04 CSI online spoken: Spoken - Plotting and analysis - final data

Kirsten Stark

26 September, 2024

Load packages

```
#library(dplyr)
library(tidyr)
library(lme4)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
library(lmerTest)
```

```
##
## Attaching package: 'lmerTest'
```

```
## The following object is masked from 'package:lme4':
##
##     lmer
```

```
## The following object is masked from 'package:stats':
##
##     step
```

```
library(Rmisc)
```

```
## Loading required package: lattice
```

```
## Warning: package 'lattice' was built under R version 4.3.2
```

```
## Loading required package: plyr
```

```r
library(Cairo)
#library(strengejacke)
library(ggplot2)
# devtools::install_github("strengejacke/sjPlot")
library(sjPlot)
```

```
## Install package "strengejacke" from GitHub (‘devtools::install_github("strengejacke/strengejacke")‘)
```

```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
options(scipen=999)

rm(list = ls())
options( "encoding" = "UTF-8" )
set.seed(99)
```

Load preprocessed data

```r
# input
input = "aphasia_final.csv"

# load data
df <- read.csv2(here::here("data","transient_data_files", input), sep=",") #%>% select(-"X")
```

Check amount of participants and trials

```r
# no. of participants:
length(unique(df$subject))
```

```
## [1] 40
```

```
# no. of trials is 160 per participant?
nrow(df) == 3*160 * length(unique(df$subject))
```

```
## [1] TRUE
```

```
#table(df$subject, df$session)

# how many non-responses
df %>% filter(VOT==0) %>% dplyr::group_by(type, subject,session) %>%
  dplyr::summarise(length(VOT))
```

```
## `summarise()` has grouped output by 'type', 'subject'. You can override using
## the `.groups` argument.
```

```
## # A tibble: 86 x 4
## # Groups:   type, subject [37]
##    type  subject session `length(VOT)`
##    <chr>   <int>   <int>         <int>
##  1 PWA       101       1             8
##  2 PWA       101       2             4
##  3 PWA       103       1            40
##  4 PWA       103       2            21
##  5 PWA       103       3            27
##  6 PWA       104       1            75
##  7 PWA       104       2            52
##  8 PWA       104       3            37
##  9 PWA       105       1            10
## 10 PWA       105       2             7
## # i 76 more rows
```

```
# table(df$VOT==0, df$subject, df$session)
```

Drop filler trials

```
all_data <- df
df <- df %>% filter(category!="Filler") %>% droplevels()
```

Add ordinal position

```
      # add position number
df <- df %>% group_by(subject, session, category) %>%
      add_count() %>%
      dplyr::mutate(PosOr = seq(1:n)) %>% dplyr::select(-n)
table(df$PosOr)
```

```
##
##    1    2    3    4    5
## 2880 2880 2880 2880 2880
```

```
#table(df$PosOr,df$session, df$subject)
```

Factorize columns

```
# factorize columns
df$VOT <- as.numeric(as.character(df$VOT))
is.numeric(df$VOT)
```

```
## [1] TRUE
```

```
df$PosOr <- as.factor(df$PosOr)
df$group <- as.factor(df$type)
df$subject <- as.factor(df$subject)
df$session <- as.factor(df$session)

# define contrasts of session: compare 1 to 2 and 1 to 3, intercept is the grand mean => simple coding
c<-contr.treatment(3)
my.coding<-matrix(rep(1/3, 6), ncol=2)
my.simple<-c-my.coding
my.simple
```

```
##            2          3
## 1 -0.3333333 -0.3333333
## 2  0.6666667 -0.3333333
## 3 -0.3333333  0.6666667
```

```
contrasts(df$session)<-my.simple
levels(df$session)
```

```
## [1] "1" "2" "3"
```

```
## Define contrast of group
contrasts(df$group) <- MASS::contr.sdif(2)
levels(df$group)
```

```
## [1] "control" "PWA"
```

```
## Define Ordinal position as continuous predictor variable
df$PosOr_cont <- as.numeric(scale(as.numeric(as.character(df$PosOr)),
                                        center = T, scale = F))
```

Errors and correct responses

Correct responses start with 1.

1 - correct.
1.1 - correct with alternative response.

1.2 - correct with phonematic paraphasia (<=25% of the word).
1.3 - correct with correct article [*].

1.4 - correct, but VOT invalid.

0 - wrong. 0.1 - wrong with phonematic paraphrasia (> 25 % of the word). 0.2 - wrong: semantic paraphrasia (word in the experiment). 0.3 - wrong: semantic paraphrasia (word not in the experiment). 0.4 - wrong: null reaction. 0.5 - wrong: replacement without connection to the word (word in the experiment)
0.6 - wrong: replacement without connection to the word (word not in the experiment).
0.7 - superordinate word.
0.8 - neologism. 0.9 - etc.

0.99 - TECHNICAL ERROR.

We will consider all responses in 1.1-1.3 for analyses of response times and responses 1.1-1.4 and 0.1-0.9 (not 1.4 and 0.99) fir analyses of error rates.

```r
## Add technical errors in the missing trials
sum(is.na(df$VOT)) # NA VOT so far are technical errors
```

```
## [1] 4
```

```r
df$error[is.na(df$VOT)] <- "99"
df$correct[is.na(df$VOT)] <- "0"

## Two trials were forgotten to be classified, but AR == 99 --> technical error?
#sum(is.na(df$correct))
#df %>% filter(is.na(correct))
df$error[is.na(df$correct) & df$AR == "99"] <- "99"
df$correct[is.na(df$correct) & df$AR == "99"] <- "0"
sum(is.na(df$correct))
```

```
## [1] 0
```

```r
## NR and 0.4 are the same -> replace this
df$error[df$error=="NR"] <- "4"

## Rename broken names
# unique(df$error)
df$error <- stringr::str_replace(df$error, ";;;;;;", "")
df$error <- stringr::str_replace(df$error, "ok", "") # subject 113, session 2, trial 121 (Couch)
df$error <- gsub("?",NA,df$error, fixed = TRUE)
# unique(df$error)

# unique(df$correct)
df$correct <- stringr::str_replace(df$correct, ";;;;;;", "")
# unique(df$correct)
```

```r
## Overall amount of correct answers
sum(df$correct != 0)
```

```
## [1] 12589
```

```
sum(is.na(df$correct)) # these are the technical errors were no audio file was recorded
```

```
## [1] 0
```

```
## Overview of correct responses
table(df$correct)
```

```
##
##     0     1   1.1   1.2   1.3   1.4
##  1811 10901  1207   124   223   134
```

```
#df$VOT[df$correct==1.4]

# Overview of incorrect responses
sum(df$correct==0, na.rm=T)
```

```
## [1] 1811
```

```
sum(df$correct == 0 & !is.na(df$error)) # here the error classification was missing
```

```
## [1] 1810
```

```
df[df$correct == 0 & is.na(df$error),]
```

```
## # A tibble: 1 x 41
## # Groups:   subject, session, category [1]
##   type  subject session trial item  category supercategory   VOT correct AR
##   <chr> <fct>   <fct>   <int> <chr> <chr>    <chr>         <dbl> <chr>   <chr>
## 1 PWA   113     2         121 couch Sitzen   Möbel          1846 0       Cous
## # i 31 more variables: error <chr>, gender <int>, age <int>, language <int>,
## #   handedness <int>, CH01 <int>, CH01_01 <int>, CH01_02 <int>, CH01_03 <int>,
## #   CH01_04 <int>, CH02 <int>, CH02_01 <int>, CH02_02 <int>, CH02_03 <int>,
## #   CH02_04 <int>, CH03 <int>, array <int>, comments <lgl>, timetotal <chr>,
## #   time_correct <chr>, SD22_OS <int>, SD22_BID <chr>, SD22_BNM <int>,
## #   SD22_FmF <int>, SD22_ScW <int>, SD22_ScH <int>, SD22_QnW <int>,
## #   OR02_01 <chr>, PosOr <fct>, group <fct>, PosOr_cont <dbl>
```

```
df$error[df$correct == 0 & is.na(df$error)] <- 1 # phonet. paraphrasia > 25 %
```

Overview of correctness classifications by group

```
df %>% group_by(type) %>% dplyr::count(correct)
```

```
## # A tibble: 12 x 3
## # Groups:   type [2]
##    type    correct     n
##    <chr>   <chr>   <int>
##  1 PWA     0        1563
##  2 PWA     1        4756
##  3 PWA     1.1       502
##  4 PWA     1.2       107
##  5 PWA     1.3       153
##  6 PWA     1.4       119
##  7 control 0         248
##  8 control 1        6145
##  9 control 1.1       705
## 10 control 1.2        17
## 11 control 1.3        70
## 12 control 1.4        15
```

Errors

```
table(df$error)
```

```
##
##    1   2   3   4   5   6   7   8   9  99
##   57 188  94 851  33  46 137  32 270 103
```

Show amount of incorrect trials per ordinal position (excluding fillers):

```
## How many incorrect (correct) non-filler trials per ordinal position?
table(df$PosOr[df$category != "Filler" & df$correct == 0],
      df$correct[df$category != "Filler" & df$correct == 0])
```

```
##
##       0
##   1 320
##   2 349
##   3 375
##   4 347
##   5 420
```

```
table(df$PosOr[df$category != "Filler" & startsWith("1", df$correct)],
      df$correct[df$category != "Filler" & startsWith("1", df$correct)])
```

```
##
##        1
##   1 2254
##   2 2200
##   3 2158
##   4 2202
##   5 2087
```

```
## How many incorrect trials (no technical errors) per ordinal position?
table(df$PosOr[df$category != "Filler" & df$correct == 0 &
                 df$error != 99])
```

```
##
##   1   2   3   4   5
## 303 333 347 331 394
```

Show amount of incorrect trials per subject

```
df %>% filter(category != "Filler") %>%
  group_by(subject, session) %>%
  dplyr::count(correct) %>%
  mutate(prop=round(n/160*100,2)) %>% #round(prop.table(n), 4)) %>%
  filter(correct == "0") %>%
  dplyr::select(-c(correct, n))
```

```
## # A tibble: 115 x 3
## # Groups:   subject, session [115]
##    subject session  prop
##    <fct>   <fct>   <dbl>
##  1 101     1        5.62
##  2 101     2        2.5
##  3 101     3        0.62
##  4 102     1        1.25
##  5 102     2        1.88
##  6 102     3        2.5
##  7 103     1       26.2
##  8 103     2       20.6
##  9 103     3       23.1
## 10 104     1       43.8
## # i 105 more rows
```

Total percentage of errors

```
sum(df$correct[df$category != "Filler"]=="0", na.rm=T)/nrow(df%>%filter(category != "Filler"))
```

```
## [1] 0.1257639
```

Summarise erroneous and correct responses

```
classification_summary <- df %>% group_by(group, session) %>% count(correct) %>%
  mutate(correct = case_when(correct == "0" ~ "wrong sum",
                             correct == "1" ~ "correct",
                             correct == "1.1" ~
                               "correct with alternative response",
                             correct == "1.2" ~
                               "correct with phonematic paraphasia (<=25% of the word)",
                             correct == "1.3" ~ "correct with correct article",
                             correct == "1.4" ~ "correct, but VOT invalid")) %>%
  rename(classification=correct)

x <- df %>% group_by(group, session) %>% count(error) %>%
  mutate(error=as.character(error)) %>%
  mutate(error=case_when(
    error == "1" ~
      "wrong with phonematic paraphrasia (> 25 % of the word)",
    error == "2" ~
```

```
      "wrong: semantic paraphrasia (word in the experiment)",
    error == "3" ~
      "wrong: semantic paraphrasia (word not in the experiment)",
    error == "4" ~
      "wrong: null reaction",
    error == "5" ~
      "wrong: replacement without connection to the word (word in the experiment) ",
    error == "6" ~
    "wrong: replacement without connection to the word (word not in the experiment)",
    error == "7" ~ "wrong: superordinate word",
    error == "8" ~ "wrong: neologism",
    error == "9" ~ "wrong: etc.",
    error == "99" ~ "TECHNICAL ERROR",
    is.na(error) ~ "sum correct")) %>%
  rename(classification = error)
(classification_summary <- rbind(classification_summary, x) %>%
  arrange(group, session))
```

```
## # A tibble: 96 x 4
## # Groups:   group, session [6]
##    group   session classification                                               n
##    <fct>   <fct>   <chr>                                                    <int>
##  1 control 1       wrong sum                                                  116
##  2 control 1       correct                                                   1917
##  3 control 1       correct with alternative response                          291
##  4 control 1       correct with phonematic paraphasia (<=25% of the word)       2
##  5 control 1       correct with correct article                                66
##  6 control 1       correct, but VOT invalid                                     8
##  7 control 1       wrong: semantic paraphrasia (word in the experiment)        15
##  8 control 1       wrong: semantic paraphrasia (word not in the experimen~     13
##  9 control 1       wrong: null reaction                                        14
## 10 control 1       wrong: replacement without connection to the word (wor~      2
## # i 86 more rows
```

```
# Export as word file
library(flextable)
huxt_word <- huxtable::huxtable(classification_summary)
huxt_word <- huxtable::set_number_format(huxt_word, round(2))
huxtable::quick_docx(
  huxt_word, file = here::here(
    "results", "tables",
    "CSI_online_aphasia_classification_summary.docx"),
  open = FALSE)

# by ordinal position
x <- df %>% group_by(group, session, PosOr) %>% count(error) %>%
  mutate(error=as.character(error)) %>%
  mutate(error=case_when(
    error == "1" ~
      "wrong with phonematic paraphrasia (> 25 % of the word)",
    error == "2" ~
      "wrong: semantic paraphrasia (word in the experiment)",
    error == "3" ~
```

```
      "wrong: semantic paraphrasia (word not in the experiment)",
    error == "4" ~
      "wrong: null reaction",
    error == "5" ~
      "wrong: replacement without connection to the word (word in the experiment) ",
    error == "6" ~
    "wrong: replacement without connection to the word (word not in the experiment)",
    error == "7" ~ "wrong: superordinate word",
    error == "8" ~ "wrong: neologism",
    error == "9" ~ "wrong: etc.",
    error == "99" ~ "TECHNICAL ERROR",
    is.na(error) ~ "sum correct")) %>%
  rename(classification = error)
(x %>% filter(group=="PWA") %>%
  arrange(group, session, PosOr))
```

```
## # A tibble: 153 x 5
## # Groups:   group, session, PosOr [15]
##    group session PosOr classification                                          n
##    <fct> <fct>   <fct> <chr>                                               <int>
##  1 PWA   1       1     "wrong: semantic paraphrasia (word in the experime~     5
##  2 PWA   1       1     "wrong: semantic paraphrasia (word not in the expe~     6
##  3 PWA   1       1     "wrong: null reaction"                                 53
##  4 PWA   1       1     "wrong: replacement without connection to the word~     2
##  5 PWA   1       1     "wrong: replacement without connection to the word~     4
##  6 PWA   1       1     "wrong: superordinate word"                             6
##  7 PWA   1       1     "wrong: neologism"                                      3
##  8 PWA   1       1     "wrong: etc."                                          22
##  9 PWA   1       1     "TECHNICAL ERROR"                                       2
## 10 PWA   1       1     "sum correct"                                         377
## # i 143 more rows
```

```
## correct responses, but VOT invalid
  # 1.4 - correct, but VOT invalid.
(df %>% filter(group=="PWA" & correct=="1.4") %>%
    group_by(group, session, PosOr) %>%
    count(correct) %>%
    arrange(group, session, PosOr))
```

```
## # A tibble: 15 x 5
## # Groups:   group, session, PosOr [15]
##    group session PosOr correct     n
##    <fct> <fct>   <fct> <chr>   <int>
##  1 PWA   1       1     1.4        15
##  2 PWA   1       2     1.4         9
##  3 PWA   1       3     1.4        12
##  4 PWA   1       4     1.4         7
##  5 PWA   1       5     1.4        11
##  6 PWA   2       1     1.4         5
##  7 PWA   2       2     1.4         7
##  8 PWA   2       3     1.4         5
##  9 PWA   2       4     1.4         4
## 10 PWA   2       5     1.4         7
## 11 PWA   3       1     1.4         9
## 12 PWA   3       2     1.4         7
## 13 PWA   3       3     1.4         4
## 14 PWA   3       4     1.4         6
## 15 PWA   3       5     1.4        11
```

Subset data for reaction time and error analyses and delete fillers

As correct reaction times will be considered:
1 - correct.
1.1 - correct with alternative response.
1.2 - correct with phonematic paraphasia (<=25% of the word).
1.3 - correct with correct article [*].

```
df %>% mutate(correct_class = case_when(
  correct == 1 ~ 1,
  correct ==1.1 ~ 1,
  correct == 1.2 ~ 1,
  correct == 1.3 ~ 1,
  correct == 1.4 ~ 0,
  correct == 0 ~ 0)) -> df

# Fillers included
# df %>% group_by(group, session) %>% dplyr::count(correct_class)
table(df$correct_class)
```

```
##
##     0     1
##  1945 12455
```

```
# Fillers excluded
# df %>% filter(category != "Filler") %>%
#   group_by(group, session) %>% dplyr::count(correct_class)
table(df$correct_class[df$category != "Filler"])
```

```
##
##     0     1
##  1945 12455
```

```
### Save data frame for RT analyses: Only correct responses and no fillers
df_RTs <- df %>% filter(correct_class == 1 & category != "Filler")
# table(df_RTs$correct_class, df_RTs$correct)
# sum(df_RTs$VOT == 0); sum(is.na(df_RTs$VOT))

df_RTs %>% group_by(group, session) %>% count()
```

```
## # A tibble: 6 x 3
## # Groups:   group, session [6]
##   group   session     n
##   <fct>   <fct>   <int>
## 1 control 1        2276
## 2 control 2        2312
## 3 control 3        2349
## 4 PWA     1        1723
## 5 PWA     2        1864
## 6 PWA     3        1931
```

```
(df_RTs %>% group_by(group, session) %>% count(correct) -> x)
```

```
## # A tibble: 24 x 4
## # Groups:   group, session [6]
##    group   session correct     n
##    <fct>   <fct>   <chr>   <int>
##  1 control 1       1        1917
##  2 control 1       1.1       291
##  3 control 1       1.2         2
##  4 control 1       1.3        66
##  5 control 2       1        2095
##  6 control 2       1.1       209
##  7 control 2       1.2         6
##  8 control 2       1.3         2
##  9 control 3       1        2133
## 10 control 3       1.1       205
## # i 14 more rows
```

```
# sum(x$n)

print(paste0("Amount of trials that went into RT analyses: ",
             nrow(df_RTs)))
```

```
## [1] "Amount of trials that went into RT analyses: 12455"
```

```
table(df_RTs$group)
```

```
##
## control     PWA
##    6937    5518
```

```
# by ordinal position
(df_RTs %>% filter(group=="PWA") %>% group_by(group, session, PosOr) %>%
    mutate(correct=case_when(correct=="1" ~ "correct",
                             correct=="1.1" ~ "correct with alternative response",
                             correct=="1.2" ~ "correct with phonematic paraphasia (<=25% of the word)",
                             correct=="1.3" ~ "correct with correct article")) %>%
    count(correct) -> x)
```

```
## # A tibble: 60 x 5
## # Groups:   group, session, PosOr [15]
##    group session PosOr correct                                                 n
##    <fct> <fct>   <fct> <chr>                                               <int>
##  1 PWA   1       1     correct                                               310
##  2 PWA   1       1     correct with alternative response                      34
##  3 PWA   1       1     correct with correct article                           14
##  4 PWA   1       1     correct with phonematic paraphasia (<=25% of the w~     4
##  5 PWA   1       2     correct                                               285
##  6 PWA   1       2     correct with alternative response                      35
##  7 PWA   1       2     correct with correct article                           15
##  8 PWA   1       2     correct with phonematic paraphasia (<=25% of the w~    10
##  9 PWA   1       3     correct                                               270
## 10 PWA   1       3     correct with alternative response                      42
## # i 50 more rows
```

As errors on the participant side will be considered:
1 - wrong with phonematic paraphrasia (> 25 % of the word).
2 - wrong: semantic paraphrasia (word in the experiment).
3 - wrong: semantic paraphrasia (word not in the experiment).
4 - wrong: null reaction.
5 - wrong: replacement without connection to the word (word in the experiment).
6 - wrong: replacement without connection to the word (word not in the experiment).
7 - superordinate word.
8 - neologism.
9 - etc.

```
df %>% mutate(error_class = case_when(
  error ==1 | error == 2 |error == 3 |
    error==4 | error==5 | error == 6 | error == 7 |
    error == 8 | error == 9 ~ 1,
  error == 99 | is.na(error) ~ 0)) -> df
# Overview including Fillers
# df %>% group_by(group, session) %>% count(error_class)
table(df$error_class)
```

```
##
##     0     1
## 12692  1708
```

```
# Overview excluding Fillers
# df %>% filter(category != "Filler") %>%
#   group_by(group, session) %>% count(error_class)
table(df$error_class[df$category != "Filler"])
```

```
##
##     0     1
## 12692  1708
```

```
#### Subset data for error analyses: All trials excluding technical errors, invalid RTs and fillers
df_errors <- df %>% filter(category != "Filler" &
                             (error != "99" | is.na(error)))

print(paste0("Amount of trials that went into RT analyses: ",
             nrow(df_errors)))
```

```
## [1] "Amount of trials that went into RT analyses: 14297"
```

```r
table(df_errors$group)
```

```
##
## control     PWA
##    7119    7178
```

RESPONSE TIMES

```r
sum(!is.na(df_RTs$error))
```

```
## [1] 0
```

Descriptives

```r
(means_final<- df_RTs %>%
   filter(category != "Filler") %>%
   Rmisc::summarySEwithin(.,"VOT",idvar = "subject",
                          withinvars = c("session", "PosOr"),
                          betweenvars = "group", na.rm = T))
```

```
##      group session PosOr   N      VOT       sd       se       ci
## 1  control       1     1 461 1297.252 386.6087 18.00616 35.38453
## 2  control       1     2 456 1278.075 368.6706 17.26458 33.92821
## 3  control       1     3 446 1352.942 419.3224 19.85549 39.02218
## 4  control       1     4 457 1329.444 375.5740 17.56861 34.52548
## 5  control       1     5 456 1347.666 395.4900 18.52052 36.39636
## 6  control       2     1 466 1204.941 327.2233 15.15832 29.78730
## 7  control       2     2 462 1244.987 364.0399 16.93667 33.28264
## 8  control       2     3 464 1213.458 289.9285 13.45959 26.44945
## 9  control       2     4 466 1244.242 296.8703 13.75225 27.02425
## 10 control       2     5 454 1266.531 329.4120 15.46007 30.38236
## 11 control       3     1 469 1199.645 323.1567 14.92199 29.32239
## 12 control       3     2 471 1173.122 282.9187 13.03621 25.61646
## 13 control       3     3 467 1208.308 305.2131 14.12358 27.75378
## 14 control       3     4 470 1232.751 306.1228 14.12039 27.74706
## 15 control       3     5 472 1263.283 349.0940 16.06835 31.57453
## 16     PWA       1     1 362 1280.805 463.4011 24.35582 47.89711
## 17     PWA       1     2 345 1352.300 518.0623 27.89154 54.85942
## 18     PWA       1     3 337 1374.660 515.0242 28.05516 55.18588
## 19     PWA       1     4 357 1334.771 468.7956 24.81129 48.79513
## 20     PWA       1     5 322 1390.007 538.3805 30.00277 59.02691
## 21     PWA       2     1 375 1168.998 438.9850 22.66909 44.57485
## 22     PWA       2     2 386 1188.463 443.3652 22.56670 44.36939
## 23     PWA       2     3 370 1212.971 435.7079 22.65138 44.54198
## 24     PWA       2     4 378 1245.738 457.2034 23.51600 46.23896
## 25     PWA       2     5 355 1308.230 496.4997 26.35147 51.82511
## 26     PWA       3     1 395 1144.157 373.7086 18.80332 36.96739
## 27     PWA       3     2 386 1183.630 457.0922 23.26538 45.74310
## 28     PWA       3     3 396 1224.247 442.1547 22.21911 43.68250
## 29     PWA       3     4 384 1223.097 437.4259 22.32230 43.88959
## 30     PWA       3     5 370 1273.044 469.5947 24.41307 48.00619
```

```
(means_final_cat<- df_RTs %>%
   filter(category != "Filler") %>%
   Rmisc::summarySEwithin(.,"VOT",idvar = "category",
                          withinvars = c("session", "PosOr"),
                          betweenvars = "group",na.rm = T))
```

```
##      group session PosOr   N      VOT       sd       se       ci
## 1  control       1     1 461 1297.378 386.0154 17.97853 35.33023
## 2  control       1     2 456 1276.834 377.8956 17.69658 34.77718
## 3  control       1     3 446 1352.899 418.0216 19.79390 38.90113
## 4  control       1     4 457 1327.830 368.6545 17.24493 33.88939
## 5  control       1     5 456 1347.895 380.1481 17.80207 34.98447
## 6  control       2     1 466 1205.679 326.2835 15.11479 29.70175
## 7  control       2     2 462 1246.112 368.7969 17.15799 33.71756
## 8  control       2     3 464 1212.429 286.3328 13.29267 26.12143
## 9  control       2     4 466 1244.625 303.0652 14.03923 27.58818
## 10 control       2     5 454 1268.504 339.2988 15.92408 31.29423
## 11 control       3     1 469 1199.602 319.6380 14.75951 29.00311
## 12 control       3     2 471 1172.419 275.1627 12.67883 24.91421
## 13 control       3     3 467 1208.130 297.3314 13.75886 27.03708
## 14 control       3     4 470 1233.023 302.7635 13.96544 27.44258
## 15 control       3     5 472 1263.295 350.2301 16.12065 31.67729
## 16     PWA       1     1 362 1284.565 539.6713 28.36449 55.78039
## 17     PWA       1     2 345 1361.937 594.3400 31.99819 62.93673
## 18     PWA       1     3 337 1367.845 597.1421 32.52840 63.98497
## 19     PWA       1     4 357 1336.280 555.5199 29.40123 57.82193
## 20     PWA       1     5 322 1365.528 590.1073 32.88540 64.69813
## 21     PWA       2     1 375 1173.590 498.9082 25.76351 50.65949
## 22     PWA       2     2 386 1185.887 507.5593 25.83409 50.79356
## 23     PWA       2     3 370 1204.526 509.7624 26.50128 52.11249
## 24     PWA       2     4 378 1242.418 522.0492 26.85131 52.79711
## 25     PWA       2     5 355 1303.733 563.8296 29.92497 58.85308
## 26     PWA       3     1 395 1159.867 431.3062 21.70137 42.66497
## 27     PWA       3     2 386 1187.291 497.2603 25.30988 49.76289
## 28     PWA       3     3 396 1225.908 509.9080 25.62384 50.37616
## 29     PWA       3     4 384 1222.496 520.3697 26.55500 52.21185
## 30     PWA       3     5 370 1278.875 536.4006 27.88614 54.83569
```

```
(means_final_wo_session <- df_RTs %>%
   filter(category != "Filler") %>%
   Rmisc::summarySEwithin(.,"VOT",idvar = "subject",
                          withinvars = c("PosOr"),
                          betweenvars = "group",na.rm = T))
```

```
##      group PosOr    N      VOT       sd        se       ci
## 1  control     1 1396 1233.645 377.5129 10.103901 19.82048
## 2  control     2 1389 1231.481 370.6296  9.944638 19.50814
## 3  control     3 1377 1256.889 376.4044 10.143497 19.89839
## 4  control     4 1393 1268.317 356.9347  9.563421 18.76027
## 5  control     5 1382 1292.193 389.9203 10.488703 20.57551
## 6      PWA     1 1132 1196.084 464.1391 13.795102 27.06687
## 7      PWA     2 1117 1237.396 516.8288 15.463937 30.34167
## 8      PWA     3 1103 1266.420 506.7019 15.256848 29.93575
## 9      PWA     4 1119 1266.373 493.1586 14.742514 28.92611
## 10     PWA     5 1047 1320.946 542.9417 16.779533 32.92538
```

```
# Export as word file
library(flextable)
huxt_word <- huxtable::huxtable(means_final)
huxt_word <- huxtable::set_number_format(huxt_word, round(2))
huxtable::quick_docx(huxt_word,
                     file = here::here("results", "tables",
                                       "CSI_online_aphasia_subject_RT_by_session.docx"),
                                       open = FALSE)
```

