Lab assignment 1: t-tests, linear regression, analyzing lexical decision task

Deadline: 26.2., 6pm

1 General lab information

This assignment consists of two parts. There are group assignments and individual assignments. Group assignments should be shown to your teacher and get graded directly *during the lab meeting*. Answers to individual questions should be submitted to Blackboard via a Blackboard quiz. If the question does not specify what type of question it is, it is a group question (there is only one individual question in this assignment).

There are 10 points in the assignment (1 per question).

More details on the whole procedure:

You work in groups of 3/4 on these exercises, with help from a teacher/teaching assistant. We expect you to work on these exercises in class time so you can work with your group and teacher. It is not acceptable to miss these classes without agreement from your group, or to repeatedly miss classes. Then you will fail the assignment, which leads to failing the course. If your group members miss lab classes without agreement from your group, please inform your teacher.

We suggest all group members doing these exercises on their individual computers simultaneously: this improves (student) learning and also makes it easier to find mistakes. Don't rely on other group members' answers if you don't understand why they are correct: this is meant to be an interactive collaboration with your group, so ask your group members to explain. If your group gets stuck on a question or different group members can't agree on an answer, ask for help from your teacher. Please share your video if bandwidth and circumstances allow. This makes for a more personal conversation.

When your group is happy with your answer, work together to finalize your answer in a document shared with the whole group. Show these answers to your teacher as you work. You can share this document with the teacher too. Your teacher will grade you as you work to monitor your progress and address problems. But we need a record of all your answers, submitted at the end of the assignment (via Blackboard). At the end, you should submit one pdf file, which also includes your R calculations and code. You could either copy-paste your R code into the file, or, better, you could use an engine for dynamic report with R like knitr:[1]. The assignment has to be submitted on Blackboard by *Friday, 6pm, February 26*. Your answer to the individual question has to be submitted by that deadline, as well.

In group questions, it is generally best to start by asking every group member's opinion. Then work on a written answer together. Then explain your answer to your teacher. You can also ask your teacher to read what you wrote, but they will often ask questions. It is likely you will then have to update this answer after talking with your teacher. Please tell your teacher what changes you made next time you talk and show them what you wrote.

Many questions build on previous questions being completed correctly, so you should be confident of your answer before using it in further questions: ask for help if you are unsure. If you get stuck and the teacher can't get help immediately, you can move on to the next topic until your teacher can help.

[1] https://yihui.org/knitr/

2 Introduction

In this assignment, you will work with data from an auditory lexical decision task. In this task, participants listen to the speaker who says a word/pseudoword and have to decide whether what they just heard was a word (by pressing one key) or not (by pressing another key). We then collect their responses. Two measures are of interest: reaction times (how much time it took them to respond) and accuracy (was their response correct or not?). More details about the task, why it is interesting, what it can reveal about the organization of the lexicon in our mind etc. can be found in the paper Tucker et al. (2019), attached to this assignment.

Your task will be to analyze selected data.

3 What will you hand in?

You will hand in a pdf file with the analysis. The pdf file should include the code you used and the code should include all the steps, from loading the csv files up to the analysis required of you in questions. Aside from that, you have to respond to individual questions on Blackboard.

4 What can you use?

You can use R and any packages you find useful (unless some questions explicitly prohibit that). Some packages are even recommended to use - you get a hint to use them in questions. You also can (and should) reuse the code present in these assignment instructions. For an ease of reuse, we put the code separately into an R (knitr) file.

