Portfolio 1

Portfolio exam - Part 1 | Methods 1 E2021, CogSci @AU

Laurits Lyngbæk

21/9/2021

Deadline: Wednesday 29/9/2021 h23:59

This is an individual portfolio assignment

Upload your Portfolio 1 assignment to the dedicated link ion Brightspace, under "Assignments". **Remember to upload the HTML knit of the markdown, and not the markdown (.Rmd) itself. No PDF knits, please.**

Please write your name in the `author` field above.

Introduction

The goal of this exam is to write a short data mining report on the CogSci Intro Week Personality Test Data in which you answer the following questions **in prose, code and graphs**.

First of all, let's start by looking at the setup chunk. If you need to load packages or set your working directory, do so here:

```
pacman::p_load(tidyverse)
pacman::p_load(pastecs)
```

Now you have to import the personality data. Once you have done so, use the `head()` function to print the first 10 lines of the data set.

```
df <- read_csv("personality_data_cleaned_2021.csv")
```

```
## New names:
## * `` -> ...1
```

```
## Rows: 48 Columns: 51
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (37): timestamp, student_number, name, gender, native_Danish, handednes...
## dbl  (13): ...1, shoesize, choose_rand_num, 2D4D, balloon, balloon_balance, ...
## date  (1): birth_day
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
df %>%
head()
```

```
## # A tibble: 6 x 51
##    ...1 timestamp  student_number name  birth_day  shoesize gender native_Danish
##   <dbl> <chr>      <chr>          <chr> <date>        <dbl> <chr>  <chr>
## 1     1 2021/08/2~ 202105598      Corn~ 1999-11-12       37 female Yes
## 2     2 2021/08/2~ 202104529      Tilde 2000-09-02       37 female Yes
## 3     3 2021/08/2~ 202108998      Rebe~ 2001-06-26       38 female Yes
## 4     4 2021/08/2~ 202109723      Sara~ 2000-04-26       37 female Yes
## 5     5 2021/08/2~ 202104528      Maja~ 2000-09-02       37 female Yes
## 6     6 2021/08/2~ 202106904      Vlada 2002-01-25       36 female No
## # ... with 43 more variables: handedness <chr>, choose_rand_num <dbl>,
## #   touch_floor <chr>, touch_hands <chr>, 2D4D <dbl>, balloon <dbl>,
## #   balloon_balance <dbl>, breathhold <dbl>, bad_choices <chr>,
## #   tongue_twist <dbl>, romberg_open <dbl>, romberg_closed <dbl>,
## #   ling_animal <chr>, ling_direct <chr>, ling_demonstr <chr>,
## #   ling_place <chr>, ling_abstract <chr>, ling_pronoun <chr>, ling_math <chr>,
## #   ling_activity <chr>, ling_adjective <chr>, ling_kiki <chr>, ...
```

Once you are done loading the data, you can start working on the questions below.

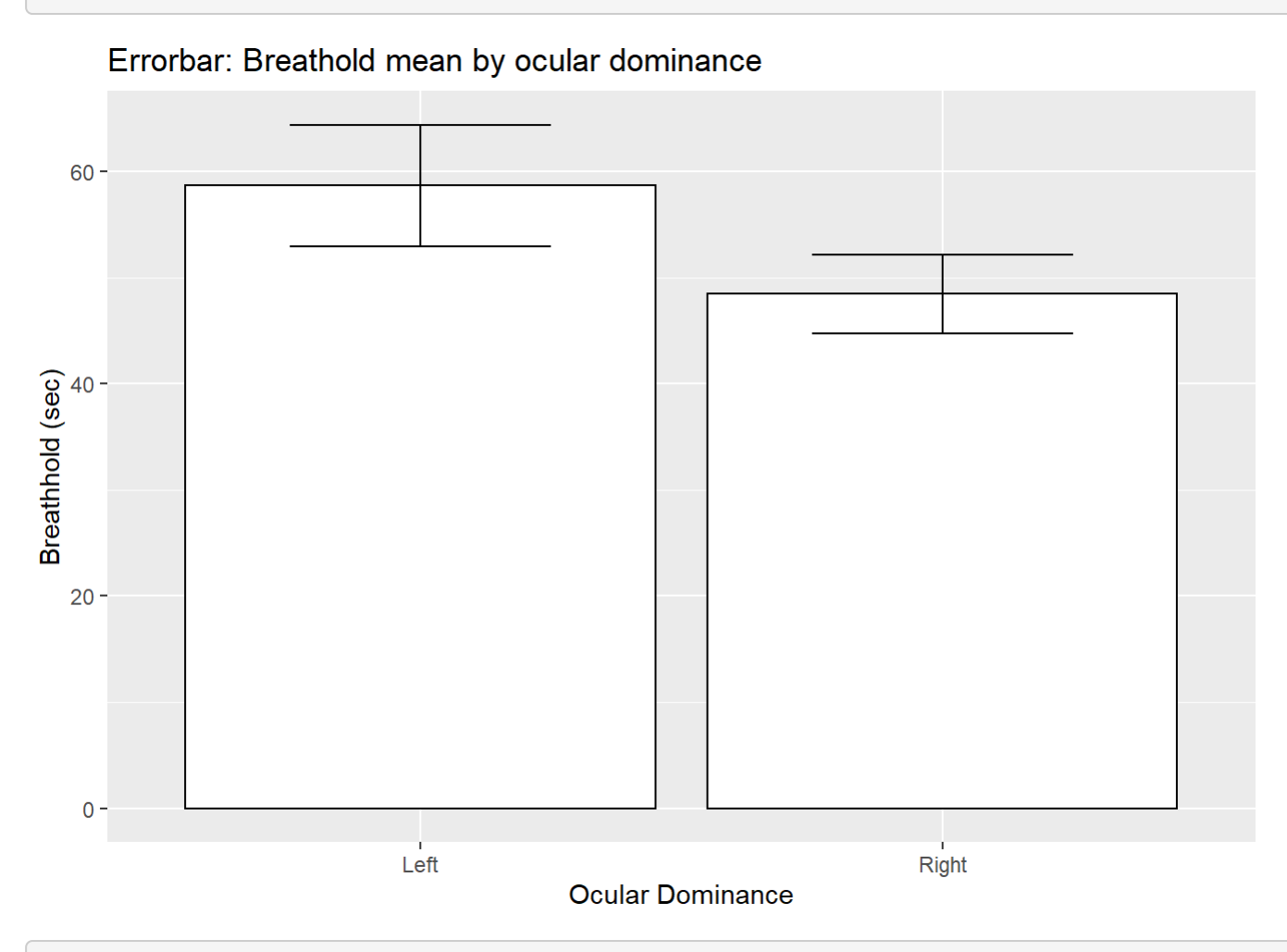
Question 1

Who can hold their breath the longest on average — those with right or left ocular dominance? Plot the data using `ggplot2` to find out. The plots should include error bars depicting the standard error of the mean: you can add these using the `geom_errorbar()` function and specifying `stat = "summary", fun.data = "mean_se"`. Then use the `mean()` and `sd()` functions *within a tidyverse pipe* to make a summary data set, in which you show mean and standard deviation of the two eye dominance groups.

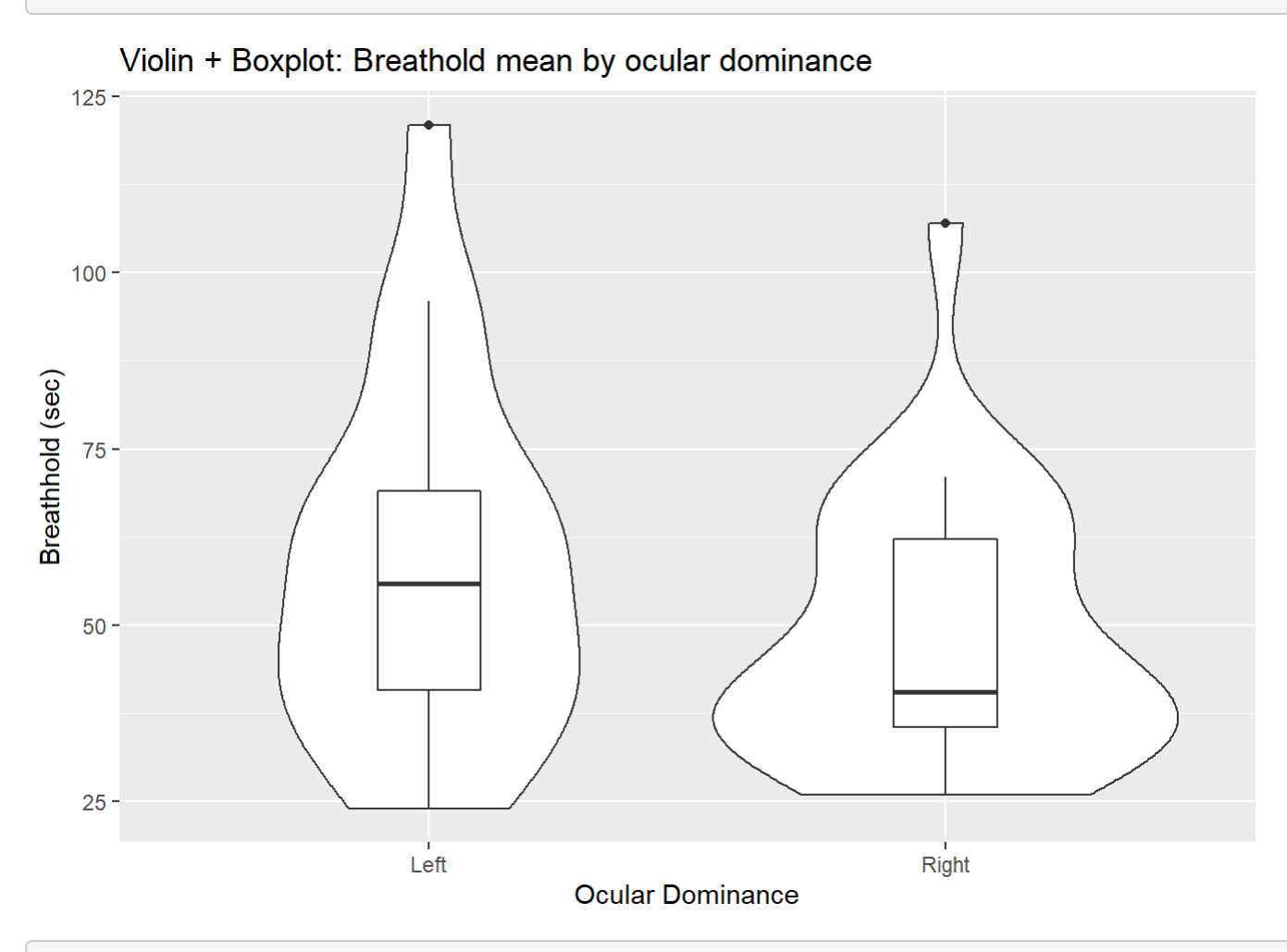
If there are people that answered other things than "Right" or "Left", then filter them out.

Bonus question: If you feel brave, you can instead try making a boxplot (`geom_boxplot()`) or a violin plot (`geom_violin()`) which are better at representing the actual distribution of the data (compared to a bar plot, which only depicts mean and standard deviation).

```
filter_df1<- df %>%
  filter(ocular_dom == "Right"| ocular_dom == "Left")

summary_df1<- filter_df1 %>%
  group_by(ocular_dom) %>%
  summarise(mean = mean(breathhold),
            sd = sd(breathhold),
            se = sd/sqrt(n()),
            max=mean+se,
            min=mean-se
            )


summary_df1 %>%
  ggplot(aes(x=ocular_dom,y=mean,ymin=min,ymax=max))+
  geom_col(fill="white",col = 'black')+
  geom_errorbar(width=0.5)+
  labs(x="Ocular Dominance",
       y="Breathhold (sec)",
       title = "Errorbar: Breathold mean by ocular dominance")
```




```
filter_df1 %>%
  ggplot(aes(x=ocular_dom,y=breathhold))+
  geom_violin()+
  geom_boxplot(width=0.2)+
  labs(x="Ocular Dominance",
       y="Breathhold (sec)",
       title = "Violin + Boxplot: Breathold mean by ocular dominance")
```




```
summary_df1
```

```
## # A tibble: 2 x 6
##   ocular_dom  mean    sd    se   max   min
##   <chr>      <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 Left        58.7  25.6  5.72  64.4  53.0
## 2 Right       48.5  18.9  3.71  52.2  44.8
```

Explain your results in plain terms here (max 3 sentences):

The errorbars are narrow, which indicates that the datas mean closely approximates the true populations mean.

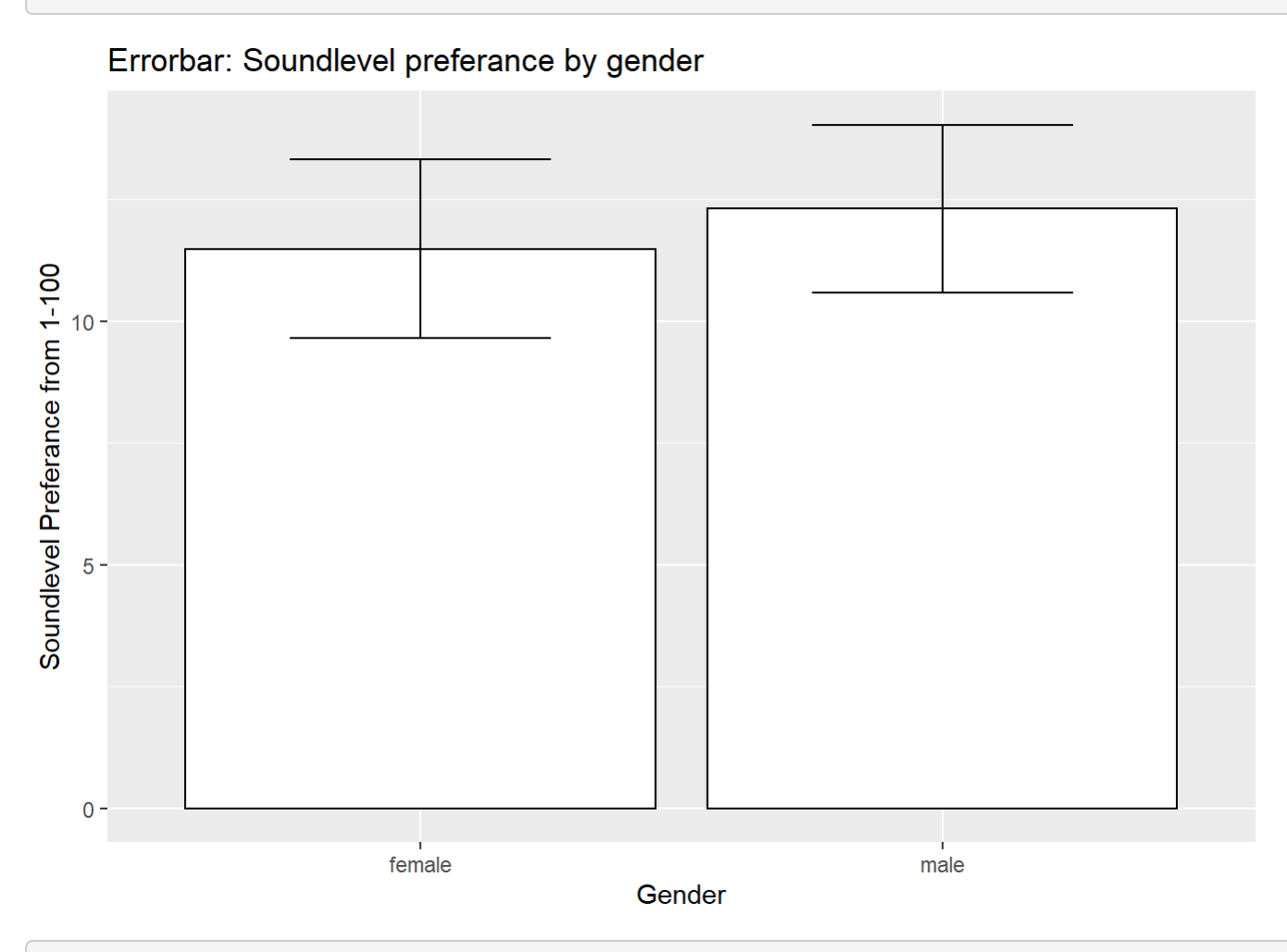
The box+violinplot indicates that the data has a few outliers.

