Part III - How To in R and RStudio

The examples in this How To have been made in R version 3.5.1 (R Core Team, 2018) and RStudio version 1.1.456 (RStudio Team, 2016). We recommend to always use the latest version by checking for updates regularly. If you are working with an earlier or later version than the one used here, however, the options should still be largely the same.

1. How To: Descriptive Statistics

1.1 Open and inspect data in R or RStudio

Almost all files can be opened in RStudio, but R cannot by default open Excel or SPSS files. Please see Practical 1C (assignment 2) for details on how to open these specific file formats.

For this How To Do Descriptive Statistics, we will use example proficiency data from two groups saved as a CSV file called 'Data_set_proficiency_scores.csv'. We can open and inspect it as follows:

```
data = read.csv("Data_set_proficiency_scores.csv", header=T, sep=",")
head(data)

##   Participant Group Profscore
## 1           1     1        73
## 2           2     2        59
## 3           3     1        92
## 4           4     2        87
## 5           5     1        79
## 6           6     2        74

str(data)

## 'data.frame':    30 obs. of  3 variables:
##  $ Participant: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Group      : int  1 2 1 2 1 2 1 2 1 2 ...
##  $ Profscore  : int  73 59 92 87 79 74 47 66 69 53 ...
```

After opening the data, you should always inspect the file and, if necessary, change the measurement scales and add the appropriate labels (also see Practical 1C, assignment 3).

As explained in Practical 1C (assignment 4), whatever format the original data is saved in, it is advisable to save your data as RDS after inspection and possible changes made.

```
data$Participant = as.factor(data$Participant)
data$Group = factor(data$Group,
          levels = c(1,2),
          labels = c("beginner", "advanced"))
saveRDS(data, file="DataProf.rds")
data = readRDS("DataProf.rds")
str(data)

## 'data.frame':    30 obs. of  3 variables:
##  $ Participant: Factor w/ 30 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10 ...
```

```
##  $ Group     : Factor w/ 2 levels "beginner","advanced": 1 2 1 2 1 2 1 2 1 2 ...
##  $ Profscore : int  73 59 92 87 79 74 47 66 69 53 ...
```

1.2 Descriptives using single calculations

Before you carry out any inferential statistics, it is important that you calculate some descriptive statistics. In R, it is very straightforward to calculate the mean, median, range (this gives you the minimum and the maximum), minimum, maximum and standard deviation. Simply use the following codes:

```
mean(File$Variable)
median(File$Variable)
range(File$Variable)
min(File$Variable)
max(File$Variable)
sd(File$Variable)
```

Calculating the mode in R is less straightforward and requires the use of a function (see Practical 2A, assignment 3a, for a brief explanation of this function:

```
which.max(tabulate(File$Variable))
```

You may often have datasets with which you would like to compare the means, medians, or standard deviations of two or more groups. In that case, you can use the code below which calculates the mean for the dependent variable (DV) as a function of the independent variable (IV):

```
aggregate(DV~IV, File, mean)
```

In order to fill in this code correctly, you will have to have a clear idea of your data and especially about the dependency relationship between variables (see Section 4.2 in Part I). This formula makes sure that the DV is split up based on the levels of the IV and the mean, or any other measure you may be interested in, is calculated for each level of the IV. For our data with proficiency scores for two different groups, we would write and obtain the following:

```
aggregate(Profscore~Group, data, mean)

##       Group Profscore
## 1 beginner  67.00000
## 2 advanced  67.33333
```

1.3 Descriptives using the psych package

A useful way to report on several descriptive statistics at once is by using the “psych” package (Revelle, 2018). You can install and call this package to your library by using the following codes:

```
install.packages("psych")
library("psych")
```

After you have called the package to your workspace (see Practical 2B for example code), you can start using the describe() function as follows:

```
describe(File$Variable)
```

This would provide the following information for our proficiency scores example:

```
describe(data$Profscore)

##    vars  n  mean   sd median trimmed   mad min max range skew kurtosis
## X1    1 30 67.17 13.3   67.5   66.96 13.34  42  93    51 0.03     -0.8
##      se
## X1 2.43
```

As you can see, describe() provides you with the most important descriptives including information on the number of variables and participants, the most common measures of central tendency (mean, median, and mode), and measures for dispersion (*SD* and range).

1.4 Creating a Table in Markdown

Once you have calculated all scores, it is important to report them in an informative way. You can create your own table in Word, but you can also do this in R Markdown (Allaire et al., 2018) as we did during the practicals in Part II.

As explained in more detail in Practical 2C (assignment 2a), you will have to replace the x's in the example table below by the values you calculated in order to create a table in R Markdown. .

```
value | Overall | Beginner | Advanced
- | - | - | -
mean | x | x | x
mode | x | x | x
median | x | x | x
sd | x | x | x
```

This returns the following table in your knitted document:

Table III.R.1: Descriptives for the overall proficiency scores as well as the scores for beginner and advanced learners separately.

value	Overall	Beginner	Advanced
mean	x	x	x
mode	x	x	x
median	x	x	x
sd	x	x	x

Table III.R.1 would be used when reporting overall scores, together with the scores for both groups. You can of course add as many rows and columns as you want, just make sure you use the proper amount of pipes (|) to separate the values in each row.

1.5 Creating a simple scatterplot or boxplot

Before you carry out any inferential statistics, it is also important that you visually inspect your data by making plots. When you are dealing with interval variables only and you mainly want to examine potential relationships between variables, a scatterplot is the best option. A simple scatterplot can be created by using the following code:

```
plot(File$Variable1, File$Variable2, main="Title of the plot",
   xlab="Variable 1 ", ylab="Variable 2")
```

A plot that is important for describing data in which groups are to be compared is a boxplot. You can use the following code to create a simple boxplot:

```
boxplot(DV ~ IV, data=File, main="Title of your boxplot", xlab="Label for the x-axis", ylab=" Label for the y-axis ")
```

Note again that the tilde (~) is used to split up the DV values on the basis of the levels of the IV. The DV will be plotted on the y-axis and the IV will be plotted on the x-axis. This boxplot would be a good choice for the proficiency data to compare the distribution of the two groups side by side, as in Figure III.R.1 below.

```
boxplot(Profscore ~ Group, data=data, main="Proficiency Scores of Group 1 and 2", xlab="Group", ylab="Proficiency Score")
```

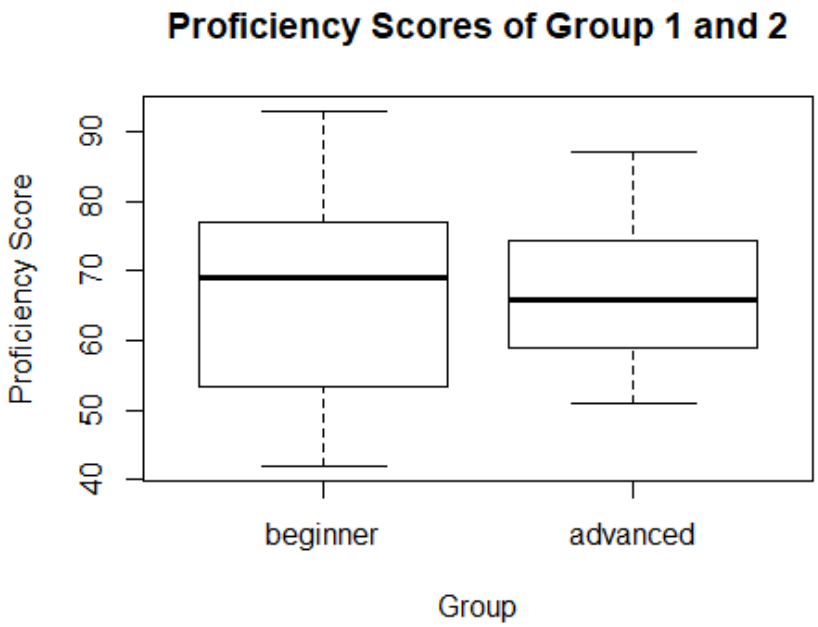

Figure III.R.1: Boxplot showing the dispersion in proficiency scores in group 1 (left) and group 2 (right).

Never forget to caption your tables and figures! As explained in Practical 2C, you can just type your text and create a bold title using asterisks as follows: .

So, `**Figure X** Your caption goes here` will become the following in your knitted document:

Figure X Your caption goes here

In case you are creating your report in Word, you can easily export your figures as PNG or PDF (or whatever format you prefer) using the following code:

```
dev.new()
png("NamePlot.png")
plot(File$Variable1, File$Variable2)
dev.off()
```

The first line, `dev.new()` is used to open a new device. The second line creates a PNG file and specifies the filename ("NamePlot.png"). You can change this to, for example, `pdf("NamePlot.pdf")` to create a PDF instead (or use JPEG or TIFF or any other format). The third line of code is the actual plot, which in this case is a scatterplot, and you can replace

this line with code for any plot you may want to export. The final line dev.off() closes the device, after which you should be able to locate and open your exported figure in the folder you are working from.

1.6 Calculating z-scores

Sometimes you would want to report on *z*-scores, for example in case you want to know how many standard deviations each participant is away from the mean. You can use the code below to calculate the z-scores for all values of a variable at once:

```
scale(File$Variable, center = TRUE, scale = TRUE)
```

Another option would be to simply look up individual *z*-scores, or add the *z*-scores as a column to your existing dataset and subsequently create subsets (also see Practical 3A, assignments 2a and 4a, for details and examples):

```
File$zscore <- scale(File$Variable, center = TRUE, scale = TRUE)SubsetName <- subset(data, group == "groupname in your data")
```

In addition to the descriptives mentioned here, it is important to check the assumptions to find out whether you are allowed to conduct the (parametric) test you were aiming for. Checking assumptions will be the main topic of the next How To.

2. How To: Check Assumptions

Before you carry out inferential statistics, you need to look at the descriptive statistics and check the relevant assumptions for the different parametric tests. This How To will only deal with the practical aspect of how to check some of the assumptions in R. For a list and details about all of the assumptions, see Section 4.6 in Part I of the book.

2.1 Checking for normality interval data

Imagine you are interested in two groups of language learners, a beginner group and an advanced group. They have proficiency scores on a scale from 1 to 100 obtained by means of a multiple choice test. For this design, which can be analysed using an Independent Samples *t*-test, we would have to check normality for each group separately.

A good place to start is by looking at histograms of the data (per group!) and to check whether their shapes approximately follow the bell-shaped curve of the normal distribution. You can make subsets or use the following code to create a histogram like the one plotted in Figure III.R.2 of one of the group's proficiency scores with a normal curve plotted over it.

```
hist(data$Profscore[data$Group ==“beginner”], prob=TRUE, xlab = "Proficiency Score", main = "Distribution of beginner's proficiency scores")
curve(dnorm(x, mean=mean(data$Profscore[data$Group ==“beginner”]), sd=sd(data$Profscore[data$Group ==“beginner”])), add=TRUE)
```

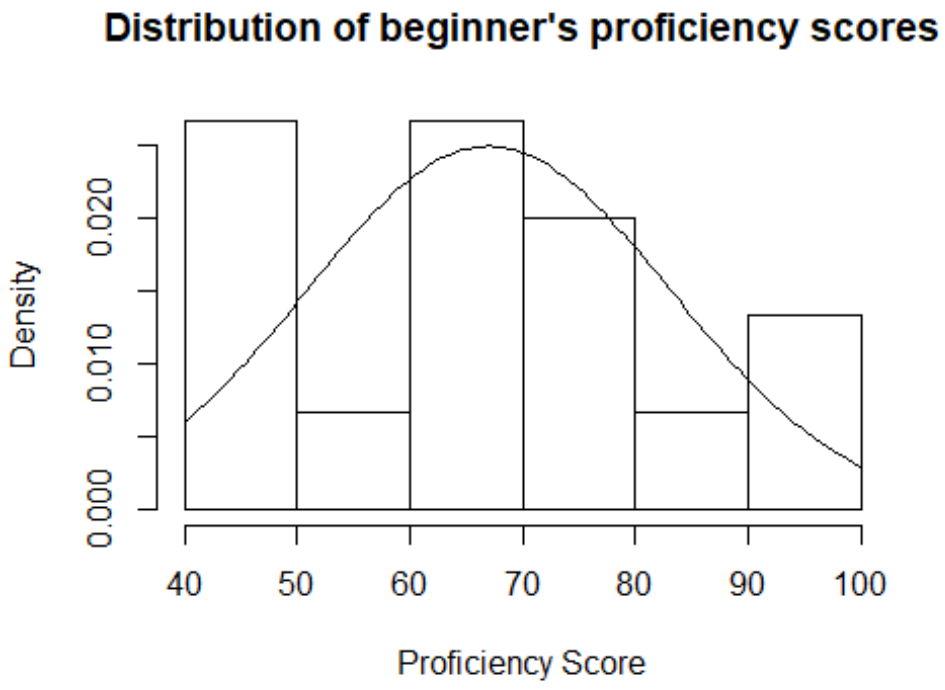

Figure III.R.2 Probabilities of proficiency scores for the beginner learners.

