Final A3P1

Klara, Pernille, Søren, Clement & Julia

10/7/2020

Assignment 3 - Part 1 - Assessing voice in schizophrenia

Individuals with schizophrenia (SCZ) tend to present voice atypicalities. Their tone is described as "inappropriate" voice, sometimes monotone, sometimes croaky. This is important for two reasons. First, voice could constitute a direct window into cognitive, emotional and social components of the disorder, thus providing a cheap and relatively non-invasive way to support the diagnostic and assessment process (via automated analyses). Second, voice atypicalities play an important role in the social impairment experienced by individuals with SCZ, and are thought to generate negative social judgments (of unengaged, slow, unpleasant interlocutors), which can cascade in more negative and less frequent social interactions.

Several studies show *significant* differences in acoustic features by diagnosis (see meta-analysis in the readings), but we want more. We want to know whether we can diagnose a participant only from knowing the features of their voice.

The corpus you are asked to analyse is a relatively large set of voice recordings from people with schizophrenia (just after first diagnosis) and matched controls (on gender, age, education). Each participant watched several videos of triangles moving across the screen and had to describe them (so you have several recordings per person). We have already extracted the pitch once every 10 milliseconds as well as several duration related features (e.g. number of pauses, etc).

N.B. For the fun of it, I threw in data from 3 different languages: 1) Danish (study 1-4); 2) Mandarin Chinese (Study 5-6); 3) Japanese (study 7). Feel free to only use the Danish data, if you think that Mandarin and Japanese add too much complexity to your analysis.

In this assignment (A3), you will have to discuss a few important questions (given the data you have). More details below.

Part 1 - Can we find a difference in acoustic features in schizophrenia? 1) Describe your sample number of studies, number of participants, age, gender, clinical and cognitive features of the two groups. Furthemore, critically assess whether the groups (schizophrenia and controls) are balanced. N.B. you need to take studies into account.

2) Describe the acoustic profile of a schizophrenic voice: which features are different? E.g. People with schizophrenia tend to have high-pitched voice, and present bigger swings in their prosody than controls. N.B. look also at effect sizes. How do these findings relate to the meta-analytic findings?

3) Discuss the analysis necessary to replicate the meta-analytic findings Look at the results reported in the paper (see meta-analysis in the readings) and see whether they are similar to those you get. 3.1) Check whether significance and direction of the effects are similar 3.2) Standardize your outcome, run the model and check whether the beta's is roughly matched (matched with hedge's g) which fixed and random effects should be included, given your dataset? E.g. what about language and study, age and gender? Discuss also how studies and languages should play a role in your analyses. E.g. should you analyze each study individually? Or each language individually? Or all together? Each of these choices makes some assumptions about how similar you expect the studies/languages to be. *Note* that there is no formal definition of replication (in statistical terms).

Your report should look like a methods paragraph followed by a result paragraph in a typical article (think the Communication and Cognition paper)

Part 2 - Can we diagnose schizophrenia from voice only? 1) Discuss whether you should you run the analysis on all studies and both languages at the same time You might want to support your results either by your own findings or by that of others 2) Choose your best acoustic feature from part 1. How well can you diagnose schizophrenia just using it? 3) Identify the best combination of acoustic features to diagnose schizophrenia using logistic regression. 4) Discuss the "classification" process: which methods are you using? Which confounds should you be aware of? What are the strength and limitation of the analysis?

Bonus question: Logistic regression is only one of many classification algorithms. Try using others and compare performance. Some examples: Discriminant Function, Random Forest, Support Vector Machine, Penalized regression, etc. The packages caret and glmnet provide them. Tidymodels is a set of tidyverse style packages, which take some time to learn, but provides a great workflow for machine learning.

Learning objectives

- Critically design, fit and report multilevel regression models in complex settings
- Critically appraise issues of replication

Overview of part 1

In the course of this part 1 of Assignment 3 you have to: - combine the different information from multiple files into one meaningful dataset you can use for your analysis. This involves: extracting descriptors of acoustic features from each pitch file (e.g. mean/median, standard deviation / interquartile range), and combine them with duration and demographic/clinical files - describe and discuss your sample - analyze the meaningful dataset to assess whether there are indeed differences in the schizophrenic voice and compare that to the meta-analysis

There are three pieces of data:

1- Demographic data (https://www.dropbox.com/s/e2jy5fyac18zld7/DemographicData.csv?dl=0). It contains

- Study: a study identifier (the recordings were collected during 6 different studies with 6 different clinical practitioners in 2 different languages)
- Language: Danish, Chinese and Japanese
- Participant: a subject ID
- Diagnosis: whether the participant has schizophrenia or is a control
- Gender
- Education
- Age
- SANS: total score of negative symptoms (including lack of motivation, affect, etc). Ref: Andreasen, N. C. (1989). The Scale for the Assessment of Negative Symptoms (SANS): conceptual and theoretical foundations. The British Journal of Psychiatry, 155(S7), 49-52.
- SAPS: total score of positive symptoms (including psychoses, such as delusions and hallucinations): http://www.bli.uzh.ch/BLI/PDF/saps.pdf
- VerbalIQ: https://en.wikipedia.org/wiki/Wechsler_Adult_Intelligence_Scale
- NonVerbalIQ: https://en.wikipedia.org/wiki/Wechsler_Adult_Intelligence_Scale
- TotalIQ: https://en.wikipedia.org/wiki/Wechsler_Adult_Intelligence_Scale

2. Articulation.txt (https://www.dropbox.com/s/vuyol7b575xdkjm/Articulation.txt?dl=0). It contains, per each file, measures of duration:

- soundname: the name of the recording file
- nsyll: number of syllables automatically inferred from the audio
- npause: number of pauses automatically inferred from the audio (absence of human voice longer than 200 milliseconds)
- dur (s): duration of the full recording
- phonationtime (s): duration of the recording where speech is present
- speechrate (nsyll/dur): average number of syllables per second
- articulation rate (nsyll / phonationtime): average number of syllables per spoken second
- ASD (speakingtime/nsyll): average syllable duration

3. One file per recording with the fundamental frequency of speech extracted every 10 milliseconds (excluding pauses): https://www.dropbox.com/sh/bfnzaf8xgxrv37u/AAD2k6SX4rJBHo7zzRML7cS9a?dl=0

- time: the time at which fundamental frequency was sampled
- f0: a measure of fundamental frequency, in Herz

NB. the filenames indicate: - Study: the study, 1-6 (1-4 in Danish, 5-6 in Mandarin Chinese) - D: the diagnosis, 0 is control, 1 is schizophrenia - S: the subject ID (NB. some controls and schizophrenia are matched, so there is a 101 schizophrenic and a 101 control). Also note that study 5-6 have weird numbers and no matched participants, so feel free to add e.g. 1000 to the participant ID in those studies. - T: the trial, that is, the recording ID for that participant, 1-10 (note that study 5-6 have more)

Getting to the pitch data

You have oh so many pitch files. What you want is a neater dataset, with one row per recording, including a bunch of meaningful descriptors of pitch. For instance, we should include “standard” descriptors: mean, standard deviation, range. Additionally, we should also include less standard, but more robust ones: e.g. median, iqr, mean absoluted deviation, coefficient of variation. The latter ones are more robust to outliers and non-normal distributions.