Calculate the main effects

```
## Ordinal position effect
x <- df_RTs %>%
   filter(category != "Filler") %>%
   Rmisc::summarySEwithin(.,"VOT",idvar = "subject",
                          withinvars = "PosOr", #c("session", "PosOr"),
                         # betweenvars = "group",
                         na.rm = T)
((x$VOT[2]-x$VOT[1])+(x$VOT[3]-x$VOT[2])+(x$VOT[4]-x$VOT[3])+
    (x$VOT[5]-x$VOT[4]))/4
```

```
## [1] 21.94008
```

```
((x$VOT[5]-x$VOT[1]))/4
```

```
## [1] 21.94008
```

```
## session effect
x <- df_RTs %>%
   filter(category != "Filler") %>%
   Rmisc::summarySEwithin(.,"VOT",idvar = "subject",
                          withinvars = "session", #c("session", "PosOr"),
                         # betweenvars = "group",
                         na.rm = T)
(x$VOT[2]-x$VOT[1])
```

```
## [1] -101.4722
```

```
(x$VOT[3]-x$VOT[1])
```

```
## [1] -118.8204
## group effect
x <- df_RTs %>%
   filter(category != "Filler") %>%
   Rmisc::summarySE(.,"VOT",#idvar = "subject",
                          #withinvars = "session", #c("session", "PosOr"),
                          groupvars = "group",
                         na.rm = T)
(x$VOT[2]-x$VOT[1])
```

```
## [1] 173.2174
```

Calculate mean and SD of ordinal position effects

```
# First calculate the OrdPos effect for each subject and each category separately
df_RTs$OrdPos_manual <- NA_real_
isEmptyNumeric <- function(x) {
    return(identical(x, numeric(0)))
}


for (i in unique(df_RTs$subject)) {
  for(j in unique(df_RTs$category)) {
    for(l in unique(df_RTs$session)) {
      for (k in 2:5) {
        x <- df_RTs$VOT[df_RTs$subject==i &
                             df_RTs$category==j &
                             df_RTs$session==l &
                             df_RTs$PosOr==k]
        xM1 <- df_RTs$VOT[df_RTs$subject==i &
                             df_RTs$category==j &
                             df_RTs$session==l &
                             df_RTs$PosOr==k-1]
      if(!isEmptyNumeric(x) & !isEmptyNumeric(xM1)){
        if(!is.na(x) &  !is.na(xM1)){
              df_RTs$OrdPos_manual[df_RTs$subject==i &
                             df_RTs$category==j &
                             df_RTs$session==l &
                             df_RTs$PosOr==k] <- x - xM1
      }
      }
      }
    }
  }
}

# return summarySEwithin
 df_RTs %>%
   filter(category != "Filler") %>%
   Rmisc::summarySEwithin(.,"OrdPos_manual",
                          idvar = "subject",
                          withinvars = c("session"),
                          betweenvars = "group",
                          na.rm = T)
```

```
##     group session    N OrdPos_manual       sd       se       ci
## 1 control       1 1726      20.18442 587.7520 14.14731 27.74768
## 2 control       2 1790      20.34033 501.1588 11.84538 23.23223
## 3 control       3 1842      22.63550 468.9750 10.92710 21.43081
## 4     PWA       1 1106      16.76849 788.1171 23.69808 46.49832
## 5     PWA       2 1292      23.61023 685.6362 19.07491 37.42122
## 6     PWA       3 1353      22.18591 673.0508 18.29780 35.89517
```

```
 # without session
  df_RTs %>%
   filter(category != "Filler") %>%
   Rmisc::summarySE(.,measurevar="OrdPos_manual",
                          groupvars = "group",
                          na.rm = T)
```

```
##     group    N OrdPos_manual       sd       se       ci
## 1 control 5358      14.21911 425.1656 5.808403 11.38683
## 2     PWA 3751      30.87817 582.7884 9.515626 18.65631
```

```
  df_RTs %>%
   filter(category != "Filler") %>%
   filter(session=="1") %>%
   Rmisc::summarySE(.,measurevar="OrdPos_manual",
                          groupvars = "group",
                          na.rm = T)
```

```
##     group    N OrdPos_manual       sd       se       ci
## 1 control 1726      13.44786 480.4111 11.56359 22.68013
## 2     PWA 1106      26.30561 644.0762 19.36688 38.00001
```

```
   # Rmisc::summarySEwithin(.,"OrdPos_manual",
   #                         idvar = "subject",
   #                         withinvars = c("session"),
   #                         betweenvars = "group",
   #                         na.rm = T)
  # without group
   df_RTs %>%
   filter(category != "Filler") %>%
   Rmisc::summarySEwithin(.,"OrdPos_manual",
                          idvar = "subject",
                          withinvars = c("session"),
                          #betweenvars = "group",
                          na.rm = T)
```

```
##   session    N OrdPos_manual       sd        se       ci
## 1       1 2832      18.85037 673.0059 12.646555 24.79739
## 2       2 3082      21.71110 585.5095 10.546725 20.67933
## 3       3 3195      22.44511 564.3771  9.984676 19.57702
```

Trial types within correct responses

```
df %>% group_by(group) %>% count(correct)
```

```
## # A tibble: 12 x 3
## # Groups:   group [2]
##    group   correct     n
##    <fct>   <chr>   <int>
##  1 control 0         248
##  2 control 1        6145
##  3 control 1.1       705
##  4 control 1.2        17
##  5 control 1.3        70
##  6 control 1.4        15
##  7 PWA     0        1563
##  8 PWA     1        4756
##  9 PWA     1.1       502
## 10 PWA     1.2       107
## 11 PWA     1.3       153
## 12 PWA     1.4       119
```

```
df %>% group_by(group,session) %>% count(correct)
```

```
## # A tibble: 36 x 4
## # Groups:   group, session [6]
##    group   session correct     n
##    <fct>   <fct>   <chr>   <int>
##  1 control 1       0         116
##  2 control 1       1        1917
##  3 control 1       1.1       291
##  4 control 1       1.2         2
##  5 control 1       1.3        66
##  6 control 1       1.4         8
##  7 control 2       0          84
##  8 control 2       1        2095
##  9 control 2       1.1       209
## 10 control 2       1.2         6
## # i 26 more rows
```

Plotting

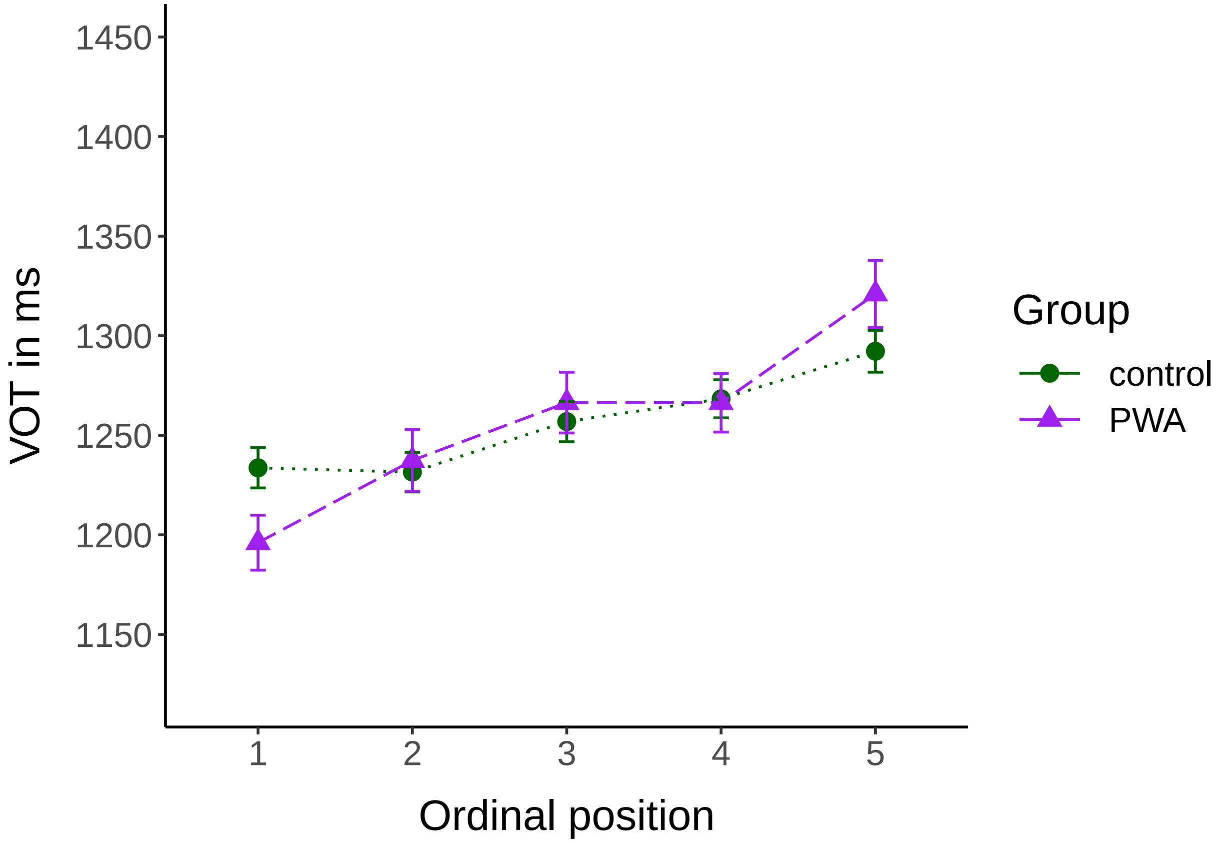
Make plots suitable for APA format, font sizes can be adjusted

```
apatheme <- theme_bw()+
  theme(plot.title=element_text(size=22,hjust = .5),# (family="Arial",size=22,hjust = .5),
        panel.grid.major=element_blank(), panel.grid.minor=element_blank(),
        panel.border=element_blank(),axis.line=element_line(),
        text=element_text(size=16))# text=element_text(family="Arial",size=16))

control_color <- "dark green" # #DDAA33"
PWA_color <- "purple" ##004488"
```

RTs across session, by ordinal position and group Line graph (only correct trials, without fillers): RTs, split by group but summarised across sessions

```
(plot_vot <- means_final_wo_session %>%
    ggplot(., aes(x=PosOr, y=VOT,color=group, group=group)) +
    geom_point(aes(shape=group), size=3)+
    scale_shape_manual(values=c(16,17))+
    stat_summary(aes(linetype=group),fun=mean,  geom="line", size = 0.5) +
    scale_linetype_manual(values=c("dotted", "longdash"))+
    geom_errorbar(aes(ymin=VOT-se, ymax=VOT+se, color=group), width =.1) +
    scale_color_manual(values=c(control_color, PWA_color))+
    apatheme+
    scale_y_continuous(limits = c(1120, 1450),
                       breaks =seq(1150,1450, by = 50)) +
    labs(x="Ordinal position",y ="VOT in ms", colour="Group",
         linetype="Group",
         shape="Group") +
    theme(
    axis.title.y = element_text(margin = margin(0,10,0,0)),
    axis.title.x = element_text(margin = margin(10,0,0,0)),
    legend.key.width = unit(1, "cm")))
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

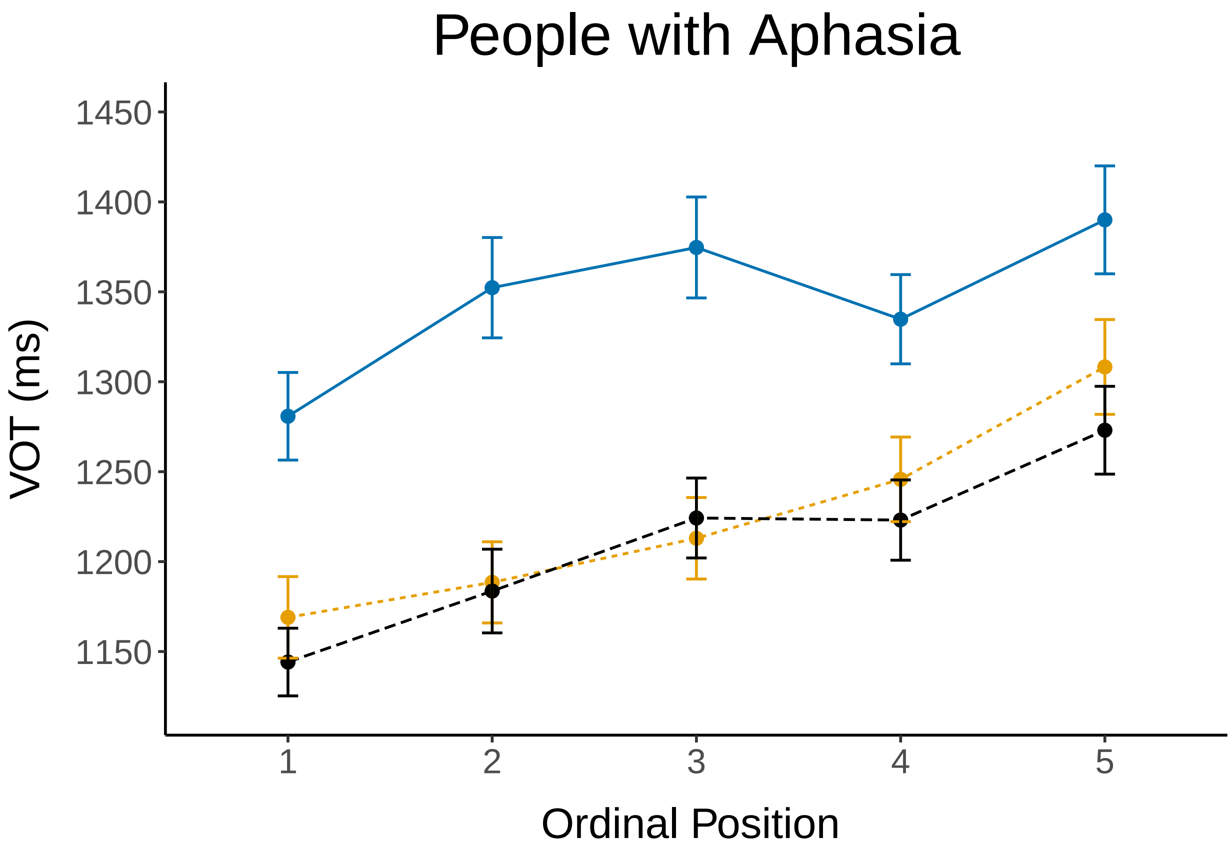


```
filename <- "CSI_online_aphasia_spoken_plot_rt_across_sessions.pdf"
ggsave(plot_vot, filename =
         here::here("results", "figures", filename),
       width = 18, height = 13, units = "cm",
       dpi = 300, device = cairo_pdf)
```

```


means_final<- df_RTs %>%
   filter(category != "Filler") %>%
   Rmisc::summarySEwithin(.,"VOT",idvar = "subject",
                          withinvars = c("PosOr", "session"),
                          betweenvars = "group", na.rm = T)
override.linetype<-c("longdash", "dashed", "dotted")
(plot_rt_repetition_PWA <- means_final %>% filter(group=="PWA") %>%
    mutate(session=case_when(session == "1" ~ "day 1",
                             session == "2" ~ "day 2",
                             session == "3" ~ "day 8")) %>%
    ggplot(., aes(x=PosOr, y=VOT, group=session, color = session)) +
    geom_point(size = 2)+
    stat_summary(aes(linetype=session),fun=mean,
                 geom="line", size = 0.5) +
    scale_linetype_manual(values=c("longdash", "dashed", "dotted"))+
    scale_color_manual(values=c("#0072B2", "#E69F00", "#000000"))+
    geom_errorbar(aes(ymin=VOT-se, ymax=VOT+se, group = session),
                   width =.1) +
    apatheme+
    scale_y_continuous(limits = c(1120, 1450),
                         breaks =seq(1150,1450, by = 50)) +
    labs(x="Ordinal Position ",y ="VOT (ms)", colour="Session",
         linetype="Session",
         title = "People with Aphasia") +
    theme(
    axis.title.y = element_text(margin = margin(0,10,0,0)),
    axis.title.x = element_text(margin = margin(10,0,0,0)),
    legend.key.width = unit(1, "cm"),
    legend.position="none")+
  guides(color=guide_legend(
     override.aes=list(linetype=override.linetype)))+
    scale_linetype(guide="none"))
```

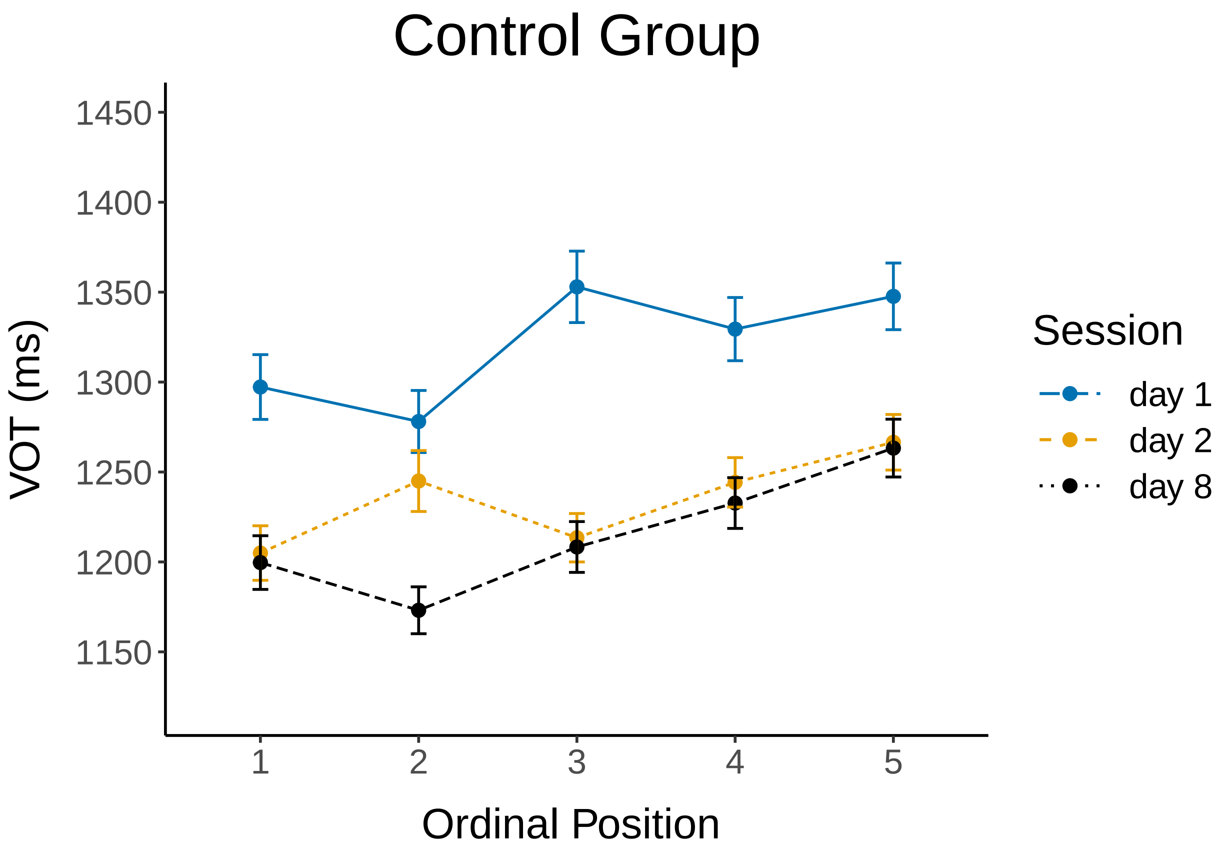
RTs by Group, session, and ordinal position

```
## Scale for linetype is already present.
## Adding another scale for linetype, which will replace the existing scale.
```



```
(plot_rt_repetition_control <- means_final %>%
    filter(group=="control") %>%
    mutate(session=case_when(session == "1" ~ "day 1",
                                  session == "2" ~ "day 2",
                                 session == "3" ~ "day 8")) %>%
    ggplot(., aes(x=PosOr, y=VOT, group=session, color = session)) +
    geom_point(size = 2)+
    stat_summary(aes(linetype=session),fun=mean,
                 geom="line", size = 0.5) +
    scale_linetype_manual(values=c("longdash", "dashed", "dotted"))+
    scale_color_manual(values=c("#0072B2", "#E69F00", "#000000"))+
    geom_errorbar(aes(ymin=VOT-se, ymax=VOT+se, group = session),
                  width =.1) +
    apatheme+
    scale_y_continuous(limits = c(1120, 1450),
                       breaks =seq(1150,1450, by = 50)) +
    labs(x="Ordinal Position ",y ="VOT (ms)", colour="Session", linetype="Session",
         title = "Control Group") +
    theme(
    axis.title.y = element_text(margin = margin(0,10,0,0)),
    axis.title.x = element_text(margin = margin(10,0,0,0)),
    legend.key.width = unit(1, "cm"))+
   guides(color=guide_legend(
     override.aes=list(linetype=override.linetype)))+
    scale_linetype(guide="none"))
```

```
## Scale for linetype is already present.
## Adding another scale for linetype, which will replace the existing scale.
```



```
plots <- cowplot::plot_grid(
  plot_rt_repetition_PWA,plot_rt_repetition_control,
  nrow = 1, ncol=2,  rel_widths = c(0.81,1), #rel_height = c(1,1),
  margin(1,1,1,1),
  labels = c("A", "B"),label_size = 34,
  label_fontfamily = "Helvetica", label_y = 1.01,
  label_x=-0.03)
```

```
## Warning in as_grob.default(plot): Cannot convert object of class
## marginsimpleUnitunitunit_v2 into a grob.
```

```
filename <- "CSI_online_aphasia_spoken_plot_rt_by_repetition.pdf"
ggsave(plots, filename =
         here::here("results", "figures", filename),
       width = 25, height = 13, units = "cm",
       dpi = 300, device = cairo_pdf)
#embedFonts(file = here::here("results", "figures", filename))
```

```

# means_subject <- df_RTs %>%
#    filter(category != "Filler") %>%
#    summarySEwithin(.,"VOT",withinvars = c("subject","session", "PosOr"),
#                    betweenvars="group")
# (means_subject <- means_subject %>%
#   group_by(subject) %>%
#   dplyr::mutate(VOT_norm = VOT - first(VOT)))
#
# (boxplot <-
#   ggplot() +
#   ## boxplot
#   geom_boxplot(data=means_subject, aes(x = PosOr,y =VOT_norm,
#                                        color=group),
#                #colour = "grey",
#                width = 0.3,fatten = 1)+
#   # ### individual means
#   geom_jitter(data=means_subject, aes(x = PosOr,y =VOT_norm, color=group),
#               position = position_dodge(0.6),
#               shape=19,
#               #color = "dark grey",
#               size=2)+
#   ### group means
#   stat_summary(data=means_subject, aes(x = PosOr,y =VOT_norm,
#                                        color=group),
#                fun=mean, geom="point",
#                #colour = "black",
#                shape=18, size=5)+
#   ### line
#   stat_summary(data=means_subject, aes(x = PosOr,y =VOT_norm,
#                                        color=group, group=group),
#                fun=mean, geom="line",
#                #colour = "black",
#                linetype = "longdash")+
#
#   ## other stuff
#   #scale_y_continuous(breaks = seq(600, 1300, by = 50))+
#   labs(x="Ordinal Position",y ="Normalized RTs (ms)")+
#   apatheme +
#   theme(
#     axis.title.y = element_text(margin = margin(0,10,0,0)),
#     axis.title.x = element_text(margin = margin(10,0,0,0))) +
#   coord_equal(ratio = 1/100))
#
# filename <- "CSI_online_aphasia_spoken_boxplot.pdf"
# ggsave(boxplot, filename =
#          here::here("results", "figures", filename),
#        width = 18, height = 18, units = "cm",
#        dpi = 300, device = cairo_pdf)
# #embedFonts(file = here::here("results", "figures", filename))
```

Normalized boxplot Export plot grid

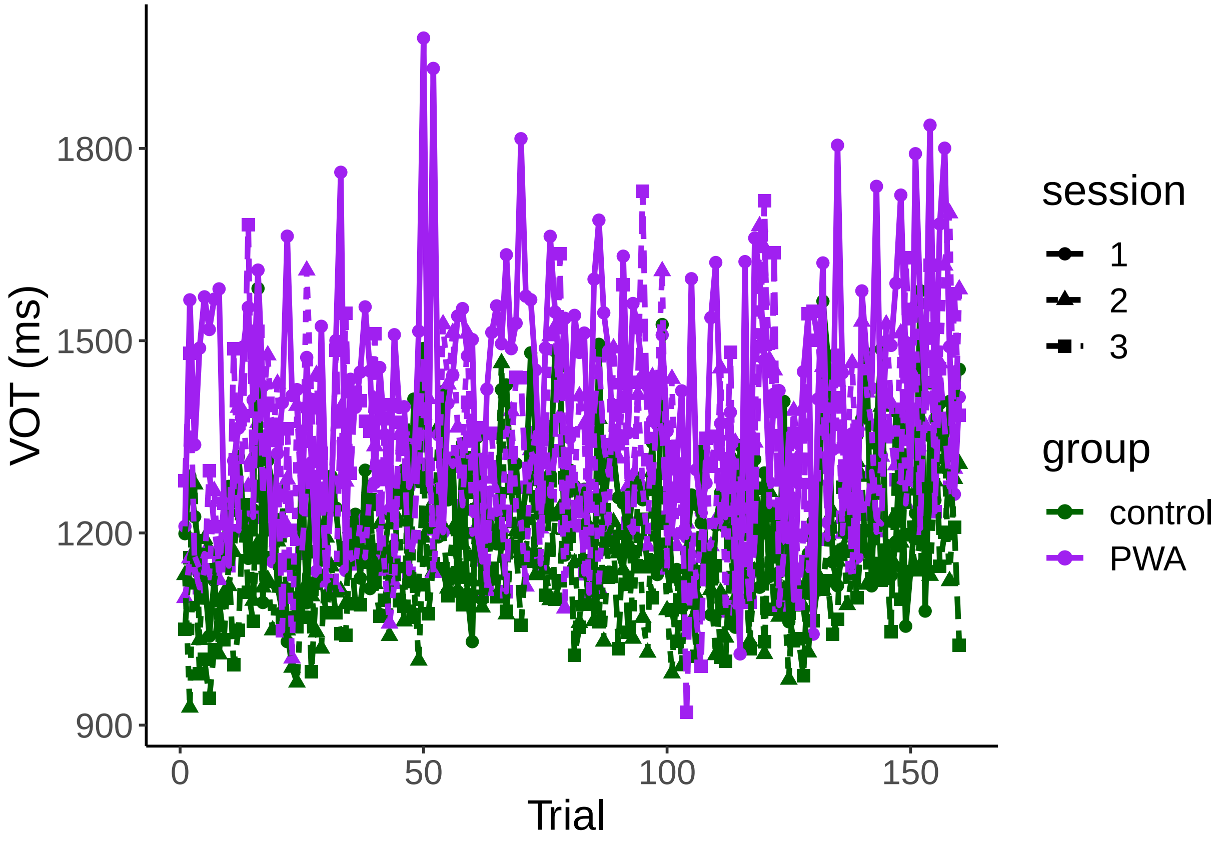
```
# cowplot::plot_grid(plot_rt, boxplot,
#            nrow = 1, labels = c("A", "B"), label_fontfamily = "Arial") %>%
#   ggsave(filename = here::here("results", "figures",
#                                 "CSI_online_aphasia_typing_RTs_and_normalized_RTs"),
#          width = 18, height = 13, units = "cm", dpi = 300,
#          device = cairo_pdf)
#embedFonts(file = here::here("results", "figures", "CSI_online_typing_RTs_and_normalized_RTs"))
```

```
# (plot_rt_fillers <- df %>%
#     mutate(kind = case_when(category == "Filler" ~"Filler",
#                             category != "Filler" ~"Experimental")) %>%
#     ggplot(., aes(x=PosOr, y=timing.01, group=kind, color=kind)) +
#     stat_summary(fun=mean,  geom="point", size = 2)+
#     stat_summary(fun=mean,  geom="line", size = 1) +
#     apatheme+
#     labs(x="Ordinal Position ",y ="RT (ms)", color = "Trial type")+
#   annotate(geom="text", x=1.5, y=1350, label="n = 30",
#            color="black", size = 8))
#
# filename <- "CSI_online_typing_plot_rt_with_fillers.pdf"
# ggsave(plot_rt_fillers, filename =
#          here::here("results", "figures", filename),
#        width = 18, height = 13, units = "cm",
#        dpi = 300, device = cairo_pdf)
# embedFonts(file = here::here("results", "figures", filename))
```

. . . with fillers for control

Control: Plot RTs accross the experiment All correct trials (Excluding filler)

```
(plot_RTs_all <- ggplot(data=df_RTs, aes(x=trial, y=VOT,
                                         linetype=session,shape=session,
                                         color=group)) +
  stat_summary(aes(color=group, shape=session),fun=mean,
               geom="point", size = 2)+
  stat_summary(aes(color=group, linetype=session),fun=mean,
               geom="line", size = 1) +
  apatheme+
  labs(x="Trial ",y ="VOT (ms)")+
  scale_color_manual(values=c(control_color, PWA_color)))
```



```
filename <- "CSI_online_aphasia__spoken_plot_rts_across_experiment.pdf"
ggsave(plot_RTs_all, filename =
         here::here("results", "figures", filename),
       width = 18, height = 13, units = "cm",
       dpi = 300, device = cairo_pdf)
#embedFonts(file = here::here("results", "figures", filename))
```

Inferential statistics

Contrast coding *Center predictor variable*
Across both groups.