5 Data preparation

We start by loading useful packages (dplyr for data manipulation and ggplot2 for graphics) and loading data as data frames and checking the structure of the data frames.

```
library(dplyr)
library(ggplot2)

itemdata <- read.csv("MALD1_SelectedItemData.csv", sep = "\t")
str(itemdata)

## 'data.frame': 36347 obs. of  23 variables:
##  $ Item         : Factor w/ 36347 levels "a","aabrihz",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ WAV          : Factor w/ 36347 levels "A.wav","aabrihz.wav",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ Pronunciation: Factor w/ 35963 levels " IH","AA BR IH Z",..: 1505 2 3 5 4 7 6 8 9 10 ...
##  $ IsWord       : logi  TRUE FALSE FALSE FALSE FALSE FALSE ...
##  $ StressPattern: int  0 NA NA NA NA NA NA NA NA NA ...
##  $ NumSylls     : int  1 2 3 3 2 3 4 3 3 3 ...
##  $ NumPhones    : int  1 5 7 8 6 7 9 6 7 7 ...
##  $ Duration     : int  359 680 568 800 632 672 753 672 704 672 ...
##  $ OrthUP       : int  2 NA NA NA NA NA NA NA NA NA ...
##  $ PhonND       : int  79 0 0 0 0 0 0 0 0 0 ...
##  $ OrthND       : int  72 NA NA NA NA NA NA NA NA NA ...
##  $ POS          : Factor w/ 8 levels "Adjective","Adverb",..: 3 NA NA NA NA NA NA NA NA NA ...
##  $ AllPOS       : Factor w/ 307 levels "#N/A","Adjective",..: 92 NA NA NA NA NA NA NA NA NA ...
##  $ FreqSUBTLEX  : int  1041179 0 0 0 0 0 0 0 0 0 ...
##  $ FreqCOCA     : int  5822980 0 0 0 0 0 0 0 0 0 ...
##  $ FreqCOCAspok : int  1259642 0 0 0 0 0 0 0 0 0 ...
##  $ FreqGoogle   : num  9.08e+09 0.00 0.00 0.00 0.00 ...
##  $ PhonUP       : int  2 5 3 3 3 4 4 4 3 5 ...
##  $ StressCat    : Factor w/ 6 levels "Final","Initial",..: 5 NA NA NA NA NA NA NA NA NA ...
##  $ Dbet         : Factor w/ 35938 levels "&b^lE","&b^lIS^n",..: 889 3028 3056 3057 3058 3064 3063 30
##  $ PhonLev      : num  5.98 5.96 6.78 7.53 6.4 ...
##  $ NumMorphs    : int  1 NA NA NA NA NA NA NA NA NA ...
##  $ OrthLev      : num  7.04 NA NA NA NA ...
```

```
head(itemdata)
```

```
##           Item              WAV       Pronunciation IsWord StressPattern NumSylls NumPhones
## 1            a            A.wav                 AH0   TRUE             0        1         1
## 2      aabrihz      aabrihz.wav          AA BR IH Z  FALSE            NA        2         5
## 3   aadsaxsaxl   aadsaxsaxl.wav    AA D S AH S AH L  FALSE            NA        3         7
## 4 aadshaxsneyt aadshaxsneyt.wav AA D SH AH S N EY T  FALSE            NA        3         8
## 5     aadsihks     aadsihks.wav       AA D S IH K S  FALSE            NA        2         6
## 6   aagaxrawnt   aagaxrawnt.wav    AA G AH R AW N T  FALSE            NA        3         7
##   Duration OrthUP PhonND OrthND      POS                                              AllPOS
## 1      359      2     79     72 Function Article.Adverb.Letter.To.Noun.Preposition.Adjective
## 2      680     NA      0     NA     <NA>                                                <NA>
## 3      568     NA      0     NA     <NA>                                                <NA>
## 4      800     NA      0     NA     <NA>                                                <NA>
## 5      632     NA      0     NA     <NA>                                                <NA>
## 6      672     NA      0     NA     <NA>                                                <NA>
##   FreqSUBTLEX FreqCOCA FreqCOCAspok FreqGoogle PhonUP StressCat     Dbet  PhonLev NumMorphs
## 1     1041179  5822980      1259642 9081174698      2      None        ^ 5.975207         1
## 2           0        0            0          0      5      <NA>    abrIz 5.958685        NA
## 3           0        0            0          0      3      <NA>  ads^s^l 6.783204        NA
## 4           0        0            0          0      3      <NA> adS^snet 7.526640        NA
## 5           0        0            0          0      3      <NA>   adsIks 6.404667        NA
## 6           0        0            0          0      4      <NA>  ag^r$nt 6.889174        NA
##    OrthLev
## 1 7.039643
## 2       NA
## 3       NA
## 4       NA
## 5       NA
## 6       NA
```

```
responsedata <- read.csv("MALD1_SelectedResponseData.csv", sep = "\t")
str(responsedata)
```