The data in the histogram clearly deviates from a normal distribution, but this is likely due to the small sample size (see Practical 3A, assignment 6, for details).

Next, look at the values for skewness and kurtosis and how much they deviate from zero. As we only have 30 participants and 15 learners per group in this particular dataset, we will look at the standardized values for kurtosis and skewness (see Practical 3 for an explanation of why these values are more appropriate in this case).The easiest way to obtain these values in R is by using the stat.desc() function from the package "pastecs" (Grosjean and Ibanez, 2018).

The basic code (after loading the package) is:

```
stat.desc(FileName$VariableName, basic=FALSE, norm=TRUE)
```

And for our dataset, we should then do the following:

```
by(data$Profscore, data$Group, stat.desc, basic=FALSE, norm=TRUE)

## data$Group: beginner
##        median          mean       SE.mean  CI.mean.0.95          var
##  69.000000000  67.000000000   4.131182236   8.860504665 256.000000000
##       std.dev      coef.var      skewness      skew.2SE      kurtosis
##  16.000000000   0.238805970   0.003613281   0.003114257  -1.235721842
##      kurt.2SE     normtest.W     normtest.p
##  -0.551220031   0.951436325   0.547352164
## ------------------------------------------------------------
## data$Group: advanced
##        median          mean       SE.mean  CI.mean.0.95          var
##   66.0000000   67.3333333    2.7109420    5.8143922  110.2380952
##      std.dev      coef.var      skewness      skew.2SE      kurtosis
##   10.4994331    0.1559322    0.1480921    0.1276394   -1.1831915
##     kurt.2SE    normtest.W    normtest.p
##   -0.5277878    0.9734684    0.9056797
```

For the current dataset, the skew.2SE and kurt.2SE values are all well within the desired range, suggesting an approximately normal distribution for both groups.

Finally, you could perform a Shapiro-Wilk for each group and make sure that these are non-significant to ascertain that your data are normally distributed.

To obtain the Shapiro-Wilk results, you should fill in the details in the code:

```
shapiro.text(File$Variable)
```

For the current dataset, we would type and obtain the following:

```
shapiro.test(data$Profscore[data$Group =="beginner"])

##
##  Shapiro-Wilk normality test
##
## data:  data$Profscore[data$Group == "beginner"]
## W = 0.95144, p-value = 0.5474

shapiro.test(data$Profscore[data$Group =="advanced"])

##
##  Shapiro-Wilk normality test
```

```
##
## data:  data$Profscore[data$Group == "advanced"]
## W = 0.97347, p-value = 0.9057
```

For both groups, the significance values are *p*s > 0.54. The chance of incorrectly rejecting the null-hypothesis is thus rather large in both cases. Therefore, we do not have to reject the null-hypothesis and can conclude that the data of both the groups are approximately normally distributed. Note that the value and significance of Shapiro-Wilk is also provided by the stat.desc function used above as normtest.W and normtest.p, respectively.

It is always important to use various ways to check your data for normality. Table II.R.2 in Practical 3 of the book provides a rough guideline on how to check for normality with different sample sizes.

Please note that, for linear regression, the assumption of normality applies to the residuals (see Section 6.2 in Part I for details)!

In case of violations of normality, you should opt for a non-parametric alternative. See Sections 5.2.1, 5.3.2 (especially Table I.5.9), and 7.4 (especially Table I.7.9) for an overview of tests that should be used in case one or more assumption of are violated.

2.2 Homogeneity of variance / Homoscedasticity

For mean comparisons, homogeneity of variance is an important assumptions. For correlations and regression analysis, a similar assumption is referred to as homoscedasticity.

2.2.1 Homogeneity of variance

For the dataset with beginner and advanced learners, we want the variation to be similar in both groups and we can use Levene's Test from the "car" package (Fox and Weisberg, 2011) to assess equality of variance (see Section 4.6 and 5.3.2 in Part I for more details on this assumption).

You should use this code after loading the "car" package:

```
leveneTest(DV~IV, data=DataSet)
```

For us, it would now look as follows:

```
leveneTest(Profscore ~ Group, data=data)

## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  1  1.8138 0.1889
##       28
```

The output for our example shows that the chance of incorrectly rejecting the null-hypothesis is relatively large (19%). Therefore, we can assume equal variances in this case. If the test had been significant, we would have to use a Welch's adjustment (see Section 5.3.2 and 7.4 in Part I and Practical 3A, assignment 8a, in Part II for details).

2.2.2 Homoscedasticity

Homoscedasticity can best be assessed by creating a scatterplot and, if preferred, adding a regression line to the plot. Imagine a researcher wants to examine the relationship between

age of acquisition and proficiency score. We could create a scatterplot as in Figure III.R.3 to examine homoscedasticity (see Section 6.4 in Part I for details on the concept of homoscedasticity and on how to assess homoscedasticity from a scatterplot).

```
plot(data$age, data$profsc, xlab="Age of Acquisition",ylab="Proficiency Score", main="Relationship between Proficiency and Age of Acquisition")
abline(lm(data$profsc ~ data$age))
```

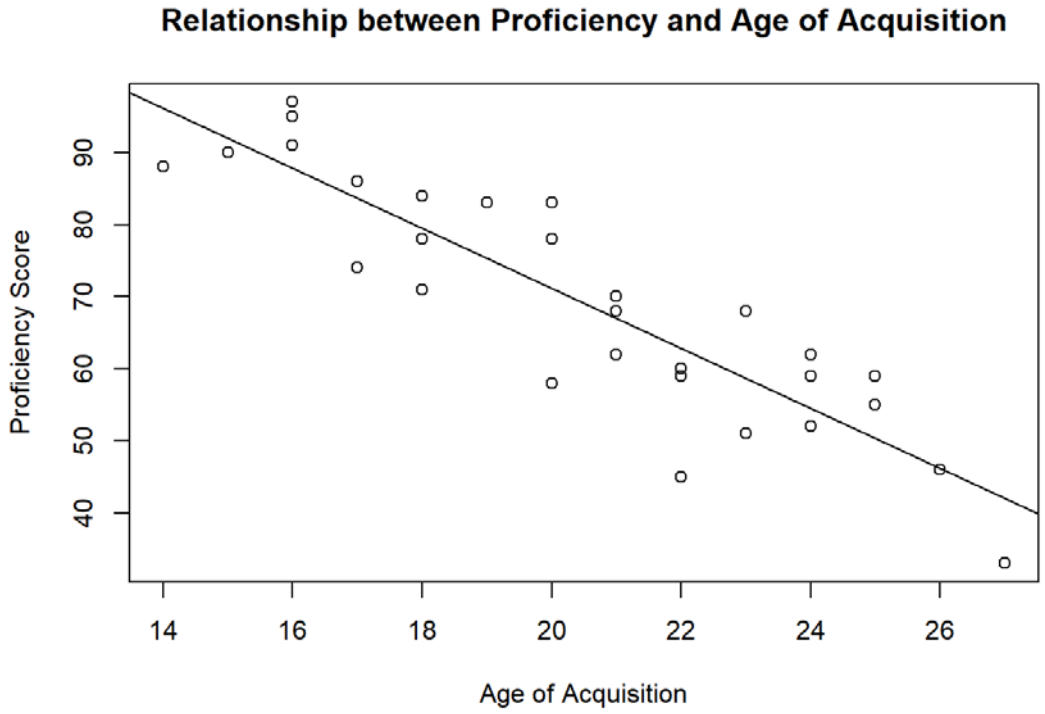

Figure III.R.3 Scatterplot visualising the relationship between age (x-axis) and proficiency score (y-axis)

Figure III.R.3 suggests no problems with respect to homoscedasticity. Note that, for linear regression, the assumption of homoscedasticity applies to the residuals of your model (also see 'How To Do a Regression Analysis')!

For correlations, in case of violations of homoscedasticity, it is often better to opt for a Spearman Rho (ρ) or Kendall's Tau (τ) instead of a Pearson (see Section 5.2.1 in Part I for an explanation of when to use which one).

2.3 Linearity and multicollinearity

For correlations and linear regression, linearity is another important assumption. Figure II.R.3 clearly reveals a linear relationship between age and proficiency score. See section 4.6 and 6.4 in the book for more information about linearity and multicollinearity.

In the case of a relationship that is monotonic, but not linear, it would be better to perform a Spearman or Kendall instead of a Pearson correlation (see Section 4.6 and 5.2.1 in Part I).

3. How To: Correlation Analysis

Please note that there are some important assumptions and prerequisites for both Pearson's *r* (parametric) as well as for Spearman Rho (ρ)and Kendall's Tau (τ) (non- parametric). These assumptions are explained in detail in Section 4.6 and 5.2.1 of Part I and a summary of which assumptions to check for correlations can be found in Table I.8.2.

3.1 Correlations: plotting the data & checking assumptions

For the dataset used in this How To, the main question to be answered is whether there is a relationship between 2 interval variables, age of acquisition (AoA) and French proficiency scores (Score), so we will aim for a Pearson *r* correlation (also see Section 5.2.1 in Part I).

```
##   Learner AoA Score
## 1      17  17    22
## 2      14  14    25
## 3       4   4    26
## 4      11  11    28
## 5      18  18    28
## 6      20  20    30
```

We first want to get to know the data and this is best done by plotting them in a scatterplot using the following base code:

```
plot(File$Variable1, File$Variable2)
```

Figure III.R.4 was created as follows (also see Practical 2, Part A-4a, on how to add titles and labels):

```
plot(data$AoA, data$Score, main="Relationship between Age of Acquisition and Proficiency", xlab="Age of Acquisition", ylab="Proficiency Score")
```

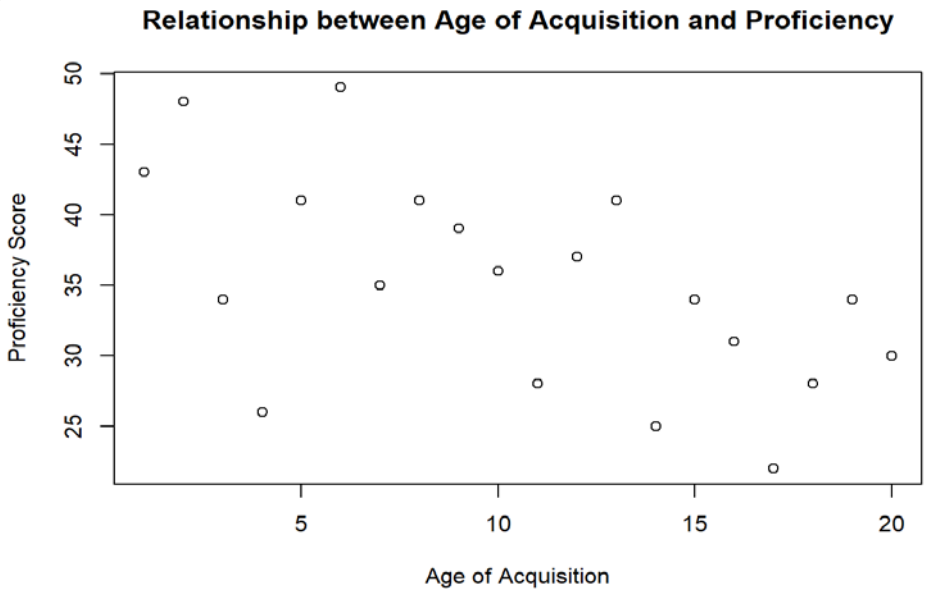

Figure III.R.4 Scatterplot showing the relationship between age of acquisition and score.