Tip: Load one file (as a sample) and: - write code to extract the descriptors - write code to extract the relevant information from the file names (Participant, Diagnosis, Trial, Study) Only then (when everything works) turn the code into a function and use map_df() to apply it to all the files. See placeholder code here for help.

```r
library(pacman)

p_load(purrr, tidyverse, reshape2, lme4, effsize)
```

```r
# Defining functions

read_pitch <- function(filename) {
    # load data
    df <- read.delim(filename, sep = '\t', header = T)

    # Parse file name to extract study, diagnosis, subject and trial
    # Finding the pattern 'Study', selecting the pattern and the next character
    Study <- str_extract(filename, 'Study.') %>%
        str_remove('Study') # Removing 'Study'

    # Finding the pattern 'D', selecting the pattern and the next character
    Diagnosis <- str_extract(filename, 'D.') %>%
        str_remove('D') # Removing 'D'
    # Finding the pattern 'S with 2 or 3 digits after it' and selecting the pattern
    Subject <-
        str_extract(filename, 'S(\\d\\d\\d|\\d\\d|\\d)') %>%
        str_remove('S') # Removing 'S'

    # Finding the position of the first T in the file name
    pos <- str_locate(filename, 'T')[1]

    # Finding the number of characters in the file name (to index the last character in the next step)
    len_split <- length(strsplit(filename, '')[[1]])

    # Saving everything in the file name from the first T to the last character
    Trial <- substr(filename, pos, len_split)

    # Removing the part after trial
    Trial <- ifelse(str_detect(Trial, '_') == T,
                    str_remove(Trial, '_f0.txt'),
                    str_remove(Trial, '.txt'))

    soundname <- ifelse(str_detect(filename, '_') == T,
                    str_remove(filename, '_f0.txt'),
                    str_remove(filename, '.txt')) %>%
        str_remove('Pitch/')

    # extract pitch descriptors (mean, sd, iqr, etc)
    mean <- mean(df$f0)
    sd <- sd(df$f0)
    iqr <- IQR(df$f0)
    median <- median(df$f0)

    # combine all this data in one dataset
    data <- c(soundname, filename, Study, Diagnosis, Subject, Trial, mean, sd, iqr, median)

    return(data)
}

# Function for adding 'Pitch/' to the file names
pastyp <- function(filename){
    filename <- paste('Pitch/', filename, sep = '')
    return(filename)
}
```

```
# Loading the mf data YAYA
pitch_data = list.files(path = "Pitch/", pattern = ".txt") %>%  # Getting a list of file names
    lapply(pastyp) %>% # Adding 'Pitch/' to the file names
    data.frame() %>% # Making the list into a data frame... I works this way - don't ask why..
    purrr::map_df(read_pitch) %>% # Mapping the read_pitch function on the file names
    t() %>% # Dunno, it has the wrong direction
    data.frame() # Making it into a data frame

# Fixing the column names
colnames(pitch_data) <- c('soundname',
                          'Filename',
                          'Study',
                          'Diagnosis',
                          'Participant',
                          'Trial',
                          'Mean',
                          'SD',
                          'IQR',
                          'Median')

# Dunno why the row names are weird, but now they are fixed
rownames(pitch_data) <- 1:nrow(pitch_data)

# Fixing classes
pitch_data$Mean <- as.numeric(pitch_data$Mean)
pitch_data$IQR <- as.numeric(pitch_data$IQR)
pitch_data$Median <- as.numeric(pitch_data$Median)
pitch_data$SD <- as.numeric(pitch_data$SD)
pitch_data$Participant <- as.numeric(pitch_data$Participant)
pitch_data$Study <- as.numeric(pitch_data$Study)

pitch_data$Diagnosis <- as.factor(pitch_data$Diagnosis)

# Loading more data, selecting only the Danish study
Dem_data <- read.csv("DemographicData.csv", sep = ';', header = T) %>%
    filter(Language == "Danish")

# Fixing classes
Dem_data$Study <- as.numeric(Dem_data$Study)

Dem_data$Gender <- as.factor(Dem_data$Gender)
Dem_data$Diagnosis <- as.factor(Dem_data$Diagnosis)

# Loading more data
Arti_data <- read.delim('Articulation.txt', sep = ',', header = T)
```

Now you need to merge demographic/clinical, duration and pitch data

```
# Participant ID had duplicates - this is solved by adding 1000 to the schizophrenic participant
Dem_data$Participant <- ifelse(Dem_data$Diagnosis == "Control",
                               as.numeric(Dem_data$Participant),
                               as.numeric(Dem_data$Participant)+1000)
pitch_data$Participant <- ifelse(pitch_data$Diagnosis == "0",
                                 as.numeric(pitch_data$Participant),
                                 as.numeric(pitch_data$Participant)+1000)

# Merging the demographic and pitch data
pitch_arti <- merge(Arti_data, pitch_data, by = 'soundname')

# Making the participants as a factor
pitch_arti$Participant <- as.factor(pitch_arti$Participant)
Dem_data$Participant <- as.factor(Dem_data$Participant)
```

```r
# Study is different for the pitch/articulation data and the demographic data
# We merge the demographic data with the pitch and articulation data and filter out the studies that ar
df <- merge(pitch_arti, Dem_data, by = "Participant") %>%
    filter(Study.x < 5 & Study.y < 5)
```