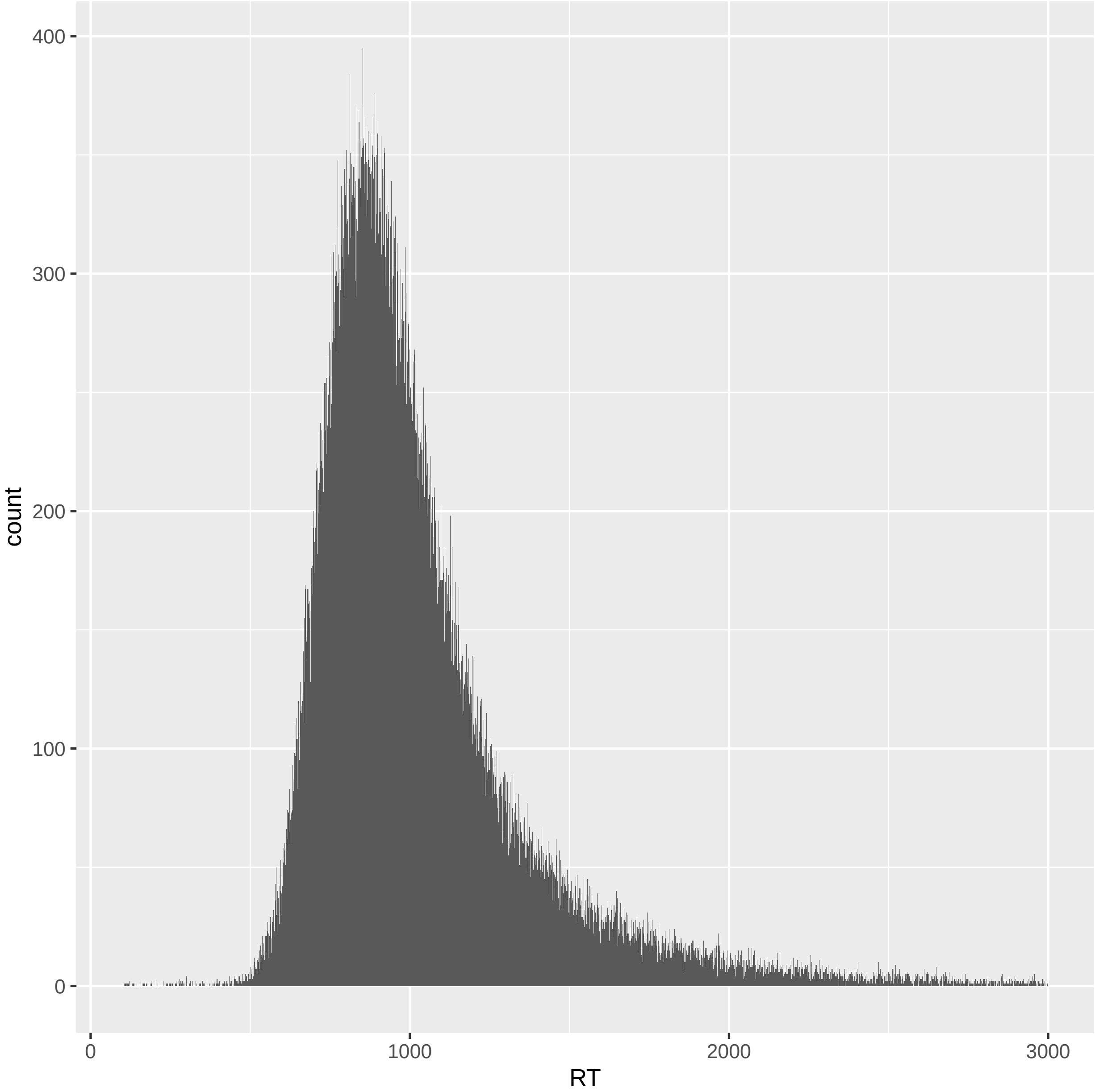
```
## 'data.frame':    172380 obs. of  9 variables:
##  $ Experiment     : Factor w/ 1 level "MALD1_sR": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Subject        : int  15292 15292 15292 15292 15292 15292 15292 15292 15292 15292 ...
##  $ Trial          : int  1 8 9 12 14 15 16 18 20 23 ...
##  $ List           : Factor w/ 91 levels "nonwordsa","nonwordsb",..: 14 14 14 14 14 14 14 14 14 14 ..
##  $ WordRunLength  : int  1 1 2 1 1 2 3 1 1 1 ...
##  $ ExperimentRunID: Factor w/ 224 levels "15292_38","15301_76",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Item           : Factor w/ 36347 levels "a","aabrihz",..: 13372 25022 19986 18371 22446 34498 256
##  $ RT             : int  2604 1517 1175 1133 1138 1076 983 886 1150 1110 ...
##  $ ACC            : logi  TRUE TRUE TRUE TRUE FALSE TRUE ...
```

```
head(responsedata)
```

```
##   Experiment Subject Trial      List WordRunLength ExperimentRunID              Item   RT   ACC
## 1   MALD1_sR   15292     1 nonwordsn            1        15292_38     gaetraxraxshax 2604  TRUE
## 2   MALD1_sR   15292     8 nonwordsn            1        15292_38          praxfehng 1517  TRUE
## 3   MALD1_sR   15292     9 nonwordsn            2        15292_38         maarmsaxld 1175  TRUE
## 4   MALD1_sR   15292    12 nonwordsn            1        15292_38 kaxngkehntaxtehdiy 1133  TRUE
## 5   MALD1_sR   15292    14 nonwordsn            1        15292_38               nuwr 1138 FALSE
## 6   MALD1_sR   15292    15 nonwordsn            2        15292_38            uwpihng 1076  TRUE
```