Figure III.R.4 shows us that the relationship is linear and homoscedastic (see Section 4.6 and 6.4 for details on these assumptions). To check normality with this particular sample, we'll use stat.desc from the "pastecs" package (Grosjean and Ibanez, 2018). Remember that Table II.R.2 in Practical 3 of the book provides a rough guideline on how to check for normality with different sample sizes.

```
stat.desc(data$AoA, basic=FALSE, norm=TRUE)

##      median        mean     SE.mean CI.mean.0.95         var
##  10.5000000  10.5000000   1.3228757   2.7688106  35.0000000
##     std.dev    coef.var    skewness     skew.2SE     kurtosis
##   5.9160798   0.5634362   0.0000000   0.0000000  -1.3809286
##    kurt.2SE  normtest.W  normtest.p
##  -0.6957635   0.9603752   0.5513717

stat.desc(data$Score, basic=FALSE, norm=TRUE)

##      median        mean     SE.mean CI.mean.0.95         var
##  34.5000000  35.1000000   1.6621166   3.4788500  55.2526316
##     std.dev    coef.var    skewness     skew.2SE     kurtosis
##   7.4332114   0.2117724   0.1284427   0.1254070  -0.9436124
##    kurt.2SE  normtest.W  normtest.p
##  -0.4754272   0.9767212   0.8851450
```

The values of skew.2SE and kurt.2SE are between -1 and 1 for both variables and both Shapiro-Wilks are non-significant, so we can continue to perform a parametric correlation: the Pearson's *r*.

When the data are not normally distributed, you should opt for a non-parametric alternative instead (see Section 5.2.1 for details).

3.2 Correlations: getting the results

To perform a correlation, you can fill in the correct variables in the following code:

```
cor.test(File$Variable1,File$Variable2,method="method of your choosing")
```

For method you can choose "pearson", "spearman" or "kendall" depending on which of these you want to use.

3.3 Correlations: interpreting the output

For our current dataset, we would fill in the following:

```
cor.test(data$AoA, data$Score, method="pearson")
```

This would provide the output in Table III.R.3:

Table III.R.3 R output of Pearson correlation

```
##
##  Pearson's product-moment correlation
##
## data:  data$AoA and data$Score
## t = -2.9593, df = 18, p-value = 0.008396
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
```

```
##  -0.8096404 -0.1734892
## sample estimates:
##        cor
## -0.5720893
```

In this case, we can see that the significance value is p = 0.008 and that our correlation of -0.572 is a moderately strong, negative correlation. For details on how to interpret this output, please see Section 5.2.1 in Part I of the book and Practical 4A, assignment 7, in Part II.

3.4 Correlations: reporting results

You can use the template explained in detail in Practical 4A (assignment 8) and report the results as follows:

A Pearson correlation analysis showed that the age at which one starts learning a foreign language and proficiency were significantly negatively related (r(18) = -0.572, p = 0.008, two-tailed, 95% CI [-0.17, -0.81]). This strong relationship suggests that the later one starts learning a foreign language, the lower their proficiency level will be, as can also be seen in Figure III.R.4.3.5 Correlations: additional useful information to check

The effect size is quite large in this study, which could potentially mean that this is a meaningful effect in terms of power, but we know that we do not have 28 participants. As explained in Practical 5A (assignment 10), we could run a power test using following code from the “pwr” package (Champely, 2018).

```
pwr.r.test(n=20, r=-0.572, sig.level = 0.008)
```

```
##
##      approximate correlation power calculation (arctangh transformation)
##
##              n = 20
##              r = 0.572
##      sig.level = 0.008
##          power = 0.5174634
##    alternative = two.sided
```

For the current study, the power is 0.52 or 52%. How many people would we need to reach the desired power of 80%?

```
pwr.r.test(r=-0.572, sig.level=0.008, power=0.8)
```

```
##
##      approximate correlation power calculation (arctangh transformation)
##
##              n = 31.51233
##              r = 0.572
##      sig.level = 0.008
##          power = 0.8
##    alternative = two.sided
```

With this *r*-value and *p*-value, we would need 32 participants to reach a power of 0.8.

4. How To: Chi-Square Analysis

Chi-square is a non-parametric test for frequency analysis and is used to assess whether there is an association between two nominal/categorical variables. Before you can run a Chi-Square test, remember that you should check the three assumptions that have to be met (see Section 5.2.3 and Practical 4B, assignment 6, for details).

4.1 The Chi-square test: entering the data

Suppose we investigated the colour preference of men and women using a questionnaire that only allowed the participants to choose between red, yellow and blue. The contingency table may look like the one in Table III.R.5.

Table III.R.5 Females and Males and their colour preference

Colour	*Female*	*Male*
Blue	29	42
Red	28	24
Yellow	23	23

We can add these frequencies and the names for the rows and columns to match the table in R using the cbind-function (also see Practical 4B, assignment 5):

```
Table <- cbind(c(29,28,23),c(42,24,23))
```

Now we should manually add the row names and column names using:

```
rownames(Table) <- c("Row1","Row2","Row3")
colnames(Table) <- c("Column1","Column2")
```

For the current dataset, we should fill this in as follows:

```
rownames(Table) <- c("Blue","Red","Yellow")
colnames(Table) <- c("Female","Male")
```

The output of this call should be the exact contingency table shown in Table III.R.5, which we can check just by typing in its name:

```
Table

##        Female Male
## Blue       29   42
## Red        28   24
## Yellow     23   23
```

To use a dataframe in long format (see Practical 4 for details), read it from a file and make sure to turn the variables into factors (also see 'How To do Descriptives').

4.2 The Chi-square test: performing the test and checking assumptions

As explained in Practical 4B (assignment 6), we prefer to use CrossTable from the "gmodels" package (Warnes, Bolker, Lumley, & Johnson, 2018) to perform the Chi-square test. After loading the package, you simply add your Table (or whatever name you gave to it) to the code below:

```
CrossTable(Table, chisq=TRUE, expected=TRUE)
```

If your data would have been formatted with one row for every individual, you would have to use the following code:

```
CrossTable(File$Variable1, File$Variable1)
```

Before looking at the χ^2-value, the assumptions need to be checked. As can be seen in the output in Table III.R.6, the expected values are all above 5 here, so we can continue to interpret the results. Please see Practical 4B, assignment 6, for suggestions on what to do in case this assumption is not met.

4.3 The Chi-square test: interpreting the output

The output obtained for the current dataset can be found in Table III.R.6.

Table III.R.6 Example Chi-Square output

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |              Expected N |
## | Chi-square contribution |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  169
##
##
##              |
##              |    Female |      Male | Row Total |
## -------------|-----------|-----------|-----------|
##         Blue |        29 |        42 |        71 |
##              |    33.609 |    37.391 |           |
##              |     0.632 |     0.568 |           |
##              |     0.408 |     0.592 |     0.420 |
##              |     0.362 |     0.472 |           |
##              |     0.172 |     0.249 |           |
## -------------|-----------|-----------|-----------|
##          Red |        28 |        24 |        52 |
##              |    24.615 |    27.385 |           |
##              |     0.465 |     0.418 |           |
##              |     0.538 |     0.462 |     0.308 |
```

```
##              |     0.350 |     0.270 |           |
##              |     0.166 |     0.142 |           |
## -------------|-----------|-----------|-----------|
##       Yellow |        23 |        23 |        46 |
##              |    21.775 |    24.225 |           |
##              |     0.069 |     0.062 |           |
##              |     0.500 |     0.500 |     0.272 |
##              |     0.287 |     0.258 |           |
##              |     0.136 |     0.136 |           |
## -------------|-----------|-----------|-----------|
## Column Total |        80 |        89 |       169 |
##              |     0.473 |     0.527 |           |
## -------------|-----------|-----------|-----------|
##
##
## Statistics for All Table Factors
##
##
## Pearson's Chi-squared test
## ------------------------------------------------------------
## Chi^2 =  2.214966     d.f. =  2     p =  0.3303895
##
##
##
```

The χ^2-value is 2.215 and the chance of incorrectly rejecting H_0 is 0.33.

Clearly, the effect in the current study did not reach significance. If the result had been significant, we would probably want to continue our analysis and calculate the effect size. Although it is not necessary here, we will show you how you would do this using the assocstats-function from the "vcd" package (Meyer Zeileis, & Hornik., 2017). After loading the package, you simply add the name of your table (here:Table) to the following code.

```
assocstats(Table)
```

```
## Loading required package: grid
```

This will provide the output in Table III.R.7.

Table III.R.7 Example Chi-Square effect size output

```
##                     X^2 df P(> X^2)
## Likelihood Ratio 2.2222  2  0.32919
## Pearson          2.2150  2  0.33039
##
## Phi-Coefficient   : NA
## Contingency Coeff.: 0.114
## Cramer's V        : 0.114
```

The value of Cramer's V, which can be found in the bottom line of Table III.R.7 is 0.114 and confirms that we are dealing with a very small effect (for details on effect sizes for Chi-square and how to interpret them, see Section 5.2.3 and Practical 4B, assignment 8).

4.4 The Chi-square test: reporting the results

Conventionally, the results of the Chi-square we performed here should be reported on as follows:

A Chi-square analysis revealed that the association between gender and colour preference was not significant χ^2 (2, N=169) = 2.22, p = 0.33.

In case of significant results, do not forget to also explicitly report on the direction of the association (e.g. men having a stronger preference for Blue), and add the effect size (see Practical 4B, assignment 9, for details on how to report on the results, and how to create symbols in Markdown).

In your results, always also include the contingency table (with observed values) and a barplot, which can be done using the following code:

```
barplot(Table, col = c("colour1", "colour2"), main = "Title", xlab = "Variable1", ylab="Variable2", beside = TRUE, legend=TRUE)
```

For the current dataset, we would type the code below, which would provide you with the barplot in Figure III.R.5.

```
barplot(Table, col = c("black", "grey", "white"), main = "Bar plot of males and females and their colour preference", xlab = "Gender", ylab="Frequency", beside = TRUE, legend.text = TRUE)
```

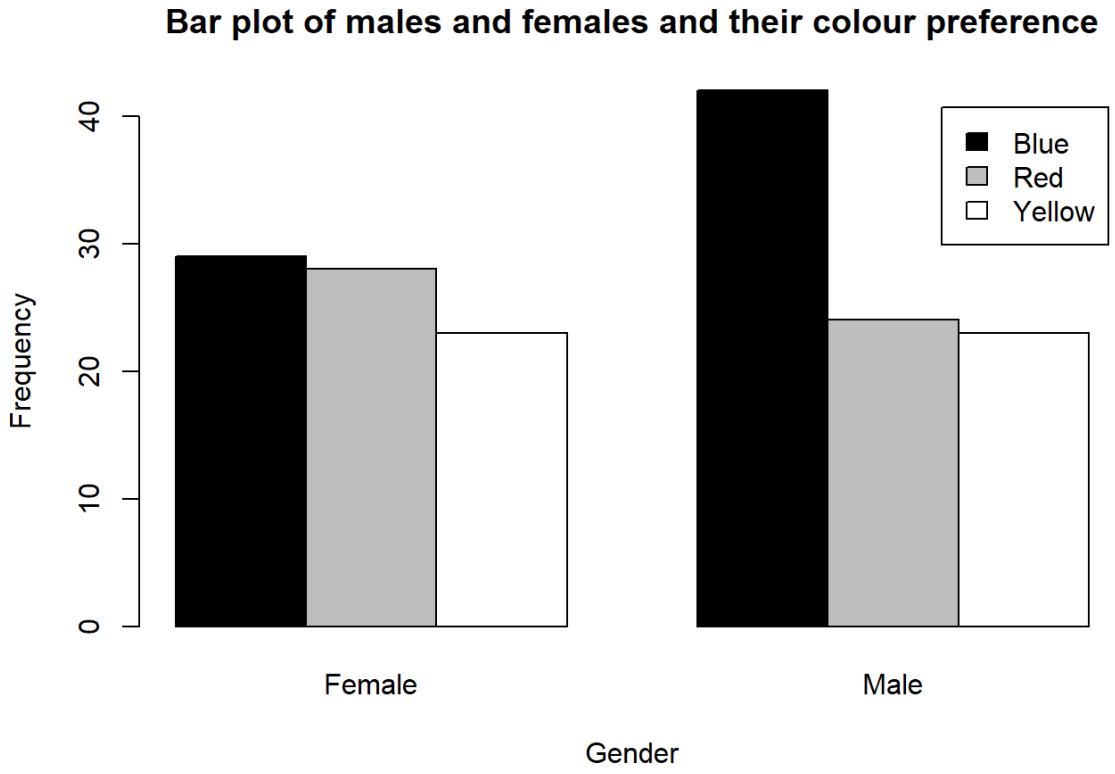

Figure III.R.5: Barplot showing the preference for blue, red, or yellow by females (left) and males (right).

5. How To: *t*-test

As with all parametric tests, please bear in mind the important assumptions and prerequisites for the *t*-test before conducting the actual test (also see Sections 4.6 and 5.3 in Part I). As mentioned in Section 5.3.2, there are several versions of the *t*-test available, all with their own set of assumptions. As an example, we will be demonstrating an *independent samples t-test* here.

5.1 The Independent Samples *t*-test: preparing for the *t*-test

First, load your data into R or enter the data manually (see Practical 4 for detailed instructions). For the present purposes, we will use a dataset comparing test scores of boys and girls:

```
head(data)

##   Subject Group score
## 1       1  boys    27
## 2       2  boys    33
## 3       3  boys    36
## 4       4  boys    25
## 5       5  boys    36
## 6       6  boys    32
```

It's good practice to always start with an inspection of the data by looking at some descriptive statistics for both groups. You could use the aggregate function to calculate each value separately for each group (see Practical 2A, assignment 5a, for details), but you can also create subsets of the data (see Practical 3A, assignment 2a, for details). Subsequently, you could obtain descriptive, for example using the describe function from the psych package (Revelle, 2018: see Practical 2B):

```
boys = subset(data, Group == "boys")
girls = subset(data, Group == "girls")describe(boys)

##         vars  n mean   sd median trimmed  mad min max range skew kurtosis
## Subject    1 20 10.5 5.92   10.5    10.5 7.41   1  20    19  0.0    -1.38
## Group*     2 20  1.0 0.00    1.0     1.0 0.00   1   1     0  NaN      NaN
## score      3 20 32.1 4.55   32.0    32.0 5.93  25  42    17  0.2    -0.85
##           se
## Subject 1.32
## Group*  0.00
## score   1.02

describe(girls)

##         vars  n mean   sd median trimmed  mad min max range skew kurtosis
## Subject    1 20 30.5 5.92   30.5   30.50 7.41  21  40    19 0.00    -1.38
## Group*     2 20  2.0 0.00    2.0    2.00 0.00   2   2     0  NaN      NaN
## score      3 20 37.9 3.18   38.0   37.81 4.45  32  45    13 0.25    -0.61
##           se
## Subject 1.32
```

```
## Group*  0.00
## score   0.71
```

The output already shows that the girls seem to score higher on average (*M*= 37.9;*SD*=3.18) than the boys (*M*= 32.1;*SD*=4.55).

5.2.1 Checking Normality

To check normality, we will look at the skew.2SE and kurt.2SE values that can be obtained using the stat.desc function from the package "pastecs" (Grosjean and Ibanez, 2018). Please note that this is not always the best option for a normality check: Table II.R.2 in Practical 3 of the book provides a guideline on how to check for normality in different circumstances.

```
by(data$score, data$Group, stat.desc, basic=FALSE, norm=TRUE)

## data$Group: boys
##       median         mean      SE.mean CI.mean.0.95          var
##   32.0000000   32.1000000    1.0179960    2.1306900   20.7263158
##      std.dev     coef.var     skewness     skew.2SE     kurtosis
##    4.5526164    0.1418261    0.1959750    0.1913432   -0.8497105
##     kurt.2SE   normtest.W   normtest.p
##   -0.4281160    0.9648240    0.6439865
## ------------------------------------------------------------
## data$Group: girls
##       median         mean      SE.mean CI.mean.0.95          var
##   38.0000000   37.9000000    0.7104483    1.4869855   10.0947368
##      std.dev     coef.var     skewness     skew.2SE     kurtosis
##    3.1772216    0.0838317    0.2456255    0.2398203   -0.6136718
##     kurt.2SE   normtest.W   normtest.p
##   -0.3091908    0.9758011    0.8692778
```

The Shapiro-Wilk tests show that the difference between the distributions of our samples and the normal distribution is not significant. In addition, the values for Skewness and Kurtosis (2SE) are within the allowed range (see Table II.R.2 for details on this interpretation). Please note that you should use a non-parametric version in case of violations of normality (see Practical 4C for details).

5.2.2 Homogeneity of Variance

We will check homogeneity of variance (see Section 4.6 in Part I) using Levene's Test from the "car" package (Fox and Weisberg, 2011):

```
leveneTest(score ~ Group, data=data)

## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value  Pr(>F)
## group  1  2.9231 0.09548 .
##       38
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The significance value for the test of homogeneity of variance for the current dataset is p = 0.09, so we can assume equal variances (see Practical 3A, assignment 7, for details on how to interpret Levene's output).

5.3 The Independent Samples *t*-test: performing the *t*-test

As the assumptions of normality and homogeneity were met for the current dataset, we could use the regular code to obtain our results:

```
t.test(File$DV ~ File$IV, var.equal = TRUE/FALSE)
```

For the current dataset, it would be filled out as follows:

```
t.test(data$score ~ data$Group, var.equal = TRUE)
```

Please note that a different code has to be used in case the data is not formatted with one column for each variable. The difference between long and wide format are discussed in more detail in Practical 3A. where you will also be able to find example codes for the paired t-test and the one-sample t-test.

5.4 The Independent Samples *t*-test: interpreting the output

The above code will provide the output in Table III.R.8

Table III.R.8 Example output of an Independent Samples *t*-test

```
##
##  Two Sample t-test
##
## data:  data$score by data$Group
## t = -4.6722, df = 38, p-value = 3.676e-05
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -8.313066 -3.286934
## sample estimates:
##  mean in group boys mean in group girls
##                32.1                37.9
```

The output again shows that the girls scored higher than the boys and now also reveals that this difference is significant.

For a detailed explanation on how to interpret the output of a t-test, please see Practical 4C assignment 9.

5.5 The Independent Samples *t*-test: calculating the effect size

As explained in detail in Chapter 4, we also want to know the size of the effect. We will use the formula by suggested by Field, Miles, and Field (2012:385) to obtain r^2, but you can also choose to calculate cohen's *d* (see Practical 4C, assignment 11, for details on both these effects sizes and how to calculate them):

```
test <- t.test(data$score ~ data$Group, var.equal = TRUE)
t <- test$statistic[[1]] #df <- test$parameter[[1]]
(r2 <- t^2/(t^2+df))

## [1] 0.364859
```

The effect size is large (see Table I.5.10 for details on how to interpret these effect sizes) and this information, either with the value of r^2 or the value of *d,* should be added to the report.

5.6 The Independent Samples *t*-test: reporting the results

On average, the girls scored higher (*M*= 37.9;*SD*=3.18) than the boys (*M*= 32.1;*SD*=4.55) on an intelligence test. This difference was significant ($t(38) = -4.67$; $p < .001$), 95% CI [-8.3, -3.2] and is also visible in Figure III.R.6. The effect size was large, $r^2 = 0.36$.

A simple boxplot can be made as discussed in 'How To Do Descriptives':

```
boxplot(score ~ Group, data=data, main="Total scores for girls and boys", xlab="Gender", ylab="Total score")
```

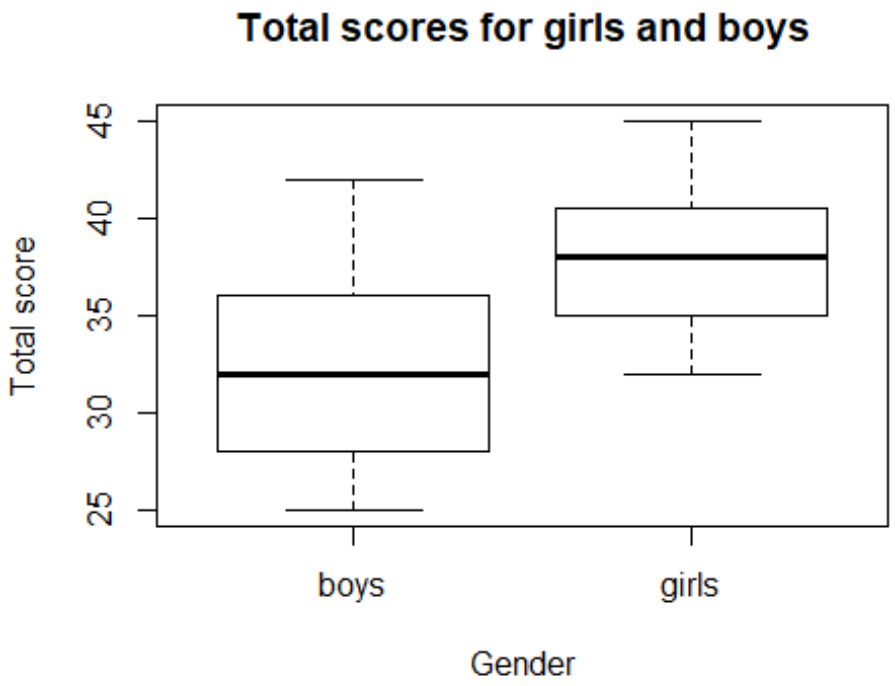

Figure III.R.6: The distribution of the intelligence score for girls and boys.

6. How To: Simple Regression Analysis

Regression is a statistical model used for assessing relationships. More specifically, it helps to predict to which extent one or more independent variables contribute to the value of a dependent variable. Regression has its own particular list of assumptions that has been dealt with in detail in Section 6.4 of Part I. As the assumptions mostly apply to the residuals of the model, they are generally checked after fitting the model.

6.1 Simple Regression: opening and inspecting the data

Imagine a researcher is interested in the impact of age on the results of a simple lexical decision task. The data look as follows:

```
str(RT)

## 'data.frame':    82 obs. of  3 variables:
##  $ Participant: Factor w/ 82 levels "1","2","3","4",..: 1 2 3 13 7 8 9 10 11 12 ...
##  $ Age        : int  31 25 28 55 35 22 37 22 23 48 ...
##  $ RT         : int  501 509 509 550 558 561 587 591 599 620 ...
```

To explore the data and the potential relationship between the variables, it would be helpful to create a scatterplot. Here, we will use the qplot function from the "ggplot2" package (Wickham, 2016, also see Practical 5C, assignment 4)

```
qplot(Age, RT, data = RT)
```

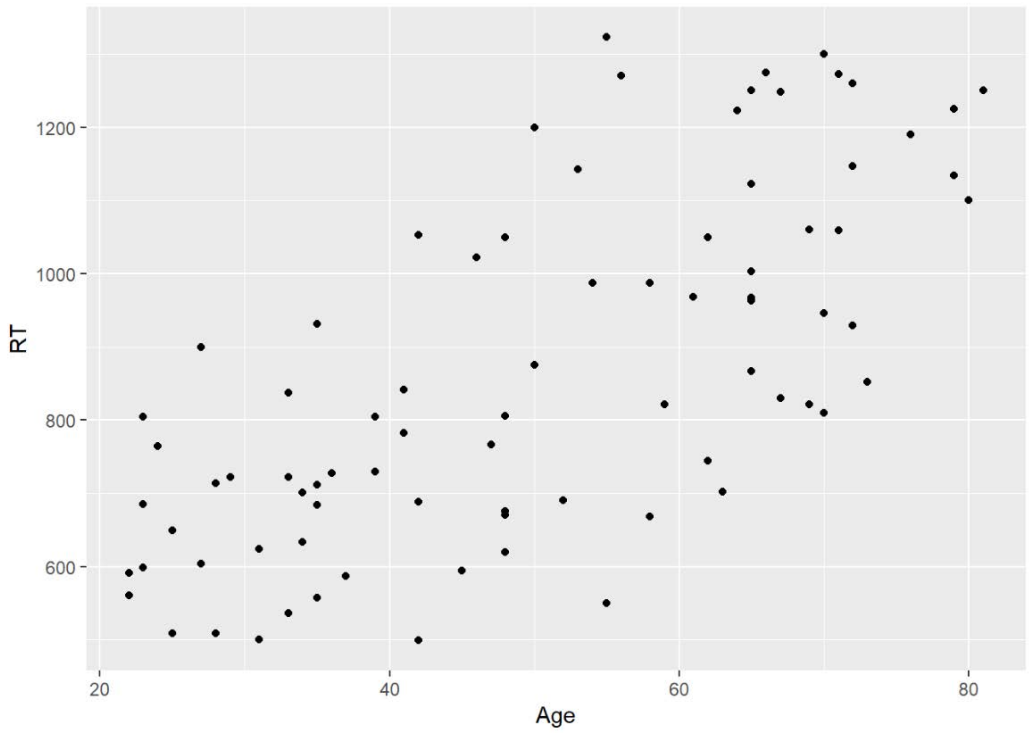

Figure III.R.7: Scatterplot showing the relationship between age (x-axis) and RT (y-axis).

The pattern in Figure III.R.7 reveals that there seems to be a linear positive relationship between age and RT in the lexical decision task.

6.2 Simple Regression: fitting the model and interpreting the output

A simple regression model can be built by using the following code:

```
ModelName = lm(DV~IV, data=Filename)
```

We will fill in our DV (RT) and IV (Age) to build our first linear model (lm1):

```
lm1 = lm(RT~Age, data =RT)
```

Remember that we will have to ask for a summary() of the model to get the results (see Practical 5B, assignment 6).