```r
write.csv2(df_RTs,
  here::here("data","transient_data_files",
             "RT_data_final"),
  sep=",")
```

```
## Warning in write.csv2(df_RTs, here::here("data", "transient_data_files", :
## attempt to set 'sep' ignored
```

```r
df_RTs$PosOr.cont <- scale(as.numeric(as.character(df_RTs$PosOr)),
                                         center = T, scale = F)
table(df_RTs$PosOr.cont)
```

```
##
##  -1.98458450421517 -0.984584504215175 0.0154154957848252   1.01541549578483
##               2528               2506               2480               2512
##   2.01541549578483
##               2429
```

```r
mean(df_RTs$PosOr.cont); sd(df_RTs$PosOr.cont)
```

```
## [1] -0.000000000000000002197888
```

```
## [1] 1.412368
```

For PWA only

```r
df_RTs_PWA <- df_RTs %>% filter(group=="PWA") %>% droplevels()
df_RTs_PWA$PosOr.cont <- scale(as.numeric(as.character(df_RTs_PWA$PosOr)),
                                         center = T, scale = F)
table(df_RTs_PWA$PosOr.cont)
```

```
##
##  -1.96955418629938 -0.969554186299384 0.0304458137006161   1.03044581370062
##               1132               1117               1103               1119
##   2.03044581370062
##               1047
```

```r
mean(df_RTs_PWA$PosOr.cont); sd(df_RTs_PWA$PosOr.cont)
```

```
## [1] -0.000000000000000001123241
```

```
## [1] 1.40862
```

Compute further contrasts

```r
# define contrasts of session: compare 1 to 2 and 1 to 3, intercept is the grand mean => simple coding
c<-contr.treatment(3)
my.coding<-matrix(rep(1/3, 6), ncol=2)
my.simple<-c-my.coding
my.simple
```

```
##            2          3
## 1 -0.3333333 -0.3333333
## 2  0.6666667 -0.3333333
## 3 -0.3333333  0.6666667
```

```
contrasts(df_RTs$session)<-my.simple
levels(df_RTs$session)
```

```
## [1] "1" "2" "3"
```

```
contrasts(df_RTs_PWA$session)<-my.simple
levels(df_RTs_PWA$session)
```

```
## [1] "1" "2" "3"
```

```
## Define contrast of group
contrasts(df_RTs$group) <- MASS::contr.sdif(2)
levels(df_RTs$group)
```

```
## [1] "control" "PWA"
```

```
levels(df_RTs_PWA$group)
```

```
## [1] "PWA"
```

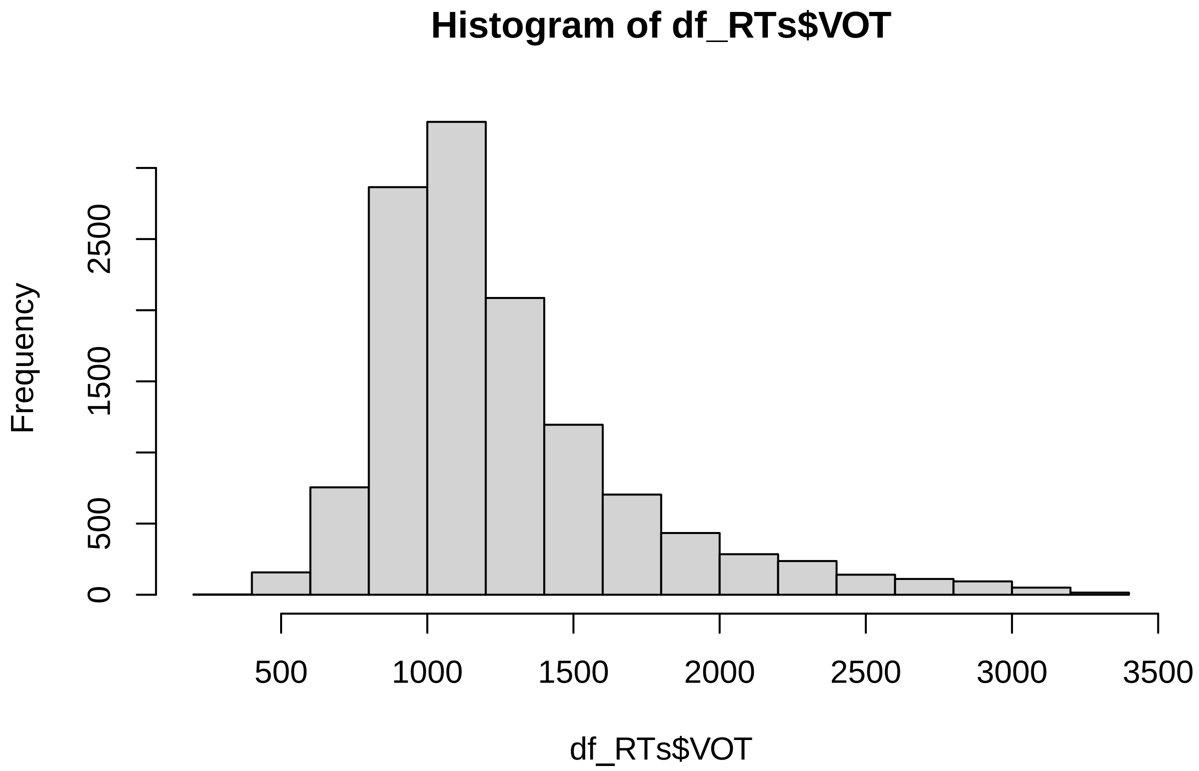
didLMERconverge function

```
## This function provides a better convergence check for lme4 v>1.0 models, which have a nasty habit of
didLmerConverge = function(lmerModel){
  relativeMaxGradient=signif(max(abs(with(
    lmerModel@optinfo$derivs,solve(Hessian,gradient)))),3)
  if (relativeMaxGradient < 0.001) {
    cat(sprintf("\tThe relative maximum gradient of %s is less than our 0.001 criterion.\n\tYou can saf
  }
  else {
    cat(sprintf("The relative maximum gradient of %s exceeds our 0.001 criterion.\nThis looks like a rea
  }
}

#didLmerConverge(m1)
```

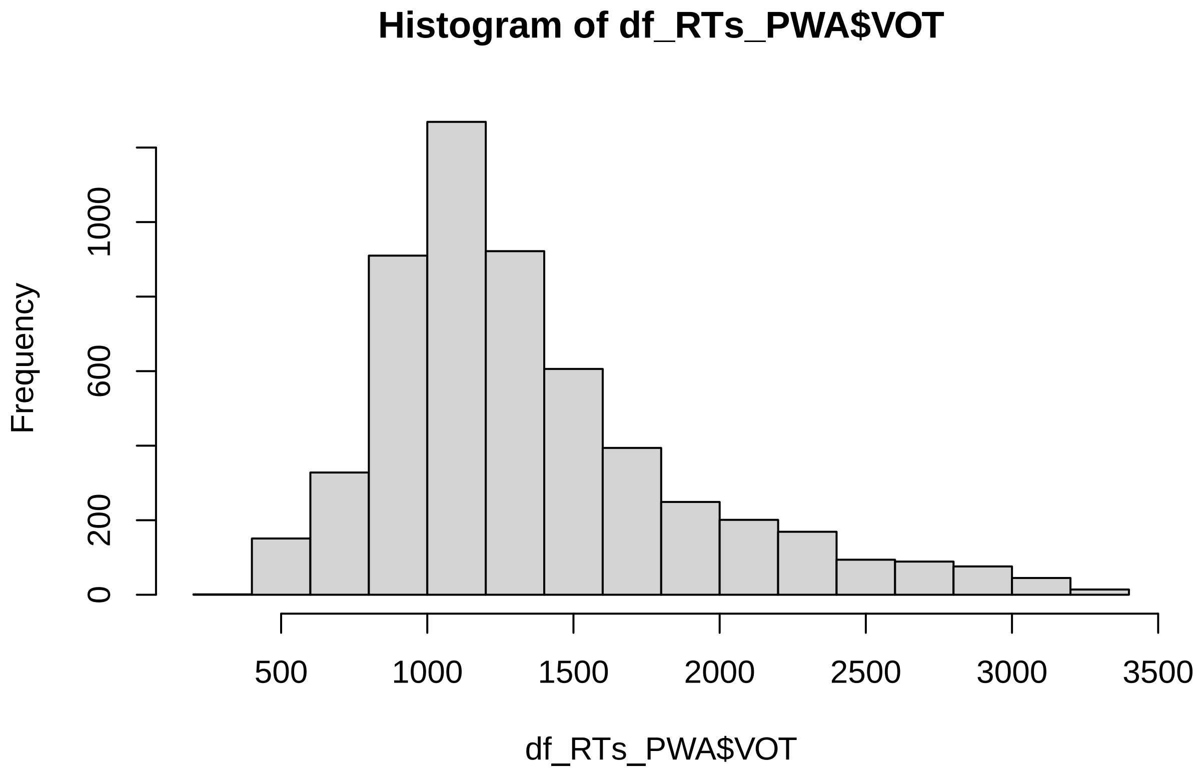
Check distribution of data Are the data normally distributed or do they need to be converted? Does a Gamma function fit the data better?

Histogram of the reaction time data

```
hist(df_RTs$VOT)
```



```r
hist(df_RTs_PWA$VOT)
```



Exclude unrealistically short reaction times < 200 ms

```
sum(df_RTs$VOT < 200)
```

```
## [1] 0
```

```
df_RTs <- df_RTs %>% filter(VOT >=200)

sum(df_RTs_PWA$VOT < 200)
```

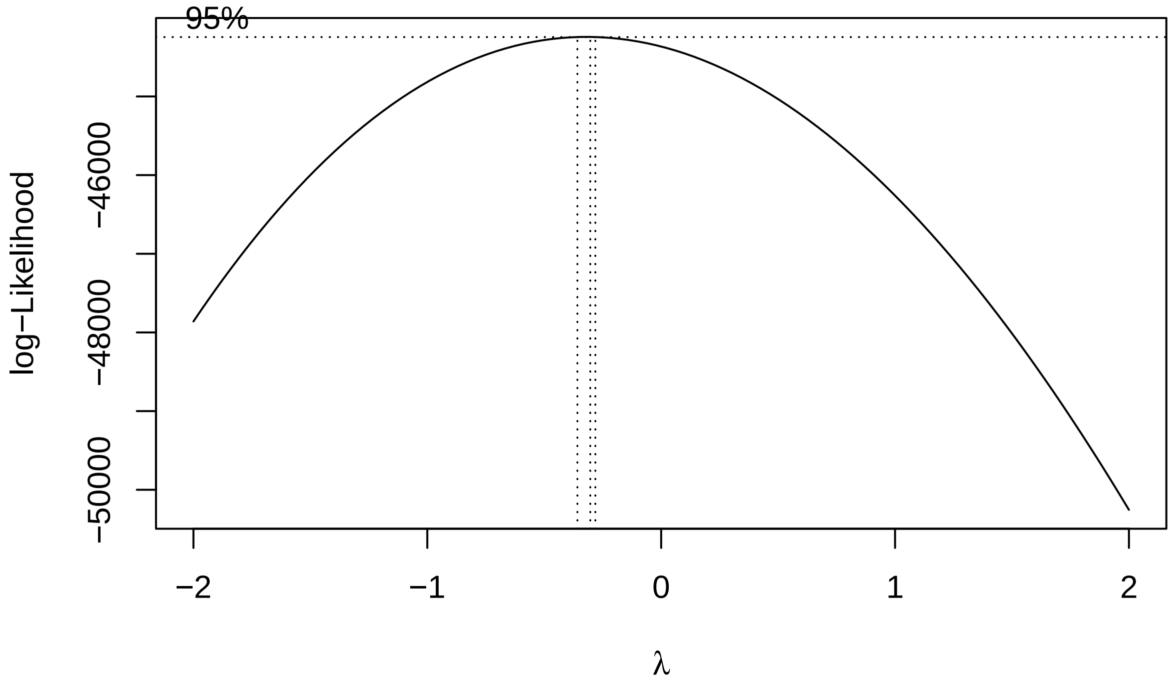
```
## [1] 0
```

```
df_RTs_PWA <- df_RTs_PWA %>% filter(VOT >=200)
```

LMMs: Transformed RTs In our pre-registration, we planned to conduct a GLMM with a Gamma distribution to account for the non-normality of the data. However, the standard errors seem suspiciously small and additional analyses showed that the GLMM doesn't converge with other types of analyses (see Appendix). Therefore, we decided to conduct an LMM with transformed RTs instead.

Analysis with factors Ordinal position x Session x Group *Box-cox test*
(common transformations: -2 -> 1/(Y^2), -1 -> 1/y, -0.5 -> 1/(sqrt(y))), 0 -> log(y), 0.5 -> sqrt(y), *1 -> y*, 2 -> y^2, 3 -> y^3)

```
MASS::boxcox(df_RTs$VOT ~ df_RTs$PosOr*df_RTs$session*df_RTs$group)
```



```
## Box-Cox suggests log transformation
df_RTs$lVOT <- log(df_RTs$VOT)
```

Compute full model, then compute step-wise reduction

```
library(lmerTest)
# m2_lmm <- lmer(lVOT ~ PosOr.cont*session*group +
#                 (PosOr.cont*session|subject) +
#                (PosOr.cont*session*group|category),
#               data = df_RTs,
#              control=lmerControl(optimizer = "bobyqa"))
# didLmerConverge(m2_lmm)
```

Model fails to converge –> Reduce

```
# 1) Increase optimizer iterations
# m2_lmm <- lmer(lVOT ~ PosOr.cont*session*group +
#                 (PosOr.cont*session|subject) +
#                (PosOr.cont*session*group|category),
#               data = df_RTs,
#              control=lmerControl(optimizer = "bobyqa",
#                                    optCtrl = list(maxfun = 2e5)))
```

```r
# didLmerConverge(m2_lmm)

# 2) Omit correlation parameters as model still fails to converge
# m2_lmm <- afex::lmer_alt(lVOT ~ PosOr.cont*session*group +
#                 (PosOr.cont*session||subject) +
#                (PosOr.cont*session*group||category),
#               data = df_RTs,
#              control=lmerControl(optimizer = "bobyqa",
#                                  optCtrl = list(maxfun = 2e5)))

# 3) Model fit is still singular -> Further reduce the model
# m2_lmm <- afex::lmer_alt(lVOT ~ PosOr.cont*session*group +
#                 (PosOr.cont+session||subject) +
#                (PosOr.cont+group||category),
#               data = df_RTs,
#              control=lmerControl(optimizer = "bobyqa",
#                                  optCtrl = list(maxfun = 2e5)))
# 4) Does the model also converge when correlation parameters are included - yes!
m2_lmm <- lmer(lVOT ~ PosOr.cont*session*group +
               (PosOr.cont+session|subject) +
              (PosOr.cont+group|category),
             data = df_RTs,
            control=lmerControl(optimizer = "bobyqa",
                                optCtrl = list(maxfun = 2e5)))
# rePCA(m2_lmm)
didLmerConverge(m2_lmm)
```

```
##  The relative maximum gradient of 0.00000257 is less than our 0.001 criterion.
##  You can safely ignore any warnings about a claimed convergence failure.
```

```r
## Warnings can be ignored

summary(m2_lmm)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: lVOT ~ PosOr.cont * session * group + (PosOr.cont + session |
##     subject) + (PosOr.cont + group | category)
##    Data: df_RTs
## Control: lmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 200000))
##
## REML criterion at convergence: 1158.2
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.8980 -0.6647 -0.1747  0.4804  5.6443
##
## Random effects:
##  Groups   Name        Variance   Std.Dev. Corr
##  subject  (Intercept) 0.03572310 0.189006
##           PosOr.cont  0.00010406 0.010201  0.18
##           session2    0.00339920 0.058303 -0.13  0.24
##           session3    0.00406330 0.063744 -0.49  0.26  0.64
##  category (Intercept) 0.00884981 0.094073
##           PosOr.cont  0.00002358 0.004856 0.07
##           group2-1    0.00090450 0.030075 0.08  0.16
##  Residual             0.06154884 0.248090
## Number of obs: 12455, groups:  subject, 40; category, 24
##
## Fixed effects:
##                                   Estimate   Std. Error           df t value
## (Intercept)                       7.116562     0.035604    58.495400 199.883
## PosOr.cont                        0.016922     0.002489    30.170032   6.800
## session2                         -0.076082     0.010856    38.392327  -7.008
## session3                         -0.089211     0.011561    36.616785  -7.717
## group2-1                          0.157635     0.060276    38.561991   2.615
## PosOr.cont:session2               0.004635     0.003924 12260.584027   1.181
## PosOr.cont:session3               0.003648     0.003895 12262.086513   0.937
## PosOr.cont:group2-1               0.006992     0.004565    37.104902   1.532
## session2:group2-1                -0.025605     0.021711    38.391896  -1.179
## session3:group2-1                -0.020079     0.023122    36.616755  -0.868
## PosOr.cont:session2:group2-1      0.013305     0.007848 12258.935598   1.695
## PosOr.cont:session3:group2-1      0.003095     0.007790 12261.681674   0.397
##                                           Pr(>|t|)
## (Intercept)                  < 0.0000000000000002 ***
## PosOr.cont                          0.00000014849 ***
## session2                            0.00000002268 ***
## session3                            0.00000000345 ***
## group2-1                                   0.0127 *
## PosOr.cont:session2                        0.2376
## PosOr.cont:session3                        0.3489
## PosOr.cont:group2-1                        0.1341
## session2:group2-1                          0.2455
## session3:group2-1                          0.3908
## PosOr.cont:session2:group2-1               0.0901 .
## PosOr.cont:session3:group2-1               0.6912
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Correlation of Fixed Effects:
##             (Intr) PsOr.c sessn2 sessn3 grp2-1 PsO.:2 PsO.:3 PO.:2- s2:2-1
## PosOr.cont   0.114
## session2    -0.096  0.134
## session3    -0.361  0.151  0.612
## group2-1     0.006  0.008 -0.001 -0.002
## PsOr.cnt:s2  0.000 -0.011  0.005  0.003  0.000
## PsOr.cnt:s3  0.000 -0.020  0.003  0.003  0.000  0.515
## PsOr.cn:2-1  0.001  0.069  0.001  0.002  0.126 -0.011 -0.016
## sssn2:gr2-1 -0.001  0.001  0.051  0.027 -0.114  0.007  0.005  0.146
## sssn3:gr2-1 -0.001  0.002  0.027  0.040 -0.426  0.005  0.007  0.164  0.612
## PsOr.:2:2-1  0.000 -0.010  0.007  0.005  0.000  0.129  0.075 -0.012  0.005
## PsOr.:3:2-1  0.000 -0.014  0.005  0.007  0.000  0.075  0.126 -0.022  0.003
##             s3:2-1 PO.:2:
## PosOr.cont
## session2
## session3
## group2-1
## PsOr.cnt:s2
## PsOr.cnt:s3
## PsOr.cn:2-1
## sssn2:gr2-1
## sssn3:gr2-1
## PsOr.:2:2-1  0.003
## PsOr.:3:2-1  0.003  0.515
```

```
anova(m2_lmm)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##                            Sum Sq Mean Sq NumDF   DenDF F value        Pr(>F)
## PosOr.cont                 2.8457 2.84571     1    30.2 46.2349 0.00000014849 ***
## session                    4.1782 2.08908     2    37.5 33.9418 0.00000000389 ***
## group                      0.4210 0.42096     1    38.6  6.8394       0.01266 *
## PosOr.cont:session         0.0949 0.04746     2 12260.9  0.7711       0.46251
## PosOr.cont:group           0.1444 0.14438     1    37.1  2.3457       0.13411
## session:group              0.0877 0.04386     2    37.5  0.7125       0.49693
## PosOr.cont:session:group 0.1958 0.09790     2 12259.7  1.5906       0.20385
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
saveRDS(m2_lmm,  file = here::here("results", "tables", "CSI_online_aphasia_SessionxGroup_control_lmm_V
tab_model(m2_lmm,transform = NULL,
          show.re.var = F,#T,
          show.stat = T,show.r2 = T,show.icc = T,
          title = "LMM of VOTs Predicted by Ordinal Position and Session",
          dv.labels = "Vocal Onset Time (log-transformed)",
          #string.pred = "",
          df.method = "satterthwaite",
          string.stat = "t-Value",
          file = here::here(
            "results", "tables",
            "CSI_online_aphasia_spoken_SessionxGroup_lmm_VOT.html"))
```

LMM of VOTs Predicted by Ordinal Position and Session

Vocal Onset Time (log-transformed)

Predictors

Estimates

CI

t-Value

p

(Intercept)

7.12

7.05 – 7.19

199.88

<0.001

PosOr cont

0.02

0.01 – 0.02

6.80

<0.001

session [2]

-0.08

-0.10 – -0.05

-7.01

<0.001

session [3]

-0.09

-0.11 – -0.07

-7.72

<0.001

group2-1

0.16

0.04 – 0.28

2.62

0.013

PosOr cont × session [2]

0.00

-0.00 – 0.01

1.18

0.238

PosOr cont × session [3]

0.00

-0.00 – 0.01

0.94

0.349

PosOr.cont:group2-1

0.01

-0.00 – 0.02

1.53

0.134

session2:group2-1

-0.03

-0.07 – 0.02

-1.18

0.246

session3:group2-1

-0.02

-0.07 – 0.03

-0.87

0.391

PosOr.cont:session2:group2-1

0.01

-0.00 – 0.03

1.70

0.090

PosOr.cont:session3:group2-1

0.00

-0.01 – 0.02

0.40

0.691

ICC

0.43

N subject

40

N category

24

Observations

12455

Marginal R2 / Conditional R2

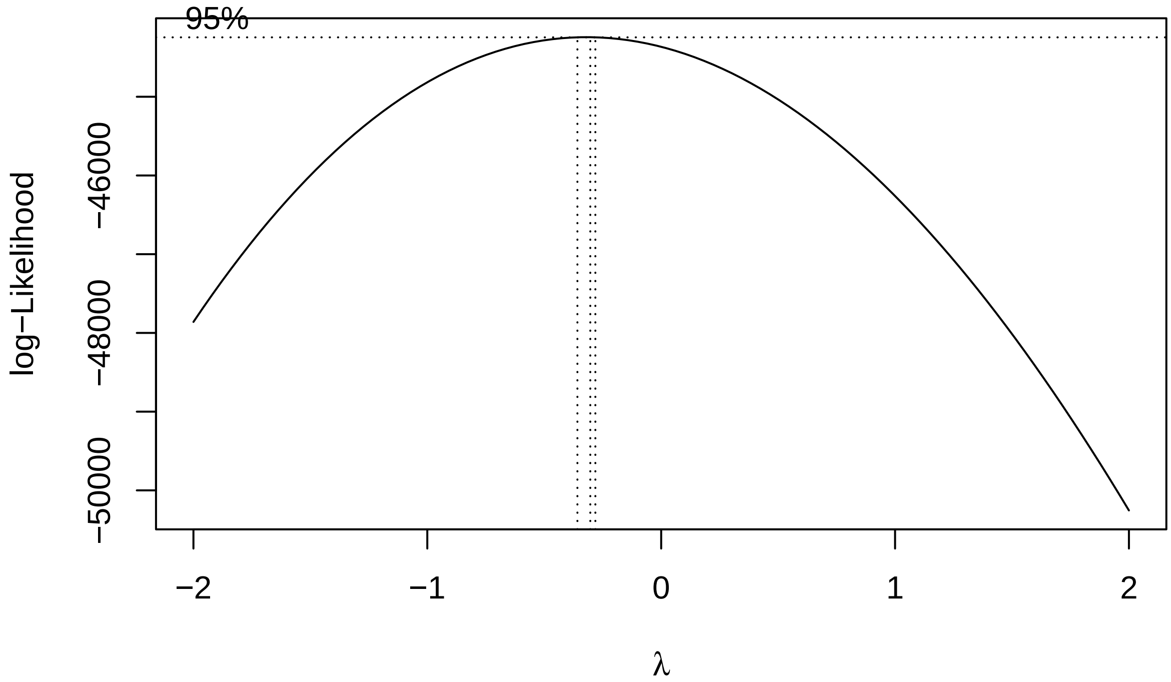
0.070 / 0.468