Afterwards, we visually check one variable (RT=reaction times).

```
g1 <- ggplot(responsedata, aes(RT))
g1 <- g1 + geom_bar()

g1
```



The data frame responsedata includes responses. The following columns are relevant: Item=what word/pseudoword was tested; RT=reaction times to the item; ACC=accuracy (TRUE - correct response, i.e., *yes* for a word, *no* for a pseudoword; FALSE - incorrect response). The data frame itemdata includes information about individual items, including frequencies based on several different corpora, stress patterns of words, whether the element is a word or not, average distance from words based on phonological Levensteihn distance etc. For detailed descriptions of what columns in these files mean, check Tucker et al. (2019).

Q1: Merge data and report basic information

Merge the two data frames (responsedata and itemdata) into one data frame. Afterwards, report descriptive summaries for reaction times (RT) and for frequencies (use FreqCOCAspok, which is a spoken corpus of American English). Finally, report descriptive summaries of RT for words and pseudowords (i.e., depending on whether IsWord is TRUE or FALSE) for the whole dataset and also for Subjects 15351, 16854 and 170373. For descriptive summaries it is enough you report means and spread of data - e.g., variances or standard deviations, or just a range of data. You can also provide histograms (for example, by replicating the code we have above; for RTs this is already done).

```
# Hint: try to use dplyr and a family of join functions, and group_by and summarise from the same
# package.  Of course, other functions might be useful and needed.

# Check help of these functions and check dplyr for details.
```

6 Cohen's d

If everything was done correctly, you should have found out that the distribution of RTs differs depending on whether people responded to a word or to a pseudoword. This might make sense to you - it seems that responding to a pseudoword takes more time. We will now be investigating this effect further. Roughly, we want to address the following question: can we conclude with reasonable confidence that in population responses to words are faster than responses to pseudowords? We will further qualify and specify this question as we proceed.

We will start by investigating Cohen's d. This is a standardized measure of effect size: it measures the strength of difference between two means. The formula is calculated as follows:

$$d = \frac{\bar{x_1} - \bar{x_2}}{s} \tag{1}$$

where $\bar{x_1}$ is the mean of data x_1 (i.e., RTs for words) and $\bar{x_2}$ is the mean of data for x_2 (RTs for non-words) and s is a pooled standard deviation, which has been calculated as shown below in Cohen's original work (note: there are various ways of calculate s, we will use this one). Assume that there are n_1 observations for x_1 and n_2 observations for x_2, i.e., $|x_1| = n_1$ and $|x_2| = n_2$. $var(x)$ is the variance of x (you can get it in **R** by using the function **var**). Then:

$$s = \sqrt{\frac{(n_1 - 1) \cdot var(x_1) + (n_2 - 1) \cdot var(x_2)}{n_1 + n_2 - 2}} \tag{2}$$

This formula can be simplified if the length of x_1 is equal to the length of x_2, i.e., $n_1 = n_2$, which is the case for all our computations below. I leave it to you to make the algebraic simplifications. You can make this simplification and use it - throughout this assignment, it will be the case that $n_1 = n_2$.

Q2: Implement cohen's d

Implement cohen's d as a function in R. That is, you have to fill in the body of the function (what is put in as ...) that you have here below. As said above, it is enough to implement the simplified version (one in which the length of x_1 and x_2 is the same). *Do not use any extra packages that already have cohen.d!*[2]

```r
cohend <- function(x1, x2) {
    ...
}
```

After the implementation, test your function and report collected Cohen's d on four cases discussed below. Along that, report whether the effect size is small, medium or large ($|d| < 0.5$ is small, $|d| < 0.8$ is medium, above that is large).