Table III.R.9: Output of a simple linear regression model in which RT is modelled by Age.

```
##
## Call:
## lm(formula = RT ~ Age, data = RT)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -363.84 -140.36   -6.70   84.66  409.16
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  383.129     57.686   6.642 3.43e-09 ***
## Age            9.649      1.089   8.864 1.62e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 170.3 on 80 degrees of freedom
## Multiple R-squared:  0.4955, Adjusted R-squared:  0.4892
## F-statistic: 78.57 on 1 and 80 DF,  p-value: 1.625e-13
```

The summary reveals a significant effect of age: for every year added to age, the RT slows down with 9.6 ms. The Multiple R-squared reveals that about 50% of the variance can be explained by our model, which is quite good.

For more details on how to interpret the output, including all other values, please see Section 6.2 and, in particular, the explanation of Table I.6.3 in Part I of the book.

6.3 Simple Regression: checking assumptions

6.3.1 Linearity and Homoscedasticity

The best way to check linearity and homoscedasticity is by creating a residuals plot by filling in the name of your model ('ModelName') in the following line of code:

```
plot(fitted(ModelName), residuals(ModelName))
```

The below line is the filled out version for the current example and this code provides you with a residuals plot like the one in Figure III.R.8.

```
plot(fitted(lm1), residuals(lm1))
```

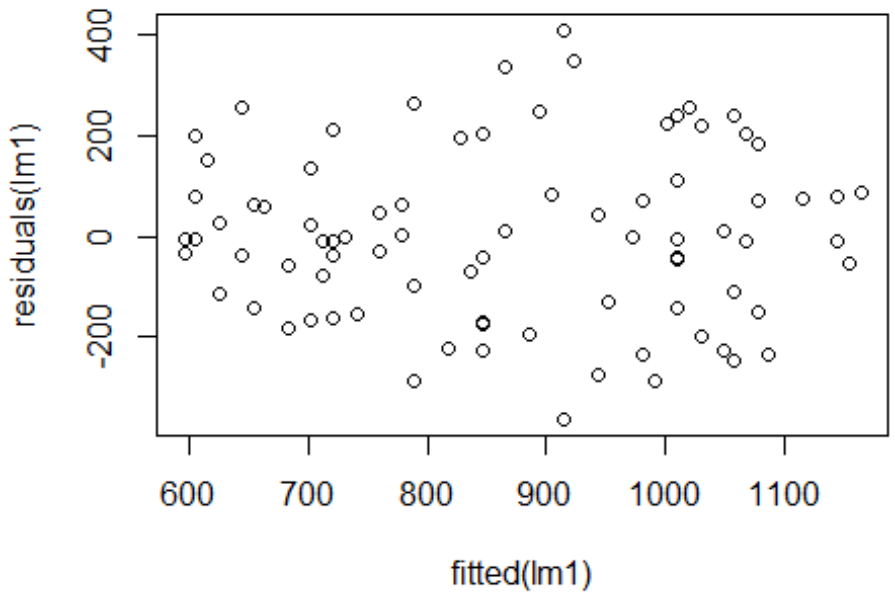

Figure III.R.8: Scatterplot showing the residuals and their deviations from the fitted values.

Figure III.R.8 confirms that the relationship between age and RT is linear (as could also be seen in Figure III.R.7) and the residuals do not reveal any signs of heteroscedasticity (see Section 6.2 and Practical 5B, assignment 9b, for an explanation on how to interpret residual plots). To be more certain about this second assumption, we could do the non Constant Variance error test from the "car" package (Fox and Weisberg, 2011). Here, the test indeed shows that homoscedasticity can be assumed:

```
ncvTest(lm1)

## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 1.565204, Df = 1, p = 0.2109
```

6.3.2 Normality of the Residuals

To assess whether the model's residuals are approximately normally distributed, one can choose to create a histogram or a quantile-quantile plot (a Q-Q plot) of the residuals. The latter can be done as follows:

```
qqnorm(residuals(ModelName))
```

The filled out version below provides you with a Q-Q plot like the one in Figure III.R.9.

```
qqnorm(residuals(lm1))
```

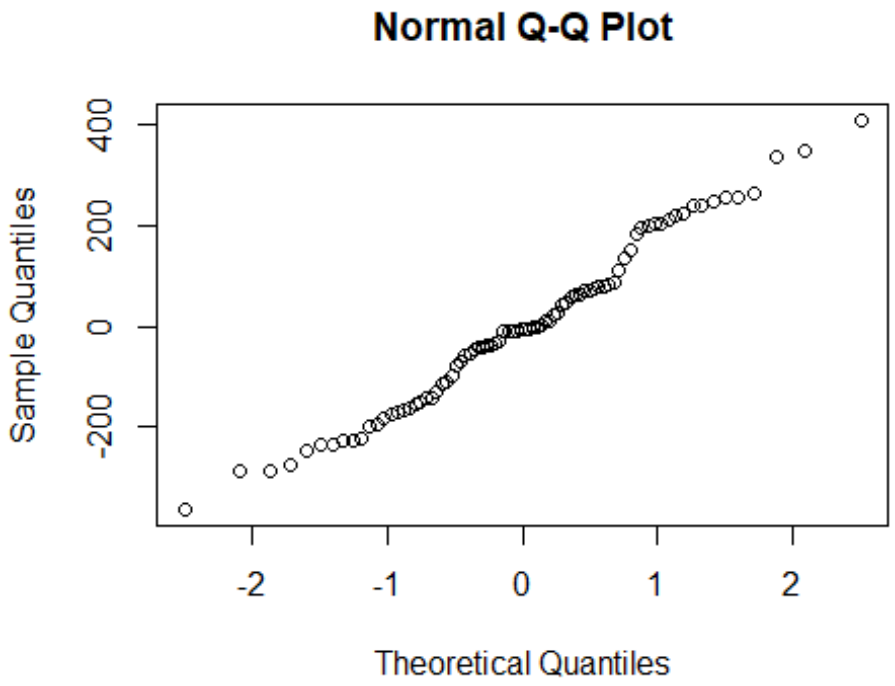

Figure III.R.9: A Q-Q plot of the residuals of our model.

The qq plot in Figure III.R.9 indicates that the residuals approximate the normal distribution (see Practical 5B, assignment 9c, for an explanation on how to interpret Q-Q plots).

Note that the same conclusion would have been on the basis of a Shapiro-Wilk on the residuals:

```
shapiro.test(residuals(lm1))

##
##  Shapiro-Wilk normality test
##
## data:  residuals(lm1)
## W = 0.98305, p-value = 0.353
```

6.4 Simple Regression: reporting the results

We suggest to report the unstandardized coefficients and all associated values in the output in Table III.R.9 in a table, but include information on the model, such as the R^2 value, in your report (see Practical 5B, assignment 10, for details):

We constructed a linear model of reaction time as a function of age. This model was significant ($F(1,80)=78.57$, $p<.001$) and explained 50% of the variance in the data (multiple R-squared). Regression coefficients are shown in Table III.R.10. The positive coefficient for age reveals that, as age increases the reaction times also increase significantly. To be precise, for every added year in age, the RT increases with 9.65 milliseconds.

Table III.R.10: Regression coefficients for the linear model of reaction times as a function of age.

	Estimate	**SE**	***t*-value**	***p*-value**
Intercept	383.129	57.686	6.642	3.43e-09 ***

Age	9.649	1.089	8.864	1.62e-13 ***

To support your conclusion, it is common practice to also add a (reference to a) figure visualizing the effect found.

7. How To: Multiple Regression Analysis

In Part I, Section 6.3, we mostly discussed multiple regression with continuous predictor variables and, for this How To, we will add an example in which we have one continuous and one categorical predictor variable as this is a relatively common design in linguistic research. To avoid unnecessary complications, we will use the same dataset as we used in 'How To Do a Simple Regression Analysis' with the only difference of an added nominal/categorical independent variable. To understand this How To, we suggest you first read Chapter 6 and make sure you understand 'How To Do a Simple Regression Analysis'.

7.1 Multiple Regression: opening and inspecting the data

The dataset below shows that one variable has been added to the data we used in 'How to Do a Simple Regression Analysis': Frequency. Here, this refers to word frequency with half of the participants responding to high frequency words and half of the participants responding to low frequency words.

```
str(RT)

## 'data.frame':    82 obs. of  4 variables:
##  $ Participant: Factor w/ 82 levels "1","2","3","4",..: 1 2 3 13 7 8 9 10 11 12 ...
##  $ Age        : int  31 25 28 55 35 22 37 22 23 48 ...
##  $ Frequency  : Factor w/ 2 levels "high","low": 1 1 1 1 1 1 1 1 1 1 ...
##  $ RT         : int  501 509 509 550 558 561 587 591 599 620 ...
```

As explained in Practical 5C, assignment 4, situations involving multiple independent or predictor variables benefit from more detailed visualisations. Especially to be able to visualize potential interactions (see Chapter 7 for details), we will use the qplot function from the "ggplot2" package (Wickham, 2016) to create the informative scatterplot below (see Practical 5C, assignment 4, for details):

```
qplot(Variable x-axis, Variable y-axis, colour = IV2, data = FileName)
```

If we fill in the correct information, we will obtain the plot in Figure III.R.10.

```
qplot(Age, RT, colour = Frequency, data = RT)
```

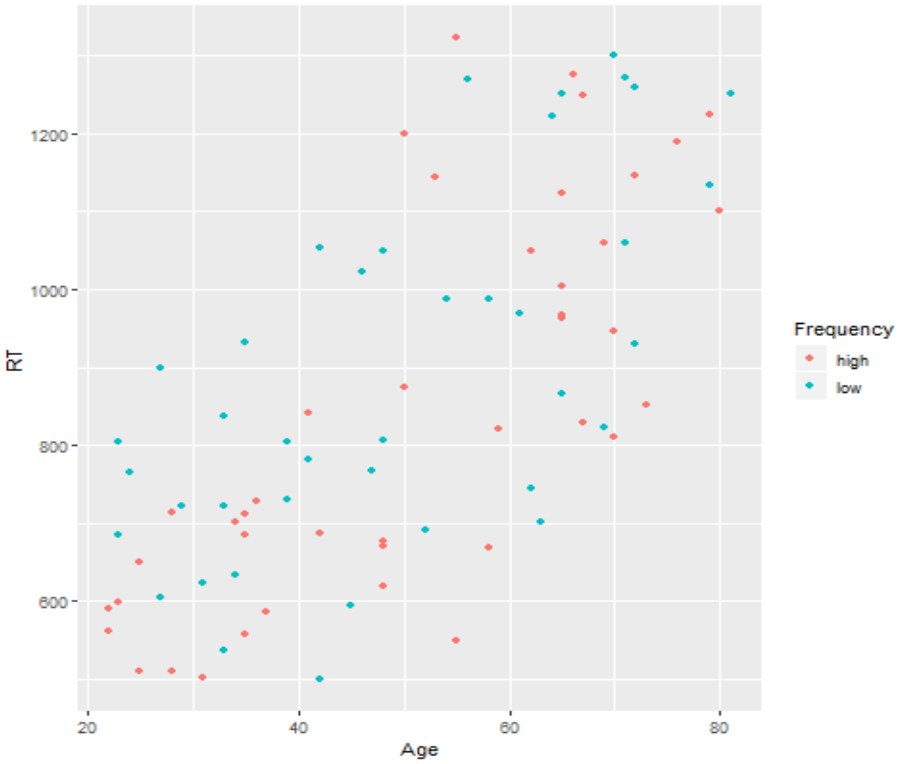

Figure III.R.10: Scatterplot showing the relationship between age (x-axis) and RT (y-axis) with separated colours for high and low frequency words.

The pattern in Figure III.R.10 suggests a linear positive relationship between age and RT, but no clear effect of word frequency.

7.2 Multiple Regression: building regression models

We will first build a model with main effects only::

```
ModelName = lm(DV~IV1 + IV2, data=Filename)
```

We will add our variables RT (DV), Age (IV1) and Frequency (IV2) as in the code below and use summary() to obtain the output in Table III.R.11.

```
lm2 = lm(RT~Age + Frequency, data =RT)
```

Table III.R.11 R output for the multiple regression model with main effects only

```
##
## Call:
## lm(formula = RT ~ Age + Frequency, data = RT)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -335.22 -111.91    4.91  104.85  437.78
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   350.509     60.697   5.775 1.45e-07 ***
## Age             9.722      1.079   9.008 9.37e-14 ***
## Frequencylow   59.413     37.294   1.593    0.115
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 168.7 on 79 degrees of freedom
## Multiple R-squared:  0.5112, Adjusted R-squared:  0.4988
## F-statistic: 41.31 on 2 and 79 DF,  p-value: 5.255e-13
```

There is a significant positive effect of age (p < .001: the older, the higher the RT), but no effect of frequency (p = .115). As you would expect, the RT's in response to low frequency words are slower, but not significantly slower.

For details on how to interpret the output of a multiple regression, see Chapter 6 in Part I, Sections 6.2-6.4. For an explanation on how to interpret the effect of a categorical predictor in a regression model, please see Practical 5C, assignment 7.

To assess whether there is a combined effect of age and word frequency, can use the following code (see Practical 5C, assignment 8, for details):

```
lm3 = lm(RT~Age*Frequency, data =RT)
```

Using summary(lm3) will provide the output in Table III.R.12.

Table III.R.12 R output for the Multiple Regression model with both main and interaction effects

```
summary(lm3)

##
## Call:
## lm(formula = RT ~ Age * Frequency, data = RT)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -338.52 -111.64    8.02   97.23  434.48
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       310.440     79.807   3.890  0.00021 ***
## Age                10.511      1.484   7.081 5.53e-10 ***
## Frequencylow      143.593    114.744   1.251  0.21452
## Age:Frequencylow   -1.682      2.168  -0.776  0.44011
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 169.1 on 78 degrees of freedom
## Multiple R-squared:  0.5149, Adjusted R-squared:  0.4963
## F-statistic:  27.6 on 3 and 78 DF,  p-value: 2.876e-12
```

Table III.R.12 shows again that there is an effect of age and no effect of word frequency. It additionally shows that there is no significant interaction between age and word frequency.

It often helps to visualize potential interactions using the "visreg" package (Breheny and Burchett, 2017) (also see Practical 5C, assignment 8):

```
visreg(ModelName, xvar=IV1, by=IV2)
```

For the current dataset, this would become:

```
visreg(lm3,xvar="Age", by="Frequency")
```

This will provide the output in Figure III.R.11.

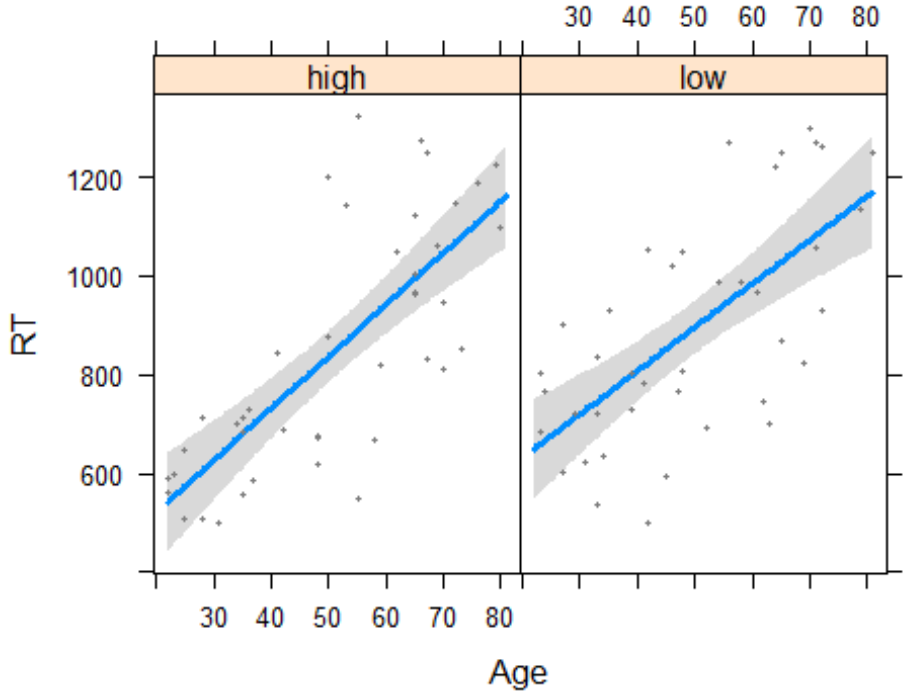

Figure III.R.11: Plot of the slopes of the effect of age for high frequent (left) and low frequent (right) words.

The effect, i.e. the slope, of age looks very much the same for both high and low frequency words, confirming the absence of an interaction.

7.3 Multiple Regression: comparing regression models

As explained in Practical 5C, assignment 11, we can also use ANOVAs to compare models that differ only with respect to one added variable (also see e.g. Baayen, 2008, Chapter 6, p. 183 onwards). Our multiple regression analysis revealed no significant effect of word frequency, but a significant effect of age. Let us compare our initial simple regression model (Table III.R.9) to the model in which we have added frequency (Table III.R.11).

```
anova(lm1, lm2)

## Analysis of Variance Table
##
## Model 1: RT ~ Age
## Model 2: RT ~ Age + Frequency
##   Res.Df     RSS Df Sum of Sq     F Pr(>F)
## 1     80 2319316
## 2     79 2247124  1     72191 2.538 0.1151
```

The results of the ANOVA shows that the RSS, the residual sum of squares, is lower for the second model, but that it does not constitute a significant improvement (p = 0.115). Hence, it is better to stick with the simpler model with only the main significant effect of age (see Practical 5C, assignment 11, for details on how to interpret such a comparison).

7.4 Multiple Regression: checking assumptions

Even though we did not find an effect of or interaction with word frequency, we will use our second model as an example to check assumptions. Note that, except for multicollinearity, the assumptions and how to check them is identical to those of a simple regression and we will go through the steps relatively quickly here. For a detailed explanation of the assumptions and how to check them, see Section 6.4 and Practical 5B and 5C.

We will first make a residuals plot to obtain Figure III.R.12.

```
plot(fitted(lm2), residuals(lm2))
```

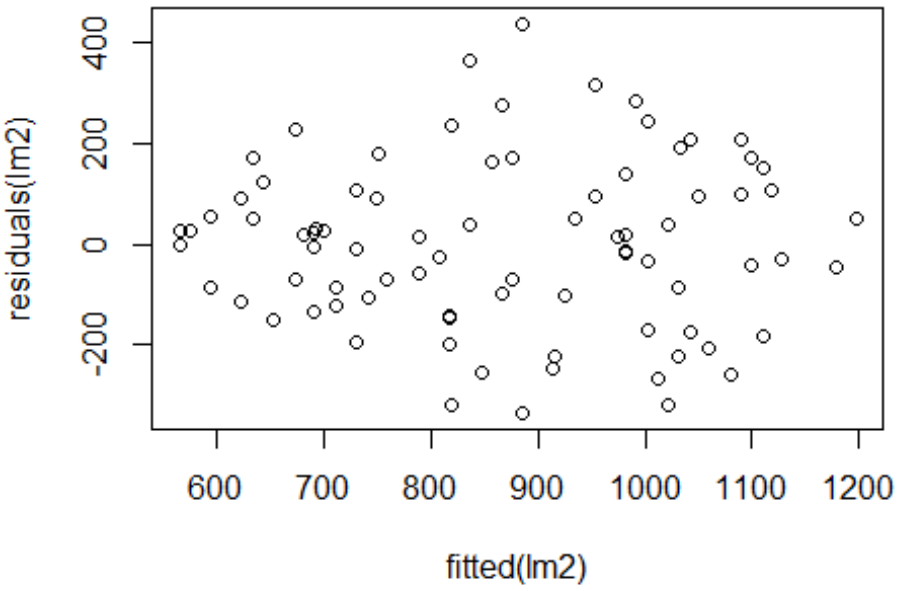

Figure III.R.12: Scatterplot showing the residuals and their deviations from the fitted values.

The relationship between age and RT is linear (also see Figure III.R.10 and III.R.11) and the residuals plot in Figure III.R.12 does not reveal any strong signs of heteroscedasticity. To be sure, we can do the NCV error test (see Practical 5C, assignment 9, for details).

```
ncvTest(lm2)

## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 2.318025, Df = 1, p = 0.12788
```

The test confirms that we can assume homoscedasticity.

```
qqnorm(residuals(lm2))
```

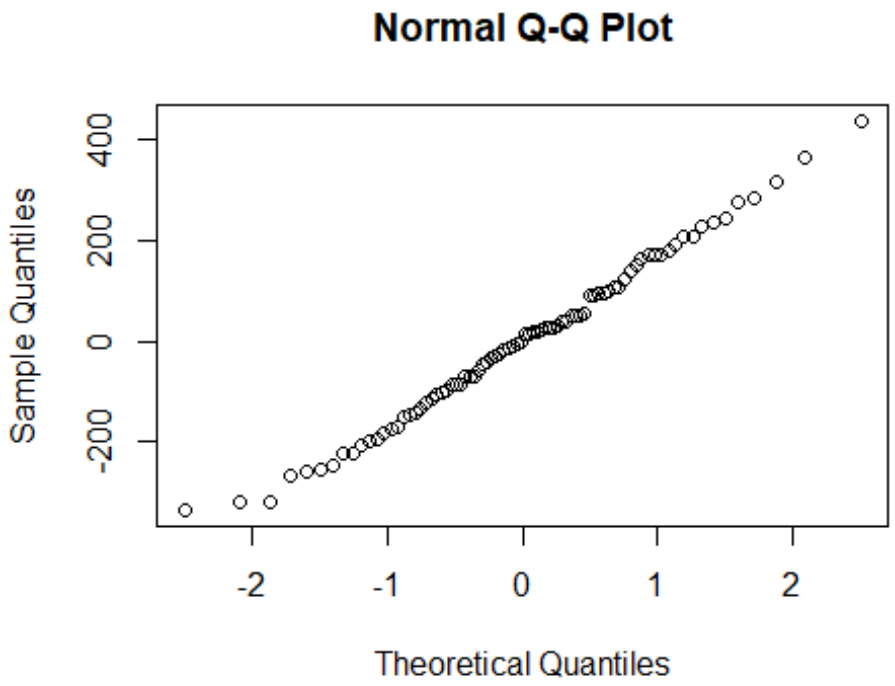

Figure III.R.13: A qqplot of the residuals of the model containing both main effects.

The qqplot in Figure III.R.13 indicates that the data approximate the normal distribution and this interpretation is confirmed by a Shapiro-Wilk:

```
shapiro.test(residuals(lm2))

##
##  Shapiro-Wilk normality test
##
## data:  residuals(lm2)
## W = 0.99189, p-value = 0.8927
```

Based on common sense, we would not expect any correlations between age and word frequency, but this can also be checked using a *t*-test (also see 'How To Do a *t*-test'). See Practical 5C, assignment 12d, for details on why to use a *t*-test in this particular case as well as which other tests to use in case your variables were measured on a different scale:

```
t.test(RT$Age~RT$Frequency)

##
##  Welch Two Sample t-test
##
## data:  RT$Age by RT$Frequency
## t = 0.37846, df = 79.989, p-value = 0.7061
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -6.215129  9.134176
## sample estimates:
## mean in group high  mean in group low
##           50.80952           49.35000
```

The *t*-test confirms suggests that collinearity is not a problem in this dataset. We could double-check this using Variance Inflation Errors by using the following code from the car package (Fox and Weisberg, 2011):

```
car::vif(RT.lm)
```

As explained in Practical 5C, assignment 12d, the outcome reveals no collinearity issues in this particular dataset.

7.5 Multiple Regression: reporting the results

Note that we would have probably opted for the simple model as frequency did not significantly impact RT. For the sake of clarity, we will show a sample report for the multiple regression model that you can use as a basis for your reports. All values in the report below can be found in the output in III.R.11 (see Practical 5B, assignment 10, for details).

We constructed a linear model of reaction time as a function of age and word frequency. This model was significant ($F(2,79)=41.31$, $p<.001$) and explained 50% of the variance in the data (adjusted R-squared). Regression coefficients are shown in Table III.R.13. The positive coefficient for age reveals that, as age increases the reaction times also increase significantly. This pattern can also be seen in Figure III.R.10. The positive coefficient for low frequency words indicates a higher, though not significantly higher, reaction time in response to low frequency words as compared to high frequency words.

Table III.R.13: Regression coefficients for the linear model of reaction times as a function of age and word frequency.

	Estimate	**SE**	***t*-value**	***p*-value**
Intercept	350.509	60.697	5.775	1.45e-07 ***
Age	9.722	1.079	9.008	9.37e-14 ***
Frequency = low	59.413	37.294	1.593	0.115

8. How To: one-way ANOVA

To show you how to perform a one-way ANOVA in R, we will be comparing three age groups and their proficiency score: younger (11-30 yrs), adult (31-50), and older (51-70) people. The important assumptions and prerequisites for the one-way ANOVA are explained in Sections 4.6 and 7.2.

8.1 One-way ANOVA: opening and inspecting the data

The dataset we will be using is the following:

```
## 'data.frame':    60 obs. of  3 variables:
##  $ Subject : Factor w/ 60 levels "subject 1","subject 10",..: 16 10 20 4 33 57 22 58 5 47 ...
##  $ AgeGroup: Factor w/ 3 levels "young","adult",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Score   : int  57 61 63 66 66 67 68 71 76 77 ...
```

8.2 One-way ANOVA: checking assumptions

As the group sizes are smaller than 25 (see Table II.R.2 in Part II for details), we will focus on the outcomes of the stat.desc function from the "pastecs" package (Grosjean and Ibanez, 2018).

```
by(prof$Score, prof$AgeGroup, stat.desc, basic=FALSE, norm=TRUE)

## prof$AgeGroup: young
##       median         mean      SE.mean CI.mean.0.95          var
##  77.50000000  76.35000000   2.40643326   5.03672271 115.81842105
##      std.dev     coef.var     skewness     skew.2SE     kurtosis
##  10.76189672   0.14095477  -0.20700930  -0.20211673  -1.48818873
##     kurt.2SE   normtest.W   normtest.p
##  -0.74980517   0.91075083   0.06587081
## --------------------------------------------------------
## prof$AgeGroup: adult
##       median         mean      SE.mean CI.mean.0.95          var
##   67.5000000   66.3500000    2.1900192    4.5837629   95.9236842
##      std.dev     coef.var     skewness     skew.2SE     kurtosis
##    9.7940637    0.1476121   -0.1334530   -0.1302989   -1.4572504
##     kurt.2SE   normtest.W   normtest.p
##   -0.7342173    0.9368918    0.2093169
## --------------------------------------------------------
## prof$AgeGroup: old
##       median         mean      SE.mean CI.mean.0.95          var
##   56.5000000   56.1000000    3.2313432    6.7632791  208.8315789
##      std.dev     coef.var     skewness     skew.2SE     kurtosis
##   14.4510062    0.2575937   -0.2810360   -0.2743939   -0.6722751
##     kurt.2SE   normtest.W   normtest.p
##   -0.3387174    0.9378816    0.2185985
```

The data of the first group do not entirely approximate normality ($W = 0.91$, $p = 0.066$). Although the p-value is not below 0.05, the smartest decision might be to perform a non-parametric test anyway, which would be a Kruskal-Wallis (see Practical 6B, assignment 9, for details on how to perform this Kruskal-Wallis test).

Still, an ANOVA can handle slight deviations from normality pretty well, especially if the design is balanced, so we will - for the purposes of this How To - continue with the ANOVA. Before doing so, we will check homogeneity using leveneTest from the "car" package (Fox and Weisberg, 2011).

```
leveneTest(Score ~ AgeGroup, data=prof)

## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  2     0.4 0.6722
##       57
```

Levene's Test shows that we can continue with the parametric one-way ANOVA (see Practical 3A, assignment 7, for details on how to interpret Levene's output).

In case of violations of homogeneity, you can perform Welch's ANOVA (see Practical 6B, assignment 9, for details).

8.3 One-way ANOVA: obtaining descriptives and creating a boxplot

Let us briefly look at some descriptives using describe from the "psych" package (Revelle, 2018) before continuing:

```
by(FileName$DV, FileName$IV, describe)
```

Filling in the correct variables will provide the following output:

```
by(prof$Score, prof$AgeGroup, describe)

## prof$AgeGroup: young
##    vars  n  mean    sd median trimmed   mad min max range  skew kurtosis
## X1    1 20 76.35 10.76   77.5   76.88 15.57  57  90    33 -0.21    -1.49
##      se
## X1 2.41
## ------------------------------------------------------------
## prof$AgeGroup: adult
##    vars  n  mean   sd median trimmed  mad min max range  skew kurtosis
## X1    1 20 66.35 9.79   67.5   66.56 12.6  49  81    32 -0.13    -1.46
##      se
## X1 2.19
## ------------------------------------------------------------
## prof$AgeGroup: old
##    vars  n mean    sd median trimmed   mad min max range  skew kurtosis
## X1    1 20 56.1 14.45   56.5   56.69 11.12  28  79    51 -0.28    -0.67
##      se
## X1 3.23
```

Let's put this in a table (Table III.R.14) and create a boxplot (Figure III.R.14) to get to know the data better (see Practical 2C, assignment 2a, and Practical 2A, assignment 5d for more information on how to create tables and boxplots, respectively).

Table III.R.14: Descriptives for the three age groups.

	Young	*Adult*	*Old*
mean	76.35	66.35	56.1

minimum	57	49	28
maximum	90	81	79
sd	10.76	9.79	14.45

```
boxplot(Score~AgeGroup, data=prof,  xlab="Age Group", ylab="Score", main="Boxplot Scores different Age Groups",col=c("darkgrey", "lightgrey", "white"))
```

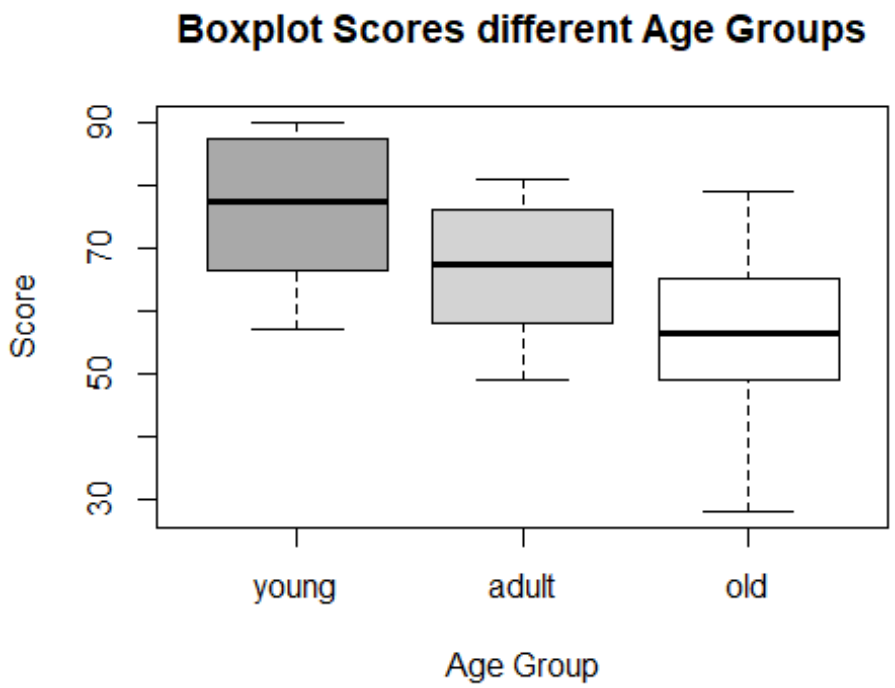

Figure III.R.14: Boxplot with the dispersion of scores for young, adult, and older people.

8.4 One-way ANOVA: performing the test and interpreting the output

You can use the function aov()to conduct the test:

```
aov(DV ~ IV, data=Yourdataset)
```

As explained in Practical 6B, assignment 9, it is good to store the results of the test in a variable (here: m1). As with a regression model, the output of the model is obtained by using summary(ModelName):

```
m1 = aov(Score ~ AgeGroup, data=prof)
summary(m1)
```

This will provide the output in Table III.R.15.

Table III.R.15 R Output of a one-way ANOVA

```
##              Df Sum Sq Mean Sq F value   Pr(>F)
## AgeGroup      2   4101  2050.4   14.63 7.47e-06 ***
## Residuals    57   7991   140.2
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

In this case, the results show that there are significant differences between the three different groups (see Section 7.2 for details on how to interpret ANOVA output).

Of course, we still need to conduct a post-hoc test and we will opt for the most well-known Tukey test (see Section 7.2 in Part I for more details on what all these value mean and how to interpret them). The Tukey is performed by adding your model's name to the code `TukeyHSD()` as we have done below:

```
TukeyHSD(m1)

##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = Score ~ AgeGroup, data = prof)
##
## $AgeGroup
##                 diff       lwr        upr     p adj
## adult-young -10.00 -19.01014  -0.9898605 0.0262373
## old-young   -20.25 -29.26014 -11.2398605 0.0000039
## old-adult   -10.25 -19.26014  -1.2398605 0.0221204
```

The results indicate that the mean proficiency of the adult and young differs significantly (p = .026), that the old and young group differ significantly (p < .001), and also that the old and adult group differ significantly (p = .022).

8.5 One-way ANOVA: calculating the effect size

To get the effect size r^2 or η^2 (eta-squared) for ANOVA (see Section 7.5 of Part I), we have two options. First of all, you can calculate it manually by looking at the output and using the equation to calculate the effect size that was also discussed in Section 7.5 of Part I.

```
summary(m1)

##             Df Sum Sq Mean Sq F value   Pr(>F)
## AgeGroup     2   4101  2050.4   14.63 7.47e-06 ***
## Residuals   57   7991   140.2
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Based on the values in the output (see Section 7.5 for a detailed explanation), we fill out the following equation:

```
4101/(4101+7991)

## [1] 0.3391499
```

The second way to calculate the effect size is by using the variable in which we stored the results of our ANOVA-test and ask R for the corresponding value of r^2:

```
summary.lm(m1)$r.squared

## [1] 0.3391435
```

Whichever option you prefer, your results will show a large effect size of f^2 or η^2 equalling .34.

8.6 One-way ANOVA: reporting the results

You can use the sample format discussed in Practical 6B, assignment 12, to report the results:

There was a large significant effect of age group on proficiency scores, $F(2, 57) = 14.63$, $p < 0.001$, $\eta^2 = .34$. A Tukey post hoc analysis revealed that the youngest group performed significantly better ($M = 76.35$, $SD = 10.76$) as compared to both the middle ($M = 66.35$, $SD = 9.79$), $p = 0.03$, 95% CI [-19, -0.99], and the oldest group ($M = 56.1$, $SD = 14.45$), $p < 0.001$, 95% CI [-29.3, -11.2]. The middle group also did significantly better than the oldest group at $p = 0.02$, 95% CI [-19.3, -1.2]. This effect is also illustrated in Figure III.R.14.

8.7 One-way ANOVA: additional things that might be worth checking

We can additionally check the power of our experiment (also see Section 4.7). As explained in Practical 6B, assignment 10, we need the cohen's *f* value that we can obtain with the "sjstats" package (Lüdecke, 2018) and then use that value to perform a power analysis using the "pwr" package (Champely, 2018).

```
cohens_f(m1)

##       term  cohens.f
## 1 AgeGroup 0.7163714

pwr.anova.test(k=3, n=20, f=0.72, sig.level=7.47e-06)

##
##      Balanced one-way analysis of variance power calculation
##
##               k = 3
##               n = 20
##               f = 0.72
##       sig.level = 7.47e-06
##           power = 0.6005543
##
## NOTE: n is number in each group
```

The power is 0.6, which is not as high as we would like it to be. How many people would we need to obtain the desired power of 80%?

```
pwr.anova.test(k=3, f=0.72, sig.level=7.47e-06, power=0.8)

##
##      Balanced one-way analysis of variance power calculation
##
##               k = 3
##               n = 24.08511
##               f = 0.72
##       sig.level = 7.47e-06
##           power = 0.8
```

```
##
## NOTE: n is number in each group
```

Groups of 25 each would already be enough.

It should also be noted here that is seems rather strange that the researcher decided to put the variable age in (rather arbitrary?) groups instead of using a measure of age as a continuous/interval variable. Also, please remember that a simple regression with one nominal variable could also have answered our question!

9. How To: Factorial ANOVA

This How To will show an example of a factorial ANOVA with two independent nominal variables, so the test we will perform is a two-way ANOVA. The assumptions are the same as those for the one-way ANOVA, but note that (as also discussed in Practical 6C, assignment 6) you should compare and check the distribution in each group or combination of groups.

As there is a lot of overlap between a one-way and a factorial ANOVA, it is advisable to read Chapter 7 in Part I and go through the 'How To Do a One-Way ANOVA' first.

9.1 Two-way ANOVA: opening and inspecting the data

We will use example data from practical 7(f), where the question was whether encouragement (yes or no) and/or gender (boy or girl) impacted the number of marbles (interval variable) toddler's put in a vase. Remember that we are not only testing a pair of hypotheses for every IV, but also for their combined effect (also see Practical 6C, assignment 3)!

The file we will be using has the following format:

```
## 'data.frame':    56 obs. of  4 variables:
##  $ Child    : Factor w/ 56 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ Group    : Factor w/ 2 levels "enc","noenc": 1 1 1 1 1 1 1 1 1 1 ...
##  $ NrMarbles: int  8 2 8 4 3 1 9 0 4 8 ...
##  $ Gender   : Factor w/ 2 levels "b","g": 1 1 2 1 2 1 2 2 1 2 ...
```

9.2 Two-way ANOVA: obtaining descriptives and checking assumptions

To plot the data, you can opt for a simple boxplot (see 'How To do Descriptives') and add two IV's as in the following code:

```
boxplot(DataFile$DV ~ DataFile$IV1 + DataFile$IV2)
```

Here we will create a nice barplot with the package "ggplot2" (Wickham, 2016).

Now, you can fill in the correct names and use the following function:

```
ggplot(FileName, aes(IV1, DV, fill = IV2)) +
  geom_bar(stat = 'identity', position = 'dodge')
```

Of course, you can also opt for a boxplot, change colours, add titles, change angles and much more! Our code and the plot it creates are to be found below in Figure III.R.15.

```
ggplot(data, aes(Group, NrMarbles, fill = Gender)) +
  geom_bar(stat = 'identity', position = 'dodge')
```

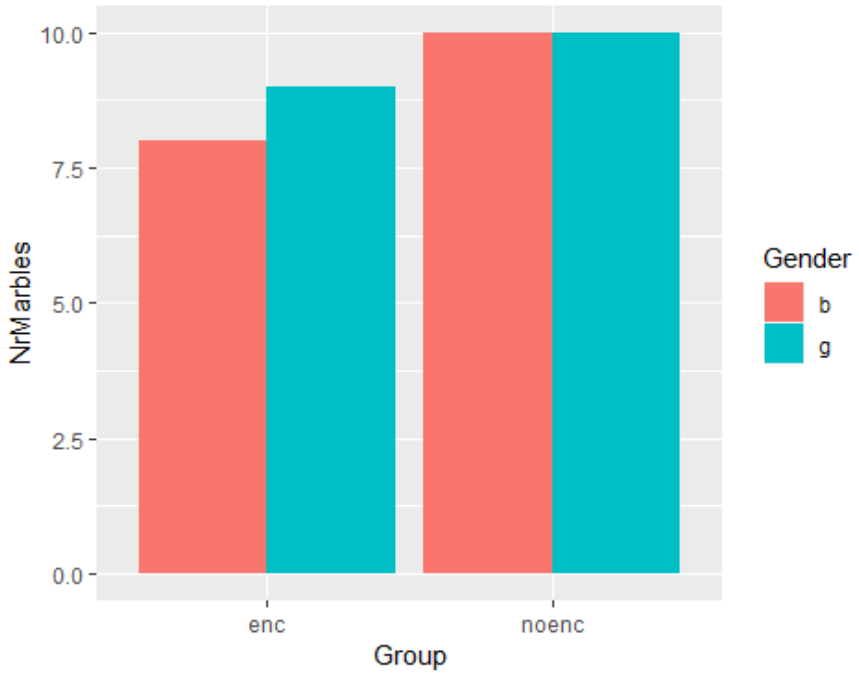

Figure III.R.15: Barplot showing the number of marbles in the group that was encouraged (left) and the group that did not receive any encouragement (right) separated into boys and girls.

As we are now not only interested in main effects, but also in interaction effects, we would like to get the descriptives for every combination of the levels and we will do so using a combination of the by() code and a list() function (see Practical 6C, assignment 5, for details):

```
by(DataFile$DV, list(DataFile$IV1, DataFile$IV2), describe)
```

We have filled it out for the current dataset and then obtain the following results:

```
by(data$NrMarbles, list(data$Group, data$Gender), describe)

## : enc
## : b
##    vars  n mean   sd median trimmed  mad min max range skew kurtosis   se
## X1    1 15 3.67 2.55      4    3.62 2.97   0   8     8  0.3    -1.25 0.66
## ------------------------------------------------------------
## : noenc
## : b
##    vars  n mean  sd median trimmed  mad min max range  skew kurtosis   se
## X1    1 15 5.93 2.4      6       6 1.48   1  10     9 -0.16    -0.71 0.62
## ------------------------------------------------------------
## : enc
## : g
##    vars  n mean   sd median trimmed  mad min max range  skew kurtosis   se
## X1    1 13 4.92 2.66      6       5 2.97   0   9     9 -0.14    -1.21 0.74
## ------------------------------------------------------------
## : noenc
## : g
##    vars  n mean   sd median trimmed  mad min max range  skew kurtosis  se
## X1    1 13 5.85 2.51      7    5.91 2.97   1  10     9 -0.39    -0.81 0.7
```

You can also find separate values using, for example:

```
by(data$NrMarbles, list(data$Group, data$Gender), mean)

## : enc
## : b
## [1] 3.666667
## ------------------------------------------------------------
## : noenc
## : b
## [1] 5.933333
## ------------------------------------------------------------
## : enc
## : g
## [1] 4.923077
## ------------------------------------------------------------
## : noenc
## : g
## [1] 5.846154
```

We will put these values in a table resulting in Table III.R.16.

Table III.R.16: Descriptives for the four subgroups.

	Enc - boy	*noenc - boy*	*enc - girl*	*noenc - girl*
mean	3.67	5.93	4.92	5.85
minimum	0	1	0	1
maximum	8	10	9	10
sd	2.55	2.40	2.66	2.51

We will use the same list() function in combination with the stat.desc function from the "pastecs" package to obtain all necessary information on normality (Grosjean and Ibanez, 2018).

```
by(data$NrMarbles, list(data$Group, data$Gender), stat.desc, basic=FALSE, norm=TRUE)

## : enc
## : b
##      median        mean     SE.mean CI.mean.0.95         var
##   4.0000000   3.6666667   0.6594851   1.4144549   6.5238095
##     std.dev    coef.var    skewness     skew.2SE     kurtosis
##   2.5541749   0.6965932   0.2996224   0.2582420  -1.2522315
##    kurt.2SE  normtest.W  normtest.p
##  -0.5585845   0.9361353   0.3362499
## ------------------------------------------------------------
## : noenc
## : b
##      median        mean     SE.mean CI.mean.0.95         var
##   6.0000000   5.9333333   0.6208034   1.3314908   5.7809524
##     std.dev    coef.var    skewness     skew.2SE     kurtosis
##   2.4043611   0.4052294  -0.1573625  -0.1356294  -0.7057396
##    kurt.2SE  normtest.W  normtest.p
##  -0.3148102   0.9734248   0.9051330
## ------------------------------------------------------------
## : enc
## : g
##      median        mean     SE.mean CI.mean.0.95         var
##   6.0000000   4.9230769   0.7378202   1.6075722   7.0769231
```

```
##      std.dev     coef.var     skewness     skew.2SE     kurtosis
##    2.6602487    0.5403630   -0.1446271   -0.1173281   -1.2135854
##     kurt.2SE   normtest.W   normtest.p
##   -0.5095354    0.9481880    0.5711035
## ------------------------------------------------------------
## : noenc
## : g
##       median         mean      SE.mean CI.mean.0.95          var
##    7.0000000    5.8461538    0.6965681    1.5176915    6.3076923
##      std.dev     coef.var     skewness     skew.2SE     kurtosis
##    2.5115120    0.4296007   -0.3933970   -0.3191417   -0.8126080
##     kurt.2SE   normtest.W   normtest.p
##   -0.3411812    0.9436855    0.5063608
```

The above output suggests that we can assume that the data of all subgroups are approximately normally distributed.

We will also have to check homogeneity of variance for the combination of all subgroups, which can be done using the interaction() function in LeveneTest from the “car” package (Fox and Weisberg, 2011), as was explained in Practical 6C, assignment 7):

```
leveneTest(data$NrMarbles, interaction(data$Group, data$Gender), center = median)

## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  3  0.1227 0.9463
##       52
```

Levene’s test is non-significant, suggesting that we can assume equality of variance.

9.3 Two-way ANOVA: performing the test and interpreting the output

As explained in detail in Practical 6C, assignment 7, when interested in a possible interaction, you need to perform a type III ANOVA. As this is not the default in R, we first have to create contrasts and then ask for type III ANOVAs explicitly using the “car” package (Fox and Weisberg, 2011).

For a detailed explanation on this and the code below, please see Practical 6C, assignment 7.

```
contrasts(data$Group) <- c(-1, 1)
contrasts(data$Gender) <- c(-1, 1)
model = aov(NrMarbles ~ Group*Gender, data=data)
```

Now that we have created contrasts, we should ask for the output of type III ANOVA by typing:

```
Anova(model, type="III")
```

This will provide the output in Table III.R.17.

Table III.R.17 R output for a type III factorial ANOVA

```
## Anova Table (Type III tests)
##
## Response: NrMarbles
##              Sum Sq Df  F value  Pr(>F)
```

```
## (Intercept)  1444.76  1 225.6882 < 2e-16 ***
## Group          35.43  1   5.5344 0.02247 *
## Gender          4.76  1   0.7436 0.39246
## Group:Gender    6.29  1   0.9820 0.32630
## Residuals     332.88 52
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Table III.R.17 shows that there is a significant main effect of Group (p =.022). There is, however, no main effect of Gender (p = .39) nor is there an interaction (Group:Gender, p= .33) between the two variables.

9.4 Two-way ANOVA: calculating the effect size

As explained in Section 7.5 of Part I, most people prefer to report omega-squared (ω^2) for both one- way and factorial ANOVAs. The value of ω^2 is easily obtained using the following code from the "sjstats' package (Lüdecke, 2018).

```
omega_sq(model, partial = TRUE)

##           term partial.omegasq
## 1        Group          0.080
## 2       Gender         -0.005
## 3 Group:Gender          0.000
```

The code partial=TRUE provides the partial version of (ω^2), which partials out other effects and is hence the one you need when you have multiple independent variables.

We are dealing with a medium effect of Group and, as we already knew, no effect of Gender nor an interaction between the two variables (for details on how to interpret (partial) ω^2, see Practical 6C, assignment 8).

9.5 Two-way ANOVA: reporting the results

We could use the following text to report on the results:

There was a significant main effect of encouragement on the number of marbles children put into the box, $F(1,52) = 5.53$, $p = 0.022$ (also see Table III.R.16 and Figure III.R.15). On average, the toddlers who were not encouraged scored higher (M = 5.89; SD = 2.409) than the toddlers who received encouragement (M =4.25; SD =2.633). This effect was of a medium size, pω^2 = 0.080. There was no significant difference between the boys and girls (p = 0.39) nor was there an interaction between gender and encouragement (p=0.33).

References

Allaire, J.J., Xie, Y., McPherson, J., Luraschi, J., Ushey, K., Atkins, A., Wickham, H., Cheng, J., Chang, W., & Iannone, R. (2018). Rmarkdown: Dynamic Documents for R (version 1.10) [Computer software]. Available from https://CRAN.R-project.org/package=rmarkdown

Baayen, R. (2008). *Analyzing Linguistic Data: A Practical Introduction to Statistics using R*. Cambridge: Cambridge University Press. doi:10.1017/CBO9780511801686

Breheny, P., & Burchett, W. (2017). Visualization of Regression Models Using visreg. *The R Journal, 9*(2), 56-71.

Champely, S. (2018). pwr: Basic Functions for Power Analysis (version 1.2-2) [Computer software]. Available from https://CRAN.R-project.org/package=pwr

Field, A., Miles, J., & Field, Z. (2012). *Discovering statistics using R.* London: Sage Publications Ltd.

Fox, J., & Weisberg, S. (2011). *An R companion to applied regression* (2nd ed.). Thousand Oaks, CA: Sage.

Grosjean, P., & Ibanez, F. (2018). pastecs: Package for Analysis of Space-Time Ecological Series (version 1.3.21) [Computer software]. Available at https://CRAN.R-project.org/package=pastecs

Lüdecke, D. (2018). sjstats: Statistical Functions for Regression Models. (version 0.17.1) [Computer software]. Available at http://doi.org/10.5281/zenodo.1284472

Meyer, D., Zeileis, A., & Hornik, K. (2017). vcd: Visualizing Categorical Data. (version 1.4-4) [Computer software].

R Core Team (2018). R: A language and environment for statistical computing (version 3.5.1) [Computer software]. Available at https://www.R-project.org/

Revelle, W. (2018). psych: Procedures for Psychological, Psychometric, and Personality Research (version 1.8.4) [Computer software]. Available at https://CRAN.R-project.org/package=psych

RStudio Team (2016). RStudio: Integrated Development for R. RStudio (version 1.1.456) [Computer software] Available at http://www.rstudio.com/

Warnes, G. R., Bolker, B., Lumley, T., & Johnson, R. C. (2018). gmodels: Various R Programming Tools for Model Fitting (version 2.18.1) [Computer software]. Available at https://CRAN.R-project.org/package=gmodels

Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis* (2nd ed.). Springer International Publishing.