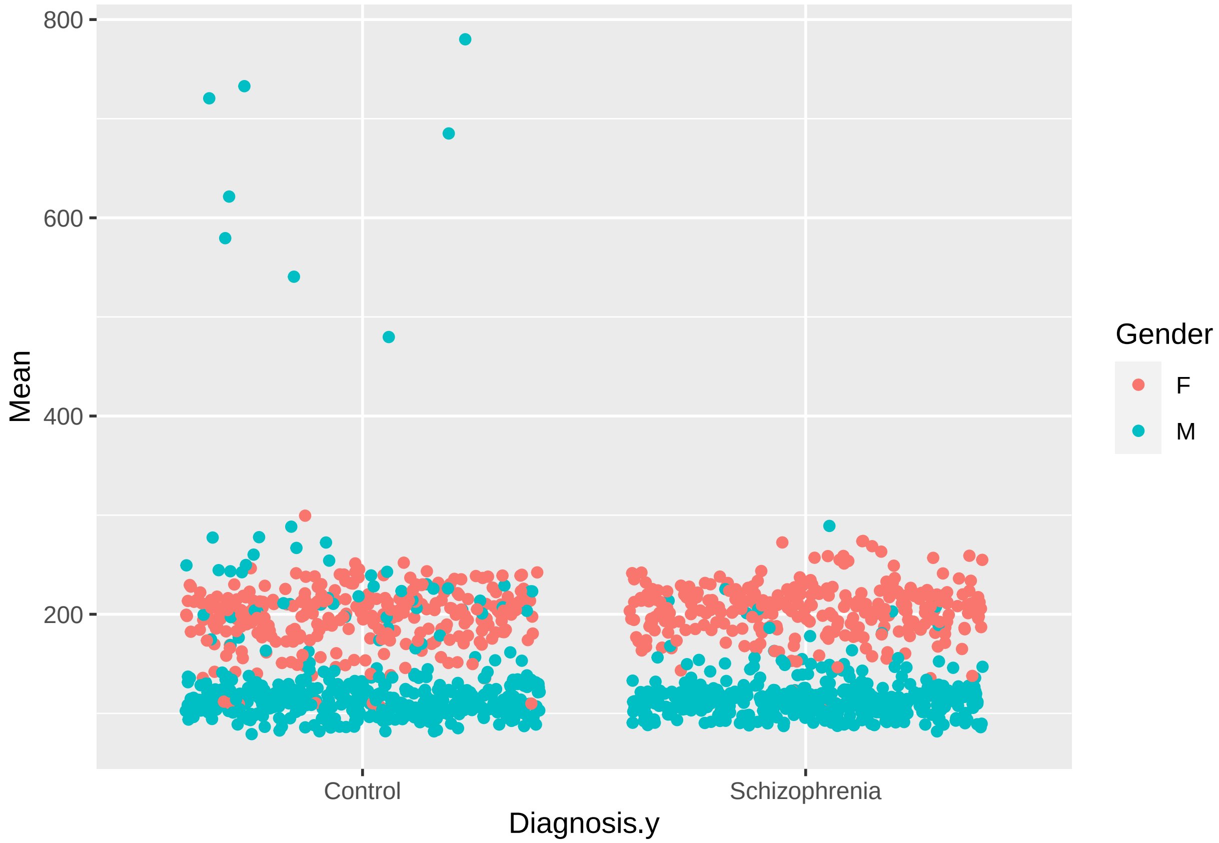
Cleaning the merged data

```r
# There are a lot of NA's in the data set
# First we select only the variables we are going to use in the description and analysis of the data an
df <- df %>% select(
    c(
        Diagnosis.x,
        Diagnosis.y,
        Participant,
        SANS,
        SAPS,
        Gender,
        Study.y,
        Trial,
        npause,
        dur..s.,
        phonationtime..s.,
        IQR,
        speechrate..nsyll.dur.,
        Age,
        Mean
    )
) %>%
    na.omit()

# For some of the participants the duration of the trial and the phonation time was not the same, but t

# Logic map of where duration and phonation time and duration is the same (where the number of pauses s
map1 <- df$dur..s. == df$phonationtime..s.

# Logic map of where the number of pauses is 0
map2 <- df$npause == 0

# changing the number of pauses to NA where the two logic maps don't match
df$npause <- ifelse(map1 == map2, df$npause, NA)

# Omitting the NA's created above
df <- df %>% na.omit()

# Creating a variable for the pause duration
df <- df %>%
    mutate(pause_dur = (dur..s. - phonationtime..s.)/npause)

# Setting pause duration to 0 where the number of pauses is 0
df$pause_dur <- ifelse(df$npause == 0, 0, df$pause_dur) %>% as.numeric()

df <- df %>%
    mutate(pause_dur_scaled = scale(pause_dur)) %>% # Scaling the pause duration
    mutate(IQR_scaled = scale(IQR)) %>% # Scaling the IQR
    mutate(SpeechRate_scaled = scale(speechrate..nsyll.dur.)) %>% # Scaling the speech rate variable
    mutate(ProportionSpokenTime = phonationtime..s./dur..s.) %>% # Creating a variable for proportion o
    mutate(ProportionSpokenTime_scaled = scale(phonationtime..s./dur..s.)) %>% # Scaling the proportion
    mutate(SAPS_scaled = scale(SAPS)) %>% # Scaling the SAPS
    mutate(Mean_scaled = scale(Mean)) # Scaling the SANS

# Making diagnosis a factor
df$Diagnosis.x <- as.factor(df$Diagnosis.x)

#write.csv(df, 'npause_fixed.csv')
```