Question 2

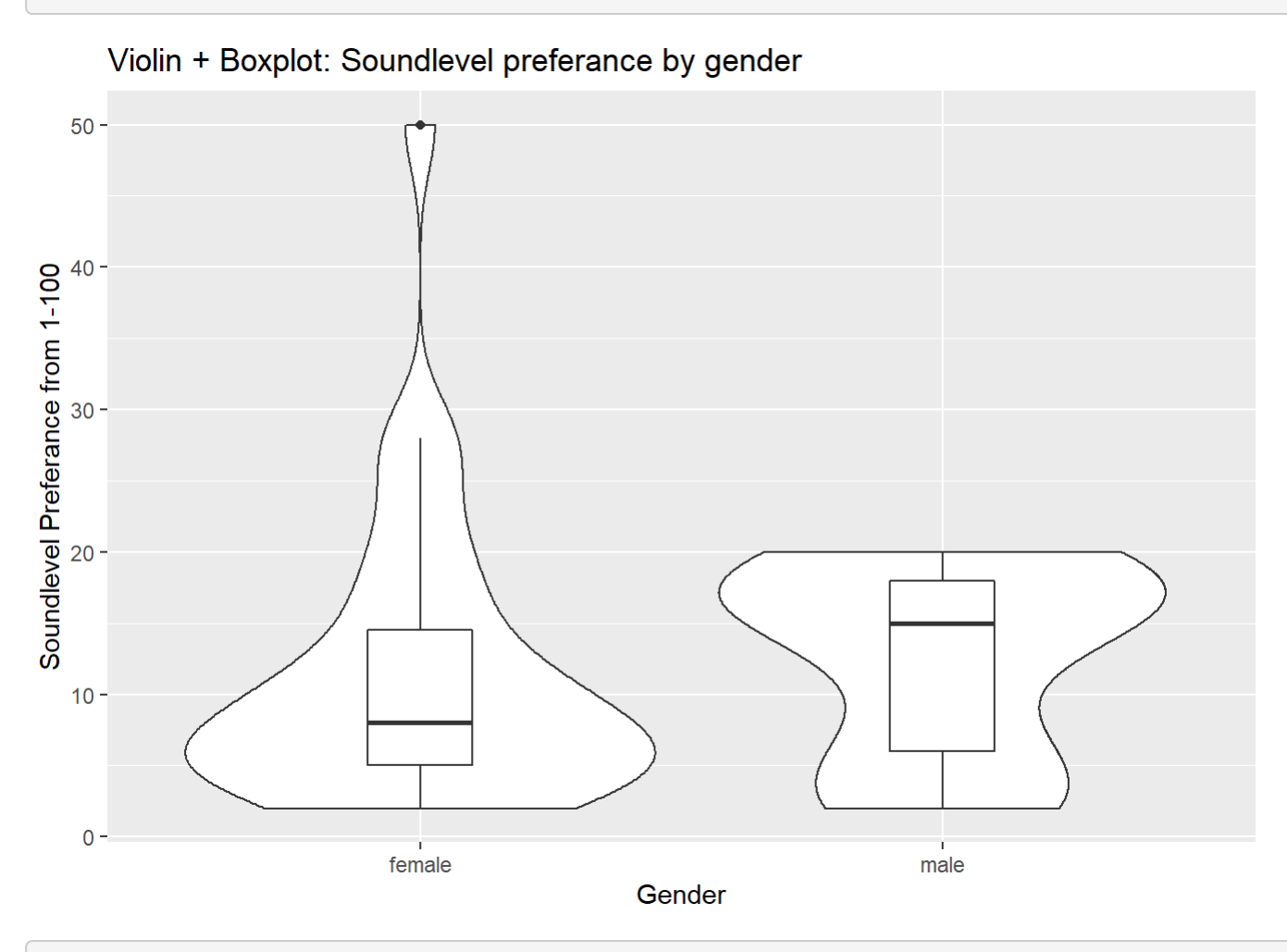
Who likes silence vs. noise best – by gender? Also in this case you should plot the data using `ggplot2` (including error bars depicting the standard error of the mean), then use the `mean()` and `sd()` functions to find mean and standard deviation of the two genders (still making a summary data set with tidyverse and pipes).

Bonus question: If you feel brave, you can instead try making a boxplot (`geom_boxplot()`) or a violin plot (`geom_violin()`) which are better at representing the actual distribution of the data (compared to a bar plot, which only depicts mean and standard deviation).

```
filter_df2 <- df %>%
  filter(!is.na(sound_level_pref))

summary_df2<- filter_df2 %>%
  group_by(gender) %>%
  summarise(mean = mean(sound_level_pref),
            sd = sd(sound_level_pref),
            se = sd/sqrt(n()),
            max=mean+se,
            min=mean-se
            )

summary_df2 %>%
  ggplot(aes(x=gender,y=mean,ymin=min,ymax=max))+
  geom_col(fill="white",col = 'black')+
  geom_errorbar(width=0.5)+
  labs(x="Gender",
       y="Soundlevel Preferance from 1-100",
       title = "Errorbar: Soundlevel preferance by gender")
```




```
filter_df2 %>%
  ggplot(aes(x=gender,y=sound_level_pref))+
  geom_violin()+
  geom_boxplot(width=0.2)+
  labs(x="Gender",
       y="Soundlevel Preferance from 1-100",
       title = "Violin + Boxplot: Soundlevel preferance by gender")
```




```
summary_df2
```

```
## # A tibble: 2 x 6
##   gender  mean    sd    se   max   min
##   <chr>  <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 female  11.5 10.2   1.84  13.3  9.65
## 2 male    12.3  6.89  1.72  14.0 10.6
```

Explain your results in plain terms here (max 3 sentences):

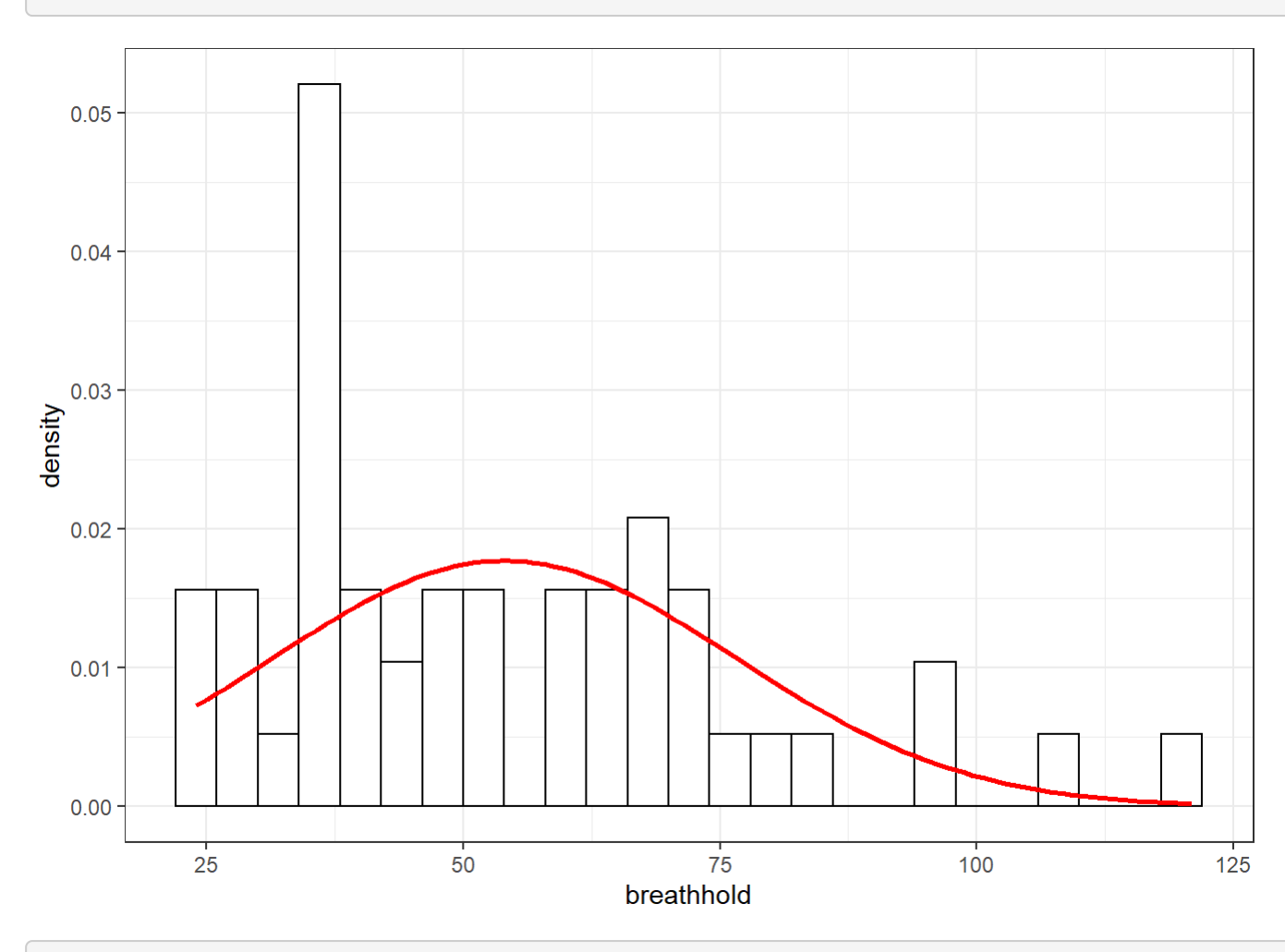
The errorbars overlap rather much, indicating that the population might have the same mean for both genders.

The box+violinplot indicates that only the female data has outliers.

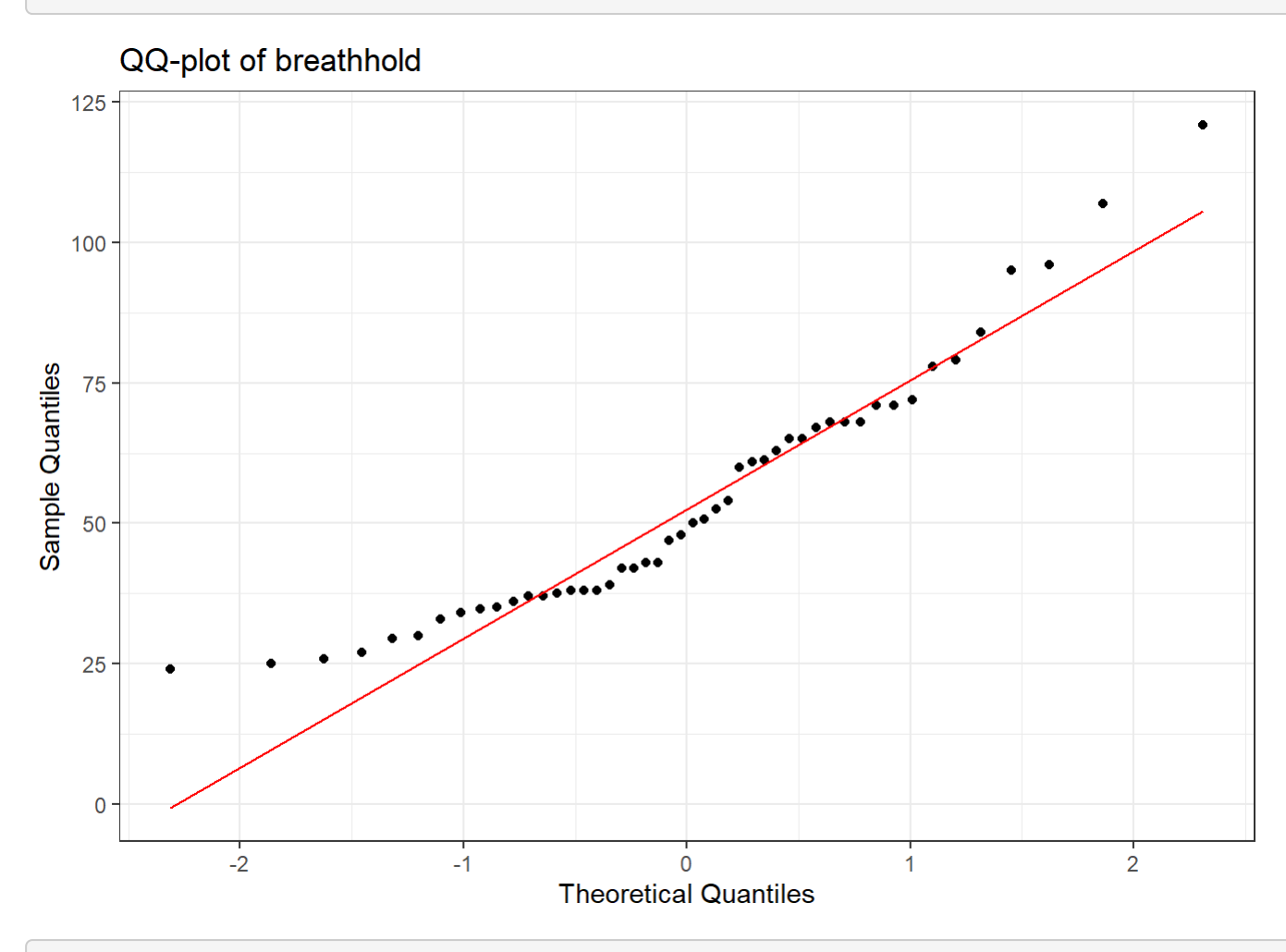
Question 3

Is the `breathhold` variable normally distributed? Provide both visual (histogram and QQ-plot) and numeric (Shapiro-Wilk test and skewness/kurtosis values) support for your answer.

```
ggplot(df, aes(breathhold)) +
 geom_histogram(aes(y = ..density..), binwidth=4, colour = "black", fill = "white") +
 stat_function(fun = dnorm, args = list(mean = mean(df$breathhold),
 sd = sd(df$breathhold)), colour = "red", size = 1) +
 theme_bw()
```




```
df %>%
  ggplot(aes(sample = breathhold))+
  stat_qq()+
  stat_qq_line(color="red")+
   labs(x = "Theoretical Quantiles",
        y = "Sample Quantiles",
        title = "QQ-plot of breathhold")+
  theme_bw()
```




```
round(pastecs::stat.desc(cbind(Breathhold=df$breathhold), basic = FALSE, norm = TRUE,desc= FALSE), digits = 2)
```

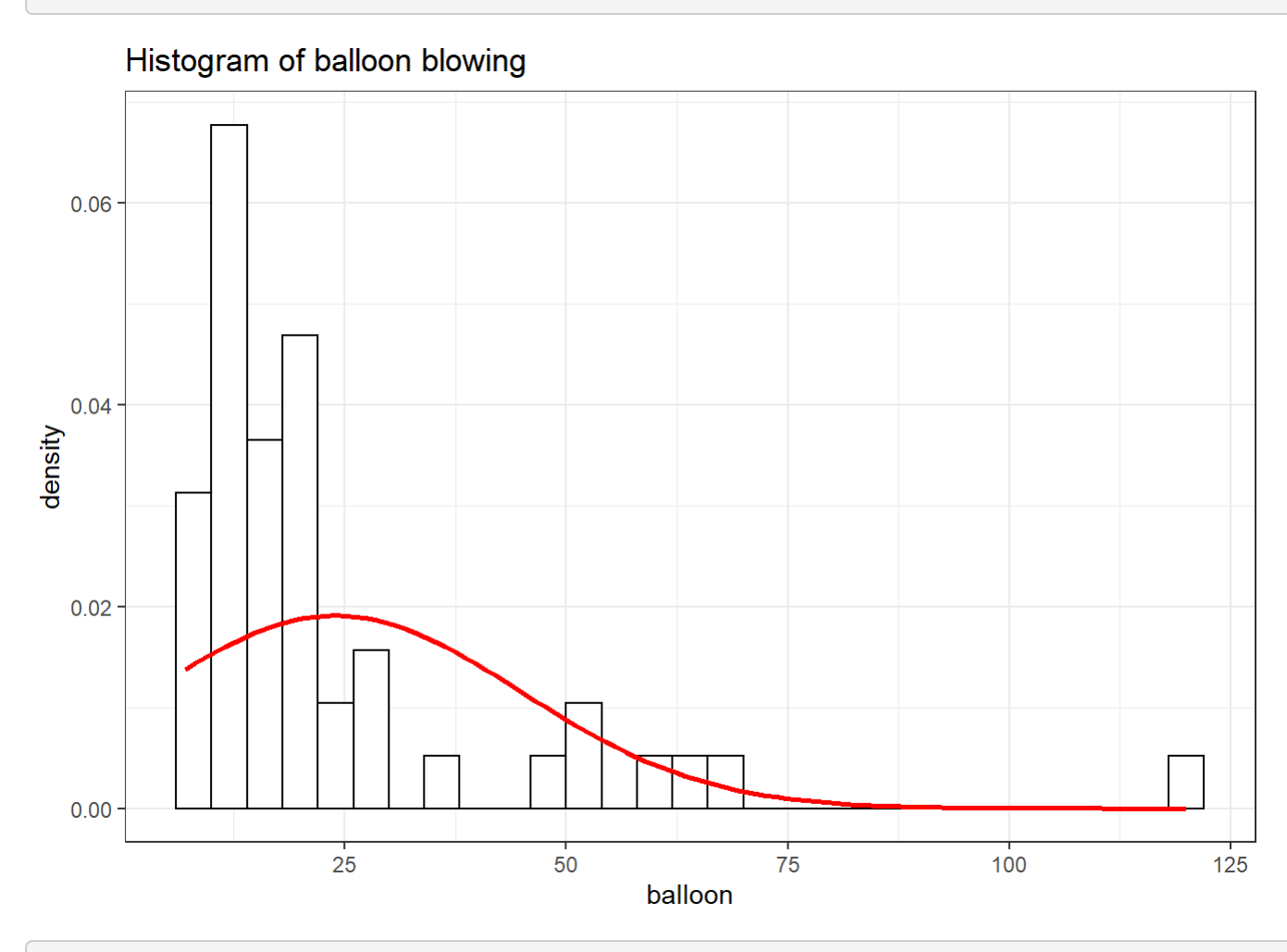
```
##            Breathhold
## skewness         0.89
## skew.2SE         1.29
## kurtosis         0.31
## kurt.2SE         0.23
## normtest.W       0.92
## normtest.p       0.00
```