```
## Check model
#performance::check_model(m2_lmm)
# ggResidpanel::resid_panel(m2_lmm, smoother = TRUE,
#                              qqbands = TRUE, type = "pearson")
performance::model_performance(m2_lmm)
```

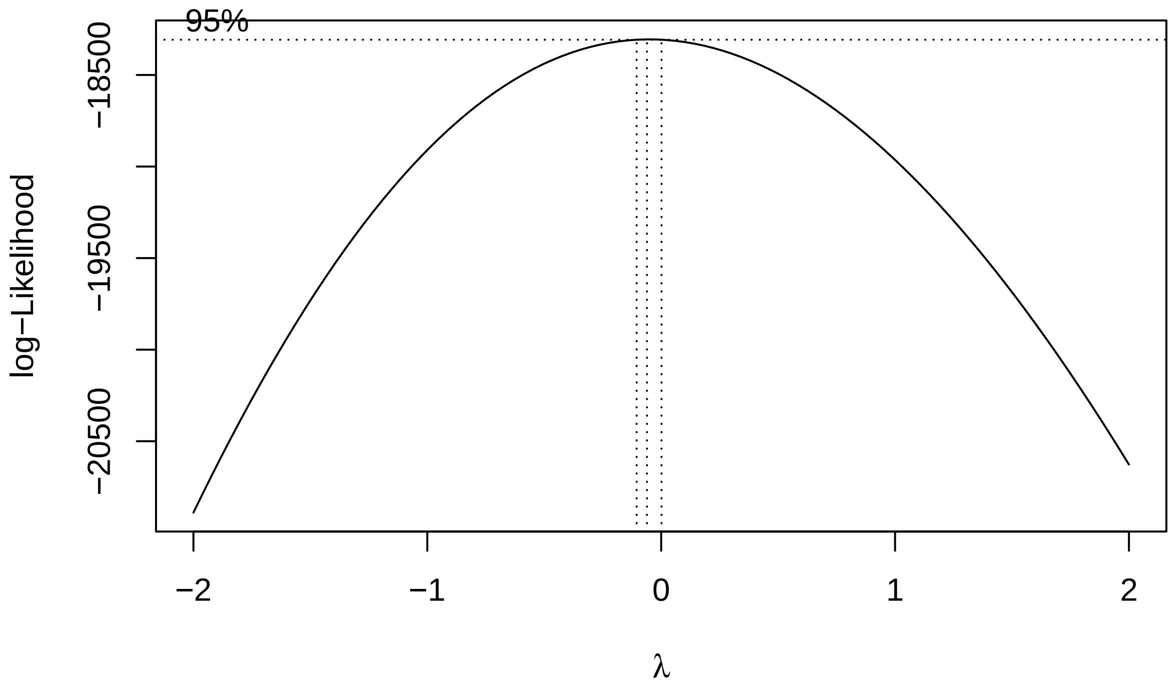
```
## # Indices of model performance
##
## AIC      |     AICc |      BIC | R2 (cond.) | R2 (marg.) |   ICC |  RMSE | Sigma
## -----------------------------------------------------------------------------------
## 1216.212 | 1216.352 | 1431.678 |      0.468 |      0.070 | 0.428 | 0.246 | 0.248
```

PWA only

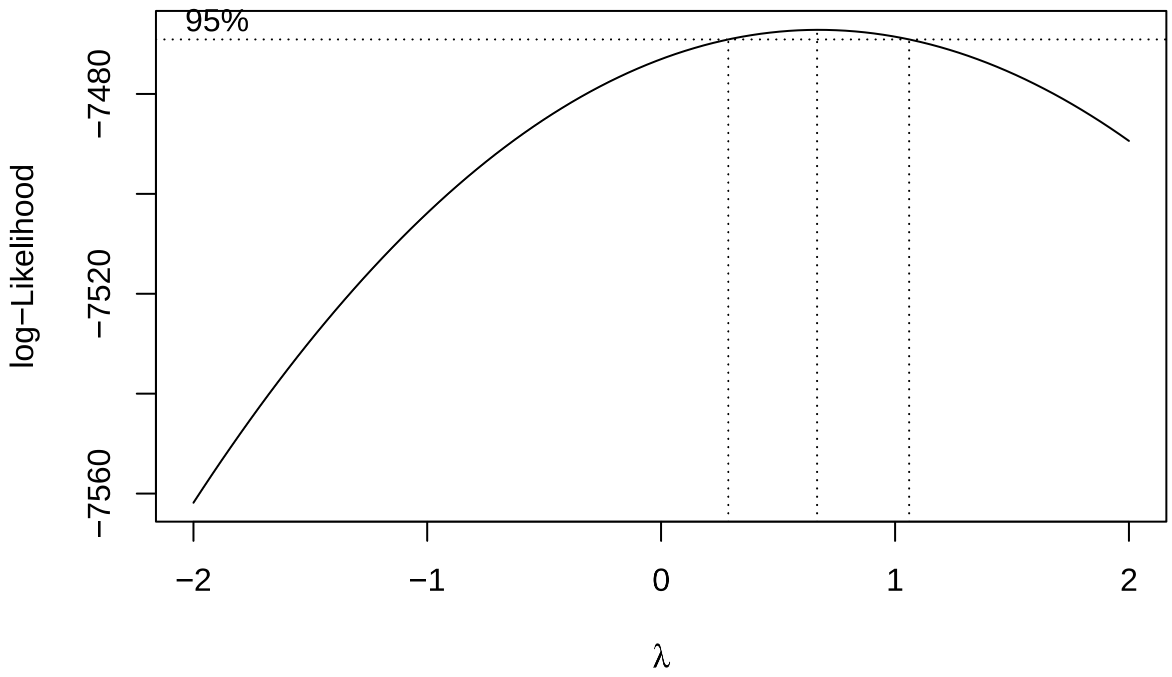
```
MASS::boxcox(df_RTs$VOT ~ df_RTs$group*df_RTs$PosOr*df_RTs$session)
```



```
MASS::boxcox(df_RTs_PWA$VOT ~ df_RTs_PWA$PosOr*df_RTs_PWA$session)
```



```
## Box-Cox suggests log transformation --> compute with log-transformed RTs as s control analysis
MASS::boxcox(log(df_RTs_PWA$VOT)~  df_RTs_PWA$PosOr*df_RTs_PWA$session)
```



```
df_RTs_PWA$VOTlog <- log(df_RTs_PWA$VOT)
```

Compute full model, then compute step-wise reduction until model convergence

```
library(lmerTest)
# m1_lmm_PWA <- lmer(VOTlog ~ PosOr.cont*session +
#                (PosOr.cont*session|subject) +
#               (PosOr.cont*session|category),
#             data = df_RTs_PWA,
#           control=lmerControl(optimizer = "bobyqa"))
# didLmerConverge(m1_lmm_PWA)
```

Model fails to converge –> Reduce

```
# 1) Increase optimizer iterations
# m1_lmm_PWA <- lmer(VOTlog ~ PosOr.cont*session +
#                (PosOr.cont*session|subject) +
#               (PosOr.cont*session|category),
#             data = df_RTs_PWA,
#           control=lmerControl(optimizer = "bobyqa",
#                                     optCtrl = list(maxfun = 2e5)))
# didLmerConverge(m1_lmm_PWA)

# 2) Omit correlation parameters as model still fails to converge
# m1_lmm_PWA <- afex::lmer_alt(VOTlog ~ PosOr.cont*session +
#                (PosOr.cont*session||subject) +
#               (PosOr.cont*session||category),
#              data = df_RTs_PWA,
#             control=lmerControl(optimizer = "bobyqa",
#                                   optCtrl = list(maxfun = 2e5)))
# rePCA(m1_lmm_PWA)

# 3) The model still has a singular fit -> reduce
# m1_lmm_PWA <- afex::lmer_alt(VOTlog ~ PosOr.cont*session +
#                (PosOr.cont*session||subject) +
#               (PosOr.cont*session-PosOr.cont||category),
#              data = df_RTs_PWA,
#             control=lmerControl(optimizer = "bobyqa",
#                                   optCtrl = list(maxfun = 2e5)))
# m1_lmm_PWA <- afex::lmer_alt(VOTlog ~ PosOr.cont*session +
#                (PosOr.cont*session||subject) +
#               (PosOr.cont*session-PosOr.cont-session||category),
#              data = df_RTs_PWA,
#             control=lmerControl(optimizer = "bobyqa",
#                                   optCtrl = list(maxfun = 2e5)))
# m1_lmm_PWA <- afex::lmer_alt(VOTlog ~ PosOr.cont*session +
#                (PosOr.cont*session||subject) +
#               (1|category),
#              data = df_RTs_PWA,
#             control=lmerControl(optimizer = "bobyqa",
#                                   optCtrl = list(maxfun = 2e5)))
# m1_lmm_PWA <- afex::lmer_alt(VOTlog ~ PosOr.cont*session +
#                (PosOr.cont+session||subject) +
#               (1|category),
#              data = df_RTs_PWA,
#             control=lmerControl(optimizer = "bobyqa",
#                                   optCtrl = list(maxfun = 2e5)))

# 4) Test whether the model also converges including correlation parameters -> yes
m1_lmm_PWA <- lmer(VOTlog ~ PosOr.cont*session +
               (PosOr.cont+session|subject) +
              (1|category),
             data = df_RTs_PWA,
            control=lmerControl(optimizer = "bobyqa",
                                 optCtrl = list(maxfun = 2e5)))
didLmerConverge(m1_lmm_PWA)
```

```
##  The relative maximum gradient of 0.00000429 is less than our 0.001 criterion.
##  You can safely ignore any warnings about a claimed convergence failure.
```

```
## Warnings can be ignored

summary(m1_lmm_PWA)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: VOTlog ~ PosOr.cont * session + (PosOr.cont + session | subject) +
##     (1 | category)
##    Data: df_RTs_PWA
## Control: lmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 200000))
##
## REML criterion at convergence: 1705.5
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.4857 -0.6809 -0.1843  0.5084  5.0807
##
## Random effects:
##  Groups   Name        Variance  Std.Dev. Corr
##  category (Intercept) 0.0092445 0.09615
##  subject  (Intercept) 0.0616484 0.24829
##           PosOr.cont  0.0001271 0.01128   0.35
##           session2    0.0009406 0.03067  -0.26  0.07
##           session3    0.0035352 0.05946  -0.68 -0.14  0.21
##  Residual             0.0760739 0.27581
## Number of obs: 5518, groups:  category, 24; subject, 20
##
## Fixed effects:
##                        Estimate  Std. Error          df t value
## (Intercept)            7.195573    0.059036   23.613808 121.884
## PosOr.cont             0.020682    0.003710   17.682005   5.575
## session2              -0.088051    0.011650   16.035853  -7.558
## session3              -0.099095    0.016309   17.569075  -6.076
## PosOr.cont:session2    0.011439    0.006552 5438.077615   1.746
## PosOr.cont:session3    0.005320    0.006495 5440.067017   0.819
##                                    Pr(>|t|)
## (Intercept)         < 0.0000000000000002 ***
## PosOr.cont                    0.00002905 ***
## session2                      0.00000113 ***
## session3                      0.00001071 ***
## PosOr.cont:session2               0.0809 .
## PosOr.cont:session3               0.4128
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) PsOr.c sessn2 sessn3 PsO.:2
## PosOr.cont   0.224
## session2    -0.145  0.031
## session3    -0.523 -0.079  0.341
## PsOr.cnt:s2  0.000 -0.021  0.004  0.003
## PsOr.cnt:s3 -0.001 -0.035  0.003  0.003  0.522
```

```
anova(m1_lmm_PWA)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##                    Sum Sq Mean Sq NumDF  DenDF F value       Pr(>F)
## PosOr.cont         2.3644 2.36438     1   17.7 31.0800 0.0000290471 ***
## session            5.4010 2.70050     2   17.5 35.4984 0.0000006973 ***
## PosOr.cont:session 0.2328 0.11641     2 5437.7  1.5303       0.2166
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
saveRDS(m1_lmm_PWA,  file = here::here(
  "results", "tables", "CSI_online_aphasia_PWA_control_lmm_VOT.RDS"))
tab_model(m1_lmm_PWA,transform = NULL,
          show.re.var = F, show.stat = T,show.r2 = F,show.icc = F,
          title = "LMM of VOTs Predicted by Ordinal Position and Session",
          df.method = "satterthwaite",
          dv.labels = "Vocal Onset Time (log-transformed)",
          #string.pred = "",
          string.stat = "t-Value",
          file = here::here(
            "results", "tables",
            "CSI_online_aphasia_PWA_control_lmm_VOT.html"))
```

LMM of VOTs Predicted by Ordinal Position and Session

Vocal Onset Time (log-transformed)

Predictors

Estimates

CI

t-Value

p

(Intercept)

7.20

7.07 – 7.32

121.88

<0.001

PosOr cont

0.02

0.01 – 0.03

5.57

<0.001

session [2]

-0.09

-0.11 – -0.06

-7.56

<0.001

session [3]

-0.10

-0.13 – -0.06

-6.08

<0.001

PosOr cont × session [2]

0.01

-0.00 – 0.02

1.75

0.081

PosOr cont × session [3]

0.01

-0.01 – 0.02

0.82

0.413

N subject

20

N category

24

Observations

5518

```
## check model fit
#performance::check_model(m1_lmm_PWA)
# ggResidpanel::resid_panel(m1_lmm_PWA, smoother = TRUE,
#                            qqbands = TRUE, type = "pearson")
performance::model_performance(m1_lmm_PWA)
```

```
## # Indices of model performance
##
## AIC      |     AICc |      BIC | R2 (cond.) | R2 (marg.) |   ICC |  RMSE | Sigma
## ---------------------------------------------------------------------------------
## 1741.484 | 1741.609 | 1860.568 |      0.495 |      0.019 | 0.485 | 0.274 | 0.276
```

ERROR RATES

Descriptives

```
df_errors %>% group_by(group) %>% count(error_class) %>%
  mutate(percentage=n/(nrow(df[df$category!="Filler" &
                               df$group=="PWA",])))
```

Error types

```
## # A tibble: 4 x 4
## # Groups:   group [2]
##   group   error_class     n percentage
##   <fct>         <dbl> <int>      <dbl>
## 1 control           0  6952     0.966
## 2 control           1   167     0.0232
## 3 PWA               0  5637     0.783
## 4 PWA               1  1541     0.214
```

```
df_errors %>% group_by(group, session) %>% count(error_class)  %>%
  mutate(percentage=n/(nrow(df[df$category!="Filler" & df$group=="PWA" &
                                 df$session=="1",])))
```

```
## # A tibble: 12 x 5
## # Groups:   group, session [6]
##    group   session error_class     n percentage
##    <fct>   <fct>         <dbl> <int>      <dbl>
##  1 control 1                 0  2284     0.952
##  2 control 1                 1    77     0.0321
##  3 control 2                 0  2316     0.965
##  4 control 2                 1    57     0.0238
##  5 control 3                 0  2352     0.98
##  6 control 3                 1    33     0.0138
##  7 PWA     1                 0  1777     0.740
##  8 PWA     1                 1   609     0.254
##  9 PWA     2                 0  1892     0.788
## 10 PWA     2                 1   506     0.211
## 11 PWA     3                 0  1968     0.82
## 12 PWA     3                 1   426     0.178
```

```
df_errors %>% group_by(group) %>% count(error) %>%
  mutate(percentage=n/(nrow(df[df$category!="Filler" &
                                 df$group=="PWA",])))
```

```
## # A tibble: 20 x 4
## # Groups:   group [2]
##    group   error     n percentage
##    <fct>   <chr> <int>      <dbl>
##  1 control 1         1   0.000139
##  2 control 2        40   0.00556
##  3 control 3        32   0.00444
##  4 control 4        31   0.00431
##  5 control 5         1   0.000139
##  6 control 6         6   0.000833
##  7 control 7        35   0.00486
##  8 control 8         3   0.000417
##  9 control 9        18   0.0025
## 10 control <NA>   6952   0.966
## 11 PWA     1        56   0.00778
## 12 PWA     2       148   0.0206
## 13 PWA     3        62   0.00861
## 14 PWA     4       820   0.114
## 15 PWA     5        32   0.00444
## 16 PWA     6        40   0.00556
## 17 PWA     7       102   0.0142
## 18 PWA     8        29   0.00403
## 19 PWA     9       252   0.035
## 20 PWA     <NA>   5637   0.783
```

```
df_errors %>% group_by(group, session) %>% count(error) %>%
  mutate(percentage=n/(nrow(df[df$category!="Filler" & df$group=="PWA" &
                                 df$session=="1",])))
```

```
## # A tibble: 54 x 5
## # Groups:   group, session [6]
##    group   session error     n percentage
##    <fct>   <fct>   <chr> <int>      <dbl>
##  1 control 1       2        15   0.00625
##  2 control 1       3        13   0.00542
##  3 control 1       4        14   0.00583
##  4 control 1       6         2   0.000833
##  5 control 1       7        23   0.00958
##  6 control 1       9        10   0.00417
##  7 control 1       <NA>   2284   0.952
##  8 control 2       1         1   0.000417
##  9 control 2       2        17   0.00708
## 10 control 2       3        11   0.00458
## # i 44 more rows
```

```
table(df_errors$error_class, df_errors$error) # technical errors are not counted as errors
```

```
##
##       1   2   3   4   5   6   7   8   9
##   0   0   0   0   0   0   0   0   0   0
##   1  57 188  94 851  33  46 137  32 270
```

```
table(df_errors$error_class[is.na(df_errors$error)]) # correct responses
```

```
##
##     0
## 12589
```

```
error_overview <- data.frame(subject=factor(rep(unique(df$subject),
                                         each=5*3)),
                             group=factor(rep(c("PWA", "control"),
                                       each=20*5*3)),
                             session=factor(rep(c("1","2","3"),
                                                each=5,
                                         times=
                                           length(unique(df$subject)))),
                             PosOr=factor(rep(c("1","2","3","4","5"),
                                         times=length(unique(df$subject))*3)),
                                error_class=0)
x <- df_errors %>% group_by(subject, session, PosOr) %>%
  count(error_class) %>%
  filter(error_class==1)
for(i in 1:nrow(x)){
  error_overview$error_class[error_overview$subject==x$subject[i] &
                             error_overview$session==x$session[i] &
                             error_overview$PosOr==x$PosOr[i] ] <-
    x$n[i]
}
error_overview$percentage <- (error_overview$error_class/24)*100

(means_final_errors <- error_overview %>%
    group_by(group,session,PosOr) %>%
    summarise(count=sum(error_class), mean=mean(error_class),
              sd=sd(error_class), se=sd(error_class)/20,
              mean_p = mean(percentage),
              sd_p=sd(percentage), se_p=sd(percentage)/20))
```

Amount of errors

```
## `summarise()` has grouped output by 'group', 'session'. You can override using
## the `.groups` argument.

## # A tibble: 30 x 10
## # Groups:   group, session [6]
##    group   session PosOr count  mean    sd     se mean_p  sd_p  se_p
##    <fct>   <fct>   <fct> <dbl> <dbl> <dbl>  <dbl>  <dbl> <dbl> <dbl>
##  1 control 1       1        13  0.65 0.745 0.0373   2.71  3.10 0.155
##  2 control 1       2        14  0.7  0.733 0.0366   2.92  3.05 0.153
##  3 control 1       3        19  0.95 0.759 0.0380   3.96  3.16 0.158
##  4 control 1       4        14  0.7  0.801 0.0401   2.92  3.34 0.167
##  5 control 1       5        17  0.85 0.745 0.0373   3.54  3.10 0.155
##  6 control 2       1         9  0.45 0.686 0.0343   1.87  2.86 0.143
##  7 control 2       2        15  0.75 1.02  0.0510   3.12  4.25 0.212
##  8 control 2       3         8  0.4  0.503 0.0251   1.67  2.09 0.105
##  9 control 2       4         9  0.45 0.759 0.0380   1.87  3.16 0.158
## 10 control 2       5        16  0.8  0.834 0.0417   3.33  3.47 0.174
## # i 20 more rows
```

```
# Export as word file
library(flextable)
huxt_word <- huxtable::huxtable(means_final_errors)
huxt_word <- huxtable::set_number_format(huxt_word, round(2))
huxtable::quick_docx(huxt_word,
                     file =
                       here::here(
                         "results", "tables",
                          "CSI_online_PWA_errors_by_session.docx"),
                     open = FALSE)
```

Calculate increase mean by ordinal position, separately for each session (not controlled for random variances, weighted only per session):

```r
means_final_errors$increase_count <- NA
means_final_errors$increase_mean <- NA
for(k in 1:length(unique(means_final_errors$group))){
for(i in 1:length(unique(means_final_errors$session))){
  for(j in 2:length(unique(means_final_errors$PosOr))) {
    means_final_errors$increase_count[
      means_final_errors$session==unique(means_final_errors$session)[i] &
        means_final_errors$PosOr==unique(means_final_errors$PosOr)[j]&
                             means_final_errors$group==unique(means_final_errors$group)[k]] <-
     means_final_errors$count[means_final_errors$session==
                                unique(means_final_errors$session)[i] &
                                means_final_errors$PosOr==
                                unique(means_final_errors$PosOr)[j]&
                             means_final_errors$group==unique(means_final_errors$group)[k]] -
      means_final_errors$count[
        means_final_errors$session==
          unique(means_final_errors$session)[i] &
        means_final_errors$PosOr==
          unique(means_final_errors$PosOr)[j-1]&
                             means_final_errors$group==unique(means_final_errors$group)[k]]
    means_final_errors$increase_mean[
      means_final_errors$session==unique(means_final_errors$session)[i] &
        means_final_errors$PosOr==
        unique(means_final_errors$PosOr)[j]&
                             means_final_errors$group==unique(means_final_errors$group)[k]] <-
     means_final_errors$mean[
       means_final_errors$session==unique(means_final_errors$session)[i] &
         means_final_errors$PosOr==unique(means_final_errors$PosOr)[j]&
         means_final_errors$group==unique(means_final_errors$group)[k]] -
      means_final_errors$mean[means_final_errors$session==
                                unique(means_final_errors$session)[i] &
                           means_final_errors$PosOr==
                             unique(means_final_errors$PosOr)[j-1]&
                            means_final_errors$group==
                             unique(means_final_errors$group)[k]]
  }}}
#means_final_errors

## Calculate overall mean increase per session (weighted: all PosOrs had the same amount of trials)
mean(means_final_errors$increase_mean[
  means_final_errors$session==1], na.rm=T)
```

```
## [1] 0.28125
```

```r
means_final_errors$PosOr_effect <- NA
means_final_errors$PosOr_effect[means_final_errors$PosOr==1] <- 1
for(k in 1:length(unique(means_final_errors$group))){
for(i in 1:length(unique(means_final_errors$session))){
  means_final_errors$PosOr_effect[
    means_final_errors$session==unique(means_final_errors$session)[i] &
        means_final_errors$group==unique(means_final_errors$group)[k] &
        means_final_errors$PosOr=="1"] <-
    (means_final_errors$increase_mean[
      means_final_errors$session==unique(means_final_errors$session)[i] &
        means_final_errors$group==unique(means_final_errors$group)[k] &
        means_final_errors$PosOr=="2"]+
       means_final_errors$increase_mean[
         means_final_errors$session==
           unique(means_final_errors$session)[i] &
        means_final_errors$group==unique(means_final_errors$group)[k] &
        means_final_errors$PosOr=="3"]+
       means_final_errors$increase_mean[
         means_final_errors$session==
           unique(means_final_errors$session)[i] &
        means_final_errors$group==unique(means_final_errors$group)[k] &
        means_final_errors$PosOr=="4"]+
       means_final_errors$increase_mean[
         means_final_errors$session==
           unique(means_final_errors$session)[i] &
        means_final_errors$group==unique(means_final_errors$group)[k] &
        means_final_errors$PosOr=="5"])/4
}}
means_final_errors
```

```
## # A tibble: 30 x 13
## # Groups:   group, session [6]
##    group   session PosOr count  mean    sd     se mean_p  sd_p  se_p
##    <fct>   <fct>   <fct> <dbl> <dbl> <dbl>  <dbl>  <dbl> <dbl> <dbl>
##  1 control 1       1        13  0.65 0.745 0.0373   2.71  3.10 0.155
##  2 control 1       2        14  0.7  0.733 0.0366   2.92  3.05 0.153
##  3 control 1       3        19  0.95 0.759 0.0380   3.96  3.16 0.158
##  4 control 1       4        14  0.7  0.801 0.0401   2.92  3.34 0.167
##  5 control 1       5        17  0.85 0.745 0.0373   3.54  3.10 0.155
##  6 control 2       1         9  0.45 0.686 0.0343   1.87  2.86 0.143
##  7 control 2       2        15  0.75 1.02  0.0510   3.12  4.25 0.212
##  8 control 2       3         8  0.4  0.503 0.0251   1.67  2.09 0.105
##  9 control 2       4         9  0.45 0.759 0.0380   1.87  3.16 0.158
## 10 control 2       5        16  0.8  0.834 0.0417   3.33  3.47 0.174
## # i 20 more rows
## # i 3 more variables: increase_count <dbl>, increase_mean <dbl>,
## #   PosOr_effect <dbl>
```

Different kinds of errors per ordinal position

```
table(df_errors$error)
```

```
##
##   1   2   3   4   5   6   7   8   9
##  57 188  94 851  33  46 137  32 270
```

```
(df_errors %>% group_by(group, session, error, PosOr) %>%
    count() -> z)
```

```
## # A tibble: 232 x 5
## # Groups:   group, session, error, PosOr [232]
##    group   session error PosOr     n
##    <fct>   <fct>   <chr> <fct> <int>
##  1 control 1       2     1         1
##  2 control 1       2     2         4
##  3 control 1       2     3         6
##  4 control 1       2     4         2
##  5 control 1       2     5         2
##  6 control 1       3     1         1
##  7 control 1       3     2         2
##  8 control 1       3     3         3
##  9 control 1       3     4         5
## 10 control 1       3     5         2
## # i 222 more rows
```