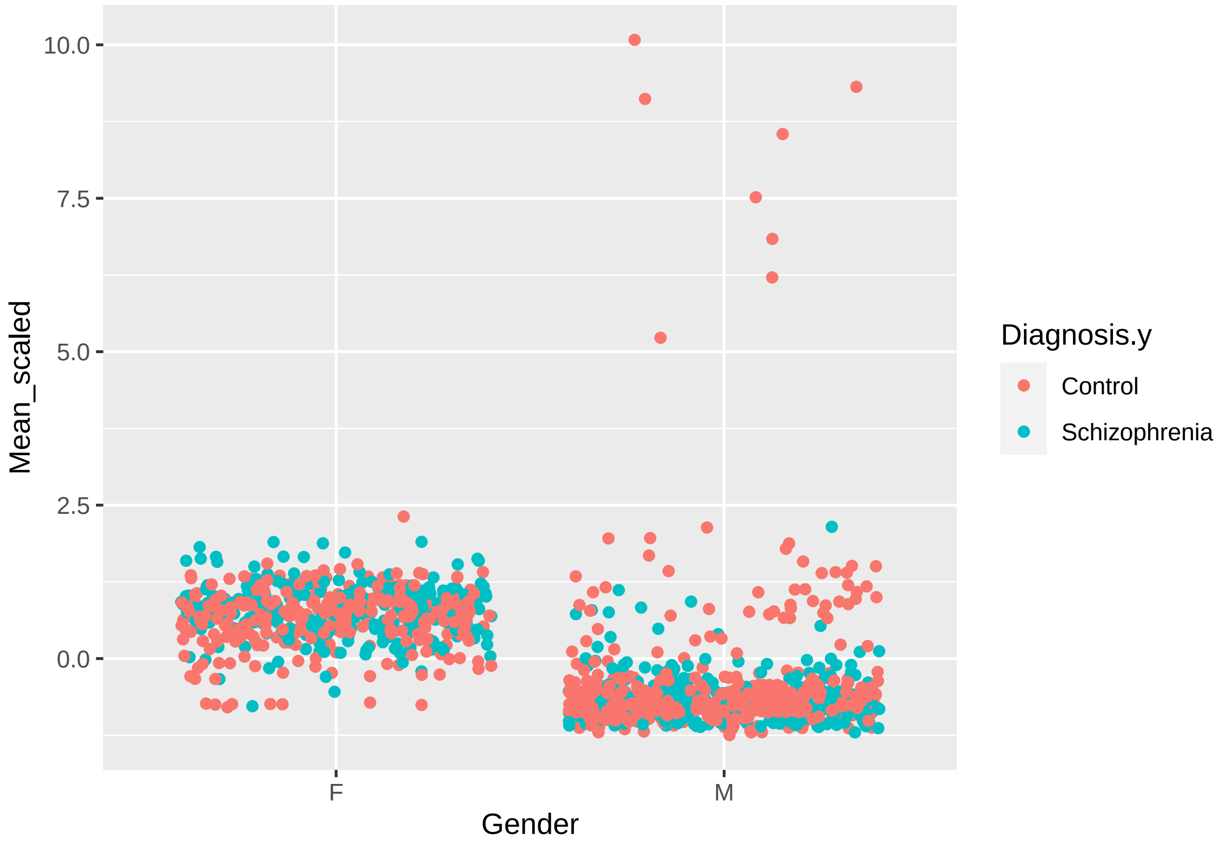
Now we need to describe our sample

First look at the missing data: we should exclude all recordings for which we do not have complete data. Then count the participants and recordings by diagnosis, report their gender, age and symptom severity (SANS, SAPS and Social) Finally, do the same by diagnosis and study, to assess systematic differences in studies. I like to use group_by() %>% summarize() for quick summaries

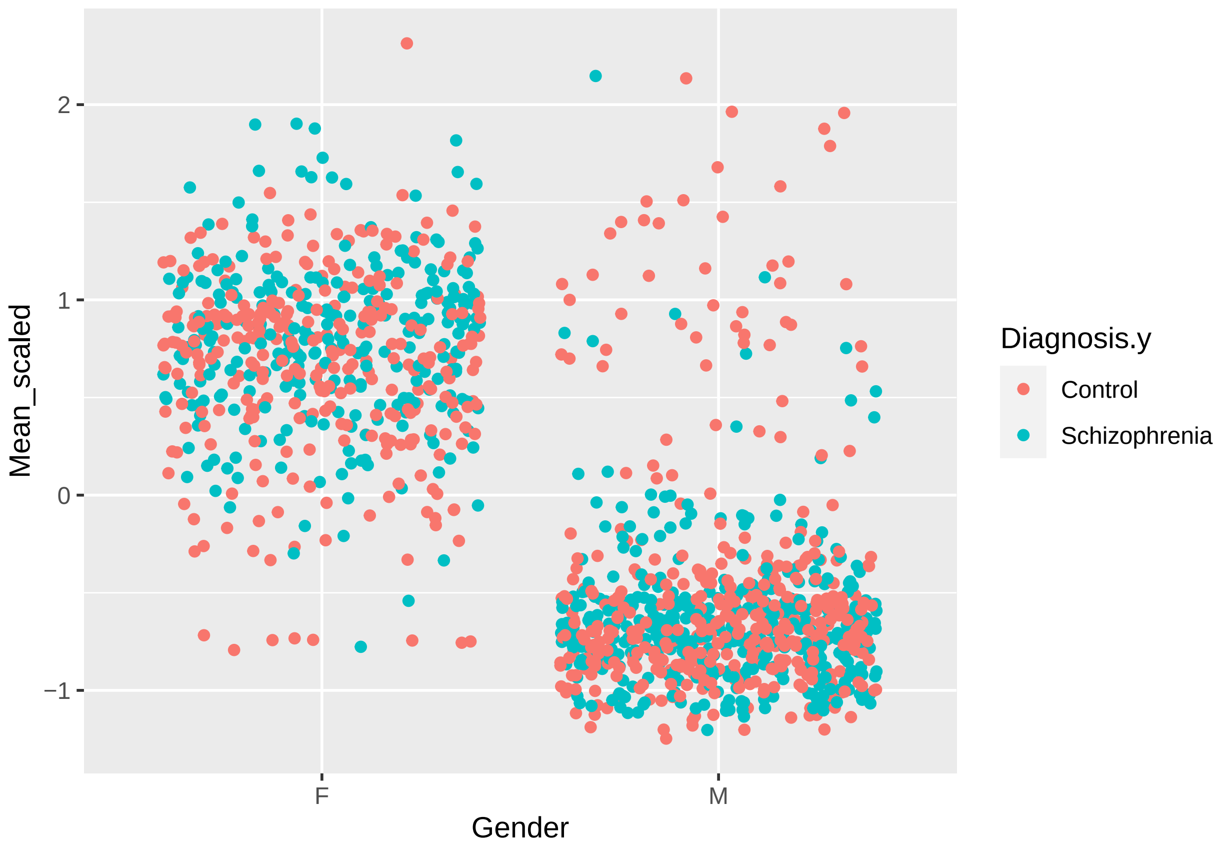
```
# Plotting the data
df %>% ggplot(aes(Diagnosis.y, Mean, color = Gender)) +
    geom_jitter()
```