Question 4

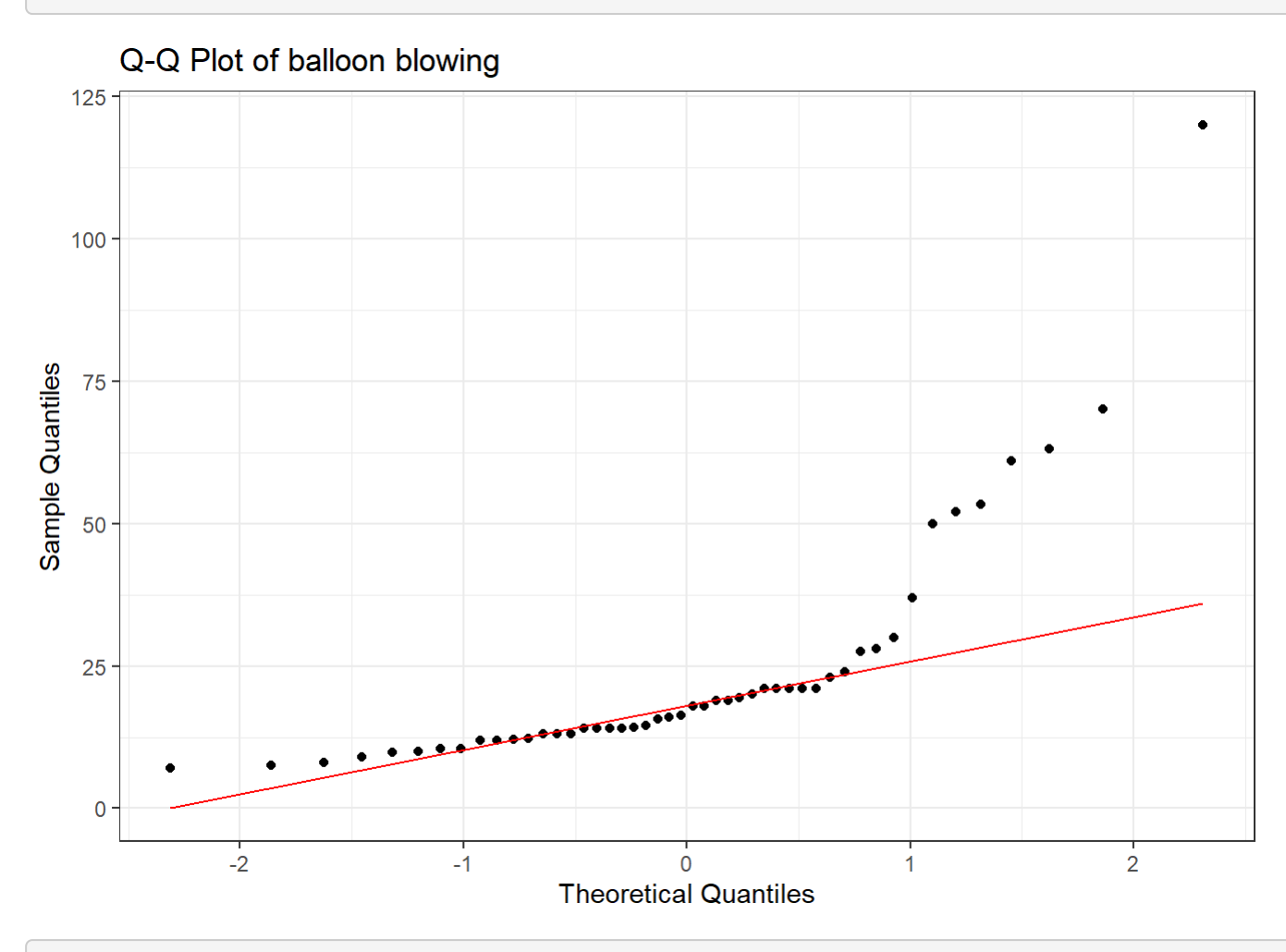
Are the two balloon reaction time variables (`balloon` and `balloon_balance`) normally distributed? Provide visual (histogram and QQ-plot) and numeric (Shapiro-Wilk test and skewness/kurtosis values) support for your answer.

If they are not, then discuss your results below.

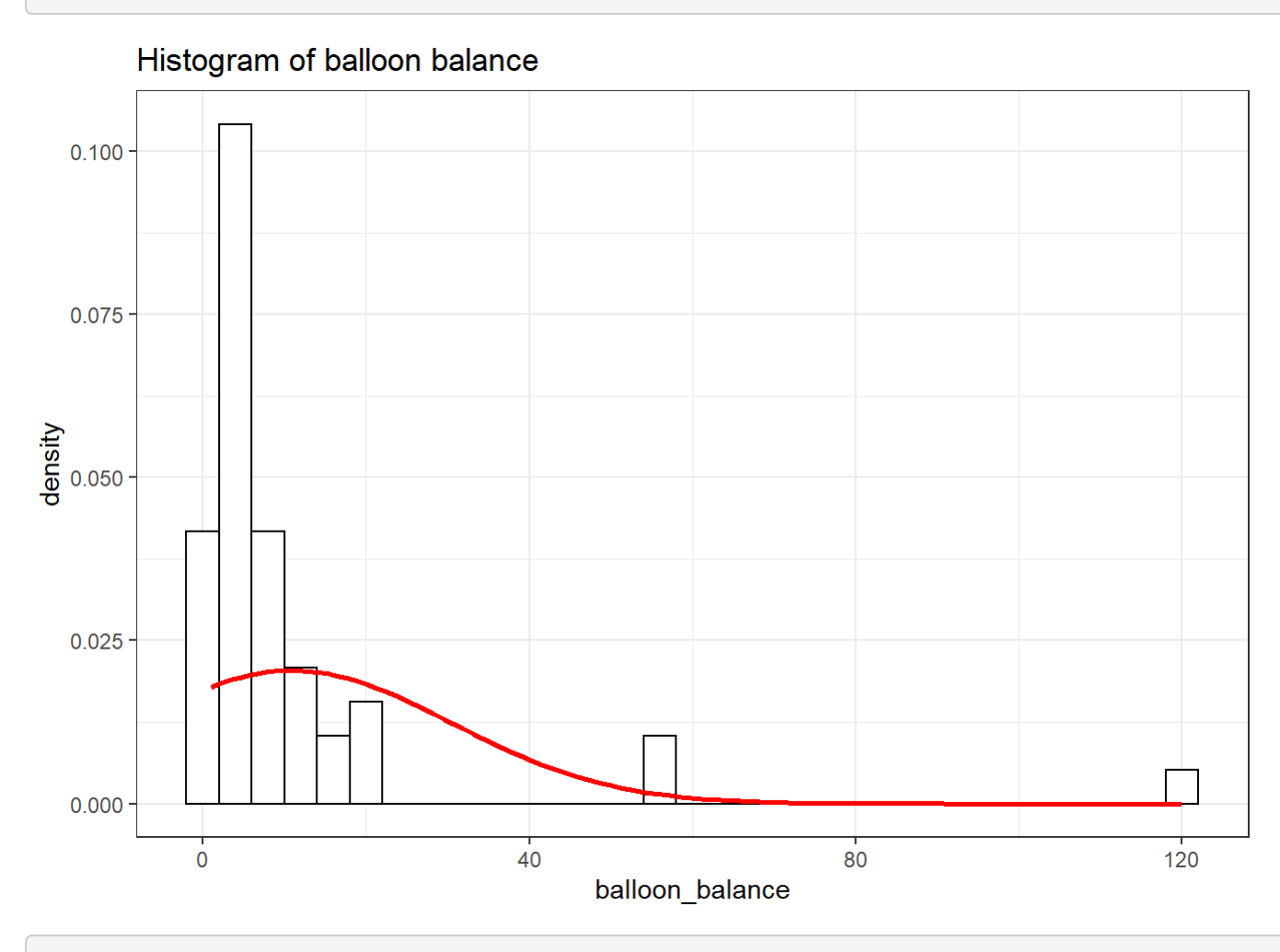
```
ggplot(df, aes(balloon)) +
 geom_histogram(aes(y = ..density..), binwidth=4, colour = "black", fill = "white") +
 stat_function(fun = dnorm, args = list(mean = mean(df$balloon),
 sd = sd(df$balloon)), colour = "red", size = 1) +
  ggtitle("Histogram of balloon blowing")+
  theme_bw()
```



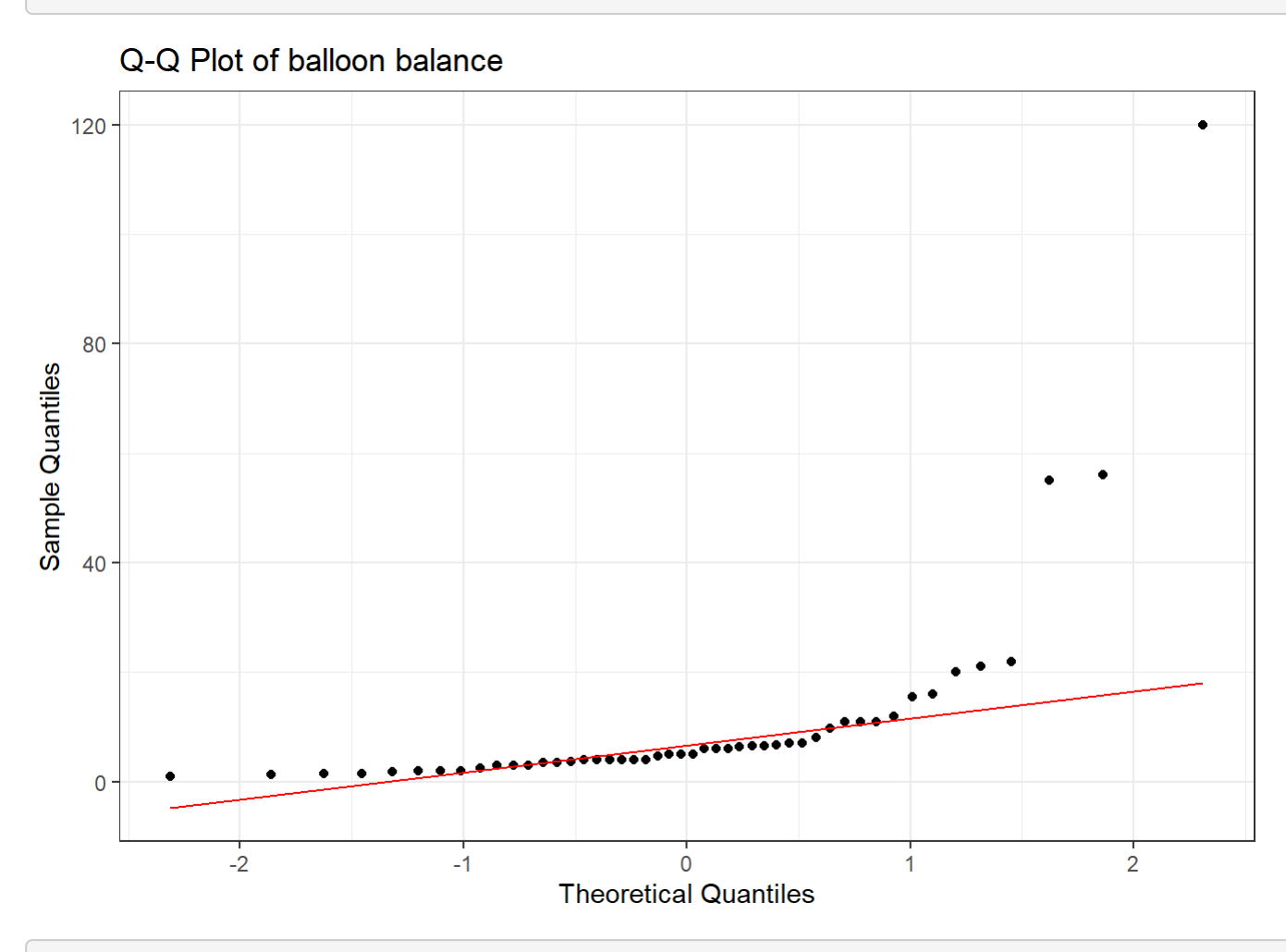

```
ggplot(df, aes(sample = balloon)) +
 stat_qq() +
 stat_qq_line(colour = "red") +
 labs(x = "Theoretical Quantiles", y = "Sample Quantiles") +
 ggtitle("Q-Q Plot of balloon blowing") +
 theme_bw()
```




```
ggplot(df, aes(balloon_balance)) +
 geom_histogram(aes(y = ..density..), binwidth=4, colour = "black", fill = "white") +
 stat_function(fun = dnorm, args = list(mean = mean(df$balloon_balance),
 sd = sd(df$balloon_balance)), colour = "red", size = 1) +
  ggtitle("Histogram of balloon balance")+
  theme_bw()
```




```
ggplot(df, aes(sample = balloon_balance)) +
 stat_qq() +
 stat_qq_line(colour = "red") +
 labs(x = "Theoretical Quantiles", y = "Sample Quantiles") +
 ggtitle("Q-Q Plot of balloon balance") +
 theme_bw()
```




```
round(pastecs::stat.desc(cbind(Balloon=df$balloon, Ballon_Balance=df$balloon_balance), basic = FALSE, norm = TRUE
,desc= FALSE), digits = 2)
```

```
##            Balloon Ballon_Balance
## skewness      2.55           4.08
## skew.2SE      3.72           5.95
## kurtosis      7.55          18.33
## kurt.2SE      5.60          13.59
## normtest.W    0.67           0.46
## normtest.p    0.00           0.00
```

Explain your results in plain terms here (max 3 sentences):

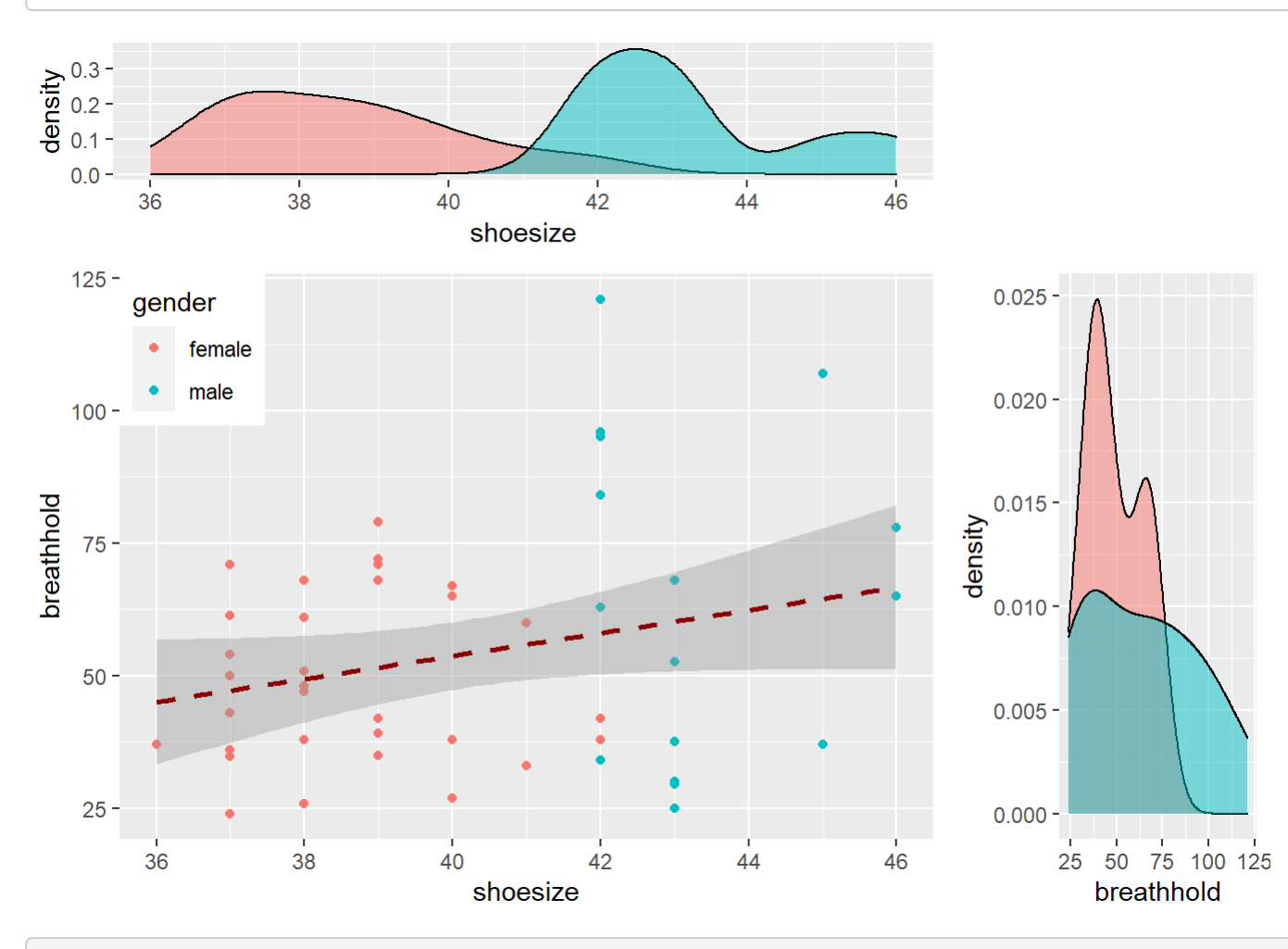
The data is not normally distributed, which we can deduct in two ways: Visually we can see that the data is skewed from the histogram and QQ-plot. Numerically we can see that skew.2SE and kurt.2SE is way above 1 for both data sets, and therefore not normally distributed data.