```r
w <- data.frame(group=rep(c("control", "PWA"), each=3*10*5),
                 session=rep(c("1","2","3"), each=10, times=2*5),
                 error=rep(seq(1:10), each=5, times=2*3),
                 PosOr=rep(seq(1:5), timed=10*3*2),
                n=NA)
for(i in 1:nrow(w)){
  l <- z %>% filter(group==w$group[i]&session==w$session[i]&
                      error==w$error[i]&PosOr==w$PosOr[i])
  if(nrow(l)!=0) {
    w$n[i] <- l$n
  }
}
# mean increase per error type and ordinal position
 y <- data.frame(group=rep(c("control", "PWA"), each=3*10),
                 session=rep(c("1","2","3"), each=10, times=2),
                 error=rep(seq(1:10), times=2*3),
                 m_increase=NA)
 w$error[is.na(w$error)] <- 10
 for(i in 1:nrow(y)){
   a1 <- w$n[w$group==y$group[i]& w$error==y$error[i] &
                       w$session==y$session[i] & w$PosOr==1]
   a2 <- w$n[w$group==y$group[i]& w$error==y$error[i] &
                       w$session==y$session[i] & w$PosOr==2]
   a3 <- w$n[w$group==y$group[i]& w$error==y$error[i] &
                       w$session==y$session[i] & w$PosOr==3]
   a4 <- w$n[w$group==y$group[i]& w$error==y$error[i] &
                       w$session==y$session[i] & w$PosOr==4]
   a5 <- w$n[w$group==y$group[i]& w$error==y$error[i] &
                       w$session==y$session[i] & w$PosOr==5]
   if(!is.na(a2)&!is.na(a1)){ y$m_increase[i] <- a2-a1 }
   if(!is.na(a3)&!is.na(a2)){ y$m_increase[i] <-
     y$m_increase[i]+a3-a2 }
   if(!is.na(a4)&!is.na(a3)){ y$m_increase[i] <-
     y$m_increase[i]+a4-a3}
   if(!is.na(a5)&!is.na(a4)){ y$m_increase[i] <-
     y$m_increase[i]+a5-a4 }
   n <- sum(!is.na(c(a1,a2,a3,a4,a5)))-1
   y$m_increase[i] <- y$m_increase[i]/n
 }
 y
```

```
##      group session error m_increase
## 1  control       1     1         NA
## 2  control       1     2  0.2500000
## 3  control       1     3  0.2500000
## 4  control       1     4  0.2500000
## 5  control       1     5         NA
## 6  control       1     6         NA
## 7  control       1     7  0.5000000
## 8  control       1     8         NA
## 9  control       1     9  0.2500000
## 10 control       1    10         NA
## 11 control       2     1         NA
## 12 control       2     2         NA
## 13 control       2     3  1.0000000
## 14 control       2     4  0.0000000
## 15 control       2     5         NA
## 16 control       2     6         NA
## 17 control       2     7         NA
## 18 control       2     8  0.0000000
## 19 control       2     9         NA
## 20 control       2    10         NA
## 21 control       3     1         NA
## 22 control       3     2         NA
## 23 control       3     3 -0.2500000
## 24 control       3     4 -0.5000000
## 25 control       3     5         NA
## 26 control       3     6         NA
## 27 control       3     7         NA
## 28 control       3     8         NA
## 29 control       3     9         NA
## 30 control       3    10         NA
## 31     PWA       1     1         NA
## 32     PWA       1     2  2.5000000
## 33     PWA       1     3  0.2500000
## 34     PWA       1     4  8.0000000
## 35     PWA       1     5  0.0000000
## 36     PWA       1     6 -0.5000000
## 37     PWA       1     7 -0.5000000
## 38     PWA       1     8  0.2500000
## 39     PWA       1     9 -0.7500000
## 40     PWA       1    10         NA
## 41     PWA       2     1 -0.7500000
## 42     PWA       2     2  0.5000000
## 43     PWA       2     3 -0.2500000
## 44     PWA       2     4  6.0000000
## 45     PWA       2     5  0.5000000
## 46     PWA       2     6 -0.2500000
## 47     PWA       2     7 -0.5000000
## 48     PWA       2     8  0.3333333
## 49     PWA       2     9 -0.5000000
## 50     PWA       2    10         NA
## 51     PWA       3     1 -0.2500000
## 52     PWA       3     2  1.0000000
## 53     PWA       3     3  0.0000000
## 54     PWA       3     4  2.0000000
## 55     PWA       3     5 -0.7500000
## 56     PWA       3     6         NA
## 57     PWA       3     7  0.2500000
## 58     PWA       3     8         NA
## 59     PWA       3     9  2.5000000
## 60     PWA       3    10         NA
```

```
 # mean across sessions
(y %>% group_by(group, error) %>%
  summarise(n=sum(m_increase), isnotna=sum(!is.na(m_increase)),
            mean_increase=mean(m_increase, na.rm=T)) ->
    increase_errors)
```

```
## 'summarise()' has grouped output by 'group'. You can override using the
## '.groups' argument.
```

```
## # A tibble: 20 x 5
## # Groups:   group [2]
##    group   error     n isnotna mean_increase
##    <chr>   <int> <dbl>   <int>         <dbl>
##  1 PWA         1 NA          2       -0.5
##  2 PWA         2  4          3        1.33
##  3 PWA         3  0          3        0
##  4 PWA         4 16          3        5.33
##  5 PWA         5 -0.25       3       -0.0833
##  6 PWA         6 NA          2       -0.375
##  7 PWA         7 -0.75       3       -0.25
##  8 PWA         8 NA          2        0.292
##  9 PWA         9  1.25       3        0.417
## 10 PWA        10 NA          0      NaN
## 11 control     1 NA          0      NaN
## 12 control     2 NA          1        0.25
## 13 control     3  1          3        0.333
## 14 control     4 -0.25       3       -0.0833
## 15 control     5 NA          0      NaN
## 16 control     6 NA          0      NaN
## 17 control     7 NA          1        0.5
## 18 control     8 NA          1        0
## 19 control     9 NA          1        0.25
## 20 control    10 NA          0      NaN
```

```
 increase_errors %>% filter(group=="PWA") %>% count()
```

```
## # A tibble: 1 x 2
## # Groups:   group [1]
##   group     n
##   <chr> <int>
## 1 PWA      10
```

```
 y %>% filter(group=="PWA") %>%
   filter(!(error %in% c(4,9,10))) %>%
   group_by(session) %>%
   summarise(mean=mean(m_increase, na.rm=T))
```

```
## # A tibble: 3 x 2
##   session    mean
##   <chr>     <dbl>
## 1 1        0.333
## 2 2       -0.0595
## 3 3        0.05
```

Calculate mean and SD of ordinal position effects in PWA group

```
# First calculate the OrdPos effect for each subject and each category separately
df_errors_PWA <- df_errors %>% filter(category!="Filler" &
                                        group=="PWA")
df_errors_PWA$OrdPos_manual <- NA_real_
isEmptyNumeric <- function(x) {
    return(identical(x, numeric(0)))
}

for (i in unique(df_errors_PWA$subject)) {
  for(j in unique(df_errors_PWA$category)) {
    for(l in unique(df_errors_PWA$session)) {
      for (k in 2:5) {
        x <- df_errors_PWA$error_class[df_errors_PWA$subject==i &
                             df_errors_PWA$category==j &
                             df_errors_PWA$session==l &
                             df_errors_PWA$PosOr==k]
        xM1 <- df_errors_PWA$error_class[df_errors_PWA$subject==i &
                             df_errors_PWA$category==j &
                             df_errors_PWA$session==l &
                             df_errors_PWA$PosOr==k-1]
      if(!isEmptyNumeric(x) & !isEmptyNumeric(xM1)){
        if(!is.na(x) &  !is.na(xM1)){
              df_errors_PWA$OrdPos_manual[df_errors_PWA$subject==i &
                             df_errors_PWA$category==j &
                             df_errors_PWA$session==l &
                             df_errors_PWA$PosOr==k] <- x - xM1
      }
      }
      }
    }
  }
}

# return summarySEwithin
 df_errors_PWA %>%
   filter(category != "Filler") %>%
   Rmisc::summarySEwithin(.,"OrdPos_manual",
                          idvar = "subject",
                          withinvars = c("session"),
                          betweenvars = c("subject", "group"),
                          na.rm = T)
```

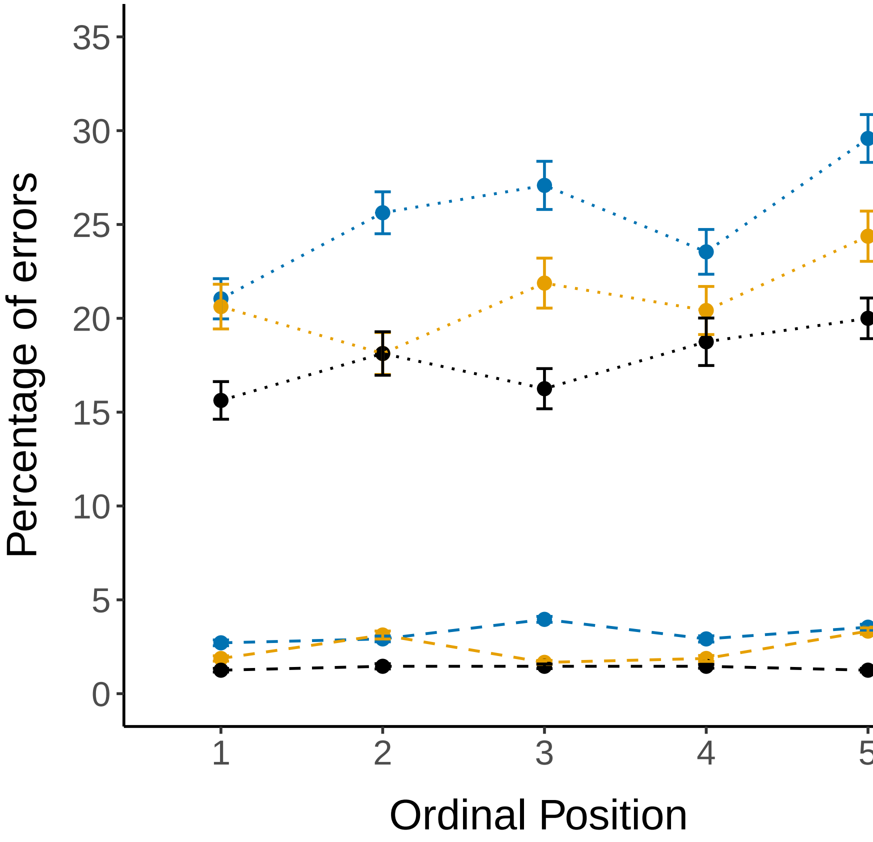
```
##    subject group session  N  OrdPos_manual        sd         se         ci
## 1      101   PWA       1 90  0.00615124635 0.3661726 0.03859798 0.07669337
## 2      101   PWA       2 95  0.01784715278 0.2184133 0.02240873 0.04449306
## 3      101   PWA       3 96  0.01795680190 0.1250000 0.01275776 0.02532735
## 4      102   PWA       1 96  0.01413859312 0.2172647 0.02217449 0.04402192
## 5      102   PWA       2 96  0.01413859312 0.2806829 0.02864708 0.05687164
## 6      102   PWA       3 96  0.01413859312 0.2172647 0.02217449 0.04402192
## 7      103   PWA       1 96  0.05580525979 0.8190776 0.08359676 0.16596055
## 8      103   PWA       2 96 -0.01711140688 0.7539370 0.07694838 0.15276184
## 9      103   PWA       3 96  0.00372192646 0.8518556 0.08694214 0.17260198
## 10     104   PWA       1 96  0.04538859312 0.7909854 0.08072961 0.16026854
## 11     104   PWA       2 96  0.01413859312 0.7633677 0.07791089 0.15467267
## 12     104   PWA       3 96 -0.01711140688 0.7947194 0.08111071 0.16102512
## 13     105   PWA       1 96  0.04886081534 0.5566583 0.05681370 0.11278945
## 14     105   PWA       2 96 -0.00322251799 0.4165570 0.04251467 0.08440229
## 15     105   PWA       3 96 -0.00322251799 0.4165570 0.04251467 0.08440229
## 16     106   PWA       1 96  0.01066637090 0.8139410 0.08307251 0.16491978
## 17     106   PWA       2 96  0.00024970423 0.7642291 0.07799880 0.15484720
## 18     106   PWA       3 96  0.03149970423 0.6629281 0.06765982 0.13432172
## 19     107   PWA       1 96  0.02108303757 0.5166631 0.05273171 0.10468568
## 20     107   PWA       2 96  0.00024970423 0.3322075 0.03390579 0.06731150
## 21     107   PWA       3 96  0.02108303757 0.3302212 0.03370306 0.06690904
## 22     108   PWA       1 96  0.02793069533 0.5306401 0.05415823 0.10751768
## 23     108   PWA       2 95 -0.00320965555 0.3787467 0.03885859 0.07715464
## 24     108   PWA       3 96  0.01751402866 0.4149746 0.04235317 0.08408168
## 25     109   PWA       1 93  0.05348885622 0.6372055 0.06607513 0.13123092
## 26     109   PWA       2 96  0.00039745837 0.6402097 0.06534113 0.12971854
## 27     109   PWA       3 93 -0.01102727281 0.7098288 0.07360582 0.14618751
## 28     110   PWA       1 96  0.01761081534 0.6877691 0.07019514 0.13935497
## 29     110   PWA       2 96  0.00719414868 0.3767503 0.03845192 0.07633671
## 30     110   PWA       3 96  0.01761081534 0.3067229 0.03130478 0.06214784
## 31     111   PWA       1 96  0.03844414868 0.6402097 0.06534113 0.12971854
## 32     111   PWA       2 96  0.00719414868 0.5179349 0.05286151 0.10494337
## 33     111   PWA       3 96 -0.00322251799 0.6150524 0.06277352 0.12462120
## 34     112   PWA       1 86 -0.00493452048 0.6072409 0.06548046 0.13019271
## 35     112   PWA       2 96  0.02268175859 0.6604424 0.06740612 0.13381807
## 36     112   PWA       3 96  0.02268175859 0.7500000 0.07654655 0.15196412
## 37     113   PWA       1 96  0.00719414868 0.5619515 0.05735393 0.11386195
## 38     113   PWA       2 96  0.01761081534 0.5756861 0.05875572 0.11664484
## 39     113   PWA       3 96  0.01761081534 0.5179349 0.05286151 0.10494337
## 40     114   PWA       1 96  0.02093683874 0.7943054 0.08106845 0.16094123
## 41     114   PWA       2 96  0.02093683874 0.7535006 0.07690383 0.15267341
## 42     114   PWA       3 93  0.00010350540 0.5417363 0.05617544 0.11156928
## 43     115   PWA       1 96  0.02802748201 0.3965310 0.04047078 0.08034465
## 44     115   PWA       2 96  0.00719414868 0.3973597 0.04055536 0.08051256
## 45     115   PWA       3 96  0.00719414868 0.3077935 0.03141404 0.06236476
## 46     116   PWA       1 96  0.00719414868 0.4528782 0.04622169 0.09176166
## 47     116   PWA       2 96  0.01761081534 0.1777047 0.01813691 0.03600631
## 48     116   PWA       3 96  0.01761081534 0.2513123 0.02564946 0.05092061
## 49     117   PWA       1 96  0.00024970423 0.6402097 0.06534113 0.12971854
## 50     117   PWA       2 96  0.03149970423 0.4528782 0.04622169 0.09176166
## 51     117   PWA       3 96  0.01066637090 0.3767503 0.03845192 0.07633671
## 52     118   PWA       1 96  0.03149970423 0.6271017 0.06400330 0.12706262
## 53     118   PWA       2 96 -0.02058362910 0.4345294 0.04434897 0.08804384
## 54     118   PWA       3 96  0.03149970423 0.3750000 0.03827328 0.07598206
## 55     119   PWA       1 94  0.01093966439 0.6951148 0.07169562 0.14237329
## 56     119   PWA       2 96  0.05216306865 0.6107588 0.06233531 0.12375124
## 57     119   PWA       3 96 -0.02075359802 0.5179349 0.05286151 0.10494337
## 58     120   PWA       1 96 -0.01033693135 0.7947194 0.08111071 0.16102512
## 59     120   PWA       2 96  0.00007973531 0.8428297 0.08602094 0.17077316
## 60     120   PWA       3 94  0.05349285588 0.7363491 0.07594861 0.15081889
```

```
  df_errors_PWA %>%
   filter(category != "Filler") %>%
   Rmisc::summarySEwithin(.,"error_class",
                          idvar = "subject",
                          withinvars = c("session"),
                          betweenvars = "group",
                          na.rm = T)
```

```
##   group session    N error_class        sd          se         ci
## 1   PWA       1 2386   0.2562733 0.4567964 0.009351633 0.01833817
## 2   PWA       2 2398   0.2101726 0.4065913 0.008302971 0.01628175
## 3   PWA       3 2394   0.1777519 0.3926172 0.008024301 0.01573530
```

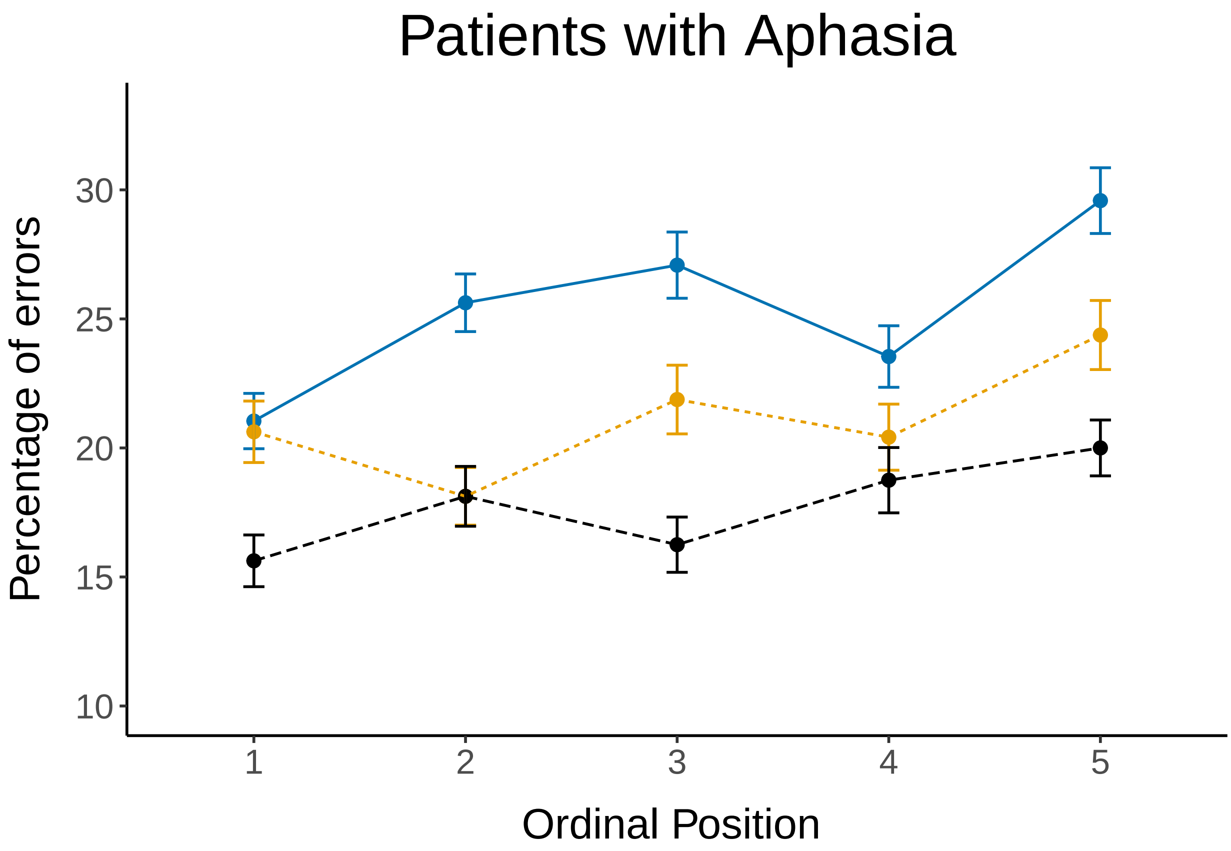
Plotting

```
means_final_errors$session_group <- paste0(means_final_errors$group,
                                           means_final_errors$session)
means_final_errors %>% rename(Session = session, Group = group) %>%
  mutate(Group = factor(Group, levels=c("PWA", "control")))->
  means_final_errors

override.linetype<-c("dotted", "dashed")
(plot_error <- means_final_errors %>%
    mutate(Session=case_when(Session == "1" ~ "day 1",
                             Session == "2" ~ "day 2",
                             Session == "3" ~ "day 8")) %>%
    ggplot(., aes(x=PosOr, y=mean_p,
                  color = Session)) +
  geom_point(size = 2)+
  stat_summary(aes(x=PosOr, y=mean_p, group=session_group,
                  color = Session, linetype=Group),
               fun=mean,  geom="line", size = 0.5) +
  scale_linetype_manual(values=c("dotted", "dashed"))+
  scale_color_manual(values=c("#0072B2", "#E69F00", "#000000"))+
  geom_errorbar(
    aes(ymin=mean_p-se_p, ymax=mean_p+se_p, group = Session), width =.1) +
  apatheme+
  scale_y_continuous(breaks = seq(0, 40, by = 5), limits=c(0,35))+
      theme(
    axis.title.y = element_text(margin = margin(0,10,0,0)),
    axis.title.x = element_text(margin = margin(10,0,0,0)),
    legend.key.width = unit(1, "cm"))+
  labs(x="Ordinal Position ",y ="Percentage of errors"))
```

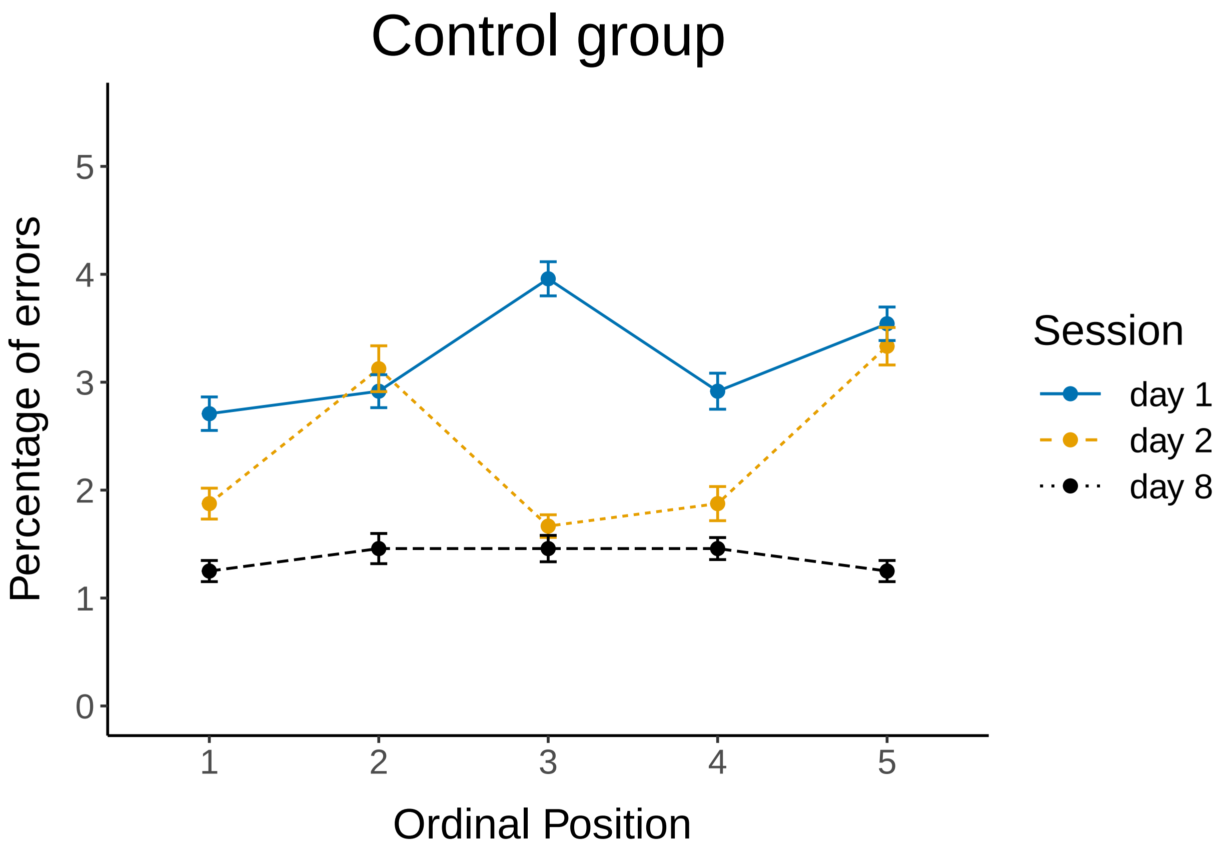


Errors by ordinal position and repetition

```
override.linetype<-c("solid", "dashed", "dotted")
(plot_error_PWA <- means_final_errors %>% filter(Group=="PWA") %>%
    mutate(Session=case_when(Session == "1" ~ "day 1",
                                 Session == "2" ~ "day 2",
                                 Session == "3" ~ "day 8")) %>%
    ggplot(., aes(x=PosOr, y=mean_p, group=Session, color = Session)) +
  geom_point( size = 2)+
      stat_summary(aes(linetype=Session),fun=mean,  geom="line",
                   size = 0.5) +
    scale_color_manual(values=c("#0072B2", "#E69F00", "#000000"))+
  geom_errorbar(aes(ymin=mean_p-se_p, ymax=mean_p+se_p, group = Session),
                width =.1) +
  apatheme+
  scale_y_continuous(breaks = seq(10,30, by = 5), limits=c(10,33))+
      theme(
    axis.title.y = element_text(margin = margin(0,10,0,0)),
    axis.title.x = element_text(margin = margin(10,0,0,0)),
    legend.key.width = unit(1, "cm"),
    legend.position="none")+
  guides(color=guide_legend(
    override.aes=list(linetype=override.linetype)))+
  labs(x="Ordinal Position ",y ="Percentage of errors",
       title="Patients with Aphasia"))
```

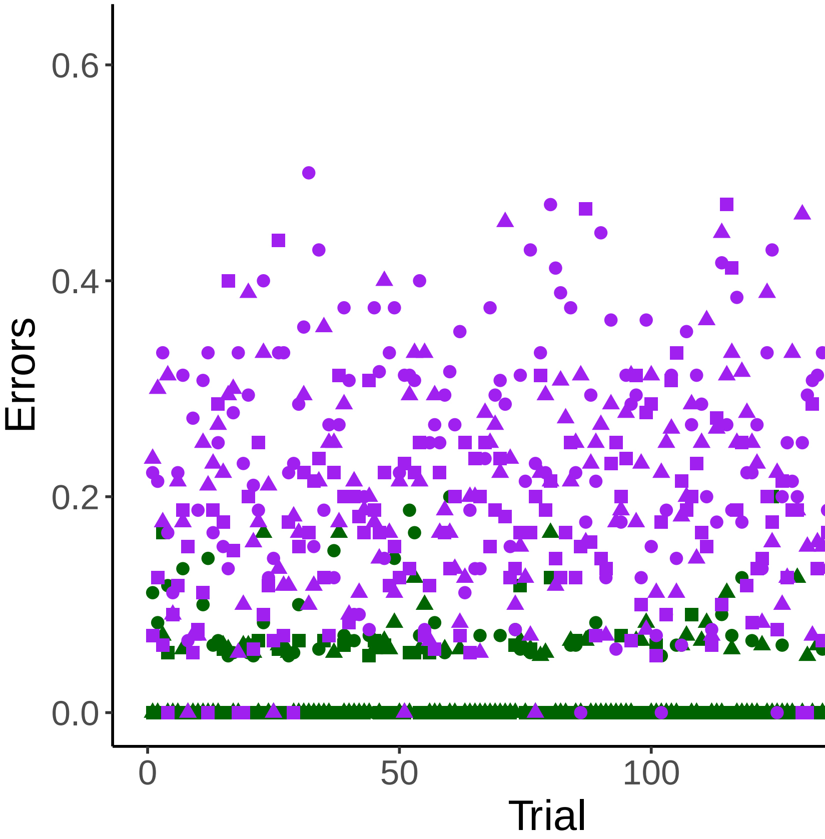