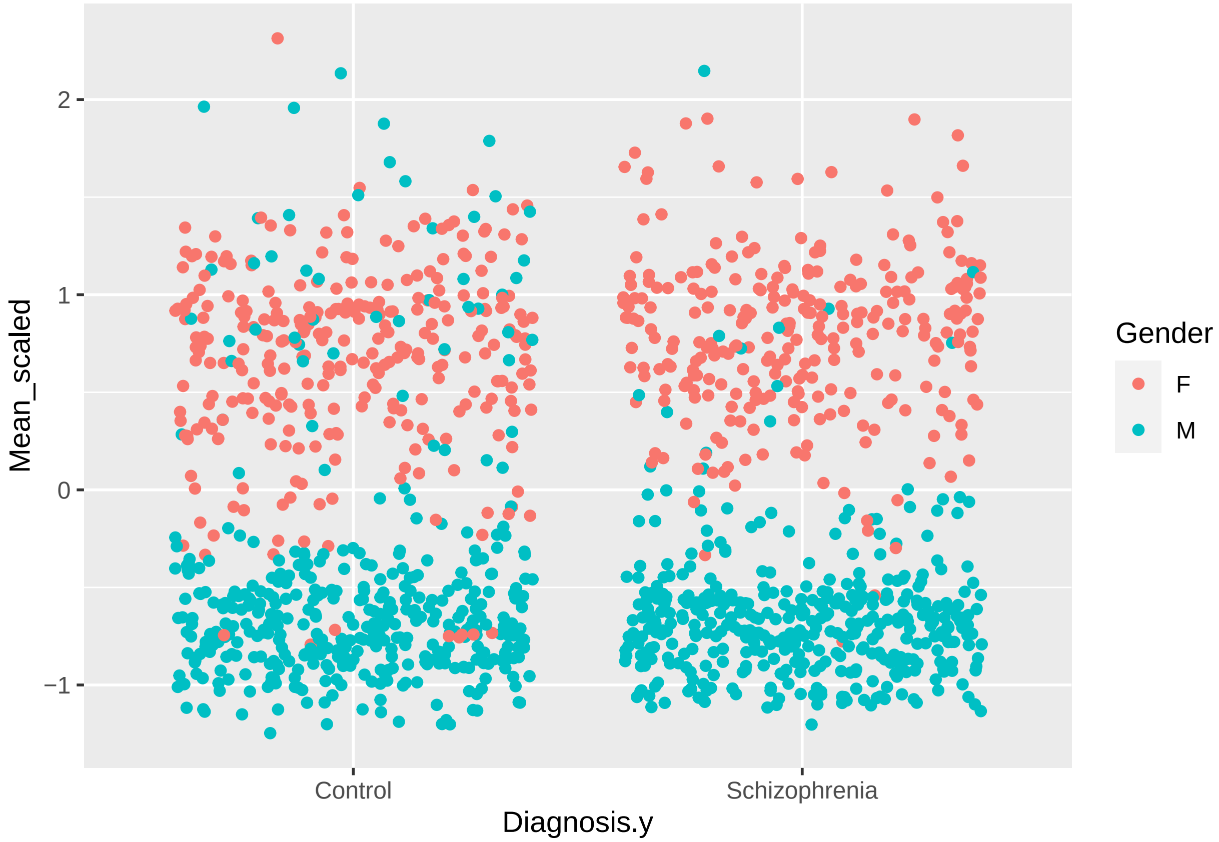
```r
df %>% ggplot(aes(Gender, Mean_scaled, color = Diagnosis.y)) +
    geom_jitter()
```



```r
# Omitting obvious outliers and plotting again
df <- df %>% filter(Mean_scaled < 3)

df %>% ggplot(aes(Gender, Mean_scaled, color = Diagnosis.y)) +
    geom_jitter()
```



```
df %>% ggplot(aes(Diagnosis.y, Mean_scaled, color = Gender)) +
    geom_jitter()
```



```
# Describing our sample
df %>% group_by(Diagnosis.y) %>% summarize(n()) # Count of diagnosis
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 2 x 2
##   Diagnosis.y   `n()`
##   <fct>         <int>
## 1 Control         722
## 2 Schizophrenia   694
```

```
df %>% group_by(Diagnosis.y) %>% summarize(mean(SAPS)) # Mean SAPS by diagnosis
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 2 x 2
##   Diagnosis.y   `mean(SAPS)`
##   <fct>                <dbl>
## 1 Control             0.0817
## 2 Schizophrenia      10.3
```

```
df %>% group_by(Diagnosis.y) %>% summarize(mean(SANS))# Mean SANS by diagnosis
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
## # A tibble: 2 x 2
##   Diagnosis.y   `mean(SANS)`
##   <fct>                <dbl>
## 1 Control              0.413
## 2 Schizophrenia        9.52
```

```
df %>% group_by(Diagnosis.x, Gender) %>% summarize(n()) # Gender distribution by diagnosis
```

```
## `summarise()` regrouping output by 'Diagnosis.x' (override with `.groups` argument)
```

```
## # A tibble: 4 x 3
## # Groups:   Diagnosis.x [2]
##   Diagnosis.x Gender `n()`
##   <fct>       <fct>  <int>
## 1 0           F        299
## 2 0           M        423
## 3 1           F        279
## 4 1           M        415
```

```
df %>% group_by(Gender) %>% summarize(n()) # General gender distribution
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 2 x 2
##   Gender `n()`
##   <fct>  <int>
## 1 F        578
## 2 M        838
```

```
df %>% group_by(Participant) %>% slice(1) %>% group_by(Gender) %>% summarize(mean(Age), sd(Age)) # Mean
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 2 x 3
##   Gender `mean(Age)` `sd(Age)`
##   <fct>        <dbl>     <dbl>
## 1 F             22.6      3.36
## 2 M             24.2      3.79
```

```
df %>% group_by(Participant) %>% slice(1) %>% group_by(Diagnosis.y) %>% summarize(mean(Age), sd(Age)) #
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 2 x 3
##   Diagnosis.y   `mean(Age)` `sd(Age)`
##   <fct>               <dbl>     <dbl>
## 1 Control              23.4      3.81
## 2 Schizophrenia        23.6      3.60
```

```
df %>% group_by(Study.y) %>% summarize(n()) # Count of study
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 3 x 2
##   Study.y `n()`
##     <dbl> <int>
## 1       1   634
## 2       2   340
## 3       4   442
```

```
df %>% group_by(Study.y) %>% summarize(mean(SANS)) # Mean SANS by study
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 3 x 2
##   Study.y `mean(SANS)`
##     <dbl>        <dbl>
## 1       1         4.94
## 2       2         4.64
## 3       4         4.97
```

```
df %>% group_by(Study.y) %>% summarize(mean(SAPS)) # Mean SAPS by study
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 3 x 2
##   Study.y `mean(SAPS)`
##     <dbl>        <dbl>
## 1       1         5.14
## 2       2         6.94
## 3       4         3.62
```

```
df %>% group_by(Study.y) %>% summarize(mean(Age)) # Mean age by study
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 3 x 2
##   Study.y `mean(Age)`
##     <dbl>       <dbl>
## 1       1        22.8
## 2       2        23.5
## 3       4        24.6
```

```
df %>% group_by(Study.y, Diagnosis.y) %>% summarize(mean(SAPS)) # Mean SAPS by study and diagnosis
```

```
## `summarise()` regrouping output by 'Study.y' (override with `.groups` argument)
## # A tibble: 6 x 3
## # Groups:   Study.y [3]
##   Study.y Diagnosis.y   `mean(SAPS)`
##     <dbl> <fct>                <dbl>
## 1       1 Control              0
## 2       1 Schizophrenia       10.5
## 3       2 Control              0
## 4       2 Schizophrenia       14.4
## 5       4 Control              0.267
## 6       4 Schizophrenia        6.98
```

```
df %>% group_by(Study.y, Diagnosis.y) %>% summarize(mean(SANS)) # Mean SANS by study and diagnosis
```

```
## `summarise()` regrouping output by 'Study.y' (override with `.groups` argument)
```

```
## # A tibble: 6 x 3
## # Groups:   Study.y [3]
##   Study.y Diagnosis.y   `mean(SANS)`
##     <dbl> <fct>                <dbl>
## 1       1 Control               0
## 2       1 Schizophrenia        10.1
## 3       2 Control               0
## 4       2 Schizophrenia         9.63
## 5       4 Control               1.35
## 6       4 Schizophrenia         8.59
```

Now we can analyze the data

If you were to examine the meta analysis you would find that the differences (measured as Hedges' g, very close to Cohen's d, that is, in standard deviations) to be the following - pitch variability (IQR) (lower, Hedges' g: -0.55, 95% CIs: -1.06, 0.09) - proportion of spoken time (phonationtime..s. / dur..s.) (lower, Hedges' g: -1.26, 95% CIs: -2.26, 0.25) - speech rate (speechrate..nsyl.dur) (slower, Hedges' g: -0.75, 95% CIs: -1.51, 0.04) - pause duration ((dur..s. - phonationtime..s.)/npause) (longer, Hedges' g: 1.89, 95% CIs: 0.72, 3.21). (Duration - Spoken Duration) / PauseN

We need therefore to set up 4 models to see how well our results compare to the meta-analytic findings (Feel free of course to test more features) Describe the acoustic profile of a schizophrenic voice *Note* in this section you need to describe the acoustic profile of a schizophrenic voice and compare it with the meta-analytic findings (see 2 and 3 in overview of part 1).

N.B. the meta-analytic findings are on scaled measures. If you want to compare your results with them, you need to scale your measures as well: subtract the mean, and divide by the standard deviation. N.N.B. We want to think carefully about fixed and random effects in our model. In particular: how should study be included? Does it make sense to have all studies put together? Does it make sense to analyze both languages together? Relatedly: does it make sense to scale all data from all studies together? N.N.N.B. If you want to estimate the studies separately, you can try this syntax: Feature ~ 0 + Study + Study:Diagnosis + [your randomEffects]. Now you'll have an intercept per each study (the estimates for the controls) and an effect of diagnosis per each study

- Bonus points: cross-validate the models and report the betas and standard errors from all rounds to get an idea of how robust the estimates are.

```r
# Discribing acustic features of the sample
df %>% group_by(Diagnosis.y) %>% summarize(mean(ProportionSpokenTime), sd(ProportionSpokenTime)) # Mean
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 2 x 3
##   Diagnosis.y   `mean(ProportionSpokenTime)` `sd(ProportionSpokenTime)`
##   <fct>                                <dbl>                      <dbl>
## 1 Control                              0.616                      0.116
## 2 Schizophrenia                        0.603                      0.155
```

```r
df %>% group_by(Diagnosis.y) %>% summarize(mean(speechrate..nsyll.dur.), sd(speechrate..nsyll.dur.)) #
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 2 x 3
##   Diagnosis.y   `mean(speechrate..nsyll.dur.)` `sd(speechrate..nsyll.dur.)`
##   <fct>                                  <dbl>                        <dbl>
## 1 Control                                 3.09                        0.678
## 2 Schizophrenia                           2.90                        0.850
```

```r
df %>% group_by(Diagnosis.y) %>% summarize(mean(IQR), sd(IQR)) # Mean and sd for the IQR by diagnosis
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 2 x 3
##   Diagnosis.y   `mean(IQR)` `sd(IQR)`
##   <fct>               <dbl>     <dbl>
## 1 Control              39.2      66.4
## 2 Schizophrenia        25.0      36.7
```

```r
df %>% group_by(Diagnosis.y) %>% summarize(mean(pause_dur), sd(pause_dur)) # Mean and sd for the pause
```

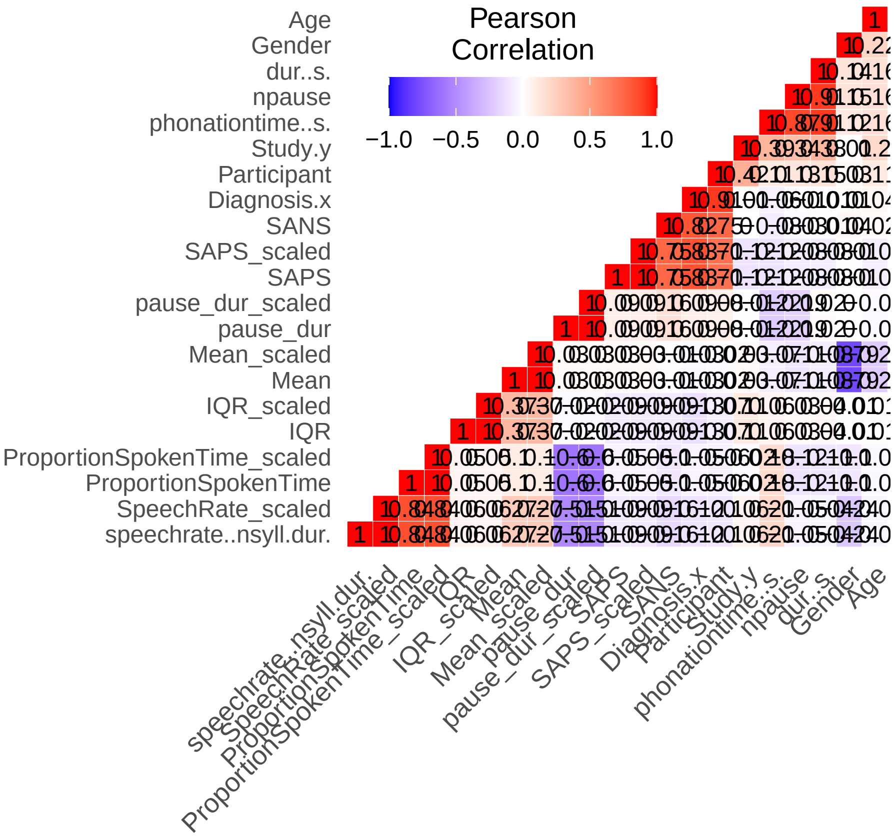
```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 2 x 3
##   Diagnosis.y   `mean(pause_dur)` `sd(pause_dur)`
##   <fct>                     <dbl>           <dbl>
## 1 Control                   0.957           0.479
## 2 Schizophrenia             1.09            0.972
```