Question 5

Shoe size could tell us something about general body size, which could also be connected to one's ability to hold your breath. In other words we predict that there is a positive relation between shoe size and how long time CogSci students can hold their breath. Try plotting the two sets of data against each other using a *scatter plot* (hint: both variables are continuous variables). You can make a scatter plot in `ggplot2` using the `geom_point()` function and plotting one variable on each axis. Use grouping in your plot to distinguish the relationship between shoe size and holding breath for males and females, since we expect males and females to have different show sizes. You can for instance use the `color` parameter within the `aes()` function to color by gender.

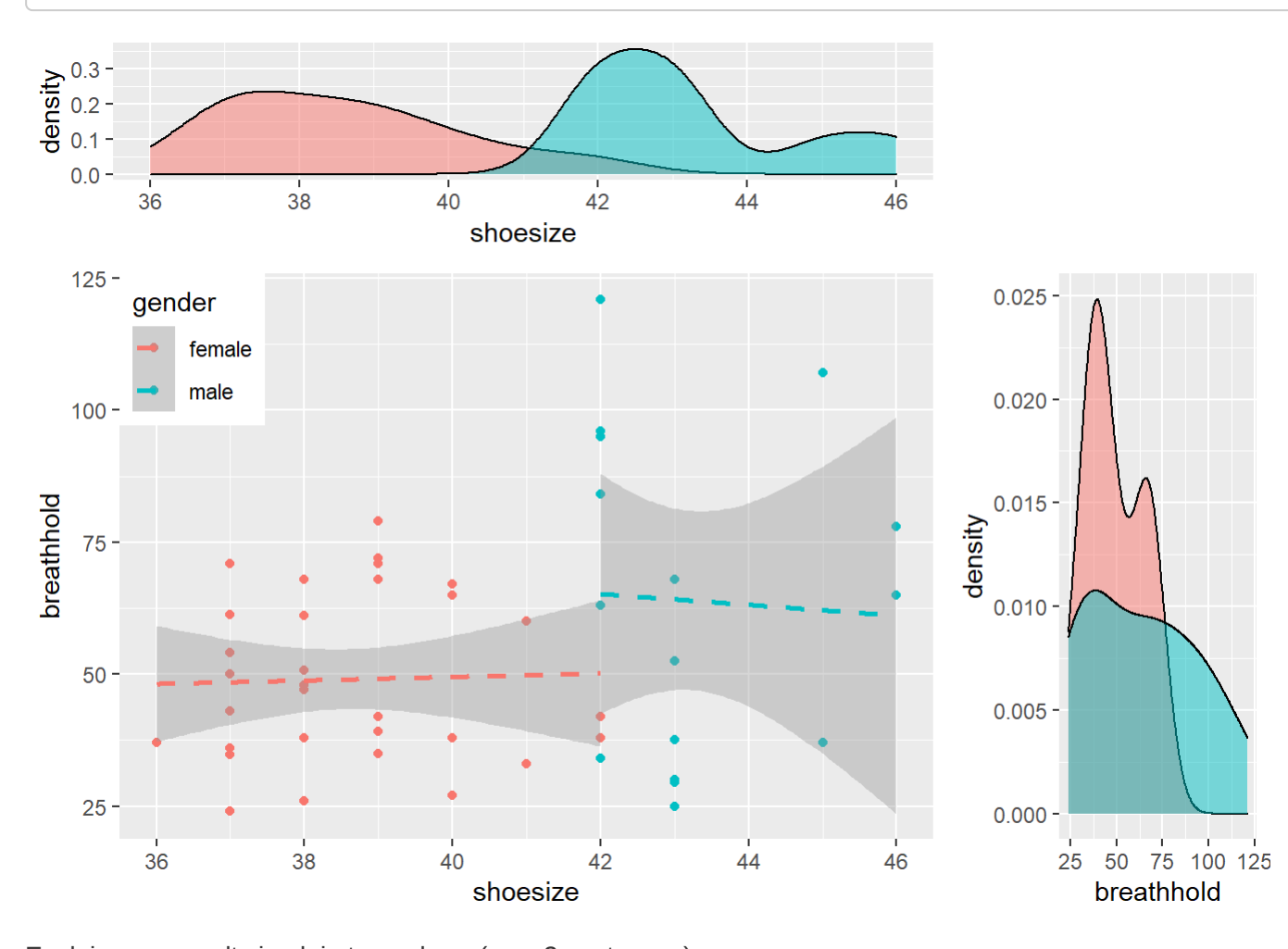
```
scatterplot<- ggplot(df,aes(shoesize,breathhold))+
  geom_point(aes(color=gender))+
  geom_smooth(method=lm,linetype="dashed",color="darkred")+
  theme(legend.position=c(0,1), legend.justification=c(0,1))

scatter_gender_reg<- ggplot(df,aes(shoesize,breathhold,color=gender))+
  geom_point()+
  geom_smooth(method=lm,linetype="dashed")+
  theme(legend.position=c(0,1), legend.justification=c(0,1))


xdensity <- ggplot(df, aes(shoesize, fill=gender)) +
  geom_density(alpha=.5) +
  theme(legend.position = "none")

ydensity <- ggplot(df, aes(breathhold, fill=gender)) +
  geom_density(alpha=.5) +
  theme(legend.position = "none")

blankPlot <- ggplot()+geom_blank(aes(1,1))+
  theme(plot.background = element_blank(),
   panel.grid.major = element_blank(),
   panel.grid.minor = element_blank(),
   panel.border = element_blank(),
   panel.background = element_blank(),
   axis.title.x = element_blank(),
   axis.title.y = element_blank(),
   axis.text.x = element_blank(),
   axis.text.y = element_blank(),
   axis.ticks = element_blank()
     )

gridExtra::grid.arrange(xdensity, blankPlot, scatterplot, ydensity,ncol=2, nrow=2, widths=c(4, 1.4), heights=c(1.
4, 4))
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
gridExtra::grid.arrange(xdensity, blankPlot, scatter_gender_reg, ydensity,ncol=2, nrow=2, widths=c(4, 1.4), heigh
ts=c(1.4, 4))
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Explain your results in plain terms here (max 3 sentences):

It seems there *might* be a positive relation between shoesize and breathhold. But if we regress by gender, we can see it seems that males on average have greater breatholding abilities and shoesizes than women, and this might be a lurking variable creating the relation.

That's all!

Portfolio 2

Portfolio 2

Studygroup 5 (Maja, Niels, Marton, Laurits & Sarah S.)

26/10/2021

Introduction

We have conducted a PsychoPy experiment. The experiment was about reading time and how out of place words (salient words) not fitting into the context of the story would possibly affect reading time.

```
knitr::opts_chunk$set(echo = T)

message = FALSE

pacman::p_load(pastecs, tidyverse, readbulk, stringr, car, ggpubr)
```

Stimuli

Our stimuli text where control/salient word is marked in **bold**

> This is a short story about Hungry Wolf. Once, a wolf was very hungry. It looked for food here and there. But it couldn't get any. At last it found a loaf of bread and piece of meat in the hole of a tree. The hungry wolf squeezed into the hole. It ate all the food. It was a woodcutter's lunch. He was on his way back to the tree to have lunch. But he saw there was no food in the hole, instead, a wolf. On seeing the woodcutter, the wolf tried to get out of the hole. But it couldn't. Its tummy was swollen. The woodcutter caught the **wolf/priest** and gave it nice beatings.

Data loading

We load in our logging data and add additional fields from the MRC database for further analysis.

```
df <- readbulk::read_bulk("logfiles", extension = ".csv", verbose = F)

df <- df %>%
  rename(word_number = X)

mrc <- read_csv("MRC_database.csv")
```

```
## Rows: 120392 Columns: 14
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): word
## dbl (13): nlet, nsyl, kf_freq, kf_ncats, kf_nsamp, tl_freq, brown_freq, fam,...
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
df <- df %>%
  mutate(word = str_to_upper(word)) %>%
  inner_join(mrc) %>%
  mutate(
    var = if_else(is.na(lag(word)),
                  TRUE,
                  lag(word) != word)) %>%
  filter(var)
```

```
## Joining, by = "word"
```

```
df <- df %>%
  mutate(name=as.factor(name)) %>%
  mutate(name=as.numeric(name)) %>%
  mutate(name=as.factor(name))
```

Variables

- **name:** Subject identification (Factor)
- **age:** Age (Int)
- **gender:** Gender of the participant (Factor)
- **condition:** control = No surprising words, salient = there will be salient words (Factor)
- **word:** The word being read (Character)
- **reading_time:** The reading time of that particular word. (Numeric)
- **word_number:** the number of letters in a word (Int)

Correlation analysis

Assumption testing

We need to examine whether or not our data is normally distributed in order to do t-tests on it. Therefore, we will do a Shapiro Wilk test on our data to get statistical evidence and also visualize it in a histogram and a qq-plot.

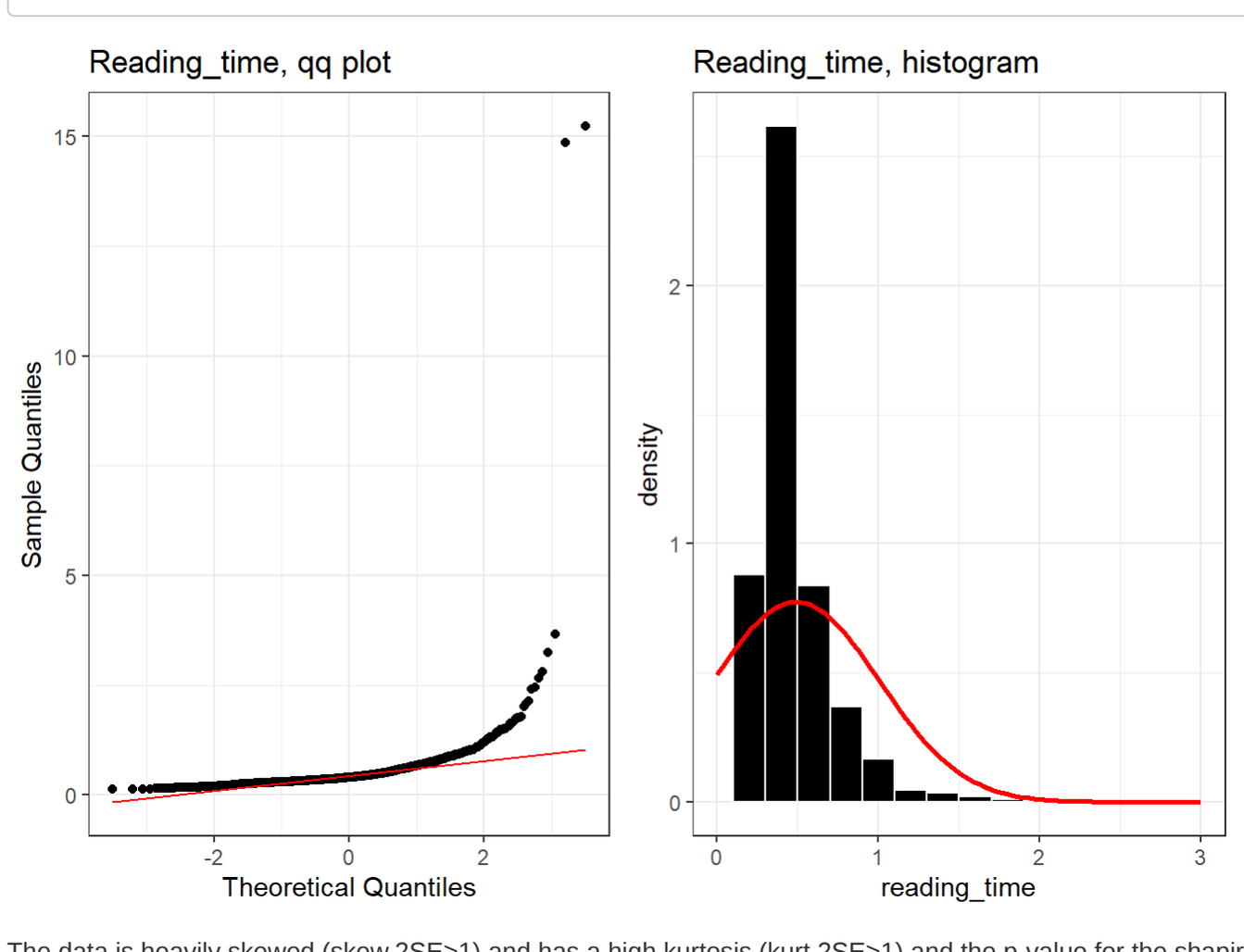
```
round(pastecs::stat.desc(cbind(df$reading_time), basic = FALSE, norm = TRUE), digits = 2)
```

```
##                   V1
## median          0.40
## mean            0.49
## SE.mean         0.01
## CI.mean.0.95    0.02
## var             0.26
## std.dev         0.51
## coef.var        1.05
## skewness       21.04
## skew.2SE      201.96
## kurtosis      580.90
## kurt.2SE     2789.04
## normtest.W      0.28
## normtest.p      0.00
```

```
qq <- df %>%
  ggplot(aes(sample = reading_time)) +
    stat_qq() +
    stat_qq_line(colour = "red") +
    labs(x = "Theoretical Quantiles", y = "Sample Quantiles") +
    ggtitle("Reading_time, qq plot") +
    theme_bw()

hist <- df %>%
  ggplot(aes(reading_time)) +
      geom_histogram(aes(y = ..density..),binwidth=0.2, colour = "white", fill = "black") +
      stat_function(fun = dnorm, args = list(mean = mean(df$reading_time), sd = sd(df$reading_time)), colour = "r
ed", size = 1) +
      ggtitle("Reading_time, histogram") + xlim(0,3) + theme_bw()
#setting xlim removes two outliers at 15 seconds RT

ggarrange(qq,hist, ncol = 2)
```

```
## Warning: Removed 4 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```



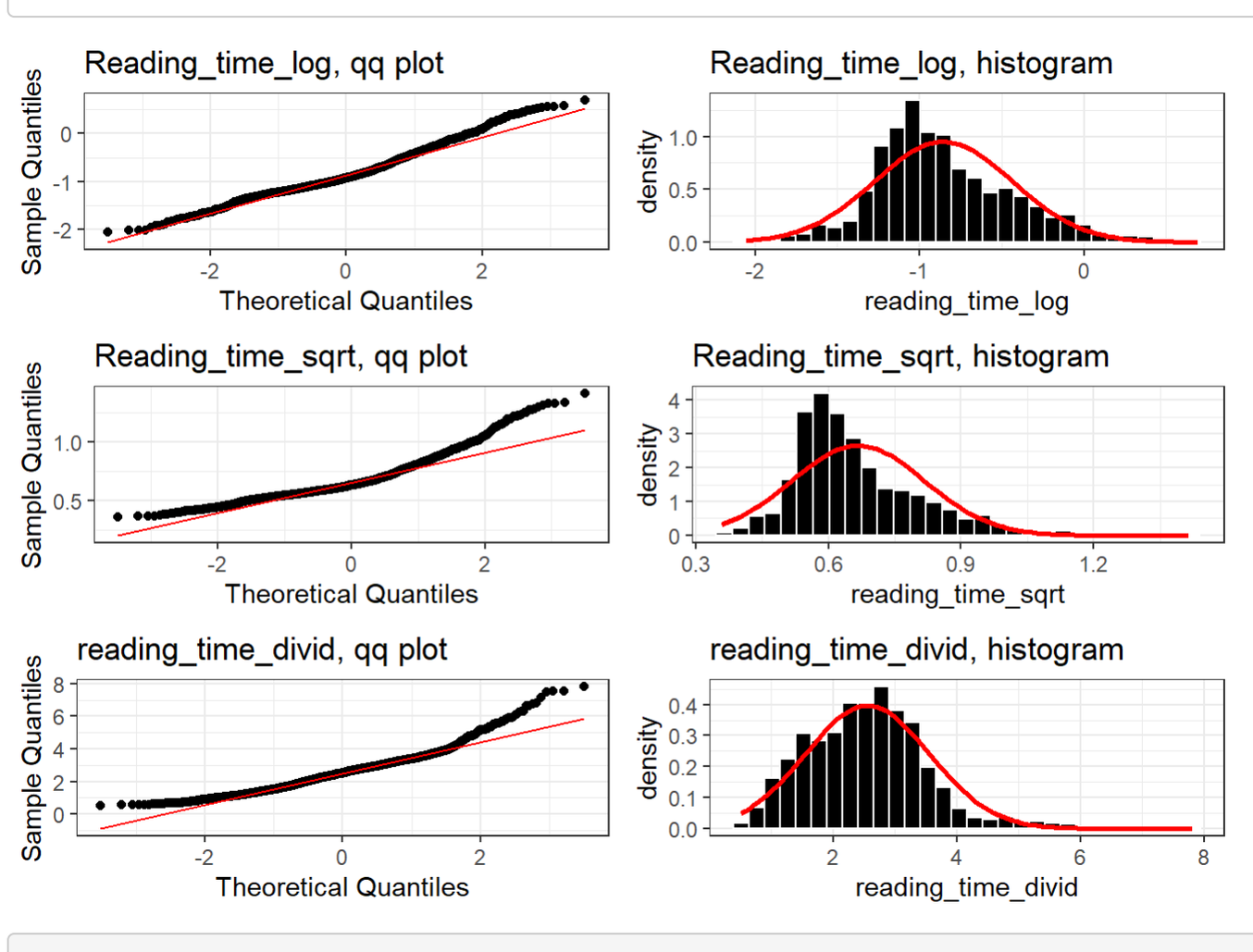
The data is heavily skewed (skew.2SE>1) and has a high kurtosis (kurt.2SE>1) and the p-value for the shapiro wilk test is below the significant level 0.05, meaning that the data is not normally distributed. The qq-plot supports that our data is not normally distributed, since the data points does not follow the linear line, which is to be expected as we are looking at reaction times. Therefore, we can try to see whether or not transforming the data and removing outliers result in a normal distribution.

```
filter_outlier <- df %>%
  filter(reading_time > mean(reading_time)-sd(reading_time)*3) %>%
  filter(reading_time < mean(reading_time)+sd(reading_time)*3)

filter_outlier <- filter_outlier %>%
  mutate(reading_time_log = log(reading_time),
        reading_time_sqrt = sqrt(reading_time),
        reading_time_divid = 1/reading_time
      )
```