1. RTs for words and pseudowords for Subject numbered 15351.
2. RTs for words and pseudowords for Subject numbered 16854.
3. RTs for words and pseudowords for Subject numbered 170373.
4. RTs for all words and pseudowords.
5. RTs for the two vectors provided below as word_15292 and pseudoword_15292 (these are a few selected responses to words and pseudowords from subject 15292).

```r
word_15292 <- c(2206, 1583, 1154, 1010, 865, 931, 1129, 683, 820, 1132, 1049, 1211, 1261, 957, 1058,
    790, 851, 1908, 1504, 1400, 924)

pseudoword_15292 <- c(677, 949, 889, 881, 917, 769, 772, 922, 1944, 881, 976, 1087, 1252, 914, 1277,
    825, 1295, 1336, 788, 885, 932)
```

7 Going beyond Cohen's d

Now, Cohen's d might look a sensible way to measure differences between means. For example, it captures the fact that 15351 has a larger d number (more effect) than 16854, corresponding to the fact that the means between words and pseudowords in the former case are more apart than in the latter case. Similarly, 170373 has a larger d than 16854 corresponding to the fact that the former has a much smaller spread of the data (as seen in its standard deviation) than the latter.

However, there's something that should make you feel uneasy about using d as a sensible proxy to answer the question of whether words and pseudowords affect RTs. Namely, we do not take into account how many data we collected. And we want to learn something about populations, not just our samples. Clearly, if we collected more data from our population, that should weigh more than collecting fewer data. But this is not the case here. For that, notice that the last data set (pseudoword/word_15292) with only 21 data points per group has almost the same d as the full complete dataset. We turn to t-values to address this issue.

8 t-values

t-values, well known and familiar to you by now, are just like Cohen's d. Unlike Cohen's d they do not tell us about differences of sample means, but differences in population means.

First, t is calculated as (caveat: this calculation works for our simple case we consider here; for other cases, e.g., with paired observations or with one sample, the calculation is different - see also the lectures for this week):

$$t = \frac{\bar{x_1} - \bar{x_2}}{SE} \tag{3}$$

[2] However, you can use such packages (for example, EFFSIZE) to double-check that your function works correctly). In doing so, be careful - some implementations might slightly differ wrt how they calculate s, so you might not get exactly identical numbers.

You can see that we standardize the distance between two means by SE, the standard error. SE is calculated as shown below, where n_1 is the number of observations in group1 (i.e., the number of observations in x_1) and n_2 in group2. You can simplify this assuming that both groups are of equal size because we will work with equal size groups.

$$SE = \sqrt{\frac{var(x_1)}{n_1} + \frac{var(x_2)}{n_2}} \tag{4}$$

Another way to understand SE is to derive it from s in Cohen's d as shown below, where s is calculated the same as in Cohen's d and we divide by the square root of sizes of x_1 and x_2. So in other words we take Cohen's d and adjust by the sizes of the samples. Again, this can be simplified if $n_1 = n_2$, as is the case in all cases below.

$$SE = s * \sqrt{1/n_1 + 1/n_2} \tag{5}$$

Q3: Implement t

Implement the t-calculation as a function. That is, fill in the body of this function:

```
tcalculation <- function(x1, x2) {

    ...

}
```

Once it is done, calculate t for:

1. RTs for words and pseudowords for Subject numbered 15351.
2. RTs for all words and pseudowords.
3. RTs for the two vectors provided below as word_15292 and pseudoword_15292 (these are a few selected responses to words and pseudowords from subject 15292).

Report t-values and say briefly why RTs for all words and pseudowords (the second question) have the highest t-value compared to pseudoword/word_15292 or even the responses of Subject numbered 15351. A brief description of the crucial intuition suffices. *For this answer, do not use the function t.test in R or any other pre-defined function that calculates t values. However, you can check that your own implementation is correct by comparing your function to t.test.*

9 Using the t-distribution to report p-values

In the Null Hypothesis Significance Testing (NHST), we study how likely it is that our results or a more extreme version of our results would have been observed under the null hypothesis. We will now assume that the null hypothesis is that the mean of the population from which x_1 is sampled does not differ from the mean of the population from which x_2 is sampled. What would then be the p value?

Here is where using t-values comes in useful: t values are accompanied by a probability distribution, so-called t-distribution. t-distribution with a parameter degrees of freedom (df) n expresses the probability that we would get such and such t-value if the data we were observing were normally distributed, the population mean was 0 and we sampled $n + 1$ data.

Let us convince ourselves that the point in the last paragraph is correct. We will do so by running a small simulation.

First, we create a function that will sample 20 random data points from a normal distribution with mean 0 and standard deviation 10 (the size of the standard deviation is not important for this simulation). This random sampling is done using the function **rnorm** (see Baayen for discussion and help function). Then, we calculate the t-value for that sample, using the definitions above but assuming just a single sample (that is, we do not compare two samples but do a one-sample t-test; see also the lecture for this week; we measure the difference of the mean of the current sample from 0, which is just the same as having the mean of the current sample in the numerator).

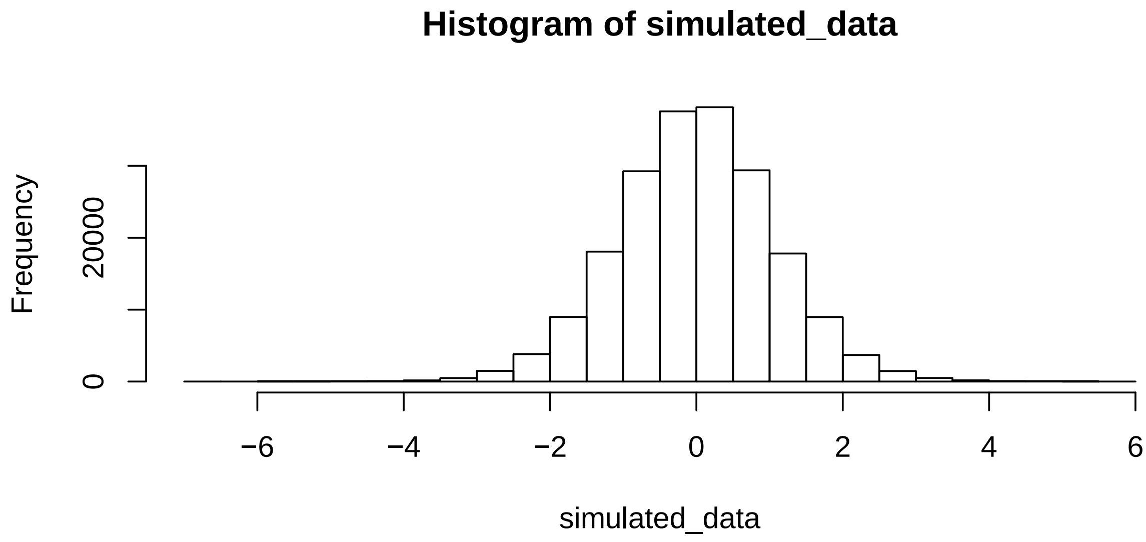
```
generate.t <- function() {

    mysample <- rnorm(20, mean = 0, sd = 10)
    tvalue <- mean(mysample)/sqrt((var(mysample)/20))
    tvalue

}
```

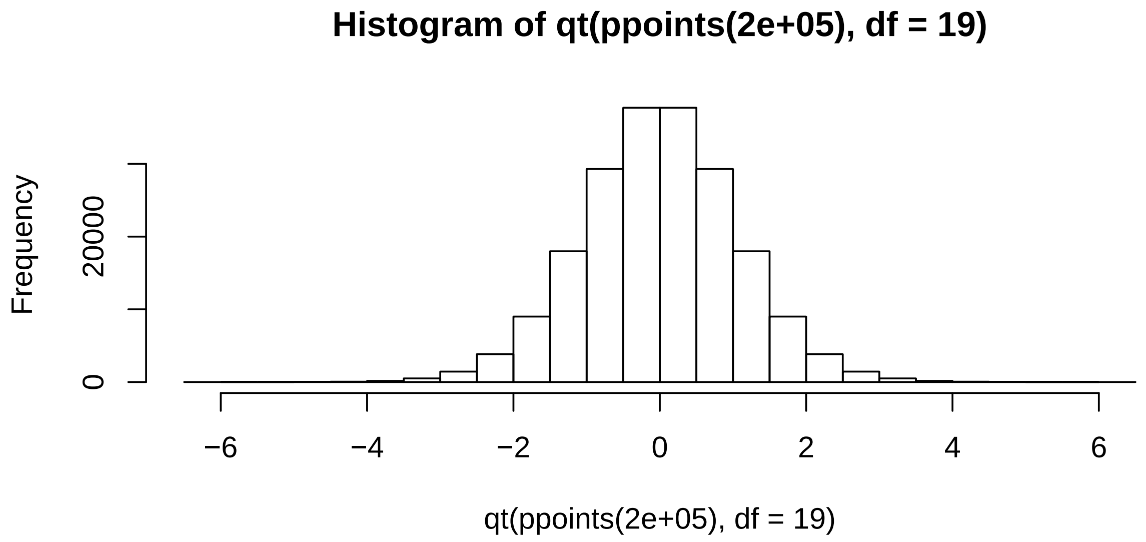
Now, we want to check that the t-values we are about to sample correspond to the probability distribution t. We do that as follows:

1. We run simulation for many times, say 200,000 times. In each simulation, we sample from the same normal distribution and store the t-value.
2. Then, we compare the collected t-values to the theoretical probability distribution t with degrees of freedom (df) = 19. Ideally, we should see a match. We first check a qqplot (see Baayen for discussion) - we expect the simulated data and the theoretical distribution to fall on a line. This is roughly the case even though extreme values at both ends might slightly fall out (due to sampling).[3] We also expect that in a histogram, we should see the same values in the simulated data and in the t-distribution.

```
simulated_data <- rep(NA, 2e+05)

for (i in 1:length(simulated_data)) {

    simulated_data[i] <- generate.t()

}

# Q-Q plot qqplot(qt(ppoints(200000), df=19), simulated_data) qqline(simulated_data, distribution=
# function(p) qt(p, df=19)) Not used - see the png plot in this assignment


# Histograms comparing t-values from simulated data and predicted based on the t-probability
# distribution
par(mfrow = c(2, 1))
hist(simulated_data)
hist(qt(ppoints(2e+05), df = 19))
```

[3] This plot is not shown here because some pdf readers have problems to render all 200,000 points. You can find it in the assignment as a separate png file.





Indeed, we see a very good match between simulated data from which t is collected and what we would expect to get just by looking at the theoretical t-distribution.

Q4: transforming data and collecting p-values

Based on what we said so far, you should be able to tie t-values that you provided in Q3 to p-values under the null hypothesis that population means between RTs of words and RTs for pseudowords do not differ, i.e., mean(wordRT)=mean(pseudowordRT). Use the t-value from Q3 for the data set word_15292 and pseudoword_15292 and use the function **pt** (with degrees of freedom = 40) to provide the answer. You can also check Baayen, section 4.1, for some relevant code on how to use **pt**.

When you are done, come back to one of the assumptions of t-probability distributions: t-values are collected from samples of *independent and normally distributed data*. We focus on the latter condition. Check if RTs in words and pseudowords are normally distributed. If not, try a transformation to get closer to normal distribution. Among transformations, it is common to consider squaring, cubing, taking an inverse, taking square root, or log-transforming data. It is fine if you find only a roughly normal distribution (no testing needed, just checking by observing a histogram is sufficient for this exercise). Once you find the best case of transformation, report t values and p values for this transformed distribution. You can decide whether you want to use one-tailed or two-tailed tests but whatever you decide, report that.

Q5: aggregating data

Even if we get normal distribution of underlying data, we still did not address the issue of independence. Are all RTs in our data set independent? Clearly not. Participants tend to differ in reaction times and so there will be dependence in reaction times of a participant. The way to avoid it is to not work with raw data but aggregations. Commonly when running a t-test on experimental data, we aggregate the dependent variable, e.g., RTs for words, per participant (that is, we get just one measure per participant, its mean RT over words). We do the same for pseudowords. Then, we calculate the t-value over these aggregated measures and then we calculate p-values. Do that for the dataset and report the results. Be careful in thinking about the type of t-test. Is this paired or unpaired?

Q6: reading about an experiment (individual question)

This question has to be answered individually. Each student submits his/her answer on Blackboard.

Imagine you read about an auditory lexical decision task experiment. The experiment says that there were 20 words and 20 pseudowords tested. 30 participants took part in the experiment. Each participant saw all the words and pseudowords. Now, the paper says: "We found a significant effect of word/pseudoword manipulation ($t = 3.594, df = 1198, p = 0.00034$)." Looking at this reported results, how did the experimentalist carry out his/her analysis (on subject-aggregated data, on non-aggregated data), and would you say that this was justified?

Q7: a pitfall for p-values

Right above, we calculated p values based on the assumption that there are 40 degrees of freedom, corresponding to the collection of 42 data points (21 for word_15292 and 21 for pseudoword_15292; for each group the degrees of freedom are 21-1, which makes 40 degrees of freedom in total). In this way, we are behaving as if the number of data points was fixed and it was only open what the values of the data points was.

However, it is quite common that experimentalists do not know in advance how many participants they want to collect. Imagine the following situation: we decided we would be collecting data for the whole day and then we will stop and check the results. It happens so that on that day, there was a 50% probability that we would collect 12 responses (6 for words, 6 for pseudowords) and a 50% probability that we would collect 42 data points (21 for words and 21 for pseudowords). In our actual sample, we happened to collect the latter amount (i.e., 42 data points) and we got results as shown in word_15292 and pseudoword_15292. What would then be the p value?

This is a slightly mind-boggling question so do not worry if you cannot answer it. If you cannot calculate the value, try to at least reason about this: do you think that the p-value will be smaller than in Q4 or bigger? In any case, note one very unintuitive aspect of p-values: they are dependent on experimenters' intentions and hypothetical situations (which might often not be explicitly stated, and might not even be implicitly considered!).

10 Linear models

So far, we worked all the time with RTs split by only one condition: IsWord. In fact, we can study more than one condition at the time. For that, we have to turn to linear regression models. First, consider the following simple model, which only looks at the regression line based on IsWord.

```
m1 <- lm(RT ~ IsWord, ...)  #put in your data here

## Error in eval(expr, envir, enclos):  ’...’  used in an incorrect context

print(summary(m1))

## Error in summary(m1):  object ’m1’ not found
```

But we can add more parameters and study how they affect the regression line that predicts RTs. For example, the following model would consider the effect of IsWord, Accuracy and their interaction. The * in the notation calculates the main effect of both factors and the interaction of the two factors. For more details about linear regression models and the notation, check Chapter 6 of Baayen.

```
m2 <- lm(RT ~ IsWord * ACC, ...)  #put in your data here

## Error in eval(expr, envir, enclos):  ’...’  used in an incorrect context

print(summary(m2))

## Error in summary(m2):  object ’m2’ not found
```

Q8: Graphically representing your results

Provide a graphical summary in which we can clearly see that IsWord affects RTs, ACC affects RTs, and the interaction of the two factors affects RTs.

Q9: What frequency is the best predictor?

There are various sources of frequency in the corpus: FreCOCA, FreqGoogle, FreqSUBTLEX and FreqCOCAspok. Find out which of these provides the best fit of the model to the dependent variable log-transformed reaction times. You can do so by comparing models in which different frequency sources are added, or by comparing how big a proportion of the variance is explained by the model (see Baayen, chapter 6, if you do not know how). After you find the answer, use the same frequency to address the following observation: it has been claimed that log-frequency of a word is a good predictor, better than a plain frequency, for log-reaction times. Is this correct? Plot the relation between the dependent and the non-transformed independent variable to see whether any clear relation can be observed and whether the relation looks linear. Then, transform the frequency to log and plot again. Then, check the resulting model.

Finally, check whether the log-transformation of all frequency data sets changes your previous answer. Which corpus of frequency is now the best predictor when we consider its log-transformation?

Q10: Exploring linear models

Explore models with at least three predictors. All the predictors you use should be significantly different from 0 and only one of them should be a frequency (p values of each those should be smaller than .05 in the summary of the model). Explore more than one parameter and report the best fitting model that you found. Discuss briefly what variables you used in your model. Did it make an intuitive sense that such variables affect reaction times? Make sure to transform reaction times: you can use log-transformation. Furthermore, you might also see that some predictors are missing or their values are non-sensical in case of some observations. In that case, remove those cases from your consideration and the final model. You don't need to find the best possible model but you should be able to get a model whose adjusted R^2 is greater than 0.1 (i.e., a case in which at least 10% of the variance in the data has been explained by the model).

References

Tucker, Benjamin V, Daniel Brenner, D Kyle Danielson, Matthew C Kelley, Filip Nenadić, and Michelle Sims. 2019. The massive auditory lexical decision (mald) database. *Behavior research methods* 51:1187–1204.