In order to avoid collinearity, we constructed a heatmap of the correlations between the variables.

```r
# Defining the functions needed for the heatmap
# Use correlation between variables as distance
reorder_cormat <- function(cormat) {
  dd <- as.dist((1 - cormat) / 2)
  hc <- hclust(dd)
  cormat <- cormat[hc$order, hc$order]
  }

# Get upper triangle of the correlation matrix
get_upper_tri <- function(cormat) {
  cormat[lower.tri(cormat)] <- NA
  return(cormat)
  }
```

```
# Getting rid of some variables and making gender numeric
heat_df <- df %>% select(-c(Diagnosis.y, Trial)) %>%
  mutate(Gender = as.numeric(as.factor(Gender))) %>%
  mutate(Diagnosis.x = as.numeric(as.factor(Diagnosis.x))) %>%
  mutate(Participant = as.numeric(as.factor(Participant)))


# Creating heatmap of correlations
heatmap <- round(cor(heat_df), 2) %>%
    reorder_cormat() %>%
    get_upper_tri() %>%
    melt(na.rm = T) %>%
    ggplot(aes(Var2, Var1, fill = value)) +
    geom_tile(color = "white") +
    scale_fill_gradient2(
        low = "blue",
        high = "red",
        mid = "white",
        midpoint = 0,
        limit = c(-1, 1),
        space = "Lab",
        name = "Pearson\nCorrelation"
    ) +
    theme_minimal() +
    theme(axis.text.x = element_text(
        angle = 45,
        vjust = 1,
        size = 10,
        hjust = 1
    )) +
    coord_fixed() + geom_text(aes(Var2, Var1, label = value),
                              color = "black",
                              size = 3) +
    theme(
        axis.title.x = element_blank(),
        axis.title.y = element_blank(),
        panel.grid.major = element_blank(),
        panel.border = element_blank(),
        panel.background = element_blank(),
        axis.ticks = element_blank(),
        legend.justification = c(1, 0),
        legend.position = c(0.6, 0.7),
        legend.direction = "horizontal"
    ) +
    guides(fill = guide_colorbar(
        barwidth = 7,
        barheight = 1,
        title.position = "top",
        title.hjust = 0.5
    ))

# Displaying heatmap
heatmap
```



```
# Building models

#Changing some variables to factors
df$Diagnosis.y <- as.factor(df$Diagnosis.y)
df$Participant <- as.factor(df$Participant)

# Predicting diagnosis by IQR including random intercepts for participants
m_pitch <-
  glmer(Diagnosis.y ~ IQR_scaled + (1 | Participant),
        data = df,
        family = "binomial",
        control = glmerControl(optimizer = "nloptwrap", calc.derivs = FALSE))

summary(m_pitch)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: Diagnosis.y ~ IQR_scaled + (1 | Participant)
##    Data: df
## Control: glmerControl(optimizer = "nloptwrap", calc.derivs = FALSE)
##
##      AIC      BIC   logLik deviance df.resid
##    369.0    384.7   -181.5    363.0     1413
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -0.07725 -0.04976 -0.03196  0.05052  0.10682
##
## Random effects:
##  Groups      Name        Variance Std.Dev.
##  Participant (Intercept) 288.9    17
## Number of obs: 1416, groups:  Participant, 173
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.1364     1.3975  -0.098    0.922
## IQR_scaled   -0.3326     0.8521  -0.390    0.696
##
## Correlation of Fixed Effects:
##            (Intr)
## IQR_scaled 0.018
```

```r
# Predicting diagnosis by proportion of spoken time including random intercepts for participants
m_pst <-
  glmer(Diagnosis.y ~ ProportionSpokenTime_scaled + (1 | Participant),
        df,
       family = "binomial",
        control = glmerControl(optimizer = "nloptwrap", calc.derivs = FALSE))

summary(m_pst)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: Diagnosis.y ~ ProportionSpokenTime_scaled + (1 | Participant)
##    Data: df
## Control: glmerControl(optimizer = "nloptwrap", calc.derivs = FALSE)
##
##      AIC      BIC   logLik deviance df.resid
##    369.4    385.2   -181.7    363.4     1413
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -0.07883 -0.04960 -0.04333  0.05109  0.09796
##
## Random effects:
##  Groups      Name        Variance Std.Dev.
##  Participant (Intercept) 279.2    16.71
## Number of obs: 1416, groups:  Participant, 173
```

```
##
## Fixed effects:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  -0.2014     1.3738  -0.147    0.883
## ProportionSpokenTime_scaled  -0.1400     0.7105  -0.197    0.844
##
## Correlation of Fixed Effects:
##             (Intr)
## PrprtnSpkT_ 0.003
```

```
# Predicting diagnosis by speech rate including random intercepts for participants
m_sr <-
  glmer(Diagnosis.y ~ SpeechRate_scaled + (1 | Participant),
       df,
       family = "binomial",
       control = glmerControl(optimizer = "nloptwrap", calc.derivs = FALSE)
      )

summary(m_sr)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: Diagnosis.y ~ SpeechRate_scaled + (1 | Participant)
##    Data: df
## Control: glmerControl(optimizer = "nloptwrap", calc.derivs = FALSE)
##
##      AIC      BIC   logLik deviance df.resid
##    368.8    384.6   -181.4    362.8     1413
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -0.07873 -0.04907 -0.03921  0.05003  0.10047
##
## Random effects:
##  Groups      Name        Variance Std.Dev.
##  Participant (Intercept) 284      16.85
## Number of obs: 1416, groups:  Participant, 173
##
## Fixed effects:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -0.1741     1.3859  -0.126    0.900
## SpeechRate_scaled  -0.3080     0.7160  -0.430    0.667
##
## Correlation of Fixed Effects:
##             (Intr)
## SpchRt_scld -0.012
```

```
# Predicting diagnosis by pause duration including random intercepts for participants
m_pd <-
  glmer(Diagnosis.y ~ pause_dur_scaled + (1 | Participant),
        df,
        family = "binomial",
        control = glmerControl(optimizer = "nloptwrap", calc.derivs = FALSE)
       )

summary(m_pd)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: Diagnosis.y ~ pause_dur_scaled + (1 | Participant)
##    Data: df
## Control: glmerControl(optimizer = "nloptwrap", calc.derivs = FALSE)
##
##      AIC      BIC   logLik deviance df.resid
##    369.2    384.9   -181.6    363.2     1413
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -0.07808 -0.04961 -0.04325  0.05128  0.09689
##
## Random effects:
##  Groups      Name        Variance Std.Dev.
##  Participant (Intercept) 278      16.67
## Number of obs: 1416, groups:  Participant, 173
##
## Fixed effects:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -0.1957     1.3712  -0.143    0.887
## pause_dur_scaled   0.2205     0.8011   0.275    0.783
##
## Correlation of Fixed Effects:
##             (Intr)
## pas_dr_scld 0.010
```

```
# Predicting diagnosis by pause duration and IQR including random intercepts for participants
m_max1 <-
  glmer(Diagnosis.y ~ IQR_scaled + pause_dur_scaled + (1 | Participant),
        df,
        family = "binomial")

# Predicting diagnosis by speech rate and IQR including random intercepts for participants
m_max2 <-
  glmer(Diagnosis.y ~ IQR_scaled + SpeechRate_scaled + (1 | Participant),
        df,
        family = "binomial")

# Predicting diagnosis by proportion of spoken time and IQR including random intercepts for participant
m_max3 <-
  glmer(Diagnosis.y ~ IQR_scaled + ProportionSpokenTime_scaled + (1 | Participant),
        df,
        family = "binomial")

# Comparing models with anova
anova(m_pitch, m_pst, m_sr, m_pd)
```

```
## Data: df
## Models:
## m_pitch: Diagnosis.y ~ IQR_scaled + (1 | Participant)
## m_pst: Diagnosis.y ~ ProportionSpokenTime_scaled + (1 | Participant)
## m_sr: Diagnosis.y ~ SpeechRate_scaled + (1 | Participant)
## m_pd: Diagnosis.y ~ pause_dur_scaled + (1 | Participant)
##         npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## m_pitch    3 368.97 384.74 -181.49   362.97
## m_pst      3 369.40 385.17 -181.70   363.40 0.0000  0          1
## m_sr       3 368.84 384.61 -181.42   362.84 0.5604  0     <2e-16 ***
## m_pd       3 369.18 384.95 -181.59   363.18 0.0000  0          1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(m_max1, m_max2, m_max3)
```

```
## Data: df
## Models:
## m_max1: Diagnosis.y ~ IQR_scaled + pause_dur_scaled + (1 | Participant)
## m_max2: Diagnosis.y ~ IQR_scaled + SpeechRate_scaled + (1 | Participant)
## m_max3: Diagnosis.y ~ IQR_scaled + ProportionSpokenTime_scaled + (1 |
## m_max3:     Participant)
##        npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## m_max1    4 286.54 307.56 -139.27   278.54
## m_max2    4 286.50 307.52 -139.25   278.50 0.0371  0     <2e-16 ***
## m_max3    4 286.61 307.63 -139.30   278.61 0.0000  0          1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Making linear models
# Predicting IQR by diagnosis including random intercepts for participants
m1 <-
  lmerTest::lmer(data = df,
                 IQR_scaled ~ Diagnosis.y + (1 | Participant),
                 REML = FALSE)

# Predicting proportion of spoken time by diagnosis including random intercepts for participants
m2 <-
  lmerTest::lmer(data = df,
                 ProportionSpokenTime_scaled ~ Diagnosis.y + (1 | Participant),
                 REML = FALSE)

# Predicting speech rate by diagnosis including random intercepts for participants
m3 <-
  lmerTest::lmer(data = df,
                 SpeechRate_scaled ~ Diagnosis.y + (1 | Participant),
                 REML = FALSE)

# Predicting pause duration by diagnosis including random intercepts for participants
m4 <-
  lmerTest::lmer(data = df,
                 pause_dur_scaled ~ Diagnosis.y + (1 | Participant),
                 REML = FALSE)
```

```r
# Calculating Cohen's D

# To use cohen's d command, we will make two subset df's separately only having Schizophrenia participa
df_Sch <- df %>%
  filter(Diagnosis.y == "Schizophrenia")

df_Con <- df %>%
  filter(Diagnosis.y == "Control")

# IQR
cohen.d(df_Sch$IQR_scaled, df_Con$IQR_scaled, na.rm = TRUE, pooled = TRUE, paired = FALSE, hedges = TRU
```

```
##
## Hedges's g
##
## g estimate: -0.2638951 (small)
## 95 percent confidence interval:
##      lower      upper
## -0.3685728 -0.1592174
```

```r
# Our g: -0.26, 95% CIs: -0.36, -0.16
# Meta g: -0.55, 95% CIs: -1.06, 0.09

# Prop of spoken time
cohen.d(df_Sch$ProportionSpokenTime_scaled, df_Con$ProportionSpokenTime_scaled, na.rm = TRUE, pooled =
```

```
##
## Hedges's g
##
## g estimate: -0.1005432 (negligible)
## 95 percent confidence interval:
##        lower        upper
## -0.204834235  0.003747818
```

```r
# Our g: -0.10, 95% CIs: -0.20, -0.03
# Meta g: -1.26, 95% CIs: -2.26, 0.25

# Speechrate
cohen.d(df_Sch$SpeechRate_scaled, df_Con$SpeechRate_scaled, na.rm = TRUE, pooled = TRUE, paired = FALSE
```

```
##
## Hedges's g
##
## g estimate: -0.2471036 (small)
## 95 percent confidence interval:
##      lower      upper
## -0.3517256 -0.1424815
```

```r
# Our g: -0.25, 95% CIs: -0.35, -0.14
# Meta g: -0.75, 95% CIs: -1.51, 0.04

# Pause duration
cohen.d(df_Sch$pause_dur_scaled, df_Con$pause_dur_scaled, na.rm = TRUE, pooled = TRUE, paired = FALSE,
##
## Hedges's g
##
## g estimate: 0.1806631 (negligible)
## 95 percent confidence interval:
##      lower      upper
## 0.07622556 0.28510064
```

```
# Our g: 0.18, 95% CIs: 0.07, 0.29
# Meta g: 1.89, 95% CIs: 0.72, 3.21
```