Now we check if the transformed data is normally distributed:

```
log_qq<- filter_outlier %>%
  ggplot(aes(sample = reading_time_log)) +
    stat_qq() +
    stat_qq_line(colour = "red") +
    labs(x = "Theoretical Quantiles", y = "Sample Quantiles") +
    ggtitle("Reading_time_log, qq plot") +
    theme_bw()

log_hist <- filter_outlier %>%
  ggplot(aes(reading_time_log)) +
      geom_histogram(aes(y = ..density..), colour = "white", fill = "black") +
      stat_function(fun = dnorm, args = list(mean = mean(filter_outlier$reading_time_log),
             sd = sd(filter_outlier$reading_time_log)), colour = "red", size = 1) +
      ggtitle("Reading_time_log, histogram") +
      theme_bw()


sqrt_qq <- filter_outlier %>%
  ggplot(aes(sample = reading_time_sqrt)) +
    stat_qq() +
    stat_qq_line(colour = "red") +
    labs(x = "Theoretical Quantiles", y = "Sample Quantiles") +
    ggtitle("Reading_time_sqrt, qq plot") +
    theme_bw()

sqrt_hist <- filter_outlier %>%
  ggplot(aes(reading_time_sqrt)) +
      geom_histogram(aes(y = ..density..), colour = "white", fill = "black") +
      stat_function(fun = dnorm, args = list(mean = mean(filter_outlier$reading_time_sqrt),
             sd = sd(filter_outlier$reading_time_sqrt)), colour = "red", size = 1) +
      ggtitle("Reading_time_sqrt, histogram") +
      theme_bw()

divid_qq <- filter_outlier %>%
  ggplot(aes(sample = reading_time_divid)) +
    stat_qq() +
    stat_qq_line(colour = "red") +
    labs(x = "Theoretical Quantiles", y = "Sample Quantiles") +
    ggtitle("reading_time_divid, qq plot") +
    theme_bw()

divid_hist <- filter_outlier %>%
  ggplot(aes(reading_time_divid)) +
      geom_histogram(aes(y = ..density..), colour = "white", fill = "black") +
      stat_function(fun = dnorm, args = list(mean = mean(filter_outlier$reading_time_divid),
             sd = sd(filter_outlier$reading_time_divid)), colour = "red", size = 1) +
      ggtitle("reading_time_divid, histogram") +
      theme_bw()

ggarrange(log_qq,log_hist,sqrt_qq,sqrt_hist,divid_qq,divid_hist, ncol = 2, nrow = 3)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
round(pastecs::stat.desc(cbind(
  "Reading_time" = df$reading_time,
  "Log_Reading_time" = filter_outlier$reading_time_log,
  "Sqrt_Reading_time" = filter_outlier$reading_time_sqrt,
  "Divid_Reading_time" = filter_outlier$reading_time_divid), basic = FALSE, norm = TRUE), digits = 2)
```

```
## Warning in cbind(Reading_time = df$reading_time, Log_Reading_time =
## filter_outlier$reading_time_log, : number of rows of result is not a multiple of
## vector length (arg 2)
```

```
##              Reading_time Log_Reading_time Sqrt_Reading_time Divid_Reading_time
## median               0.40            -0.92              0.63               2.52
## mean                 0.49            -0.86              0.67               2.56
## SE.mean              0.01             0.01              0.00               0.02
## CI.mean.0.95         0.02             0.02              0.01               0.04
## var                  0.26             0.18              0.02               1.02
## std.dev              0.51             0.42              0.15               1.01
## coef.var             1.05            -0.49              0.23               0.39
## skewness            21.04             0.57              1.27               0.86
## skew.2SE           201.96             5.45             12.19               8.27
## kurtosis           580.90             0.47              2.17               2.00
## kurt.2SE          2789.04             2.26             10.43               9.59
## normtest.W           0.28             0.97              0.91               0.96
## normtest.p           0.00             0.00              0.00               0.00
```

We can see that the transformed data is still not normally distributed. We check if the variables meet the assumptions of normality:

```
round(pastecs::stat.desc(cbind(
  "Reading_time" = df$reading_time,
  "nlet" = df$nlet,
  "kf_freq" = df$kf_freq), basic = FALSE, norm = TRUE), digits = 2)
```

```
##              Reading_time  nlet      kf_freq
## median               0.40  3.00      5262.00
## mean                 0.49  3.37     14724.41
## SE.mean              0.01  0.03       454.93
## CI.mean.0.95         0.02  0.07       892.14
## var                  0.26  2.48 456977449.95
## std.dev              0.51  1.57     21377.03
## coef.var             1.05  0.47         1.45
## skewness            21.04  1.33         1.72
## skew.2SE           201.96 12.75        16.48
## kurtosis           580.90  2.88         1.79
## kurt.2SE          2789.04 13.83         8.61
## normtest.W           0.28  0.88         0.69
## normtest.p           0.00  0.00         0.00
```

The variables are not normally distributed either.

Correlation

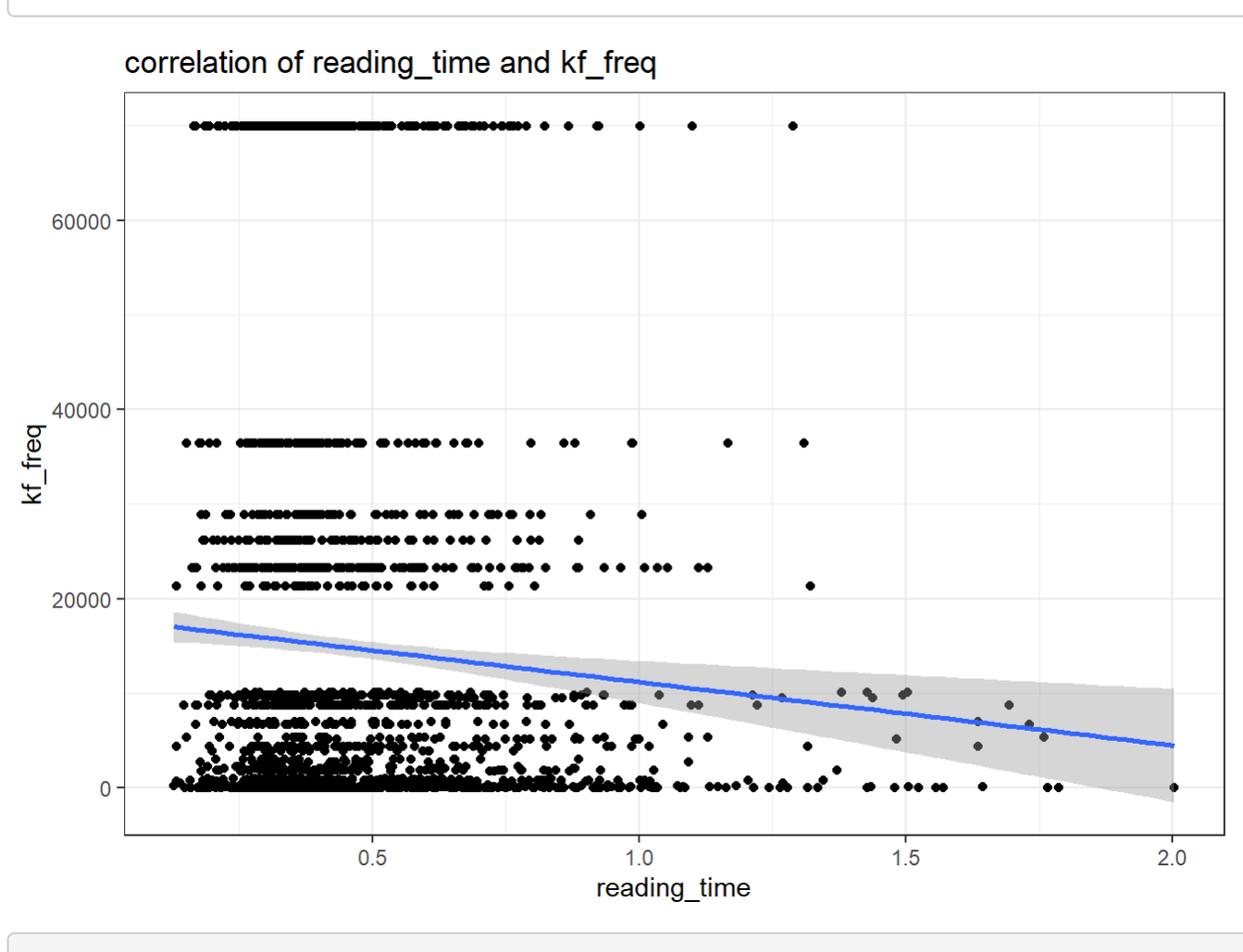
Now we can explore if a relation exists between reading times and length, frequency and ordinality of the words using correlation analysis and scatter plots with linear regression lines.

Assumptions of parametric tests: 1. Data are normally distributed 2. Variance is homogeneous across samples, groups, levels of a variable 3. Data are at least at the interval level 4. Data are independent from each other across participants or across sessions within participants

Since our data does not fit these assumptions, we need to use a non-parametric correlation test. Thus, our choice was Spearman's correlation test.

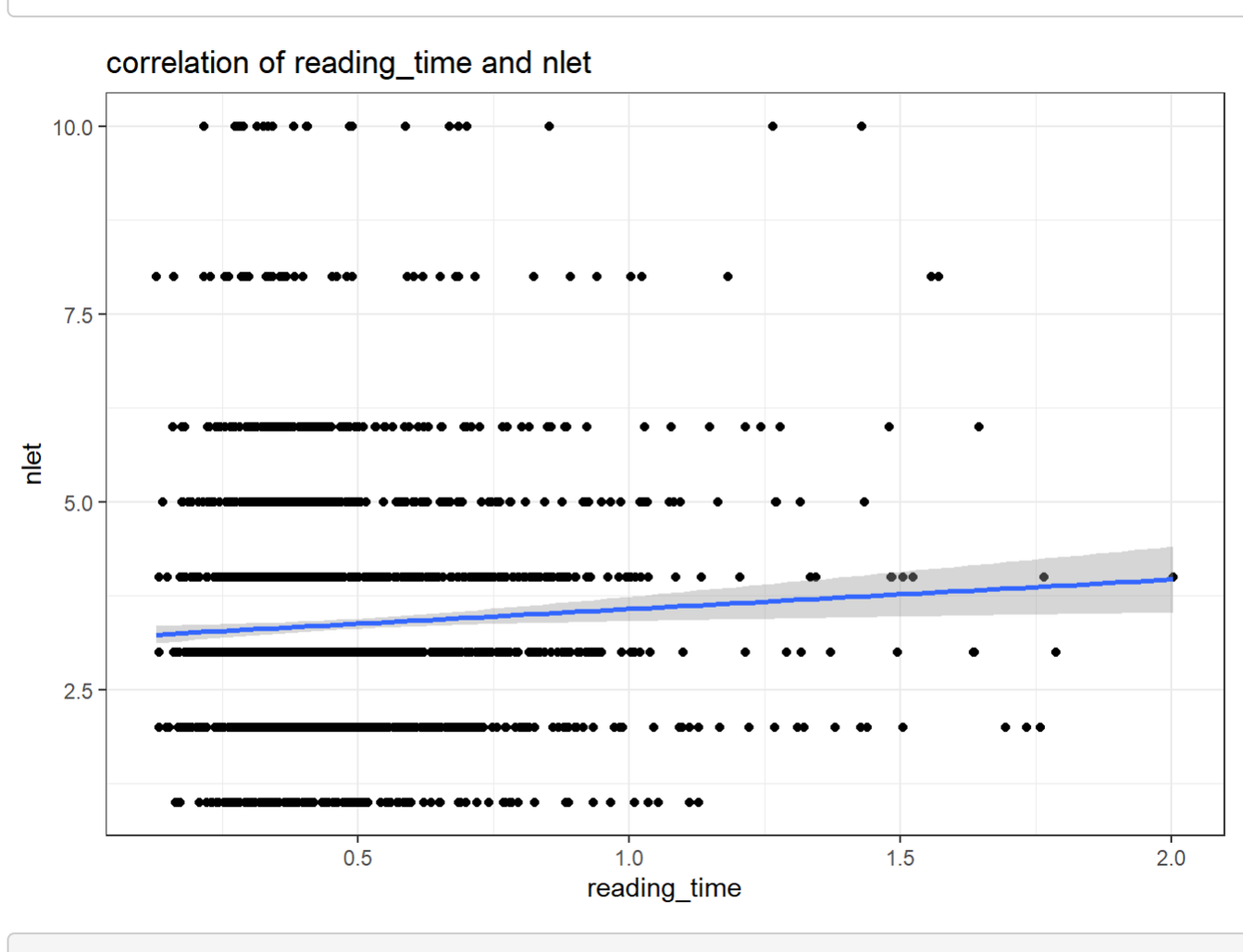
```
cor_kf_freq <- filter_outlier %>%
  ggplot() +
  aes(reading_time, kf_freq) +
  geom_point() +
  geom_smooth(method = "lm")+
  ggtitle("correlation of reading_time and kf_freq") +
    theme_bw()



cor_nlet <- filter_outlier %>%
  ggplot() +
  aes(reading_time, nlet) +
  geom_point() +
  geom_smooth(method = "lm")+
   ggtitle("correlation of reading_time and nlet") +
    theme_bw()



cor_word_number <- filter_outlier %>%
  ggplot() +
  aes(reading_time, word_number) +
  geom_point() +
  geom_smooth(method = "lm")+
   ggtitle("correlation of reading_time and word_number") +
    theme_bw()


cor_kf_freq
```

```
## `geom_smooth()` using formula 'y ~ x'
```



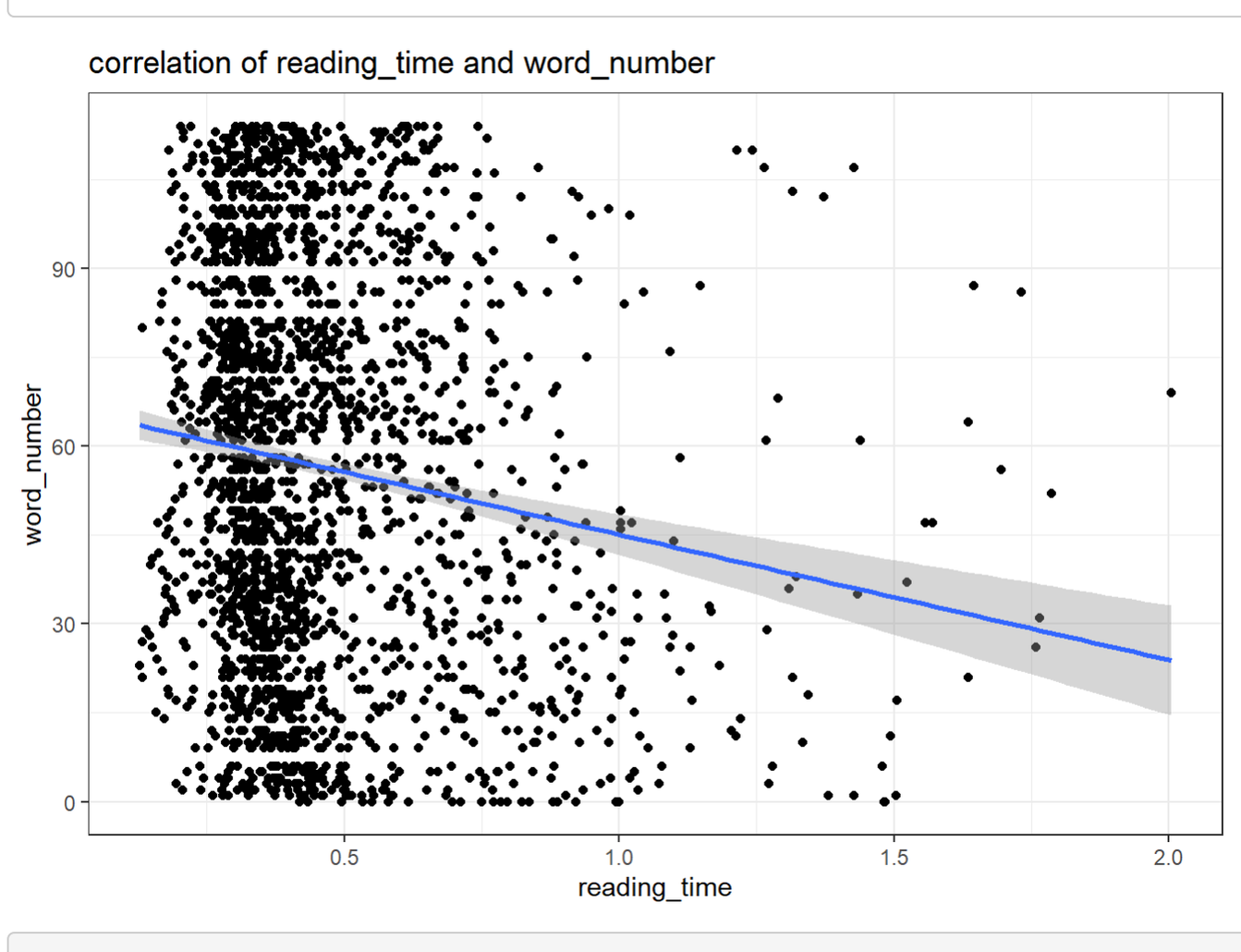

```
cor_nlet
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
cor_word_number
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
cor.test(df$reading_time, df$kf_freq, method = "spearman")
```

```
## Warning in cor.test.default(df$reading_time, df$kf_freq, method = "spearman"):
## Cannot compute exact p-value with ties
```

```
##
##  Spearman's rank correlation rho
##
## data:  df$reading_time and df$kf_freq
## S = 1848846854, p-value = 0.1517
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##         rho
## -0.03051679
```

```
cor.test(df$reading_time, df$nlet, method = "spearman")
```

```
## Warning in cor.test.default(df$reading_time, df$nlet, method = "spearman"):
## Cannot compute exact p-value with ties
```

```
##
##  Spearman's rank correlation rho
##
## data:  df$reading_time and df$nlet
## S = 1748432711, p-value = 0.2319
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.0254524
```

```
cor.test(df$reading_time, df$word_number, method = "spearman")
```

```
## Warning in cor.test.default(df$reading_time, df$word_number, method =
## "spearman"): Cannot compute exact p-value with ties
```

```
##
##  Spearman's rank correlation rho
##
## data:  df$reading_time and df$word_number
## S = 2.018e+09, p-value = 3.999e-09
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.1248057
```

From the scatterplot and the Spearman's correlation test, we can see that there is no linear relation between the variables, except for the variable word_number. There is a statistically significant linear relation between word_number and reading_time (***p-value<0.001). Furthermore, reading time seems to decrease as the experiment progresses. Looking at the rho value: reading_time and kf_freq: rho = -0.03 means that there is a weak or no relationship (p = 0.15) reading_time and nlet: rho = 0.02 means that there is a weak or no relationship (p = 0.23) reading_time and word_number: rho = -0.12 means that there is a weak or no relationship (p = 3.9e-09)*** In conclusion, there is a weak (or no) relationship between reading time and word number, as the p-value signals its significance

Hypothesis testing

Assumption testing

Normality

We create a new data frame containing the mean reading time for "salient word" and the word right after. We remove outliers of 3 standard deviations.

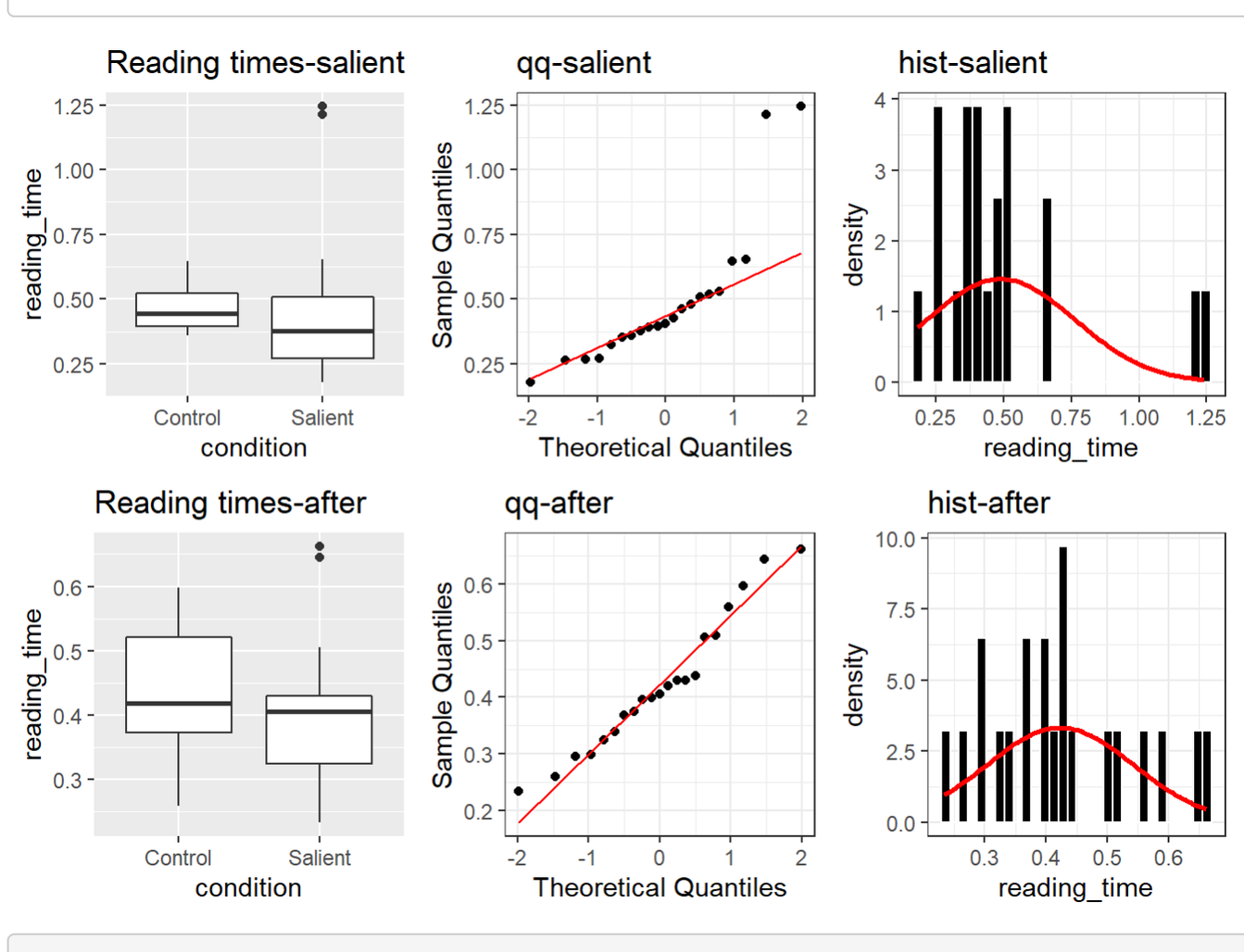
```
hyp_df <- filter_outlier %>%
  mutate(control = str_detect(toupper(File), "CONTROL")) %>%
  mutate(salience = (salience == "True")) %>%
  mutate(condition = ifelse(control, "Control", "Salient"))

hyp_df <- hyp_df[, c(5, 6, 27)]

salient_df <- hyp_df %>% filter(salience)%>%
  filter(reading_time > mean(reading_time)-sd(reading_time)*3) %>%
  filter(reading_time < mean(reading_time)+sd(reading_time)*3)


after_salient_df <- hyp_df %>%  filter(lag(salience))%>%
  filter(reading_time > mean(reading_time)-sd(reading_time)*3) %>%
  filter(reading_time < mean(reading_time)+sd(reading_time)*3)
```

Now we need to examine if our data meets the assumptions for a parametric t-test, but we need to check the assumptions separately for the two conditions, because they represent data from different groups. The assumptions: Assumes the dependent variable are normally distributed and that the variances are equal

Checking for normality

```
box_salient <- salient_df %>%
  ggplot(aes(x = condition, y = reading_time)) +
  geom_boxplot() +
  ggtitle("Reading times-salient")

box_after_salient <- after_salient_df %>%
  ggplot(aes(x = condition, y = reading_time)) +
  geom_boxplot() +
  ggtitle("Reading times-after")

hist_salient <- salient_df %>% ggplot(aes(reading_time)) +
      geom_histogram(aes(y = ..density..), colour = "white", fill = "black") +
      stat_function(fun = dnorm, args = list(mean = mean(salient_df$reading_time),
             sd = sd(salient_df$reading_time)), colour = "red", size = 1) +
      ggtitle("hist-salient") +
      theme_bw()

qq_salient <- salient_df %>%
  ggplot(aes(sample = reading_time)) +
    stat_qq() +
    stat_qq_line(colour = "red") +
    labs(x = "Theoretical Quantiles", y = "Sample Quantiles") +
    ggtitle("qq-salient") +
    theme_bw()

hist_after_salient <- after_salient_df %>% ggplot(aes(reading_time)) +
      geom_histogram(aes(y = ..density..), colour = "white", fill = "black") +
      stat_function(fun = dnorm, args = list(mean = mean(after_salient_df$reading_time),
             sd = sd(after_salient_df$reading_time)), colour = "red", size = 1) +
      ggtitle("hist-after") +
      theme_bw()

qq_after_salient <- after_salient_df %>%
  ggplot(aes(sample = reading_time)) +
    stat_qq() +
    stat_qq_line(colour = "red") +
    labs(x = "Theoretical Quantiles", y = "Sample Quantiles") +
    ggtitle("qq-after") +
    theme_bw()

ggarrange(
  box_salient, qq_salient, hist_salient,
  box_after_salient, qq_after_salient, hist_after_salient,
  ncol = 3, nrow = 2
)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
pastecs::stat.desc(
  cbind(
    "Reading time for salient words" = salient_df$reading_time,
    "Reading time for words right after" = after_salient_df$reading_time

  ), basic = FALSE, norm = TRUE, desc = FALSE) %>%
  round(digits = 2)
```

```
##            Reading time for salient words Reading time for words right after
## skewness                             1.73                               0.45
## skew.2SE                             1.73                               0.44
## kurtosis                             2.27                              -0.78
## kurt.2SE                             1.17                              -0.40
## normtest.W                           0.76                               0.96
## normtest.p                           0.00                               0.42
```

From the normality test, we can see that reading time for the words right after the salient word is approximately normally distributed (skew.2SE<1, kurt.2SE<1, Shapiro Wilk p-value>0.05). On the other hand the reading time for the salient word is not normally distributed (skew.2SE>1, kurt.2SE>1, Shapiro Wilk p-value<0.05). Thus, only reading times for the subsequent words meet the first assumption for the student's t-test.

Homoscedasticity

Now we perform Levene's test, which is an inferential statistic used to evaluate the equality of variances for a variable determined for two or more groups. *H0:* no difference in the variance among the two groups *H1:* a difference in the variance among the two groups

```
my_data = stack(list(salient_df=salient_df$reading_time, after_salient_df=after_salient_df$reading_time))
leveneTest(values ~ ind, my_data)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  1  2.0283 0.1621
##       40
```

The p-value of the test is 0.16, which is more than our significance level of 0.05. Therefore, we reject the alternative hypothesis and conclude that the variance among the two groups must be equal, thus the data is homoscedastic.

t-test

Therefore, our data does not meet the assumptions for a parametric t-test. Therefore we choose to perform a non-parametric t-test from the WRS2 package, that allow us to "trim" some part of data from tails of the distrubution in order to deal with non-normal distrubutions. Our hypotheses: *H0 (null hypothesis)* = No difference in the mean reading times in the two conditions of our reading experiment *H1 (alternative hypothesis)* = There is a difference in the mean reading times in the two conditions of our reading experiment

```
WRS2::yuen(reading_time~condition, data=salient_df)
```

```
## Call:
## WRS2::yuen(formula = reading_time ~ condition, data = salient_df)
##
## Test statistic: 0.7066 (df = 11.23), p-value = 0.49421
##
## Trimmed mean difference:  0.04926
## 95 percent confidence interval:
## -0.1038     0.2023
##
## Explanatory measure of effect size: 0.31
```

```
WRS2::yuen(reading_time~condition, data=after_salient_df)
```

```
## Call:
## WRS2::yuen(formula = reading_time ~ condition, data = after_salient_df)
##
## Test statistic: 0.9105 (df = 10.71), p-value = 0.38258
##
## Trimmed mean difference:  0.04716
## 95 percent confidence interval:
## -0.0672     0.1615
##
## Explanatory measure of effect size: 0.27
```

Conclusion

It can be concluded that there is no statistically significant difference between the means of reading time in the two conditions (p-value > 0.05). This is regardless of whether one assess reading time of the salient word or the word after. Therefore, we accept the null hypothesis; that there is no difference in the mean reading times in the two conditions of our experiment.

```
#### Word-by-word reading time experiment - Portfolio 2 ####
# Studygroup 5 (Maja, Niels, Marton, Laurtis & Sarah S.)
# October 26, 2021

from psychopy import visual, core, event, data, gui
import pandas as pd

box = gui.Dlg(title = "Choose condition")
box.addField("Condition: ", choices=["Control", "Salient"])
box.show()
if box.OK:
    Condition = box.data[0]
elif box.Cancel:
    core.quit()

box = gui.Dlg(title = "Reading experiment")
box.addField("Name: ")
box.addField("Age: ")
box.addField("Gender: ", choices=["Female", "Male", "Other" ])
box.show()
if box.OK:
    name = box.data[0]
    age = box.data[1]
    gender = box.data[2]
elif box.Cancel:
    core.quit()

win = visual.Window(fullscr = True, color = "pink")

stopwatch = core.Clock()

date = data.getDateStr()

if Condition == "Control":
    f = open("control.txt")
else:
    f = open("salient.txt")

text = f.read()
f.close()
words = text.split()

columns = ['name','age','gender', 'reading_time', 'word', 'salience']
logfile = pd.DataFrame(columns=columns)

logfile_name = "logfiles/logfile_{}_{}.csv".format(name,Condition)

instruction = '''
Welcome to the Reading Experiment!

In a moment, you will be presented with a short text, which you will read through one word at a time.
Press the space bar to move on to the next word. Read at your own pace.

Press any key to start the experiment.
'''

goodbye = '''
The experiment is done. Thank you for your participation!'''

def msg(txt):
    message = visual.TextStim(win, text = txt, alignText = "left", height = 0.08)
    message.draw()
    win.flip()
    event.waitKeys()

def present_word(word):
    stimulus = visual.TextStim(win, word)
    stimulus.draw()
    stopwatch.reset()
    win.flip()
    keys = event.waitKeys(keyList = ["escape", "space"])
    reading_time = stopwatch.getTime()
    if keys == ["escape"]:
        core.quit()
    else:
        return reading_time
```

```
def salient(word):
    return word[0] == "*"

msg(instruction)

for word in words:
    salience = salient(word)
    if salience:
        word = word[1:]
    reading_time = present_word(word)

    logfile = logfile.append({
        'name': name,
        'age': age,
        'gender': gender,
        'word': word,
        'salience': salience,
        'reading_time': reading_time}, ignore_index = True)

logfile.to_csv(logfile_name)

msg(goodbye)
```

Portfolio 3

Does priming influence word association and semantic fields?

This paper investigates how priming may affect people's semantic fields. A semantic field is a set of words related in meaning (Collins & Quillian, 1969), and this study is motivated by curiosity towards how people are affected by their surrounding environment and how their semantic networks function with regards to knowledge representation (Meyer et al.,1971). Semantic priming is the observed effect in response to a target word when preceded by a semantically related priming word, compared to an unrelated word. (Jong-Sun Lee et al., 2014).

Furthermore, this paper will contribute to the knowledge about how humans are influenced by priming, specifically which words people associate with the word "Garden", and how two different priming conditions: Botanical and Crops, may prime them in a direction to a specific semantic network(Katharina Sass et al., 2009). Based on this curiosity, the two following hypotheses have been made:

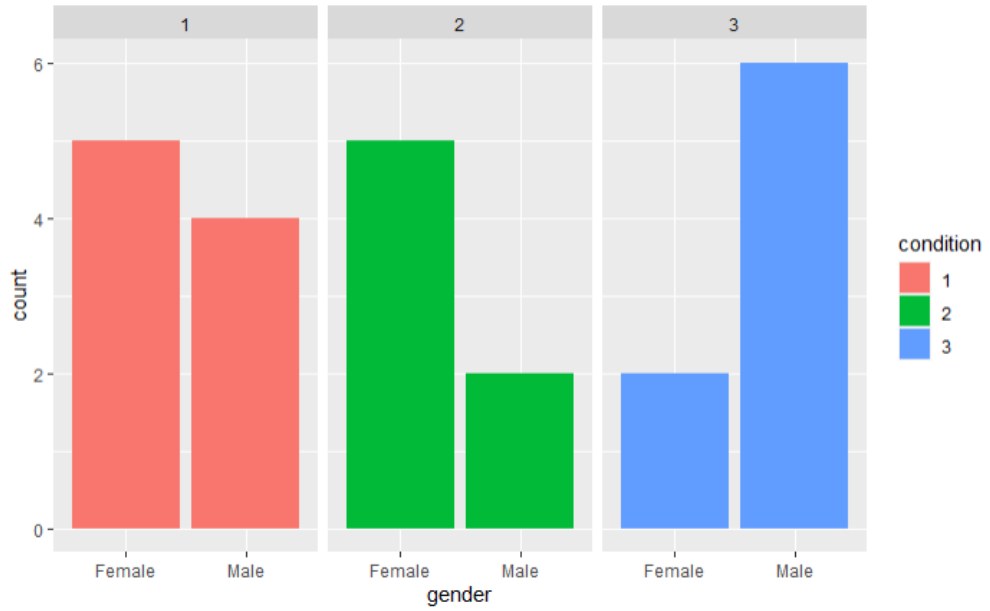
The Null-Hypothesis (H0): there is no significant difference in the semantic fields between the control condition and either the botanical priming condition or the crops priming condition

The Alternative Hypothesis (H1): there is a significant difference in the semantic fields between the control condition and either the botanical priming condition or the crops priming condition

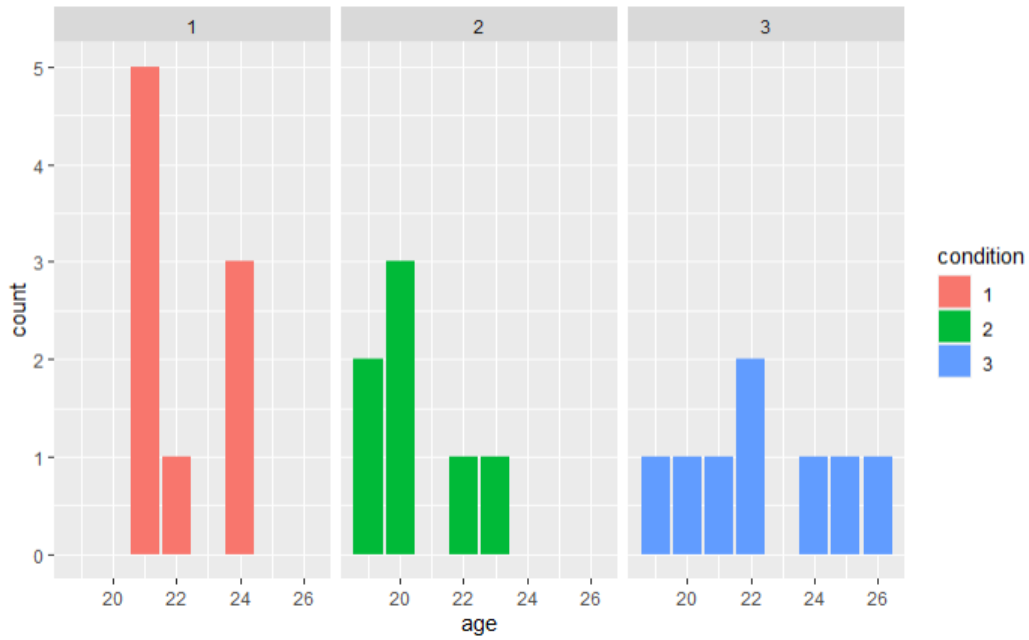
The experimental procedure to test the hypotheses, will be for participants to be exposed to either the control condition or one of the two priming conditions, and then they will have to write down as many words as possible that they associate with the stimuli presented to them.

Methods

***Participants*:** The experiment had 24 participants that were a mix between native and non-native Danish speakers, both female and male. All participants were university students, whose age varied between 19 and 25 years, with a mean age of 22 and a SD of 1.97.



Plot 1: Distribution of gender in the three conditions



Plot 2: Distribution of age in the three conditions

***Materials/Stimuli*:** The experiment had three conditions:

1. No priming; the only word that the participant saw was "Garden". This was the control condition.
2. Botanical priming; here the participants were given the hint "e.g. tree & flower".
3. Crops priming; here the participants were given the hint "e.g. carrot & tomato".

The priming categories, "Botanical" and "Crops" were chosen as conditions as they vary in their respective semantic distance from the primary semantic field "Garden". The chosen botanical example words had a semantic distance to "Garden" of M = 99.7, whereas the crops example words had a semantic distance to "Garden" of M = 144.2. These semantic distances were estimated using the Euclidean distance method.

Thus, to accept the alternative hypothesis H1, condition 2 and 3 must show a priming effect on the participants' semantic fields, for there to be a significant semantic distance compared to the control condition.

Procedure: The experiment was conducted in PsychoPy and the script was in English.

First, the participants were presented with a dialogue box, where they had to write their participant ID, age, gender and native language. In the dialogue box, the conductors of the experiment also chose one of the three conditions. Then the participants were presented with an introduction text.

Next, the participants were presented with an instruction text, which stated:

If condition = 1: *"Please write words you associate with the word: "Garden""*

If condition = 2: *"Please write words you associate with the word: "Garden" e.g., tree & flower"*

If condition = 3: *"Please write words you associate with the word: "Garden" e.g., carrot & tomato"*

The instruction text above was shown briefly for five seconds, which ensured that the participants only had limited time to come up with words associated with "Garden" before the word-association task began. Then the participants had 20 seconds to write as many words as possible that they

associate with the word "Garden". As this was a between-subjects independent measures experiment, each participant only went through one of the three conditions one time.

Analysis and results

The spread of the participants' semantic fields was measured as follows: For each participant the mean semantic distance of all the possible two-word combinations of the words they wrote was calculated. E.g. if the participant wrote the words "dirt", "tree" and "grass", the value would be: *mean(s("dirt", "tree"), s("dirt", "grass"), s("tree", "grass"))*, where *s(<word1>, <word2>)* represents the Euclidean distance of the vector mappings of word1 and word2 from the *EN_100k* database. This was chosen to be the outcome variable and the predictor variable was the different conditions of the experiment.

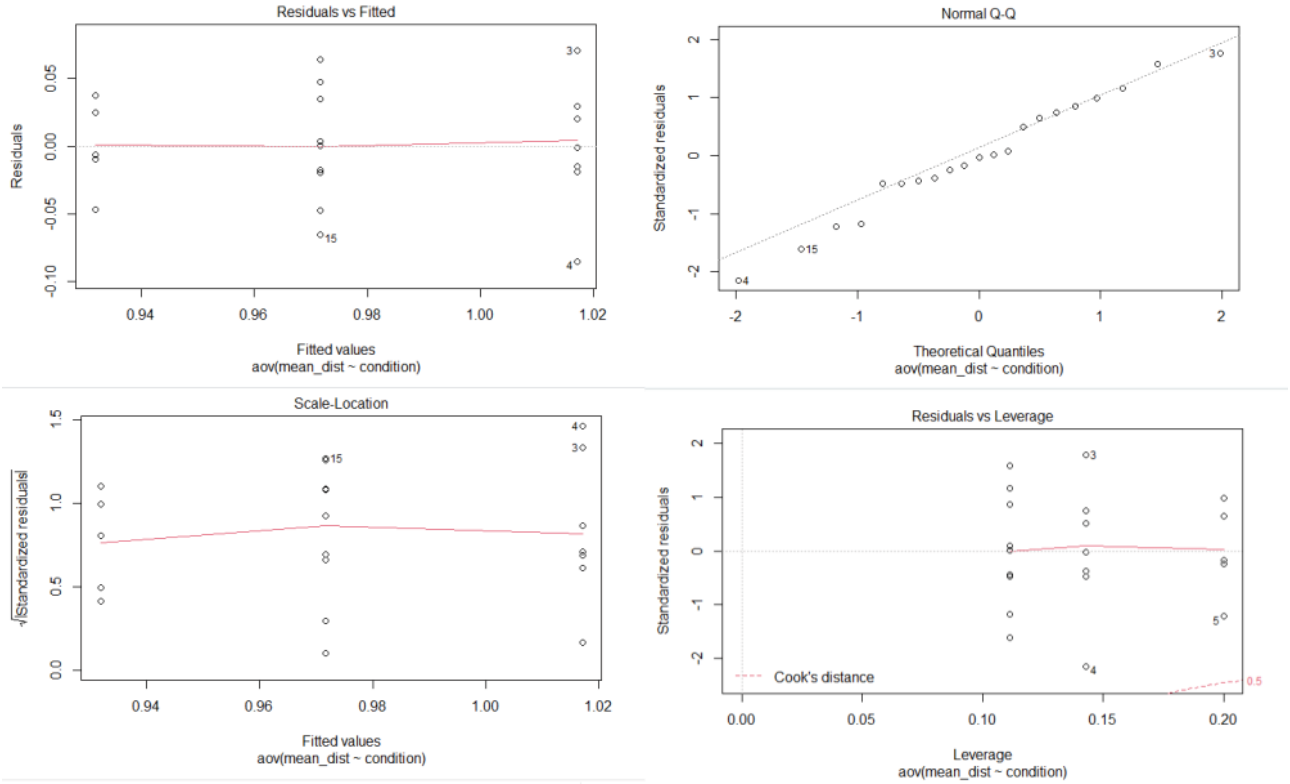
As the experiment was based on the participants manually writing the words, it inevitably led to misspellings and plural forms, which made it difficult to measure the semantic distance, as the calculations were limited to entries in the chosen semantic distance database. Therefore, misspellings and pluralism were removed from the dataset.

The data did not meet the assumptions of ANOVA. Therefore *log(), sqrt()* and inverse transformations were conducted on the data, but this proved to be ineffective. Three highly influential data points were removed from the data set. The influence metric used for removal of the aforementioned points was their Cook's distance with a threshold of 4/N, where N is the number of observations. All the data points removed were from condition 3, which reduced the number of participants in condition 3 from eight to five participants. Condition 1 had nine participants and condition 2 had seven participants.

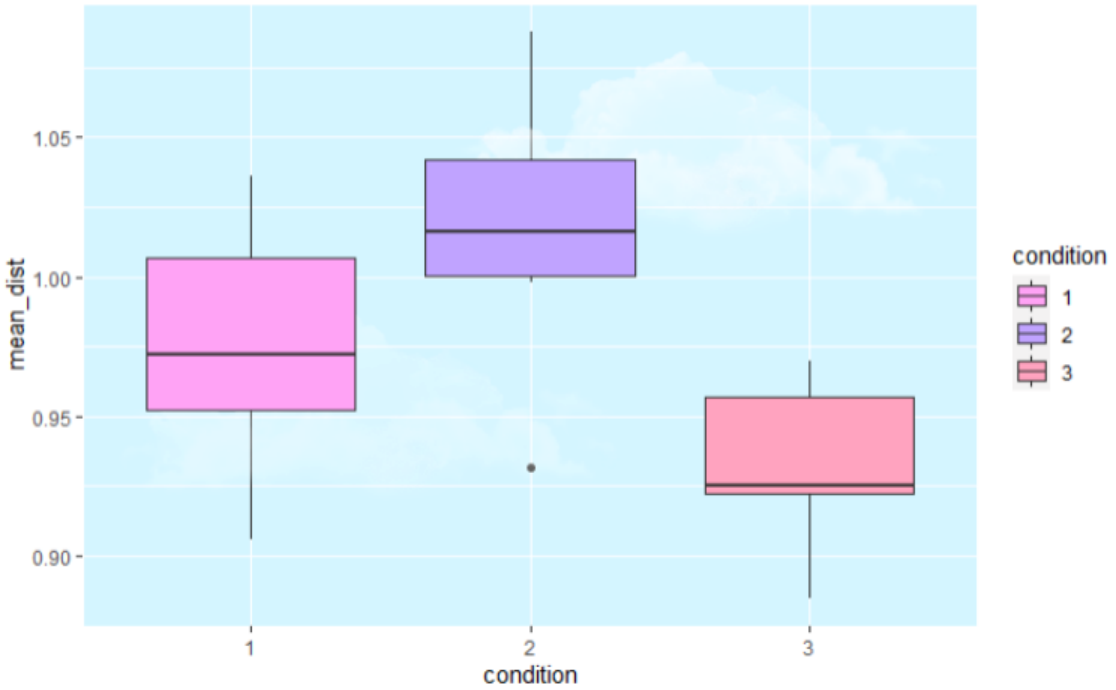
Then, the ANOVA test was performed, and the results showed that there was a significant difference between the three conditions of the experiment, $F(2) = 5.833$, $p = .0111$. Individual pairwise effects were tested with a post-hoc Bonferroni-corrected t-test. Only the difference between condition 2 and condition 3 was statistically significant ($p = .01$).

The mean and SD of semantic spread for the three conditions:

	Condition 1	**Condition 2**	**Condition 3**
Mean	0.97	1.02	0.93
SD	0.04	0.05	0.03



Plot 3: Plots over assumptions



Plot 4: Boxplot of the semantic distance from "garden" in the three conditions

Discussion

The results showed that there was no statistically significant difference in the semantic fields between the control condition and the priming conditions, meaning that the Null-Hypothesis H0 must be accepted and therefore the Alternative Hypothesis H1 was rejected. However, the results showed a statistically significant difference between priming condition 2 and 3, which was not expected. This could be explained by how the *EN_100k* database lists semantic distances, as it was not specifically made for "Garden" contexts. In the experiment, many of the participants submitted the word "flower", which could be interpreted as "flower" being highly associated with the word "Garden" in the participants' semantic fields. However, in the database the semantic distance for "flower" and "Garden" is rather large. This indicates that even if the words submitted by the participants are associated with the word "Garden" in their semantic field, the semantic similarity between them would be disregarded in the database which might have had an influence on the result. In a further study, it would be wise to generate a semantic distance database specifically made for garden words.

Also, "in general the more priming stimuli that the participant is presented to, the stronger the obtained priming effects" (Harry Reis, 2000). Hence, two priming words were chosen and not only one. One could argue that there should be more than two priming words in order for the priming effect to be stronger. However, the priming words were limited to two to ensure that the participants came up with new words associated with "garden" themselves, and not just repeated the presented priming words. This would be problematic for the analysis; were the participants primed or did they just remember the priming words and wrote them down.

Another choice that was made was having two priming conditions and not just one. This choice was made, as there were explicit hypotheses about the two conditions having differential effects. However, this was not analyzed, which could be a limitation to this study, since it could have had an significant impact on the results.

Conclusion

This study found that, in accordance with the Null-Hypothesis, H0, priming did not have a statistically significant effect on semantic fields compared to the control condition. However, there was a statistically significant difference between the two priming conditions, "Botanical" and "Crops", which indicated that priming might have an influence on word association and the structure of semantic fields, just not in the way that the hypotheses of this study anticipated.

References:

Harry T. Reis, Charles M. (2000), Handbook of Research Methods in Social and Personality Psychology - p. 264

Meyer, D. E., & Schvaneveldt, R. W. (1971). Facilitation in recognizing pairs of words: Evidence of a dependence between retrieval operations. Journal of Experimental Psychology, 90(2), 227–234. https://doi.org/10.1037/h0031564

Allan M. Collins, M. Ross Quillian (1969). Retrieval time from semantic memory. Journal of Verbal Learning and Verbal Behavior. 8(2), 240-247. 10.1016/S0022-5371(69)80069-1

Jong-Sun Lee et al (2014). The effect of word imagery on priming effect under a preconscious condition: An fMRI study. https://doi.org/10.1002/hbm.22512

Katharina Sass, Sören Krach, Olga Sachs, Tilo Kircher (2009). Lion – tiger – stripes: Neural correlates of indirect semantic priming across processing modalities, NeuroImage, 45(1), 224-236, https://doi.org/10.1016/j.neuroimage.2008.10.014.