```
(plot_error_control <- means_final_errors %>% filter(Group=="control") %>%
    mutate(Session=case_when(Session == "1" ~ "day 1",
                             Session == "2" ~ "day 2",
                             Session == "3" ~ "day 8")) %>%
    ggplot(., aes(x=PosOr, y=mean_p, group=Session, color = Session)) +
  geom_point( size = 2)+
      stat_summary(aes(linetype=Session),fun=mean,  geom="line",
                   size = 0.5) +
    scale_color_manual(values=c("#0072B2", "#E69F00", "#000000"))+
  geom_errorbar(aes(ymin=mean_p-se_p, ymax=mean_p+se_p, group = Session), width =.1) +
  apatheme+
  scale_y_continuous(breaks = seq(0, 5, by =1), limits=c(0,5.5))+
      theme(
    axis.title.y = element_text(margin = margin(0,10,0,0)),
    axis.title.x = element_text(margin = margin(10,0,0,0)),
    legend.key.width = unit(1, "cm"))+
  guides(color=guide_legend(
    override.aes=list(linetype=override.linetype)))+
  labs(x="Ordinal Position ",y ="Percentage of errors",
       title="Control group"))
```



```r
filename <- "CSI_online_spoken_plot_error.pdf"
ggsave(plot_error, filename =
         here::here("results", "figures", filename),
       width = 18, height = 13, units = "cm",
       dpi = 300, device = cairo_pdf)
#embedFonts(file = here::here("data", "verbal_CSI", "Plots", filename))

ggsave(plot_error_PWA, filename =
         here::here("results", "figures",
                    "CSI_online_spoken_plot_error_PWA.pdf"),
       width = 18, height = 13, units = "cm",
       dpi = 300, device = cairo_pdf)
ggsave(plot_error_control, filename =
         here::here("results", "figures",
                    "CSI_online_spoken_plot_error_control.pdf"),
       width = 18, height = 13, units = "cm",
       dpi = 300, device = cairo_pdf)
```

```r
(plot_errors_all <- ggplot(data=df_errors,
                        aes(x=trial, y=error_class, linetype=session,
                              shape=session, color=group)) +
   stat_summary(aes(color=group, shape=session),fun=mean,
                geom="point", size = 2)+
  apatheme+
  labs(x="Trial ",y ="Errors")+
  scale_color_manual(values=c(control_color, PWA_color)))
```



Control: Plot Errors accross the experiment

```
filename <- "CSI_online_aphasia_errors_across_experiment.pdf"
ggsave(plot_errors_all, filename =
         here::here("results", "figures", filename),
       width = 18, height = 13, units = "cm",
       dpi = 300, device = cairo_pdf)
#embedFonts(file = here::here("results", "figures", filename))
```

GLMM with binomial distribution

Contrast coding *Center predictor variable*

```
df_errors_PWA <- df_errors %>% filter(group=="PWA") %>% droplevels()
df_errors_PWA$PosOr.cont <-
  c(scale(as.numeric(as.character(df_errors_PWA$PosOr)),
        center = T, scale = F))

df_errors$PosOr.cont <-
  c(scale(as.numeric(as.character(df_errors$PosOr)),
        center = T, scale = F))
```

Contrast coding

```
# define contrasts of session: compare 1 to 2 and 1 to 3, intercept is the grand mean => simple coding
c<-contr.treatment(3)
my.coding<-matrix(rep(1/3, 6), ncol=2)
my.simple<-c-my.coding
my.simple
```

```
##            2          3
## 1 -0.3333333 -0.3333333
## 2  0.6666667 -0.3333333
## 3 -0.3333333  0.6666667
```

```
contrasts(df_errors$session)<-my.simple
levels(df_errors$session)
```

```
## [1] "1" "2" "3"
```

```
contrasts(df_errors_PWA$session)<-my.simple
levels(df_RTs_PWA$session)
```

```
## [1] "1" "2" "3"
```

```
## Define contrast of group
contrasts(df_errors$group) <- MASS::contr.sdif(2)
levels(df_errors$group)
```

```
## [1] "control" "PWA"
```

```
levels(df_errors_PWA$group)
```

```
## [1] "PWA"
```

Error analyses with factors Ordinal position x Session x Group *GLMM*

Compute the full model with the maximal random structure. If model fails to converge, increase optimizer iterations, exclude correlation parameters, andd step-wise reduce the random structure by excluding variables explaining close to zero variance. If the model converges, test whether it also converges with correlation parameters.

```
# m2_error <- glmer(error_class ~ PosOr.cont*session*group +
#                       (PosOr.cont*session|subject) +
#                       (PosOr.cont*session*group|category) ,
#                     data =df_errors, family = "binomial",
#                     control=glmerControl(optimizer = "bobyqa"))

# 2) The model fit is singular -> increase optimizer iterations
# m2_error <- glmer(error_class ~ PosOr.cont*session +
#                       (PosOr.cont*session|subject) +
#                       (PosOr.cont*session*group|category) ,
#                   data =df_errors, family = "binomial",
#                   control=glmerControl(optimizer = "bobyqa",
#                                  optCtrl = list(maxfun = 2e5)))

# 3) Further reduce by excluding correlation parameters
# m2_error <- afex::lmer_alt(error_class ~ PosOr.cont*session*group +
#                     (PosOr.cont*session||subject) +
#                     (PosOr.cont*session*group||category) ,
#                   data =df_errors, family = "binomial",
#                   control=glmerControl(optimizer = "bobyqa",
#                                  optCtrl = list(maxfun = 2e5)))

# 4) Model fit is still singular -> further reduce
# m2_error <- afex::lmer_alt(error_class ~ PosOr.cont*session*group +
#                     (PosOr.cont+session||subject) +
#                     (PosOr.cont*session*group-session-PosOr.cont:session-
#                        PosOr.cont:group-session:group||category) ,
#                   data =df_errors, family = "binomial",
#                   control=glmerControl(optimizer = "bobyqa",
#                                  optCtrl = list(maxfun = 2e5)))
# m2_error <- afex::lmer_alt(error_class ~ PosOr.cont*session*group +
#                     (PosOr.cont||subject) +
#                     (PosOr.cont*session*group-
#                     session-PosOr.cont:session-
#                        PosOr.cont:group-
#                       session:group-PosOr.cont||category) ,
#                   data =df_errors, family = "binomial",
#                   control=glmerControl(optimizer = "bobyqa",
#                                  optCtrl = list(maxfun = 2e5)))
# m2_error <- afex::lmer_alt(error_class ~ PosOr.cont*session*group +
#                     (PosOr.cont||subject) +
#                     (group||category) ,
#                   data =df_errors, family = "binomial",
#                   control=glmerControl(optimizer = "bobyqa",
#                                  optCtrl = list(maxfun = 2e5)))

# does the model also converge when correlation parameters are included? -> yes!
m2_error <- glmer(error_class ~ PosOr.cont*session*group +
                    (PosOr.cont|subject) +
                    (group|category) ,
                  data =df_errors, family = "binomial",
                  control=glmerControl(optimizer = "bobyqa",
                                 optCtrl = list(maxfun = 2e5)))

# rePCA(m2_error)
didLmerConverge(m2_error)
```

```
##  The relative maximum gradient of 0.0000163 is less than our 0.001 criterion.
##  You can safely ignore any warnings about a claimed convergence failure.
```

```
summary(m2_error)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: error_class ~ PosOr.cont * session * group + (PosOr.cont | subject) +
##     (group | category)
##    Data: df_errors
## Control: glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 200000))
##
##      AIC      BIC   logLik deviance df.resid
##   6751.3   6887.5  -3357.6   6715.3    14279
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.6525 -0.2781 -0.1491 -0.0628 19.9112
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  subject  (Intercept) 1.35055  1.1621
##           PosOr.cont  0.01236  0.1112   0.28
##  category (Intercept) 0.86949  0.9325
##           group2-1    0.91566  0.9569   -0.85
## Number of obs: 14297, groups:  subject, 40; category, 24
##
## Fixed effects:
##                              Estimate Std. Error z value             Pr(>|z|)
## (Intercept)                  -3.29230    0.27666 -11.900 < 0.0000000000000002
## PosOr.cont                    0.06105    0.03806   1.604             0.108654
## session2                     -0.35994    0.09769  -3.685             0.000229
## session3                     -0.80417    0.11210  -7.174    0.000000000000731
## group2-1                      2.88334    0.44599   6.465    0.000000000101315
## PosOr.cont:session2          -0.01056    0.06886  -0.153             0.878112
## PosOr.cont:session3          -0.05147    0.07895  -0.652             0.514418
## PosOr.cont:group2-1           0.05079    0.07378   0.688             0.491170
## session2:group2-1            -0.05772    0.19539  -0.295             0.767674
## session3:group2-1             0.20457    0.22419   0.912             0.361512
## PosOr.cont:session2:group2-1 -0.07065    0.13771  -0.513             0.607939
## PosOr.cont:session3:group2-1  0.02124    0.15787   0.135             0.892979
##
## (Intercept)                  ***
## PosOr.cont
## session2                     ***
## session3                     ***
## group2-1                     ***
## PosOr.cont:session2
## PosOr.cont:session3
## PosOr.cont:group2-1
## session2:group2-1
## session3:group2-1
## PosOr.cont:session2:group2-1
## PosOr.cont:session3:group2-1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) PsOr.c sessn2 sessn3 grp2-1 PsO.:2 PsO.:3 PO.:2- s2:2-1
## PosOr.cont   0.082
## session2     0.019 -0.011
## session3     0.052  0.012  0.384
## group2-1    -0.345  0.003 -0.017 -0.053
## PsOr.cnt:s2 -0.002  0.082 -0.071 -0.023  0.004
## PsOr.cnt:s3  0.003  0.242 -0.023 -0.023 -0.004  0.383
## PsOr.cn:2-1  0.003 -0.471  0.013 -0.013  0.105 -0.060 -0.207
## sssn2:gr2-1 -0.013  0.013 -0.630 -0.229  0.023  0.060  0.018 -0.012
## sssn3:gr2-1 -0.043 -0.012 -0.230 -0.696  0.063  0.018  0.014  0.012  0.384
## PsOr.:2:2-1  0.003 -0.054  0.060  0.018 -0.003 -0.627 -0.228  0.080 -0.071
## PsOr.:3:2-1 -0.003 -0.195  0.018  0.014  0.004 -0.228 -0.694  0.242 -0.023
##             s3:2-1 PO.:2:
## PosOr.cont
## session2
## session3
## group2-1
## PsOr.cnt:s2
## PsOr.cnt:s3
## PsOr.cn:2-1
## sssn2:gr2-1
## sssn3:gr2-1
## PsOr.:2:2-1 -0.023
## PsOr.:3:2-1 -0.023  0.383
```

```
# save model output
saveRDS(m2_error, file = here::here("results", "tables", "CSI_online_aphasia_SessionxGroup_glmm_errors.
tab_model(m2_error,transform = NULL,
          show.re.var = F, show.stat = T,show.r2 = F,show.icc = F,
          title = "GLMM (Binomial distribution) of Errors Predicted by Ordinal Position and Session,
          PWA only",
          dv.labels = "Error Rate",
          #string.pred = "",
          string.stat = "z-Value",
          file =
            here::here(
              "results", "tables",
              "CSI_online_aphasia_SessionxGroup_glmm_errors.html"))
```

GLMM (Binomial distribution) of Errors Predicted by Ordinal Position and Session, PWA only

Error Rate

Predictors

Log-Odds

CI

z-Value

p

(Intercept)

-3.29

-3.83 – -2.75

-11.90

<0.001

PosOr cont

0.06

-0.01 – 0.14

1.60

0.109

session [2]

-0.36

-0.55 – -0.17

-3.68

<0.001

session [3]

-0.80

-1.02 – -0.58

-7.17

<0.001

group2-1

2.88

2.01 – 3.76

6.46

<0.001

PosOr cont × session [2]

-0.01

-0.15 – 0.12

-0.15

0.878

PosOr cont × session [3]

-0.05

-0.21 – 0.10

-0.65

0.514

PosOr.cont:group2-1

0.05

-0.09 – 0.20

0.69

0.491

session2:group2-1

-0.06

-0.44 – 0.33

-0.30

0.768

session3:group2-1

0.20

-0.23 – 0.64

0.91

0.362

PosOr.cont:session2:group2-1

-0.07

-0.34 – 0.20

-0.51

0.608

PosOr.cont:session3:group2-1

0.02

-0.29 – 0.33

0.13

0.893

N subject

40

N category

24

Observations

14297

```
#performance::check_model(m2_error)
# ggResidpanel::resid_panel(m2_error, smoother = TRUE,
#                            qqbands = TRUE, type = "pearson")
performance::model_performance(m2_error)
```

```
## # Indices of model performance
##
## AIC      |     AICc |      BIC | R2 (cond.) | R2 (marg.) |   ICC |  RMSE | Sigma | Log_loss | Score_
## ----------------------------------------------------------------------------------------------------
## 6751.266 | 6751.314 | 6887.487 |      0.587 |      0.277 | 0.429 | 0.257 | 1.000 |    0.224 |      -
```

Make the estimates interpretable

```r
# Odds Ratio:
x <- data.frame(summary(m2_error)$coefficients)
x$Odds_Ratio <- plogis(x$Estimate)
x %>% dplyr::select(Estimate, Odds_Ratio) %>%
  mutate(Estimate=round(Estimate,2),
         Odds_Ratio=round(Odds_Ratio,2))
```

```
##                               Estimate Odds_Ratio
## (Intercept)                      -3.29       0.04
## PosOr.cont                        0.06       0.52
## session2                         -0.36       0.41
## session3                         -0.80       0.31
## group2-1                          2.88       0.95
## PosOr.cont:session2              -0.01       0.50
## PosOr.cont:session3              -0.05       0.49
## PosOr.cont:group2-1               0.05       0.51
## session2:group2-1                -0.06       0.49
## session3:group2-1                 0.20       0.55
## PosOr.cont:session2:group2-1     -0.07       0.48
## PosOr.cont:session3:group2-1      0.02       0.51
```

PWA only *GLMM*

```r
# m1_error <- glmer(error_class ~ PosOr.cont*session +
#                     (PosOr.cont*session|subject) +
#                     (PosOr.cont*session|category) ,
#                   data =df_errors_PWA, family = "binomial",
#                   control=glmerControl(optimizer = "bobyqa"))

# 2) The model fit is singular -> reduce optimizer iterations
# m1_error <- glmer(error_class ~ PosOr.cont*session +
#                     (PosOr.cont*session|subject) +
#                     (PosOr.cont*session|category) ,
#                   data =df_errors_PWA, family = "binomial",
#                   control=glmerControl(optimizer = "bobyqa",
#                                  optCtrl = list(maxfun = 2e5)))

# 3) Further reduce by excluding correlation parameters
# m1_error <- afex::lmer_alt(error_class ~ PosOr.cont*session +
#                     (PosOr.cont*session||subject) +
#                     (PosOr.cont*session||category) ,
#                   data =df_errors_PWA, family = "binomial",
#                   control=glmerControl(optimizer = "bobyqa",
#                                  optCtrl = list(maxfun = 2e5)))

# 4) Model fit is still singular -> further reduce
# m1_error <- afex::lmer_alt(error_class ~ PosOr.cont*session +
#                     (PosOr.cont*session||subject) +
#                     (1|category) ,
#                   data =df_errors_PWA, family = "binomial",
#                   control=glmerControl(optimizer = "bobyqa",
#                                  optCtrl = list(maxfun = 2e5)))
# m1_error <- afex::lmer_alt(error_class ~ PosOr.cont*session +
#                     (PosOr.cont+session||subject) +
#                     (1|category) ,
#                   data =df_errors_PWA, family = "binomial",
#                   control=glmerControl(optimizer = "bobyqa",
#                                  optCtrl = list(maxfun = 2e5)))
m1_error <- glmer(error_class ~ PosOr.cont*session +
                    (PosOr.cont |subject) +
                    (1|category) ,
                  data =df_errors_PWA, family = "binomial",
                  control=glmerControl(optimizer = "bobyqa",
                                 optCtrl = list(maxfun = 2e5)))
rePCA(m1_error)
```

```
## $category
## Standard deviations (1, .., p=1):
## [1] 0.5889541
##
## Rotation (n x k) = (1 x 1):
##      [,1]
## [1,]    1
##
## $subject
## Standard deviations (1, .., p=2):
## [1] 1.5014522 0.1293152
##
## Rotation (n x k) = (2 x 2):
##             [,1]        [,2]
## [1,] -0.99950726 -0.03138835
## [2,] -0.03138835  0.99950726
##
## attr(,"class")
## [1] "prcomplist"
```

```
didLmerConverge(m1_error)
```

```
##  The relative maximum gradient of 0.00000289 is less than our 0.001 criterion.
##  You can safely ignore any warnings about a claimed convergence failure.
```

```
summary(m1_error)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: error_class ~ PosOr.cont * session + (PosOr.cont | subject) +
##     (1 | category)
##    Data: df_errors_PWA
## Control: glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 200000))
##
##      AIC      BIC   logLik deviance df.resid
##   5336.7   5405.5  -2658.3   5316.7     7168
##
```

```
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.7692 -0.3964 -0.2423 -0.1216 10.6603
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  category (Intercept) 0.34687  0.5890
##  subject  (Intercept) 2.25215  1.5007
##           PosOr.cont  0.01893  0.1376   0.34
## Number of obs: 7178, groups:  category, 24; subject, 20
##
## Fixed effects:
##                       Estimate Std. Error z value             Pr(>|z|)
## (Intercept)           -1.86442    0.35937  -5.188 0.000000212474493786 ***
## PosOr.cont             0.08469    0.04309   1.965               0.0494 *
## session2              -0.39126    0.08430  -4.642 0.000003457979079708 ***
## session3              -0.70618    0.08763  -8.059 0.000000000000000771 ***
## PosOr.cont:session2 -0.04642    0.05957  -0.779               0.4358
## PosOr.cont:session3 -0.04163    0.06188  -0.673               0.5011
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) PsOr.c sessn2 sessn3 PsO.:2
## PosOr.cont   0.224
## session2     0.009  0.004
## session3     0.017  0.000  0.460
## PsOr.cnt:s2  0.001  0.050 -0.028 -0.014
## PsOr.cnt:s3  0.000  0.094 -0.014 -0.029  0.459
```

```r
# save model output
saveRDS(m1_error, file =
          here::here("results", "tables",
                     "CSI_online_aphasia_PWA_glmm_errors.RDS"))
tab_model(m1_error,transform = NULL,
          show.re.var = F, show.stat = T,show.r2 = F,show.icc = F,
          title = "GLMM (Binomial distribution) of Errors Predicted by Ordinal Position and Session,
          PWA only",
          dv.labels = "Error Rate",
          string.stat = "z-Value",
          file = here::here(
            "results", "tables",
            "CSI_online_aphasia_PWA_glmm_errors.html"))
```

GLMM (Binomial distribution) of Errors Predicted by Ordinal Position and Session, PWA only

Error Rate

Predictors

Log-Odds

CI

z-Value

p

(Intercept)

-1.86

-2.57 – -1.16

-5.19

<0.001

PosOr cont

0.08

0.00 – 0.17

1.97

0.049

session [2]

-0.39

-0.56 – -0.23

-4.64

<0.001

session [3]

-0.71

-0.88 – -0.53

-8.06

<0.001

PosOr cont × session [2]

-0.05

-0.16 – 0.07

-0.78

0.436

PosOr cont × session [3]

-0.04

-0.16 – 0.08

-0.67

0.501

N subject

20

N category

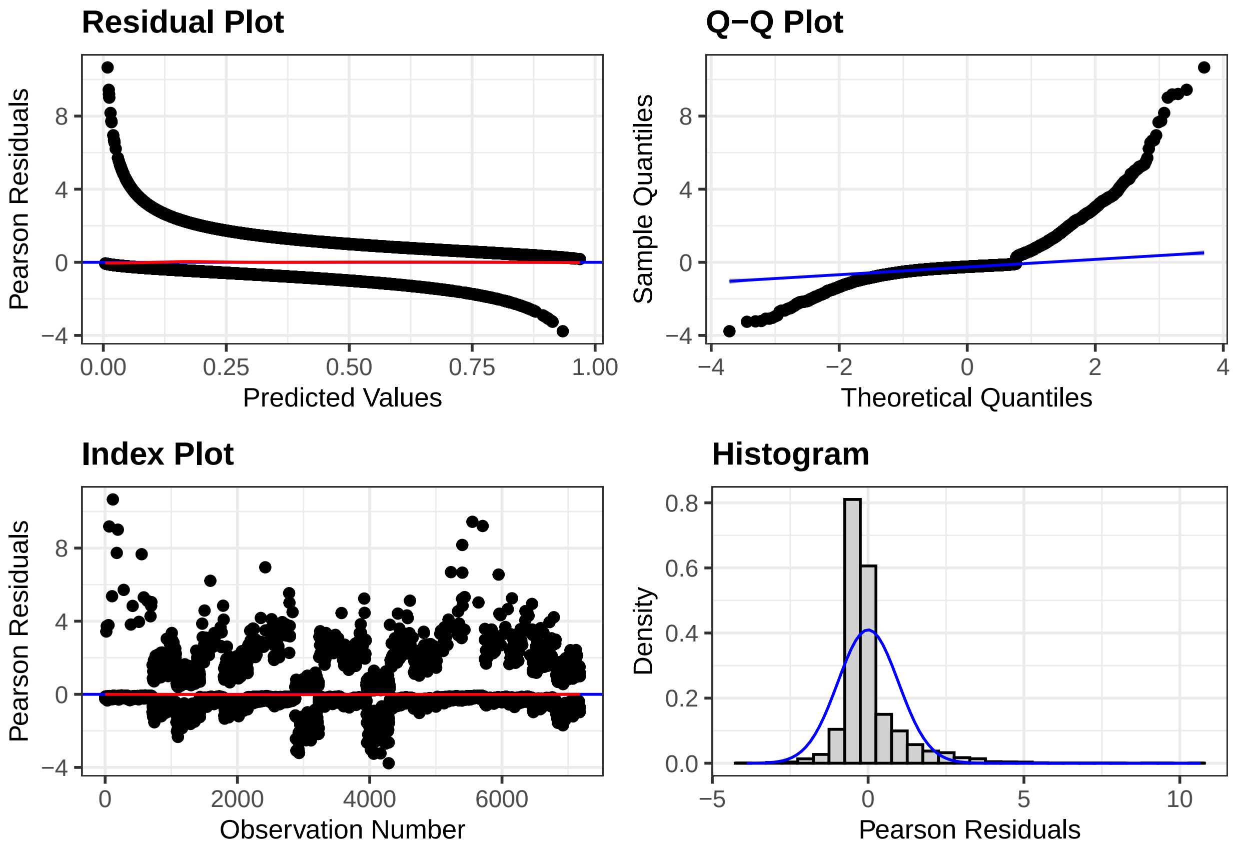
24

Observations

7178

```r
#performance::check_model(m1_error)
ggResidpanel::resid_panel(m1_error, smoother = TRUE,
                          qqbands = TRUE, type = "pearson")
```

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



```r
performance::model_performance(m1_error)
```

```
## # Indices of model performance
##
## AIC      |     AICc |      BIC | R2 (cond.) | R2 (marg.) |   ICC |  RMSE | Sigma | Log_loss | Score_
## -----------------------------------------------------------------------------------------------------
## 5336.693 | 5336.724 | 5405.481 |      0.454 |      0.016 | 0.445 | 0.332 | 1.000 |    0.356 |      -
```

Make the estimates interpretable

```r
# Odds Ratio:
x <- data.frame(summary(m1_error)$coefficients)
x$Odds_Ratio <- plogis(x$Estimate)
x %>% dplyr::select(Estimate, Odds_Ratio) %>%
  mutate(Estimate=round(Estimate,2),
         Odds_Ratio=round(Odds_Ratio,2))
```

```
##                       Estimate Odds_Ratio
## (Intercept)              -1.86       0.13
## PosOr.cont                0.08       0.52
## session2                 -0.39       0.40
## session3                 -0.71       0.33
## PosOr.cont:session2      -0.05       0.49
## PosOr.cont:session3      -0.04       0.49
```

```
# m2_error_control <- glmer(error_class ~ PosOr.cont*session +
#                        (PosOr.cont*session|subject) +
#                        (PosOr.cont*session|category) ,
#                      data =df_errors[df_errors$group=="control",],
#                      family = "binomial",
#                      control=glmerControl(optimizer = "bobyqa"))
# m2_error_control <- afex::lmer_alt(error_class ~ PosOr.cont*session +
#                        (PosOr.cont*session||subject) +
#                        (PosOr.cont*session||category) ,
#                      data =df_errors[df_errors$group=="control",],
#                      family = "binomial",
#                      control=glmerControl(optimizer = "bobyqa"))
# m2_error_control <- afex::lmer_alt(error_class ~ PosOr.cont*session +
#                        (1|subject) +
#                        (PosOr.cont*session-PosOr.cont-session||category) ,
#                      data =df_errors[df_errors$group=="control",],
#                      family = "binomial",
#                      control=glmerControl(optimizer = "bobyqa"))
m2_error_control <- glmer(error_class ~ PosOr.cont*session +
                     (1|subject) +
                     (1|category) ,
                  data =df_errors[df_errors$group=="control",],
                  family = "binomial",
                  control=glmerControl(optimizer = "bobyqa"))

# rePCA(m2_error_control)
didLmerConverge(m2_error_control)
```

Exploratory follow-up: Make sure there is enough power in the control group

```
##  The relative maximum gradient of 0.0000012 is less than our 0.001 criterion.
##  You can safely ignore any warnings about a claimed convergence failure.
```

```
summary(m2_error_control)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: error_class ~ PosOr.cont * session + (1 | subject) + (1 | category)
##    Data: df_errors[df_errors$group == "control", ]
## Control: glmerControl(optimizer = "bobyqa")
##
```

