

Note that there are 400 principal components in this dataset. We can visualize their relative importance via a scree plot, which we limit to the first 50 components for clarity, since the remaining 350 components contribute almost no variance. This indicates that we can probably discard most of the principal components without losing much of the expressivity of our dataset. We will see further down that this is indeed the case!



One of the advantages of the faces dataset is that the loadings vectors can be represented graphically, and that we can reason about them. Figure 4.8 shows the first 8 loadings vectors, represented as images. How should we interpret these images? Each loadings vector represents a particular *pattern* in the dataset of all faces: the first loadings vector, for example, captures the overall structure of a human face, while the second represents the illumination from right to left. Probably there were some photos in the dataset that were illuminated from the left or the right. Loadings vector three does the same for the top-down illumination, and loadings vectors four through eight capture particular patterns involving the eyes or the eyebrows. *By selectively "mixing" all 400 loadings vectors, we can recreate any face in the dataset.*

To finish, let's also investigate how well PCA performs as a data reduction method. By retaining only a limited number of principal components, we can build "compressed" versions of the images that contain less information but are comparable to the eye. Figure 4.9 shows two original faces from the dataset (left), together with compressed versions involving the first 10, 40, and 80 most significant principal components. The version that uses only 10 components is quite generic and it is difficult even to distinguish the male and female face. The version with 80 components, on the other hand, is very close to the original.

It is worth realizing the amount of data compression realized by using PCA. The original images had 4096 degrees of freedom, whereas the rightmost versions in Figure 4.9 are described by the coefficients of 80 loadings vectors, more than a 50-fold reduction in degrees of freedom! Clearly there are some visual artifacts that appear in the compressed versions, but the faces are clearly distinguishable, and it seems very reasonable at this point that a machine learning algorithm (for example, to classify the faces, or to do segmentation) could take these compressed images as input and still perform well.

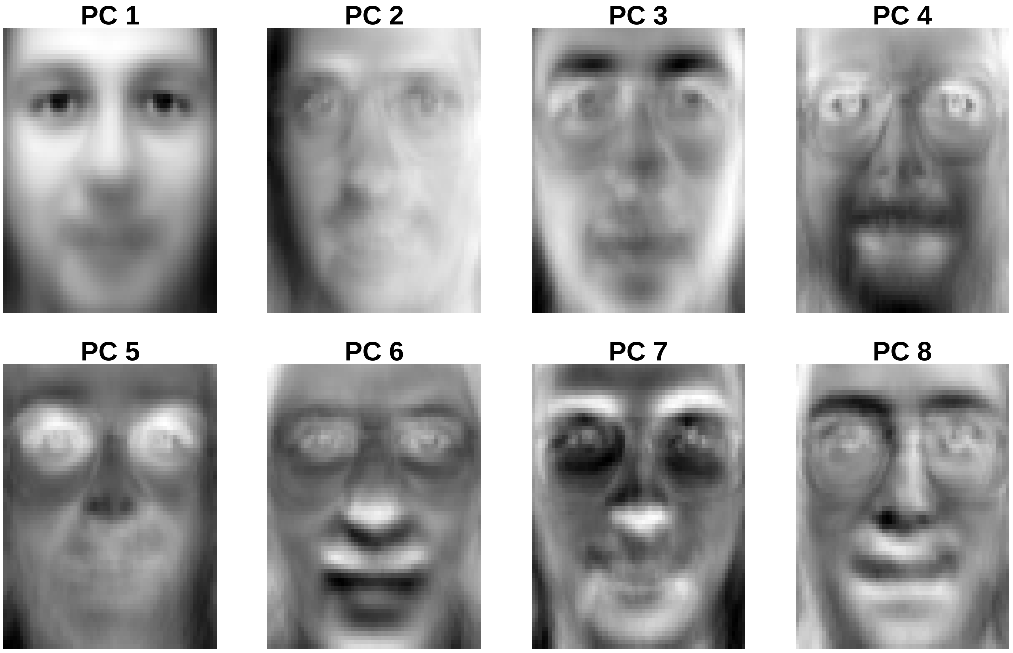


Figure 4.8: The first 8 loadings vectors of the Olivetti dataset represent particularly expressive patterns in the dataset.

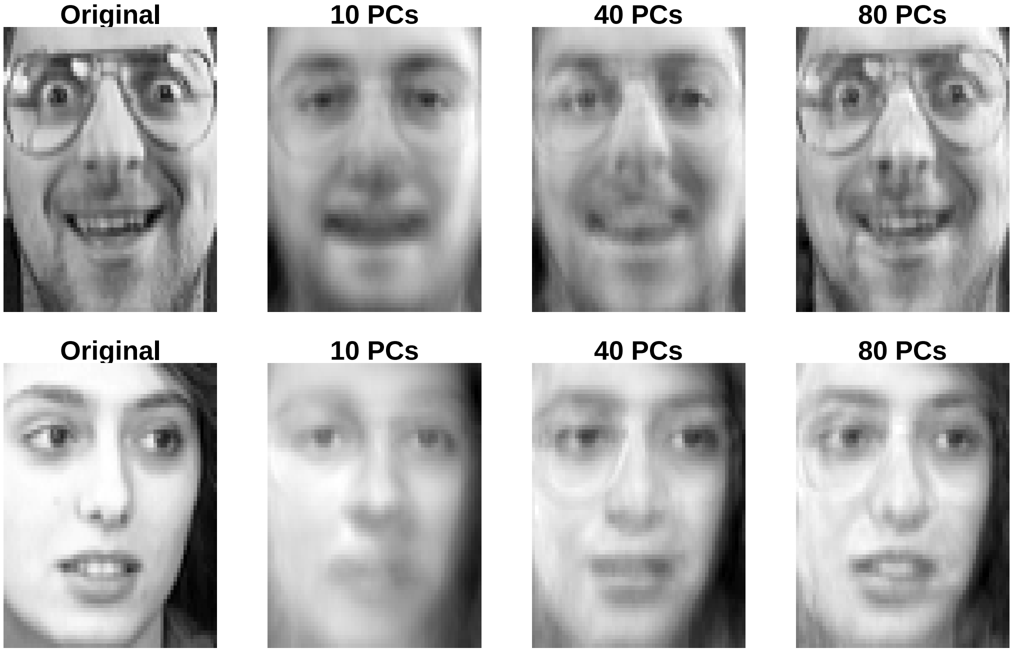


Figure 4.9: Original images (left), and 3 PCA-reduced images with increasing numbers of principal components.

References

Bishop, Christopher M. 2006. *Pattern Recognition and Machine Learning*. Vol. 2. Information Science and Statistics. Springer, New York.

Hastie, T., R. Tibshirani, and J. H. Friedman. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics. Springer.

Malavi, Derick, Amin Nikkhah, Katleen Raes, and Sam Van Haute. 2023. "Hyperspectral Imaging and Chemometrics for Authentication of Extra Virgin Olive Oil: A Comparative Approach with FTIR, UV-VIS, Raman, and GC-MS." *Foods* 12 (3): 429. https://doi.org/10.3390/foods12030429.

A Datasets

A.1 Point clouds with specific mean and covariance

The `rmvnorm` command, part of the mvtnorm package, provides a way to sample points from a multivariate normal distribution with a specific population mean and standard deviation. For example, the code snippet below generates 100 sample points from a distribution with mean μ and covariance matrix Σ given by

$$\mu = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \Sigma = \begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix}. \tag{A.1}$$

The `mean` and `sigma` parameters are the *population* mean and covariance, and the *sample* mean and covariance will be slightly different:

```
bar_x <- colMeans(samples)
bar_x
```

```
[1] 0.9661279 0.7986743
```

```
sigma_x <- cov(samples)
sigma_x
```

```
         [,1]     [,2]
[1,] 2.162411 1.201452
[2,] 1.201452 3.173997
```

In some cases, we require a dataset whose sample mean and covariance are *exactly* equal to some given parameters. It turns out that we can achieve this by means of a judiciously chosen linear transformation. Assume that we have n datapoints $x_i \in \mathbb{R}^p$, and let A be an arbitrary $p \times p$ matrix and $b \in \mathbb{R}^p$ a vector. The transformed data points

$$y_i = Ax_i + b, \tag{A.2}$$

have sample mean $\overline{y}$ and covariance matrix Σ_y given

$$\overline{y} = A\overline{x} + b \quad \text{and} \quad \Sigma_y = A\Sigma_x A^T,$$

where $\overline{x}$ and Σ_x are the sample mean and covariance matrix of the x variables.

To make the sample mean equal to a given vector μ, and the sample covariance matrix equal to a given matrix Σ, we have to solve the following equations for A and b:

$$A\overline{x} + b = \mu \quad \text{and} \quad A\Sigma_x A^T = \Sigma.$$

Let's focus on the second equation first, since it only involves the matrix A. Since the covariance matrices Σ and Σ_x are positive definite, it turns out that there exist unique upper-triangular matrices R and R_x such that

$$\Sigma_x = R_x^T R_x \quad \text{and} \quad \Sigma = R^T R.$$

This is the so-called Cholesky decomposition; we don't have to worry about the details here, since R can compute R and R_x for us. Using this decomposition, the equation for A can be written as

$$AR_x^T R_x A^T = (R_x A^T)^T R_x A^T = R^T R,$$

so that it is sufficient to find a matrix A such that $R_x A^T = R$. This we know how to do: we just multiply both sides from the left by $(R_x)^{-1}$ and take the transpose to find

$$A = (R_x^{-1} R)^T.$$

With this expression for A, we can find b by setting

$$b = \mu - (R_x^{-1} R)^T \overline{x}.$$

We can code up these equations in R as in the snippet below. This approach works well for data points in low dimensions. For higher-dimensional data it is recommended (for reasons of numerical efficiency and stability) to avoid computing the inverse R_x^{-1} directly. Instead, we can solve a system of linear equations that yield the transformed points y_i directly. We will not pursue this alternative any further.

```r
add_to_rows <- function(X, b) {
  t(apply(X, 1, function(row) row + b))
}

R <- chol(sigma)
R_x <- chol(cov(samples))
A <- t(solve(R_x) %*% R)
b <- mean - A %*% bar_x
samples_y <- samples %*% t(A)
samples_y <- add_to_rows(samples_y, b)
```

The resulting datapoints have mean and covariance exactly as given by Equation A.1:

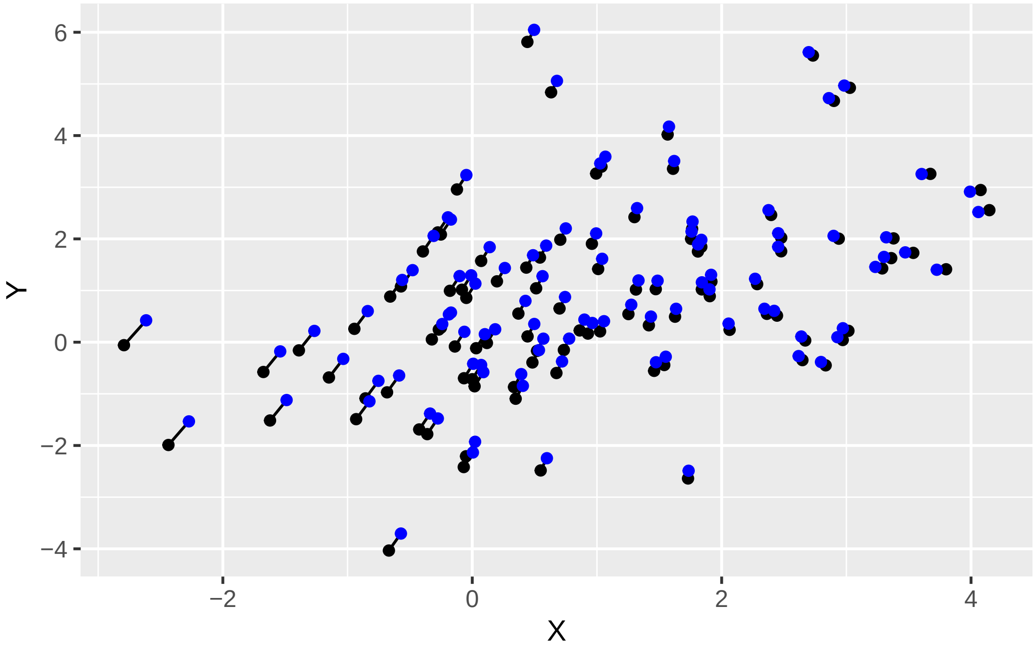
```
colMeans(samples_y)
```

```
[1] 1 1
```

```
cov(samples_y)
```

```
     [,1] [,2]
[1,]    2    1
[2,]    1    3
```

Last, we can take a look at the relative location of the original and adjusted data points. We see that our procedure selectively moves points around to make the mean and covariance matrix equal to what we imposed on it.


6
4
2
0
−2
−4
Y
−2
0
2
4
X