Portfolio 4

BSc in Cognitive Science, Aarhus University
Methods 1 E2021

PORTFOLIO 4
Mixed-effects models and logistic regression
Deadline: December 2nd, 2021

The portfolio uses two data sets: The "**Breakage Angle of Chocolate Cakes**" data set and the "**Titanic**" data set. The data sets include the following variables:

Cake:

- **replicate**: a factor with levels 1 to 15 indicating # replication of test
- **recipe**: a factor with levels A, B, and C for each of three different recipes
- [**temperature**: disregard]
- **angle**: a numeric vector giving the angle at which the cake broke
- **temp**: a numeric value of the baking temperature (degrees F)

Titanic:

- **Survived**: a numeric value indicating whether each participant survived the incident or not
- **Pclass**: a currently numeric variable with levels 1 to 3 for 1st, 2nd, and 3rd class
- **Name**: a character variable with passenger names
- **Sex**: a character variable with two levels (male/female)
- **Age**: a numeric value indicating passenger age
- [**Siblings/Spouses Aboard**: disregard]
- [**Parents/Children Aboard**: disregard]
- [**Fare**: disregard]

Analysis 1: Cake breakage

To predict the angle at which cake break, I fitted a linear mixed-effect model to predict *angle* as the outcome variable. I started with 3 models and found temperature to be the predictor variable. Recipe turned out to be a random slope and replicate to be the random intercept:

Cake_1 = angle ~ temp + (1+recipe|replicate)

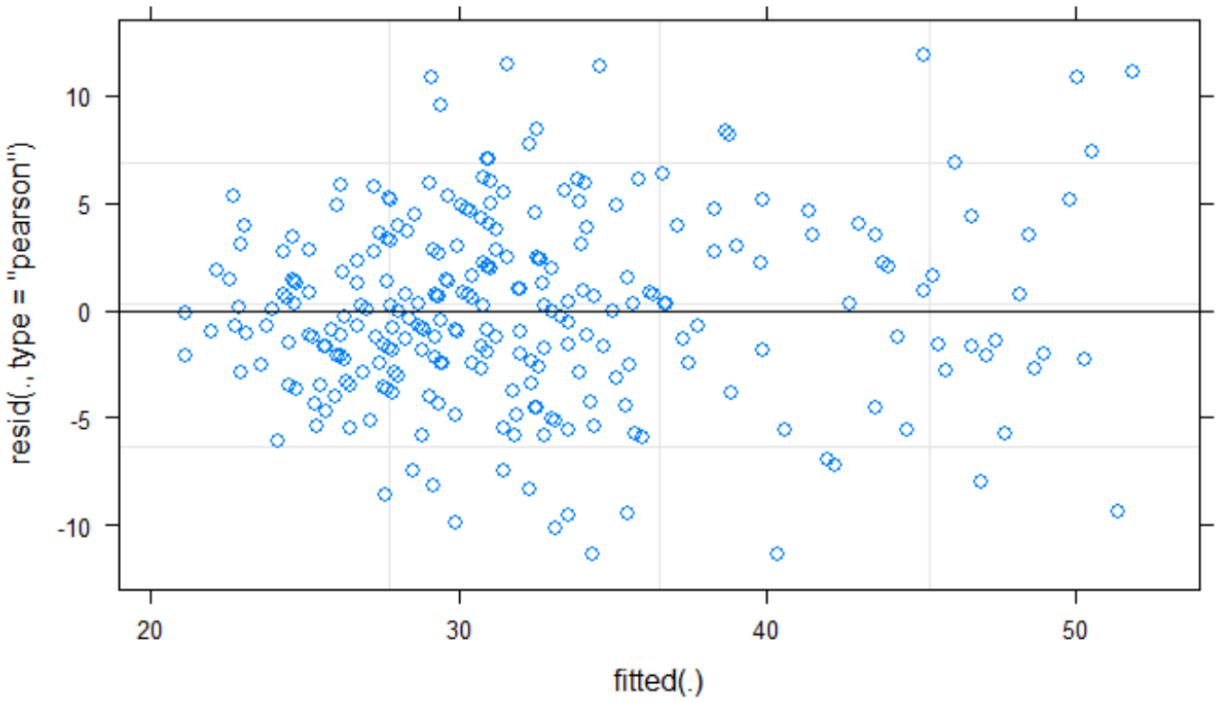
This model got chosen as it had the lowest AIC and highest conditional R^2. This means, that the angle at which cakes break is significantly predicted by temperature (beta = 0,158, SD = 0,016, t = 9,8, p = < 0.001). When temperature increases, the angle that the cake breaks at increases.

Models:	**AIC**	**R2c**
Cake_1 = angle ~ temp + (1+recipe|replicate)	*1666*	*0.702*
Cake_2 = angle ~ temp + recipe + (1|replicate)	1674	0.659
Cake_3 = angle ~ temp * recipe +(1|replicate)	1678	0.660
Cake_4 = angle ~ temp * recipe +(1|replicate) +(1|recipe)	1677	0.658

Summary output:

```
Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's method ['lmerModLmerTest']
Formula: angle ~ temp + (1 + recipe | replicate)
   Data: cake

     AIC      BIC   logLik deviance df.resid
  1666.2   1698.6   -824.1   1648.2      261

Scaled residuals:
     Min       1Q   Median       3Q      Max
-2.51095 -0.56465 -0.01979  0.62483  2.62895

Random effects:
 Groups    Name        Variance Std.Dev. Corr
 replicate (Intercept) 24.981   4.998
           recipeB      8.513   2.918    0.42
           recipeC     15.347   3.918    0.31 0.99
 Residual              20.477   4.525
Number of obs: 270, groups:  replicate, 15

Fixed effects:
            Estimate Std. Error        df t value Pr(>|t|)
(Intercept)  1.77214    3.50194 219.36537   0.506    0.613
temp         0.15803    0.01613 239.97848   9.800   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Correlation of Fixed Effects:
     (Intr)
temp -0.921
```

Check assumptions:



There is compact and unsystematic spread in the plot therefore the assumptions are fulfilled.

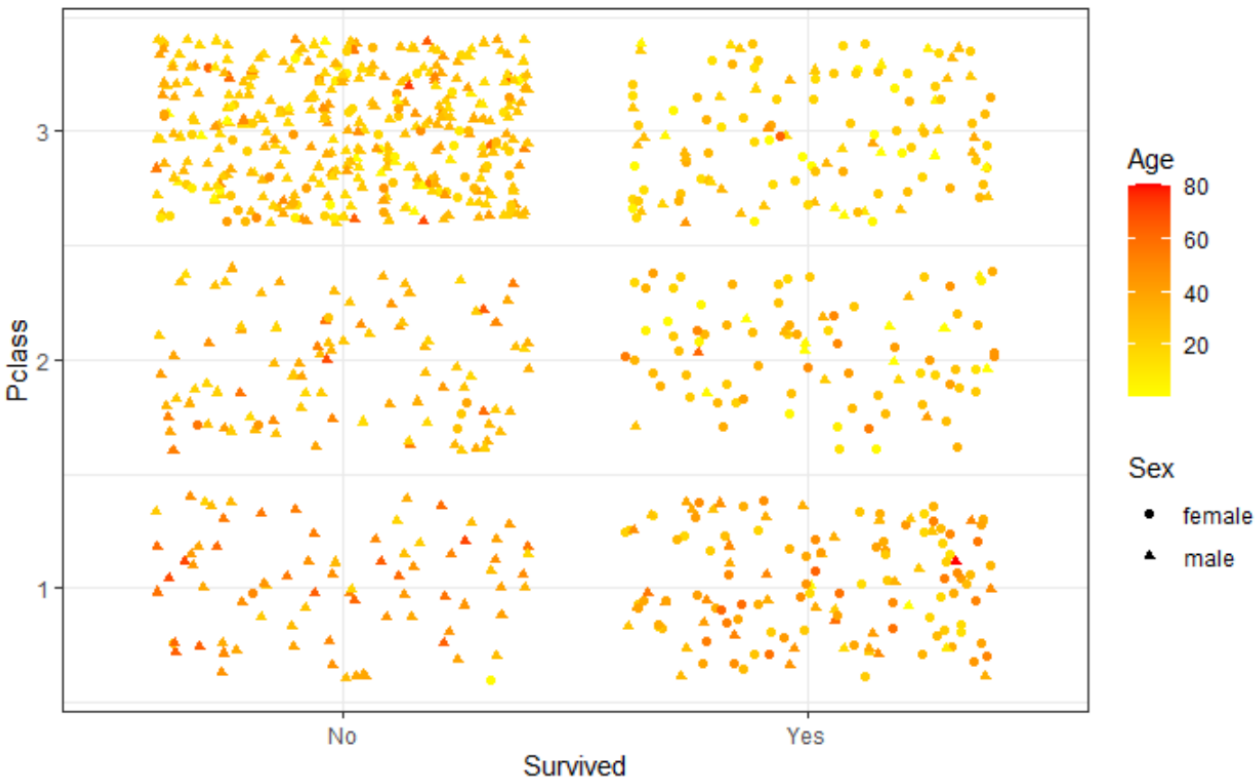
Analysis 2: Titanic survival

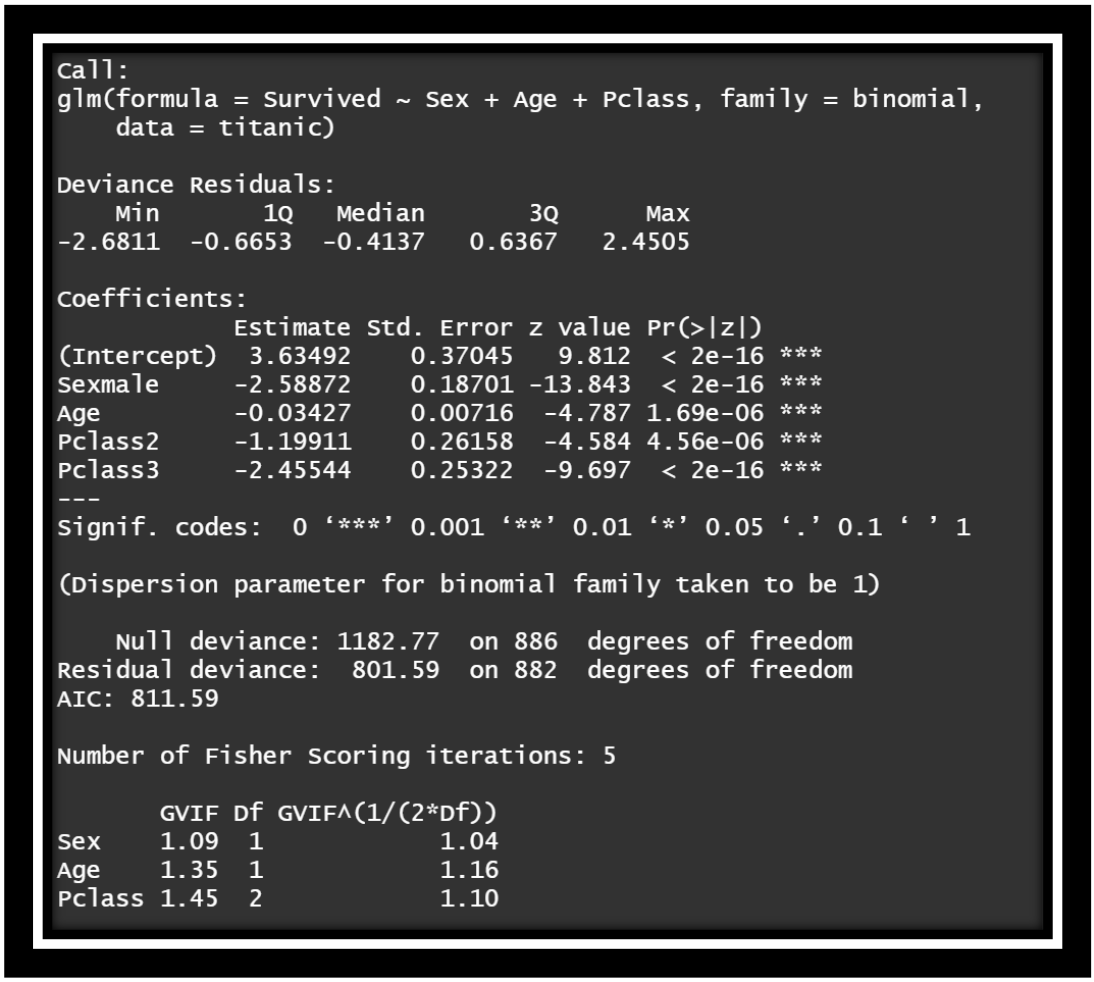
To predict the survival rate of titanic passengers I created a generalized logistic model with binomial outcomes on the titanic data set, after testing other plausible models:

Survived ~ Sex + Age + Passenger_class

As seen in figure 1 '*summary of GLM'*, the model has a baseline passenger of a *first-class female at age 0*, and all other predictors has a negative log-odds, meaning everyone has a smaller likelihood of surviving than the baseline passenger. All predictors have a significant p-value < 0.01.

When trained on a training dataset (seed (666) in r, p 0.8) the prediction accuracy on the remaining test dataset was 78 %, see figure 2 'Confusion matrix'. The training dataset had a R2 MacFadden of 0.409 and the test dataset a R2 MacFadden of 0.376.





```
Call:
glm(formula = Survived ~ Sex + Age + Pclass, family = binomial,
    data = titanic)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.6811  -0.6653  -0.4137   0.6367   2.4505

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  3.63492    0.37045   9.812  < 2e-16 ***
Sexmale     -2.58872    0.18701 -13.843  < 2e-16 ***
Age         -0.03427    0.00716  -4.787 1.69e-06 ***
Pclass2     -1.19911    0.26158  -4.584 4.56e-06 ***
Pclass3     -2.45544    0.25322  -9.697  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1182.77  on 886  degrees of freedom
Residual deviance:  801.59  on 882  degrees of freedom
AIC: 811.59

Number of Fisher Scoring iterations: 5

       GVIF Df GVIF^(1/(2*Df))
Sex    1.09  1            1.04
Age    1.35  1            1.16
Pclass 1.45  2            1.10
```

Figure 1. Summary of GLM

Table of survival:

Passengers (median age)	Probability of survival
First class female	92 %
Second class female	81 %
Third class female	60 %
First class male	41 %
Second class male	23 %
Third class male	9 %

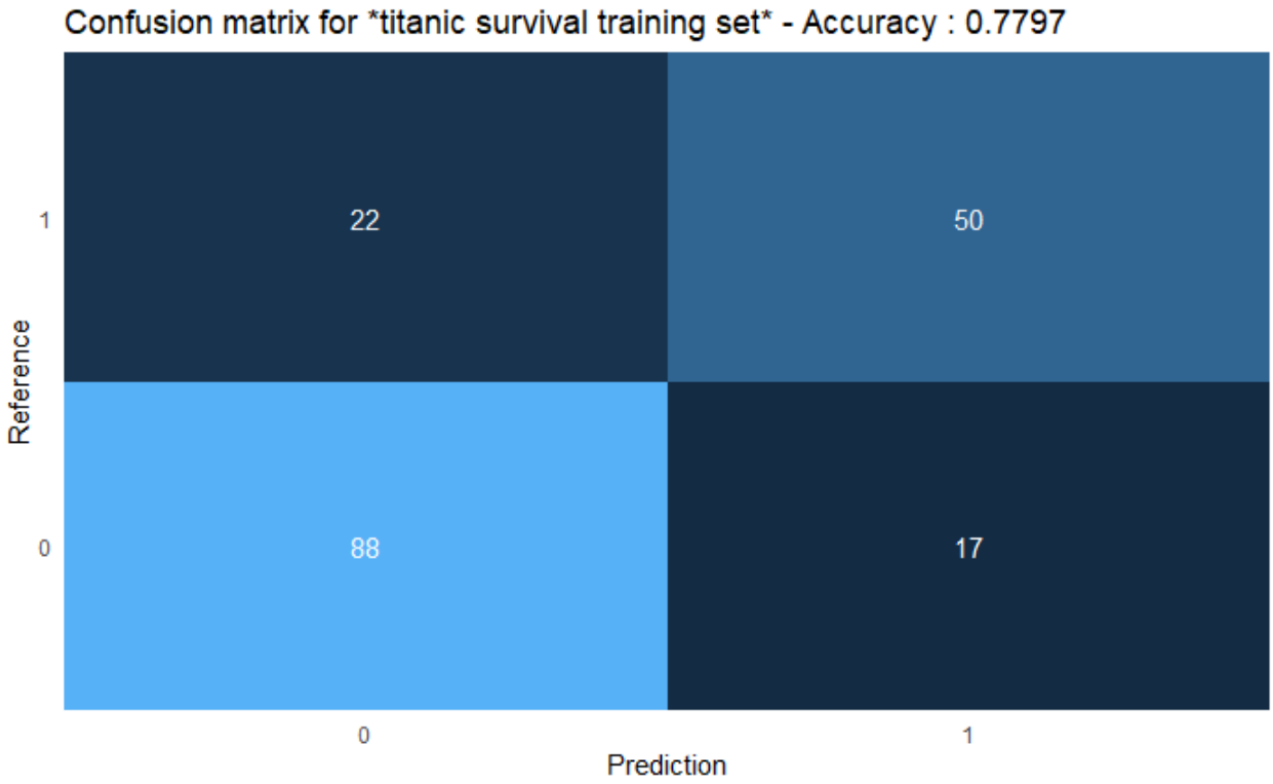


Figure 2. Confusion matrix