```
##      AIC      BIC   logLik deviance df.resid
##   1414.1   1469.1   -699.1   1398.1     7111
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -0.6002 -0.1607 -0.0987 -0.0622 22.0736
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  category (Intercept) 1.7351   1.3172
##  subject  (Intercept) 0.2353   0.4851
## Number of obs: 7119, groups:  category, 24; subject, 20
##
## Fixed effects:
##                     Estimate Std. Error z value             Pr(>|z|)
## (Intercept)         -4.61206    0.32706 -14.102 < 0.0000000000000002 ***
## PosOr.cont           0.04362    0.05789   0.753               0.4512
## session2            -0.33004    0.17645  -1.870               0.0614 .
## session3            -0.89977    0.20691  -4.349            0.0000137 ***
## PosOr.cont:session2  0.02371    0.12453   0.190               0.8490
## PosOr.cont:session3 -0.06058    0.14603  -0.415               0.6783
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Correlation of Fixed Effects:
##             (Intr) PsOr.c sessn2 sessn3 PsO.:2
## PosOr.cont  -0.012
## session2     0.029 -0.017
## session3     0.086  0.017  0.367
## PsOr.cnt:s2 -0.005  0.101 -0.082 -0.025
## PsOr.cnt:s3  0.006  0.315 -0.025 -0.021  0.367
```

```r
# save model output
saveRDS(m2_error_control, file = here::here("results", "tables", "CSI_online_aphasia_Session_control_gr
tab_model(m2_error_control,transform = NULL,
          show.re.var = F, show.stat = T,show.r2 = F,show.icc = F,
          title = "GLMM (Binomial distribution) of Errors Predicted by Ordinal Position and Session,
          Control group only",
          dv.labels = "Error Rate",
          #string.pred = "",
          string.stat = "z-Value",
          file = here::here(
            "results", "tables",
            "CSI_online_aphasia_Session_control_group_errors.html"))
```

GLMM (Binomial distribution) of Errors Predicted by Ordinal Position and Session, Control group only

Error Rate

Predictors

Log-Odds

CI

z-Value

p

(Intercept)

-4.61

-5.25 – -3.97

-14.10

<0.001

PosOr cont

0.04

-0.07 – 0.16

0.75

0.451

session [2]

-0.33

-0.68 – 0.02

-1.87

0.061

session [3]

-0.90

-1.31 – -0.49

-4.35

<0.001

PosOr cont × session [2]

0.02

-0.22 – 0.27

0.19

0.849

PosOr cont × session [3]

-0.06

-0.35 – 0.23

-0.41

0.678

N subject

20

N category

24

Observations

7119

Control analyses: Covariate tests (AAT, LEMO test, MPO, lesion size)

Add covariates individually into the converging models, and use LRT to assess which covariates exlpain additional variance when added to the main model. Then conduct one model with all covariates that explain additional variance.

Add covariates

```
read.csv2(here::here("data", "additional_data",
                     "questionnaire-test_data.csv")) -> tests


read.csv2(here::here("data", "additional_data",
                     "MRT_ROIs.csv")) -> mrt
mrt$Proband_in <- mrt$Proband_in+100
mrt$SoSci_ID <- toupper(mrt$SoSci_ID)
all(mrt$SoSci_ID %in% df_RTs_PWA$OR02_01) &
  all(df_RTs_PWA$OR02_01 %in% mrt$SoSci_ID)
```

```
## [1] TRUE
```

```
# table(df_RTs_PWA$subject)
```

Combine with PWA data

```
df_RTs_PWA$TokenTest <- NA
df_RTs_PWA$AAT <- NA
df_RTs_PWA$AAT_spontansprache <- NA
df_RTs_PWA$LEMO <- NA
df_RTs_PWA$mont_post_onset <- NA
df_RTs_PWA$LHoverall <- NA
df_RTs_PWA$ATL <- NA
df_RTs_PWA$IFG <- NA
df_RTs_PWA$MTG_ITG <- NA
df_RTs_PWA$SMG_AG <- NA
df_RTs_PWA$Precentral <- NA
df_RTs_PWA$ID <- NA
df_RTs_PWA$SoSci_ID <- NA
df_RTs_PWA$Proband_in <- NA
for(i in 1:nrow(mrt)){
  df_RTs_PWA$TokenTest[
    toupper(df_RTs_PWA$OR02_01) == toupper(tests$Code[i])] <-
    as.numeric(as.character(tests$Token.Test..Prozentrang.[i]))
  df_RTs_PWA$AAT[
    toupper(df_RTs_PWA$OR02_01) == toupper(tests$Code[i])] <-
    as.numeric(as.character(tests$AAT.Ben.gesamt..Prozentrang.[i]))
  df_RTs_PWA$AAT_spontansprache[
    toupper(df_RTs_PWA$OR02_01) == toupper(tests$Code[i])] <-
    as.numeric(as.character(tests$AAT.Untertest.Spontansprache.Semantik..Punktwert.Anzahl.korrekt.[i]))
  df_RTs_PWA$LEMO[
    toupper(df_RTs_PWA$OR02_01) == toupper(tests$Code[i])] <-
    as.numeric(as.character(tests$LEMO.V15.Syn.aud..Mit.Abl..n.40...Anz.korr[i]))
  df_RTs_PWA$months_post_onset[
    toupper(df_RTs_PWA$OR02_01) == toupper(tests$Code[i])] <-
    as.numeric(as.character(tests$MPO..months.post.onset.[i]))
  df_RTs_PWA$LHoverall[
    df_RTs_PWA$OR02_01 == mrt$SoSci_ID[i]] <-
    as.numeric(as.character(mrt$LH.Gesamt[i]))
  df_RTs_PWA$ATL[
    df_RTs_PWA$OR02_01 == mrt$SoSci_ID[i]] <-
     as.numeric(as.character(mrt$ATL[i]))
  df_RTs_PWA$IFG[
    df_RTs_PWA$OR02_01 == mrt$SoSci_ID[i]] <-
     as.numeric(as.character(mrt$IFGorb.op.tri[i]))
  df_RTs_PWA$MTG_ITG[
    df_RTs_PWA$OR02_01 == mrt$SoSci_ID[i]] <-
     as.numeric(as.character(mrt$MTG...ITG[i]))
  df_RTs_PWA$SMG_AG[
    df_RTs_PWA$OR02_01 == mrt$SoSci_ID[i]] <-
     as.numeric(as.character(mrt$SMG...AG[i]))
  df_RTs_PWA$Precentral[
    df_RTs_PWA$OR02_01 == mrt$SoSci_ID[i]] <-
     as.numeric(as.character(mrt$Precentral[i]))
  df_RTs_PWA$ID[
    df_RTs_PWA$OR02_01 == mrt$SoSci_ID[i]] <-
     mrt$ID[i]
  df_RTs_PWA$SoSci_ID[
    df_RTs_PWA$OR02_01 == mrt$SoSci_ID[i]] <-
     mrt$SoSci_ID[i]
  df_RTs_PWA$Proband_in[
    df_RTs_PWA$OR02_01 == mrt$SoSci_ID[i]] <-
     as.numeric(as.character(mrt$Proband_in[i]))
}
```

```
## Warning: Unknown or uninitialised column: ‘months_post_onset‘.
```

```
sum(!(df_RTs_PWA$OR02_01==df_RTs_PWA$SoSci_ID), na.rm=T)
```

```
## [1] 0
```

```
write.csv(df_RTs_PWA, here::here(
  "data", "additional_data",
  "CSI_online_aphasia_raw_data_for_RT_analyses_incl_MRT.csv"))



df_errors_PWA$TokenTest <- NA
df_errors_PWA$AAT <- NA
df_errors_PWA$AAT_spontansprache <- NA
```

```r
df_errors_PWA$LEMO <- NA
df_errors_PWA$mont_post_onset <- NA
df_errors_PWA$LHoverall <- NA
df_errors_PWA$ATL <- NA
df_errors_PWA$IFG <- NA
df_errors_PWA$MTG_ITG <- NA
df_errors_PWA$SMG_AG <- NA
df_errors_PWA$Precentral <- NA
df_errors_PWA$ID <- NA
df_errors_PWA$SoSci_ID <- NA
df_errors_PWA$Proband_in <- NA
for(i in 1:nrow(mrt)){
  df_errors_PWA$TokenTest[
    toupper(df_errors_PWA$OR02_01) == toupper(tests$Code[i])] <-
    as.numeric(as.character(tests$Token.Test..Prozentrang.[i]))
  df_errors_PWA$AAT[
    toupper(df_errors_PWA$OR02_01) == toupper(tests$Code[i])] <-
    as.numeric(as.character(tests$AAT.Ben.gesamt..Prozentrang.[i]))
  df_errors_PWA$AAT_spontansprache[
    toupper(df_errors_PWA$OR02_01) == toupper(tests$Code[i])] <-
    as.numeric(as.character(
      tests$AAT.Untertest.Spontansprache.Semantik..Punktwert.Anzahl.korrekt.[i]))
  df_errors_PWA$LEMO[
    toupper(df_errors_PWA$OR02_01) == toupper(tests$Code[i])] <-
    as.numeric(as.character(
      tests$LEMO.V15.Syn.aud..Mit.Abl..n.40...Anz.korr[i]))
  df_errors_PWA$months_post_onset[
    toupper(df_errors_PWA$OR02_01) == toupper(tests$Code[i])] <-
    as.numeric(as.character(tests$MPO..months.post.onset.[i]))
  df_errors_PWA$LHoverall[
    df_errors_PWA$OR02_01 == mrt$SoSci_ID[i]] <-
    as.numeric(as.character(mrt$LH.Gesamt[i]))
  df_errors_PWA$ATL[
    df_errors_PWA$OR02_01 == mrt$SoSci_ID[i]] <-
     as.numeric(as.character(mrt$ATL[i]))
  df_errors_PWA$IFG[
    df_errors_PWA$OR02_01 == mrt$SoSci_ID[i]] <-
     as.numeric(as.character(mrt$IFGorb.op.tri[i]))
  df_errors_PWA$MTG_ITG[
    df_errors_PWA$OR02_01 == mrt$SoSci_ID[i]] <-
     as.numeric(as.character(mrt$MTG...ITG[i]))
  df_errors_PWA$SMG_AG[
    df_errors_PWA$OR02_01 == mrt$SoSci_ID[i]] <-
     as.numeric(as.character(mrt$SMG...AG[i]))
  df_errors_PWA$Precentral[
    df_errors_PWA$OR02_01 == mrt$SoSci_ID[i]] <-
     as.numeric(as.character(mrt$Precentral[i]))
  df_errors_PWA$ID[
    df_errors_PWA$OR02_01 == mrt$SoSci_ID[i]] <-
     mrt$ID[i]
  df_errors_PWA$SoSci_ID[
    df_errors_PWA$OR02_01 == mrt$SoSci_ID[i]] <-
     mrt$SoSci_ID[i]
  df_errors_PWA$Proband_in[
    df_errors_PWA$OR02_01 == mrt$SoSci_ID[i]] <-
     as.numeric(as.character(mrt$Proband_in[i]))
}
```

```
## Warning: Unknown or uninitialised column: ‘months_post_onset‘.
```

```
sum(!(df_errors_PWA$OR02_01==df_errors_PWA$SoSci_ID), na.rm=T)
```

```
## [1] 0
```

```
write.csv(df_errors_PWA, here::here(
  "data", "additional_data",
  "CSI_online_aphasia_raw_data_for_error_analyses_incl_MRT.csv"))
```

Add tests individually and assess whehther they explain additional variance

AAT: Token test

RTs Add into converging model

```
df_RTs_PWA$PosOr.cont <- scale(as.numeric(as.character(df_RTs_PWA$PosOr)),
                                        center = T, scale = F)
table(df_RTs_PWA$PosOr.cont)
```

```
##
##  -1.96955418629938 -0.969554186299384 0.0304458137006161   1.03044581370062
##               1132               1117               1103               1119
##   2.03044581370062
##               1047
```

```
mean(df_RTs_PWA$PosOr.cont); sd(df_RTs_PWA$PosOr.cont)
```

```
## [1] -0.0000000000000001123241
```

```
## [1] 1.40862
```

```
# define contrasts of session: compare 1 to 2 and 1 to 3, intercept is the grand mean => simple coding
c<-contr.treatment(3)
my.coding<-matrix(rep(1/3, 6), ncol=2)
my.simple<-c-my.coding
my.simple
```

```
##            2          3
## 1 -0.3333333 -0.3333333
## 2  0.6666667 -0.3333333
## 3 -0.3333333  0.6666667
```

```r
contrasts(df_RTs_PWA$session)<-my.simple
levels(df_RTs_PWA$session)
```

```
## [1] "1" "2" "3"
```

```r
## Center token test
df_RTs_PWA$TokenTest_c <- scale(df_RTs_PWA$TokenTest, center=T, scale=T)
```

```r
m1_lmm_PWA_tt <- lmer(VOTlog ~ PosOr.cont*session*TokenTest_c +
               (PosOr.cont+session|subject) +
              (1|category),
             data = df_RTs_PWA,
            control=lmerControl(optimizer = "bobyqa",
                                 optCtrl = list(maxfun = 2e5)))
didLmerConverge(m1_lmm_PWA_tt)
```

```
##  The relative maximum gradient of 0.000000998 is less than our 0.001 criterion.
##  You can safely ignore any warnings about a claimed convergence failure.
```

```r
## Warnings can be ignored

# summary(m1_lmm_PWA_tt)
# anova(m1_lmm_PWA_tt)
#
# saveRDS(m1_lmm_PWA_tt,  file = here::here("results", "tables", "CSI_online_aphasia_PWA_lmm_VOT_plus-T
# tab_model(m1_lmm_PWA_tt,transform = NULL,
#           show.re.var = F, show.stat = T,show.r2 = F,show.icc = F,
#           title = "LMM of VOTs Predicted by Ordinal Position and Session",
#           df.method = "satterthwaite",
#           dv.labels = "Vocal Onset Time (log-transformed)",
#           #string.pred = "",
#           string.stat = "t-Value",
#           file = here::here(
#             "results", "tables",
#             "CSI_online_aphasia_PWA_control_lmm_VOT_TokenTest.html"))
```

Does the model with Token test fit the data better? -> yes!

```r
# summary(m1_lmm_PWA)
# performance::model_performance(m1_lmm_PWA)
anova(m1_lmm_PWA, m1_lmm_PWA_tt)
```

```
## refitting model(s) with ML (instead of REML)

## Data: df_RTs_PWA
## Models:
## m1_lmm_PWA: VOTlog ~ PosOr.cont * session + (PosOr.cont + session | subject) + (1 | category)
## m1_lmm_PWA_tt: VOTlog ~ PosOr.cont * session * TokenTest_c + (PosOr.cont + session | subject) + (1 |
##               npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## m1_lmm_PWA      18 1697.5 1816.5 -830.73   1661.5
## m1_lmm_PWA_tt   24 1696.5 1855.2 -824.23   1648.5 13.004  6    0.04297 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Errors Add into converging model

```
df_errors_PWA$PosOr.cont <- scale(as.numeric(as.character(df_errors_PWA$PosOr)),
                                  center = T, scale = F)
# define contrasts of session: compare 1 to 2 and 1 to 3, intercept is the grand mean => simple coding
c<-contr.treatment(3)
my.coding<-matrix(rep(1/3, 6), ncol=2)
my.simple<-c-my.coding
my.simple
```

```
##            2          3
## 1 -0.3333333 -0.3333333
## 2  0.6666667 -0.3333333
## 3 -0.3333333  0.6666667
```

```
contrasts(df_errors_PWA$session)<-my.simple
levels(df_errors_PWA$session)
```

```
## [1] "1" "2" "3"
```

```
## Center token test
df_errors_PWA$TokenTest_c <-
  scale(df_errors_PWA$TokenTest, center=T, scale=T)
```

```
m1_glmm_PWA_tt <- glmer(error_class ~ PosOr.cont*session*TokenTest_c +
                      (PosOr.cont |subject) +
                      (1|category) ,
                    data =df_errors_PWA, family = "binomial",
                    control=glmerControl(optimizer = "bobyqa",
                                   optCtrl = list(maxfun = 2e5)))
didLmerConverge(m1_glmm_PWA_tt)
```

```
##  The relative maximum gradient of 0.00000284 is less than our 0.001 criterion.
##  You can safely ignore any warnings about a claimed convergence failure.
```

```
## Warnings can be ignored

# summary(m1_glmm_PWA_tt)
# anova(m1_glmm_PWA_tt)
#
# saveRDS(m1_glmm_PWA_tt,  file = here::here("results", "tables", "CSI_online_aphasia_PWA_lmm_error_plu
# tab_model(m1_glmm_PWA_tt,transform = NULL,
#           show.re.var = F, show.stat = T,show.r2 = F,show.icc = F,
#           title = "GLMM of errors Predicted by Ordinal Position and Session",
#           dv.labels = "Errors",
#           #string.pred = "",
#           string.stat = "z-Value",
#           file = here::here(
#             "results", "tables",
#             "CSI_online_aphasia_PWA_control_glmm_error_TokenTest.html"))
```

Does the model with Token test fit the data better? -> yes!

```r
#summary(m1_lmm_PWA)
anova(m1_error, m1_glmm_PWA_tt)
```

```
## Data: df_errors_PWA
## Models:
## m1_error: error_class ~ PosOr.cont * session + (PosOr.cont | subject) + (1 | category)
## m1_glmm_PWA_tt: error_class ~ PosOr.cont * session * TokenTest_c + (PosOr.cont | subject) + (1 | cate
##                 npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## m1_error          10 5336.7 5405.5 -2658.3   5316.7
## m1_glmm_PWA_tt    16 5326.5 5436.5 -2647.2   5294.5 22.208  6    0.00111 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

AAT: Naming test

RTs Add into converging model

```r
df_RTs_PWA$PosOr.cont <- scale(as.numeric(as.character(df_RTs_PWA$PosOr)),
                                        center = T, scale = F)
table(df_RTs_PWA$PosOr.cont)
```

```
##
##   -1.96955418629938 -0.969554186299384 0.0304458137006161   1.03044581370062
##                1132               1117               1103               1119
##    2.03044581370062
##                1047
```

```r
mean(df_RTs_PWA$PosOr.cont); sd(df_RTs_PWA$PosOr.cont)
```

```
## [1] -0.00000000000000001123241
```

```
## [1] 1.40862
```

```r
# define contrasts of session: compare 1 to 2 and 1 to 3, intercept is the grand mean => simple coding
c<-contr.treatment(3)
my.coding<-matrix(rep(1/3, 6), ncol=2)
my.simple<-c-my.coding
my.simple
```

```
##            2          3
## 1 -0.3333333 -0.3333333
## 2  0.6666667 -0.3333333
## 3 -0.3333333  0.6666667
```

```r
contrasts(df_RTs_PWA$session)<-my.simple
levels(df_RTs_PWA$session)
```

```
## [1] "1" "2" "3"
## Center naming test
df_RTs_PWA$AAT_c <- scale(df_RTs_PWA$AAT, center=T, scale=T)
```

```
m1_lmm_PWA_aat <- lmer(VOTlog ~ PosOr.cont*session*AAT_c +
               (PosOr.cont+session|subject) +
              (1|category),
             data = df_RTs_PWA,
            control=lmerControl(optimizer = "bobyqa",
                                 optCtrl = list(maxfun = 2e5)))
didLmerConverge(m1_lmm_PWA_aat)
```

```
##  The relative maximum gradient of 0.0000013 is less than our 0.001 criterion.
##  You can safely ignore any warnings about a claimed convergence failure.
```

```
## Warnings can be ignored

# summary(m1_lmm_PWA_aat)
# anova(m1_lmm_PWA_aat)
#
# saveRDS(m1_lmm_PWA_aat,  file = here::here("results", "tables", "CSI_online_aphasia_PWA_lmm_VOT_plus-
# tab_model(m1_lmm_PWA_aat,transform = NULL,
#           show.re.var = F, show.stat = T,show.r2 = F,show.icc = F,
#           title = "LMM of VOTs Predicted by Ordinal Position and Session",
#           df.method = "satterthwaite",
#           dv.labels = "Vocal Onset Time (log-transformed)",
#           #string.pred = "",
#           string.stat = "t-Value",
#           file = here::here(
#             "results", "tables",
#             "CSI_online_aphasia_PWA_control_lmm_VOT_AachenAphase.html"))
```

Does the model with Naming test fit the data better? -> yes!

```
#summary(m1_lmm_PWA)
anova(m1_lmm_PWA, m1_lmm_PWA_aat)
```

```
## refitting model(s) with ML (instead of REML)
```

```
## Data: df_RTs_PWA
## Models:
## m1_lmm_PWA: VOTlog ~ PosOr.cont * session + (PosOr.cont + session | subject) + (1 | category)
## m1_lmm_PWA_aat: VOTlog ~ PosOr.cont * session * AAT_c + (PosOr.cont + session | subject) + (1 | categ
##                npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## m1_lmm_PWA       18 1697.5 1816.5 -830.73   1661.5
## m1_lmm_PWA_aat   24 1695.9 1854.6 -823.93   1647.9 13.609  6    0.03432 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Errors   Add into converging model

```
df_errors_PWA$PosOr.cont <- scale(as.numeric(as.character(df_errors_PWA$PosOr)),
                                        center = T, scale = F)

# define contrasts of session: compare 1 to 2 and 1 to 3, intercept is the grand mean => simple coding
c<-contr.treatment(3)
my.coding<-matrix(rep(1/3, 6), ncol=2)
my.simple<-c-my.coding
my.simple
```

```
##            2          3
## 1 -0.3333333 -0.3333333
## 2  0.6666667 -0.3333333
## 3 -0.3333333  0.6666667
```

```
contrasts(df_errors_PWA$session)<-my.simple
levels(df_errors_PWA$session)
```

```
## [1] "1" "2" "3"
```

```
## Center naming test
df_errors_PWA$AAT_c <- scale(df_errors_PWA$AAT, center=T, scale=T)
```

```
m1_glmm_PWA_aat <- glmer(error_class ~ PosOr.cont*session*AAT_c +
                    (PosOr.cont |subject) +
                    (1|category) ,
                  data =df_errors_PWA, family = "binomial",
                  control=glmerControl(optimizer = "bobyqa",
                                 optCtrl = list(maxfun = 2e5)))
didLmerConverge(m1_glmm_PWA_aat)
```

```
##  The relative maximum gradient of 0.00000131 is less than our 0.001 criterion.
##  You can safely ignore any warnings about a claimed convergence failure.
```

```
## Warnings can be ignored

# summary(m1_glmm_PWA_aat)
# anova(m1_glmm_PWA_aat)
#
# saveRDS(m1_glmm_PWA_aat,  file = here::here(
#   "results", "tables", "CSI_online_aphasia_PWA_lmm_error_plus-AAT.RDS"))
# tab_model(m1_glmm_PWA_aat,transform = NULL,
#           show.re.var = F, show.stat = T,show.r2 = F,show.icc = F,
#           title = "GLMM of errors Predicted by Ordinal Position and Session",
#           dv.labels = "Errors",
#           #string.pred = "",
#           string.stat = "z-Value",
#           file = here::here(
#             "results", "tables",
#             "CSI_online_aphasia_PWA_control_glmm_error_AAT.html"))
```

Does the model with Naming subtest fit the data better? -> yes!

```r
#summary(m1_lmm_PWA)
anova(m1_error, m1_glmm_PWA_aat)
```

```
## Data: df_errors_PWA
## Models:
## m1_error: error_class ~ PosOr.cont * session + (PosOr.cont | subject) + (1 | category)
## m1_glmm_PWA_aat: error_class ~ PosOr.cont * session * AAT_c + (PosOr.cont | subject) + (1 | category
##                  npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## m1_error           10 5336.7 5405.5 -2658.3   5316.7
## m1_glmm_PWA_aat    16 5318.5 5428.6 -2643.3   5286.5 30.169  6  0.0000365 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

AAT: Spontaneous speech test

RTs Add into converging model

```r
df_RTs_PWA$PosOr.cont <- scale(as.numeric(as.character(df_RTs_PWA$PosOr)),
                                         center = T, scale = F)
table(df_RTs_PWA$PosOr.cont)
```

```
##
##  -1.96955418629938 -0.969554186299384 0.0304458137006161   1.03044581370062
##               1132               1117               1103               1119
##   2.03044581370062
##               1047
```

```r
mean(df_RTs_PWA$PosOr.cont); sd(df_RTs_PWA$PosOr.cont)
```

```
## [1] -0.00000000000000001123241
```

```
## [1] 1.40862
```

```r
# define contrasts of session: compare 1 to 2 and 1 to 3, intercept is the grand mean => simple coding
c<-contr.treatment(3)
my.coding<-matrix(rep(1/3, 6), ncol=2)
my.simple<-c-my.coding
my.simple
```

```
##            2          3
## 1 -0.3333333 -0.3333333
## 2  0.6666667 -0.3333333
## 3 -0.3333333  0.6666667
```

```r
contrasts(df_RTs_PWA$session)<-my.simple
levels(df_RTs_PWA$session)
```

```
## [1] "1" "2" "3"
## Center spontaneous speech test
df_RTs_PWA$AAT_spontansprache_c <-
  scale(df_RTs_PWA$AAT_spontansprache, center=T, scale=T)
```

```
m1_lmm_PWA_aat_spontansprache <- lmer(
  VOTlog ~ PosOr.cont*session*AAT_spontansprache_c +
               (PosOr.cont+session|subject) +
              (1|category),
             data = df_RTs_PWA,
            control=lmerControl(optimizer = "bobyqa",
                                optCtrl = list(maxfun = 2e5)))
didLmerConverge(m1_lmm_PWA_aat_spontansprache)
```

```
##  The relative maximum gradient of 0.0000112 is less than our 0.001 criterion.
##  You can safely ignore any warnings about a claimed convergence failure.
```

```
## Model converges


# summary(m1_lmm_PWA_aat_spontansprache)
# anova(m1_lmm_PWA_aat_spontansprache)
#
# saveRDS(m1_lmm_PWA_spontansprache,  file = here::here("results", "tables", "CSI_online_aphasia_PWA_lm
# tab_model(m1_lmm_aat_spontansprache,transform = NULL,
#           show.re.var = F, show.stat = T,show.r2 = F,show.icc = F,
#           title = "LMM of VOTs Predicted by Ordinal Position and Session",
#           df.method = "satterthwaite",
#           dv.labels = "Vocal Onset Time (log-transformed)",
#           #string.pred = "",
#           string.stat = "t-Value",
#           file = here::here(
#             "results", "tables", "CSI_online_aphasia_PWA_control_lmm_VOT_AachenAphase_Spontansprache.
```

Does the model with Token test fit the data better? -> no, only a trend.

```
#summary(m1_lmm_PWA)
anova(m1_lmm_PWA, m1_lmm_PWA_aat_spontansprache)
```

```
## refitting model(s) with ML (instead of REML)

## Data: df_RTs_PWA
## Models:
## m1_lmm_PWA: VOTlog ~ PosOr.cont * session + (PosOr.cont + session | subject) + (1 | category)
## m1_lmm_PWA_aat_spontansprache: VOTlog ~ PosOr.cont * session * AAT_spontansprache_c + (PosOr.cont + s
##                               npar    AIC    BIC  logLik deviance  Chisq Df
## m1_lmm_PWA                      18 1697.5 1816.5 -830.73   1661.5
## m1_lmm_PWA_aat_spontansprache   24 1697.7 1856.5 -824.87   1649.7 11.732  6
##                               Pr(>Chisq)
## m1_lmm_PWA
## m1_lmm_PWA_aat_spontansprache    0.06822 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Errors Add into converging model

```
df_errors_PWA$PosOr.cont <-
  scale(as.numeric(as.character(df_errors_PWA$PosOr)),
                                        center = T, scale = F)
# define contrasts of session: compare 1 to 2 and 1 to 3, intercept is the grand mean => simple coding
c<-contr.treatment(3)
my.coding<-matrix(rep(1/3, 6), ncol=2)
my.simple<-c-my.coding
my.simple
```

```
##            2          3
## 1 -0.3333333 -0.3333333
## 2  0.6666667 -0.3333333
## 3 -0.3333333  0.6666667
```

```
contrasts(df_errors_PWA$session)<-my.simple
levels(df_errors_PWA$session)
```

```
## [1] "1" "2" "3"
```

```
## Center spontenous speech test
df_errors_PWA$AAT_spontansprache_c <-
  scale(df_errors_PWA$AAT_spontansprache, center=T, scale=T)
```

```
m1_glmm_PWA_aat_spontan <- glmer(error_class ~ PosOr.cont*session*
                                   AAT_spontansprache_c +
                    (PosOr.cont |subject) +
                    (1|category) ,
                  data =df_errors_PWA, family = "binomial",
                  control=glmerControl(optimizer = "bobyqa",
                                 optCtrl = list(maxfun = 2e5)))
didLmerConverge(m1_glmm_PWA_aat_spontan)
```

```
##  The relative maximum gradient of 0.0000073 is less than our 0.001 criterion.
##  You can safely ignore any warnings about a claimed convergence failure.
```

```
## Warnings can be ignored

# summary(m1_glmm_PWA_aat_spontan)
# anova(m1_glmm_PWA_aat_spontan)
#
# saveRDS(m1_glmm_PWA_aat_spontan,  file = here::here("results", "tables", "CSI_online_aphasia_PWA_lmm_
# tab_model(m1_glmm_PWA_aat_spontan,transform = NULL,
#           show.re.var = F, show.stat = T,show.r2 = F,show.icc = F,
#           title = "GLMM of errors Predicted by Ordinal Position and Session",
#           dv.labels = "Errors",
#           #string.pred = "",
#           string.stat = "z-Value",
#           file = here::here("results", "tables", "CSI_online_aphasia_PWA_control_glmm_error_AAT_spont
```

Does the model with spontenous speech test fit the data better? -> yes!

```
#summary(m1_lmm_PWA)
anova(m1_error, m1_glmm_PWA_aat_spontan)
```

```
## Data: df_errors_PWA
## Models:
## m1_error: error_class ~ PosOr.cont * session + (PosOr.cont | subject) + (1 | category)
## m1_glmm_PWA_aat_spontan: error_class ~ PosOr.cont * session * AAT_spontansprache_c + (PosOr.cont | su
##                          npar    AIC    BIC  logLik deviance  Chisq Df
## m1_error                   10 5336.7 5405.5 -2658.3   5316.7
## m1_glmm_PWA_aat_spontan    16 5335.9 5446.0 -2652.0   5303.9 12.753  6
##                          Pr(>Chisq)
## m1_error
## m1_glmm_PWA_aat_spontan     0.04713 *
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

LEMO

RTs Add into converging model

```
df_RTs_PWA$PosOr.cont <-
  scale(as.numeric(as.character(df_RTs_PWA$PosOr)),
        center = T, scale = F)
table(df_RTs_PWA$PosOr.cont)
```

```
##
##  -1.96955418629938 -0.969554186299384 0.0304458137006161   1.03044581370062
##               1132               1117               1103               1119
##   2.03044581370062
##               1047
```

```
mean(df_RTs_PWA$PosOr.cont); sd(df_RTs_PWA$PosOr.cont)
```

```
## [1] -0.0000000000000001123241
```

```
## [1] 1.40862
```

```
# define contrasts of session: compare 1 to 2 and 1 to 3, intercept is the grand mean => simple coding
c<-contr.treatment(3)
my.coding<-matrix(rep(1/3, 6), ncol=2)
my.simple<-c-my.coding
my.simple
```

```
##            2          3
## 1 -0.3333333 -0.3333333
## 2  0.6666667 -0.3333333
## 3 -0.3333333  0.6666667
```

```r
contrasts(df_RTs_PWA$session)<-my.simple
levels(df_RTs_PWA$session)
```

```
## [1] "1" "2" "3"
```

```r
## Center LEMO test
df_RTs_PWA$LEMO_c <- scale(df_RTs_PWA$LEMO, center=T, scale=T)
```

```r
m1_lmm_PWA_lemo <- lmer(VOTlog ~ PosOr.cont*session*LEMO_c +
               (PosOr.cont+session|subject) +
              (1|category),
             data = df_RTs_PWA,
            control=lmerControl(optimizer = "bobyqa",
                                 optCtrl = list(maxfun = 2e5)))
didLmerConverge(m1_lmm_PWA_lemo)
```

```
##  The relative maximum gradient of 0.00000141 is less than our 0.001 criterion.
##  You can safely ignore any warnings about a claimed convergence failure.
```

```r
## Warnings can be ignored

# summary(m1_lmm_PWA_lemo)
# anova(m1_lmm_PWA_lemo)
#
# saveRDS(m1_lmm_PWA_lemo,  file = here::here(
#   "results", "tables", "CSI_online_aphasia_PWA_lmm_VOT_plus-LEMO.RDS"))
# tab_model(m1_lmm_PWA_lemo,transform = NULL,
#           show.re.var = F, show.stat = T,show.r2 = F,show.icc = F,
#           title = "LMM of VOTs Predicted by Ordinal Position and Session",
#           df.method = "satterthwaite",
#           dv.labels = "Vocal Onset Time (log-transformed)",
#           #string.pred = "",
#           string.stat = "t-Value",
#           file = here::here(
#             "results", "tables",
#             "CSI_online_aphasia_PWA_control_lmm_VOT_LEMO.html"))
```

Does the model with LEMO fit the data better? -> no! (but a trend)

```r
#summary(m1_lmm_PWA)
anova(m1_lmm_PWA, m1_lmm_PWA_lemo)
```

```
## refitting model(s) with ML (instead of REML)

## Data: df_RTs_PWA
## Models:
## m1_lmm_PWA: VOTlog ~ PosOr.cont * session + (PosOr.cont + session | subject) + (1 | category)
## m1_lmm_PWA_lemo: VOTlog ~ PosOr.cont * session * LEMO_c + (PosOr.cont + session | subject) + (1 | ca
##                   npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## m1_lmm_PWA          18 1697.5 1816.5 -830.73   1661.5
## m1_lmm_PWA_lemo     24 1698.7 1857.5 -825.34   1650.7 10.776  6    0.09554 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Errors Add into converging model

```
df_errors_PWA$PosOr.cont <- scale(as.numeric(as.character(df_errors_PWA$PosOr)),
                                        center = T, scale = F)

# define contrasts of session: compare 1 to 2 and 1 to 3, intercept is the grand mean => simple coding
c<-contr.treatment(3)
my.coding<-matrix(rep(1/3, 6), ncol=2)
my.simple<-c-my.coding
my.simple
```

```
##            2          3
## 1 -0.3333333 -0.3333333
## 2  0.6666667 -0.3333333
## 3 -0.3333333  0.6666667
```

```
contrasts(df_errors_PWA$session)<-my.simple
levels(df_errors_PWA$session)
```

```
## [1] "1" "2" "3"
```

```
## Center LEMO test
df_errors_PWA$LEMO_c <- scale(df_errors_PWA$LEMO, center=T, scale=T)
```

Compute further contrasts

```
m1_glmm_PWA_lemo <- glmer(error_class ~ PosOr.cont*session*LEMO_c +
                    (PosOr.cont |subject) +
                    (1|category) ,
                  data =df_errors_PWA, family = "binomial",
                  control=glmerControl(optimizer = "bobyqa",
                                 optCtrl = list(maxfun = 2e5)))
didLmerConverge(m1_glmm_PWA_lemo)
```

```
##  The relative maximum gradient of 0.00000715 is less than our 0.001 criterion.
##  You can safely ignore any warnings about a claimed convergence failure.
```

```
## Warnings can be ignored

# summary(m1_glmm_PWA_lemo)
# anova(m1_glmm_PWA_lemo)
#
# saveRDS(m1_glmm_PWA_lemo,  file = here::here("results", "tables", "CSI_online_aphasia_PWA_lmm_error_p
# tab_model(m1_glmm_PWA_lemo,transform = NULL,
#           show.re.var = F, show.stat = T,show.r2 = F,show.icc = F,
#           title = "GLMM of errors Predicted by Ordinal Position and Session",
#           dv.labels = "Errors",
#           #string.pred = "",
#           string.stat = "z-Value",
#           file = here::here(
#             "results", "tables",
#             "CSI_online_aphasia_PWA_control_glmm_error_Lemo.html"))
```

Does the model with LEMO test fit the data better? -> no

```
#summary(m1_lmm_PWA)
anova(m1_error, m1_glmm_PWA_lemo)
```

```
## Data: df_errors_PWA
## Models:
## m1_error: error_class ~ PosOr.cont * session + (PosOr.cont | subject) + (1 | category)
## m1_glmm_PWA_lemo: error_class ~ PosOr.cont * session * LEMO_c + (PosOr.cont | subject) + (1 | catego
##                  npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## m1_error           10 5336.7 5405.5 -2658.3   5316.7
## m1_glmm_PWA_lemo   16 5340.5 5450.6 -2654.3   5308.5 8.1693  6      0.226
```

Time since stroke

RTs Add into converging model

```
df_RTs_PWA$PosOr.cont <- scale(as.numeric(as.character(df_RTs_PWA$PosOr)),
                                        center = T, scale = F)
table(df_RTs_PWA$PosOr.cont)
```

```
##
##  -1.96955418629938 -0.969554186299384 0.0304458137006161   1.03044581370062
##               1132               1117               1103               1119
##   2.03044581370062
##               1047
```

```
mean(df_RTs_PWA$PosOr.cont)
```

```
## [1] -0.00000000000000001123241
```

```
sd(df_RTs_PWA$PosOr.cont)
```

```
## [1] 1.40862
```

```
# define contrasts of session: compare 1 to 2 and 1 to 3, intercept is the grand mean => simple coding
c<-contr.treatment(3)
my.coding<-matrix(rep(1/3, 6), ncol=2)
my.simple<-c-my.coding
my.simple
```

```
##            2          3
## 1 -0.3333333 -0.3333333
## 2  0.6666667 -0.3333333
## 3 -0.3333333  0.6666667
```

```
contrasts(df_RTs_PWA$session)<-my.simple
levels(df_RTs_PWA$session)
```

```
## [1] "1" "2" "3"
## Center months post onset
df_RTs_PWA$months_post_onset_c <- scale(df_RTs_PWA$months_post_onset,
                                        center=T, scale=T)
```

```
m1_lmm_PWA_MPO <- lmer(VOTlog ~ PosOr.cont*session*months_post_onset_c +
               (PosOr.cont+session|subject) +
              (1|category),
             data = df_RTs_PWA,
            control=lmerControl(optimizer = "bobyqa",
                                 optCtrl = list(maxfun = 2e5)))
didLmerConverge(m1_lmm_PWA_MPO)
```

```
##  The relative maximum gradient of 0.00000365 is less than our 0.001 criterion.
##  You can safely ignore any warnings about a claimed convergence failure.
```

```
## Warnings can be ignored

# summary(m1_lmm_PWA_MPO)
# anova(m1_lmm_PWA_MPO)
#
# saveRDS(m1_lmm_PWA_MPO,  file = here::here(
#   "results", "tables", "CSI_online_aphasia_PWA_lmm_VOT_plus-MPO.RDS"))
# tab_model(m1_lmm_PWA_MPO,transform = NULL,
#           show.re.var = F, show.stat = T,show.r2 = F,show.icc = F,
#           title = "LMM of VOTs Predicted by Ordinal Position and Session",
#           df.method = "satterthwaite",
#           dv.labels = "Vocal Onset Time (log-transformed)",
#           #string.pred = "",
#           string.stat = "t-Value",
#           file = here::here(
#             "results", "tables",
#             "CSI_online_aphasia_PWA_control_lmm_VOT_MPO.html"))
```

Does the model with MPO fit the data better? -> no!

```
#summary(m1_lmm_PWA)
anova(m1_lmm_PWA, m1_lmm_PWA_MPO)
```

```
## refitting model(s) with ML (instead of REML)
```

```
## Data: df_RTs_PWA
## Models:
## m1_lmm_PWA: VOTlog ~ PosOr.cont * session + (PosOr.cont + session | subject) + (1 | category)
## m1_lmm_PWA_MPO: VOTlog ~ PosOr.cont * session * months_post_onset_c + (PosOr.cont + session | subjec
##                npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## m1_lmm_PWA       18 1697.5 1816.5 -830.73   1661.5
## m1_lmm_PWA_MPO   24 1703.4 1862.2 -827.69   1655.4 6.0782  6     0.4145
```

Errors Add into converging model

```
df_errors_PWA$PosOr.cont <-
  scale(as.numeric(as.character(df_errors_PWA$PosOr)),
                                        center = T, scale = F)

# define contrasts of session: compare 1 to 2 and 1 to 3, intercept is the grand mean => simple coding
c<-contr.treatment(3)
my.coding<-matrix(rep(1/3, 6), ncol=2)
my.simple<-c-my.coding
my.simple
```

```
##            2          3
## 1 -0.3333333 -0.3333333
## 2  0.6666667 -0.3333333
## 3 -0.3333333  0.6666667
```

```
contrasts(df_errors_PWA$session)<-my.simple
levels(df_errors_PWA$session)
```

```
## [1] "1" "2" "3"
```

```
## Center months post onse
df_errors_PWA$MPO_c <-
  scale(df_errors_PWA$months_post_onset, center=T, scale=T)
```

```
m1_glmm_PWA_MPO <- glmer(error_class ~ PosOr.cont*session*MPO_c +
                    (PosOr.cont |subject) +
                    (1|category) ,
                  data =df_errors_PWA, family = "binomial",
                  control=glmerControl(optimizer = "bobyqa",
                                 optCtrl = list(maxfun = 2e5)))
didLmerConverge(m1_glmm_PWA_MPO)
```

```
##  The relative maximum gradient of 0.000006 is less than our 0.001 criterion.
##  You can safely ignore any warnings about a claimed convergence failure.
```

```
## Warnings can be ignored

# summary(m1_glmm_PWA_MPO)
# anova(m1_glmm_PWA_MPO)
#
# saveRDS(m1_glmm_PWA_MPO,  file = here::here(
#   "results", "tables", "CSI_online_aphasia_PWA_lmm_error_plus-MPO.RDS"))
# tab_model(m1_glmm_PWA_MPO,transform = NULL,
#           show.re.var = F, show.stat = T,show.r2 = F,show.icc = F,
#           title =
#             "GLMM of errors Predicted by Ordinal Position and Session",
#           #df.method = "satterthwaite",
#           dv.labels = "Errors",
#           #string.pred = "",
#           string.stat = "z-Value",
#           file = here::here(
#             "results", "tables",
#             "CSI_online_aphasia_PWA_control_glmm_error_MPO.html"))
```

Does the model with MPO fit the data better? -> no!

```
#summary(m1_lmm_PWA)
anova(m1_error, m1_glmm_PWA_MPO)
```

```
## Data: df_errors_PWA
## Models:
## m1_error: error_class ~ PosOr.cont * session + (PosOr.cont | subject) + (1 | category)
## m1_glmm_PWA_MPO: error_class ~ PosOr.cont * session * MPO_c + (PosOr.cont | subject) + (1 | category
##                 npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## m1_error          10 5336.7 5405.5 -2658.3   5316.7
## m1_glmm_PWA_MPO   16 5341.8 5451.9 -2654.9   5309.8 6.8914  6      0.331
```

Lesion size

```


df_RTs_PWA$LHoverall_c <- scale(df_RTs_PWA$LHoverall, center=T, scale=F)

m1_lmm_PWA_LH <- lmer(VOTlog ~ PosOr.cont*session*LHoverall_c +
               (PosOr.cont+session|subject) +
              (1|category),
             data = df_RTs_PWA,
            control=lmerControl(optimizer = "bobyqa",
                                optCtrl = list(maxfun = 2e5)))
didLmerConverge(m1_lmm_PWA_LH)
```

RTs

```
##  The relative maximum gradient of 0.00000407 is less than our 0.001 criterion.
##  You can safely ignore any warnings about a claimed convergence failure.
```

```
## Warnings can be ignored

# summary(m1_lmm_PWA_LH)
# anova(m1_lmm_PWA_LH)
#
# saveRDS(m1_lmm_PWA_LH,  file = here::here("results", "tables",
#       "CSI_online_aphasia_PWA_lmm_VOT_plus-Lesion-size.RDS"))
# tab_model(m1_lmm_PWA_LH,transform = NULL,
#           show.re.var = F, show.stat = T,show.r2 = F,show.icc = F,
#           title = "LMM of VOTs Predicted by Ordinal Position and Session",
#           df.method = "satterthwaite",
#           dv.labels = "Vocal Onset Time (log-transformed)",
#           #string.pred = "",
#           string.stat = "t-Value",
#           file = here::here(
#             "results", "tables",
#             "CSI_online_aphasia_PWA_control_lmm_VOT_Lesion-size.html"))
```

Does the model with lesion size fit the data better? -> no!

```r
#summary(m1_lmm_PWA)
anova(m1_lmm_PWA, m1_lmm_PWA_LH)
```

```
## refitting model(s) with ML (instead of REML)

## Data: df_RTs_PWA
## Models:
## m1_lmm_PWA: VOTlog ~ PosOr.cont * session + (PosOr.cont + session | subject) + (1 | category)
## m1_lmm_PWA_LH: VOTlog ~ PosOr.cont * session * LHoverall_c + (PosOr.cont + session | subject) + (1 |
##               npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## m1_lmm_PWA      18 1697.5 1816.5 -830.73   1661.5
## m1_lmm_PWA_LH   24 1704.3 1863.1 -828.15   1656.3 5.1739  6     0.5217
```

Errors Add into converging model

```r
df_errors_PWA$PosOr.cont <- scale(as.numeric(as.character(df_errors_PWA$PosOr)),
                                         center = T, scale = F)
# define contrasts of session: compare 1 to 2 and 1 to 3, intercept is the grand mean => simple coding
c<-contr.treatment(3)
my.coding<-matrix(rep(1/3, 6), ncol=2)
my.simple<-c-my.coding
my.simple
```

```
##            2          3
## 1 -0.3333333 -0.3333333
## 2  0.6666667 -0.3333333
## 3 -0.3333333  0.6666667
```

```r
contrasts(df_errors_PWA$session)<-my.simple
levels(df_errors_PWA$session)
```

```
## [1] "1" "2" "3"
```

```r
## Center lesion size
df_errors_PWA$LHoverall_c <- scale(df_errors_PWA$LHoverall, center=T, scale=T)
```

```r
m1_glmm_PWA_LH <- glmer(error_class ~ PosOr.cont*session*LHoverall_c +
                     (PosOr.cont |subject) +
                     (1|category) ,
                   data =df_errors_PWA, family = "binomial",
                   control=glmerControl(optimizer = "bobyqa",
                                 optCtrl = list(maxfun = 2e5)))
didLmerConverge(m1_glmm_PWA_LH)
```

```
##  The relative maximum gradient of 0.00000659 is less than our 0.001 criterion.
##  You can safely ignore any warnings about a claimed convergence failure.
## Warnings can be ignored

# summary(m1_glmm_PWA_LH)
# anova(m1_glmm_PWA_LH)
#
# saveRDS(m1_glmm_PWA_LH,  file = here::here("results", "tables", "CSI_online_aphasia_PWA_lmm_error_plu
# tab_model(m1_glmm_PWA_LH,transform = NULL,
#           show.re.var = F, show.stat = T,show.r2 = F,show.icc = F,
#           title = "GLMM of errors Predicted by Ordinal Position and Session",
#           dv.labels = "Errors",
#           #string.pred = "",
#           string.stat = "z-Value",
#           file = here::here(
#             "results", "tables",
#             "CSI_online_aphasia_PWA_control_glmm_error_LH.html"))
```

Does the model with Token test fit the data better? -> yes!

```
#summary(m1_lmm_PWA)
anova(m1_error, m1_glmm_PWA_LH)
```

```
## Data: df_errors_PWA
## Models:
## m1_error: error_class ~ PosOr.cont * session + (PosOr.cont | subject) + (1 | category)
## m1_glmm_PWA_LH: error_class ~ PosOr.cont * session * LHoverall_c + (PosOr.cont | subject) + (1 | cate
##                 npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## m1_error          10 5336.7 5405.5 -2658.3   5316.7
## m1_glmm_PWA_LH    16 5333.9 5444.0 -2651.0   5301.9 14.746  6    0.02232 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Add covariates explaining significant variance in a single model

RTs Add Token Test and Naming Test into converging model.

```
cor.test(df_RTs_PWA$AAT_c, df_RTs_PWA$TokenTest_c)
```

```
##
##  Pearson's product-moment correlation
##
## data:  df_RTs_PWA$AAT_c and df_RTs_PWA$TokenTest_c
## t = 110.24, df = 5516, p-value < 0.0000000000000022
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.8209312 0.8374132
## sample estimates:
##       cor
## 0.8293525
```

```
df_RTs_PWA$PosOr.cont <- scale(as.numeric(as.character(df_RTs_PWA$PosOr)),
                                        center = T, scale = F)
# define contrasts of session: compare 1 to 2 and 1 to 3, intercept is the grand mean => simple coding
c<-contr.treatment(3)
my.coding<-matrix(rep(1/3, 6), ncol=2)
my.simple<-c-my.coding
my.simple
```

```
##            2          3
## 1 -0.3333333 -0.3333333
## 2  0.6666667 -0.3333333
## 3 -0.3333333  0.6666667
```

```
contrasts(df_RTs_PWA$session)<-my.simple
levels(df_RTs_PWA$session)
```

```
## [1] "1" "2" "3"
```

```
## Center token test and Naming test
df_RTs_PWA$TokenTest_c <- scale(df_RTs_PWA$TokenTest, center=T, scale=T)
df_RTs_PWA$AAT_c <- scale(df_RTs_PWA$AAT, center=T, scale=T)


m1_lmm_PWA_tt_aat <- lmer(VOTlog ~ PosOr.cont*session*(TokenTest_c*AAT_c)+
               (PosOr.cont+session|subject) +
              (1|category),
             data = df_RTs_PWA,
            control=lmerControl(optimizer = "bobyqa",
                                 optCtrl = list(maxfun = 2e5)))
didLmerConverge(m1_lmm_PWA_tt_aat)
```

```
##  The relative maximum gradient of 0.000124 is less than our 0.001 criterion.
##  You can safely ignore any warnings about a claimed convergence failure.
```

```
## Warnings can be ignored

summary(m1_lmm_PWA_tt_aat)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite’s method [
## lmerModLmerTest]
## Formula: VOTlog ~ PosOr.cont * session * (TokenTest_c * AAT_c) + (PosOr.cont +
##     session | subject) + (1 | category)
##    Data: df_RTs_PWA
## Control: lmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 200000))
##
## REML criterion at convergence: 1810.1
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.4968 -0.6792 -0.1915  0.5042  5.0319
##
```

```
## Random effects:
##  Groups   Name        Variance  Std.Dev. Corr
##  category (Intercept) 0.0092600 0.09623
##  subject  (Intercept) 0.0409836 0.20244
##           PosOr.cont  0.0001357 0.01165   0.27
##           session2    0.0005339 0.02311  -0.55  0.39
##           session3    0.0029467 0.05428  -0.61  0.09  0.05
##  Residual             0.0761189 0.27590
## Number of obs: 5518, groups:  category, 24; subject, 20
##
## Fixed effects:
##                                          Estimate  Std. Error          df
## (Intercept)                              7.194167    0.056486   20.594370
## PosOr.cont                               0.016263    0.004697   21.433722
## session2                                -0.071053    0.013778   17.687842
## session3                                -0.078038    0.018663   20.704486
## TokenTest_c                             -0.026982    0.073995   16.095249
## AAT_c                                   -0.198863    0.091339   16.428960
## PosOr.cont:session2                      0.015077    0.008743 5424.195608
## PosOr.cont:session3                      0.008715    0.008385 5431.728865
## TokenTest_c:AAT_c                       -0.048308    0.030970   16.515486
## PosOr.cont:TokenTest_c                  -0.001060    0.006471   21.750345
## PosOr.cont:AAT_c                         0.006637    0.009542   33.501731
## session2:TokenTest_c                    -0.014041    0.018817   18.561124
## session3:TokenTest_c                     0.003742    0.025955   20.863974
## session2:AAT_c                          -0.026912    0.028832   25.851015
## session3:AAT_c                          -0.010841    0.036113   30.139853
## PosOr.cont:TokenTest_c:AAT_c             0.004082    0.003375   39.082828
## session2:TokenTest_c:AAT_c              -0.019027    0.010650   33.041850
## session3:TokenTest_c:AAT_c              -0.015318    0.012639   34.144950
## PosOr.cont:session2:TokenTest_c         -0.014488    0.012062 5423.457225
## PosOr.cont:session3:TokenTest_c         -0.012646    0.012000 5424.747740
## PosOr.cont:session2:AAT_c               -0.005411    0.019067 5418.272739
## PosOr.cont:session3:AAT_c               -0.006923    0.018310 5438.091709
## PosOr.cont:session2:TokenTest_c:AAT_c   -0.004311    0.006971 5070.771048
## PosOr.cont:session3:TokenTest_c:AAT_c   -0.004446    0.006325 5375.875481
##                                       t value             Pr(>|t|)
## (Intercept)                           127.361 < 0.0000000000000002 ***
## PosOr.cont                              3.462             0.002280 **
## session2                               -5.157            0.0000698 ***
## session3                               -4.181             0.000432 ***
## TokenTest_c                            -0.365             0.720121
## AAT_c                                  -2.177             0.044363 *
## PosOr.cont:session2                     1.724             0.084677 .
## PosOr.cont:session3                     1.039             0.298667
## TokenTest_c:AAT_c                      -1.560             0.137757
## PosOr.cont:TokenTest_c                 -0.164             0.871368
## PosOr.cont:AAT_c                        0.696             0.491501
## session2:TokenTest_c                   -0.746             0.464916
## session3:TokenTest_c                    0.144             0.886732
## session2:AAT_c                         -0.933             0.359248
## session3:AAT_c                         -0.300             0.766084
## PosOr.cont:TokenTest_c:AAT_c            1.210             0.233685
## session2:TokenTest_c:AAT_c             -1.787             0.083187 .
## session3:TokenTest_c:AAT_c             -1.212            0.233862
## PosOr.cont:session2:TokenTest_c        -1.201            0.229720
## PosOr.cont:session3:TokenTest_c        -1.054            0.292032
## PosOr.cont:session2:AAT_c              -0.284            0.776599
## PosOr.cont:session3:AAT_c              -0.378            0.705356
## PosOr.cont:session2:TokenTest_c:AAT_c  -0.618            0.536333
## PosOr.cont:session3:TokenTest_c:AAT_c  -0.703            0.482137
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

```
##
## