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.

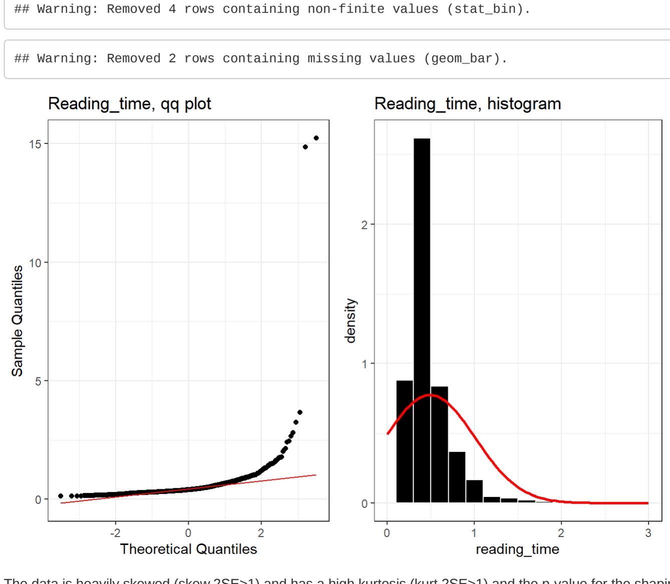
```
round(pastecs::stat.desc(cbind(df$reading_time), basic = FALSE, norm = TRUE), digits = 2)
```

```
##                   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).
```



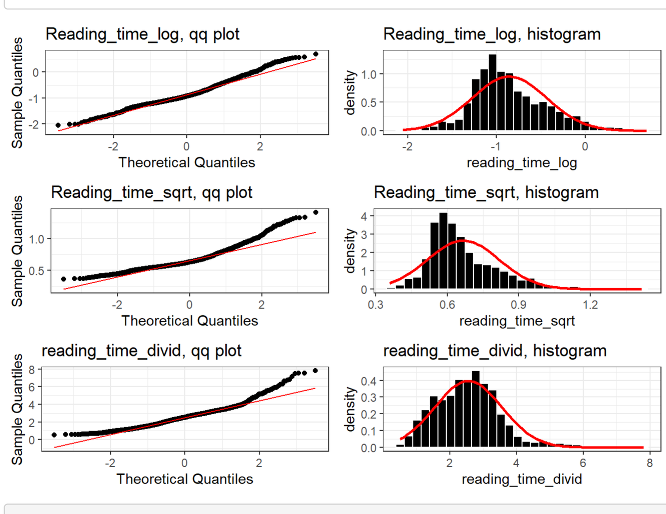
The data is heavily skewed (skew.2SE>1) and has a high kurtosis (kurt.2SE>1) and the p-value for the shapiro wilk test is below the significant level 0.05, meaning that the data is not normally distributed. The qq-plot supports that our data is not normally distributed, since the data points does not 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

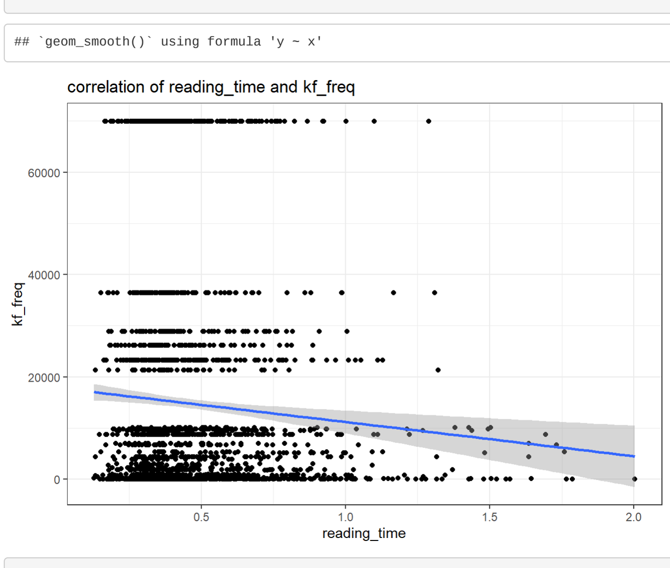
Now we can explore if a relation exists between reading times and length, frequency and ordinality of the words using correlation analysis and scatter plots with linear regression lines.

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.

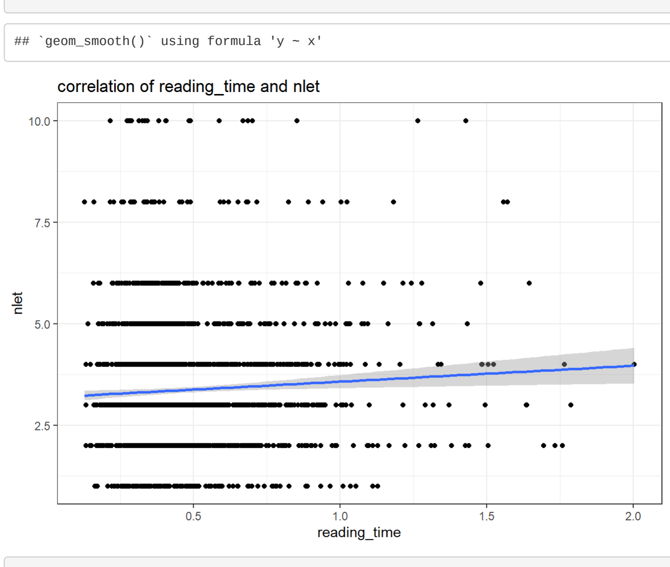
```
cor_kf_freq <- filter_outlier %>%
  ggplot() +
  aes(reading_time, kf_freq) +
  geom_point() +
  geom_smooth(method = "lm")+
  ggtitle("correlation of reading_time and kf_freq") +
    theme_bw()



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'
```



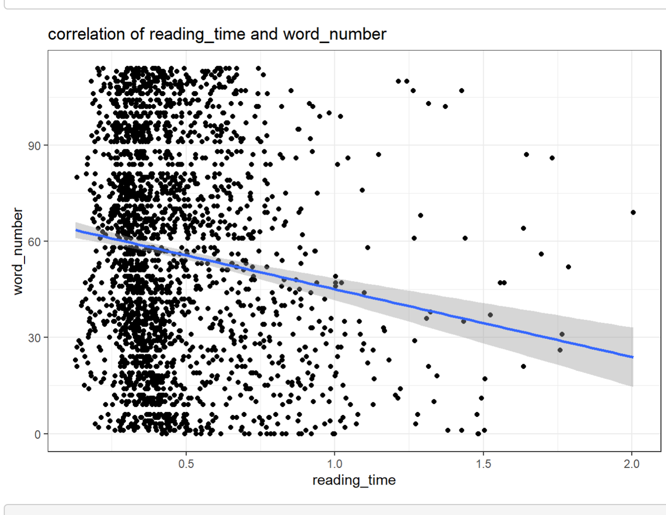

```
cor_nlet
```

```
## `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.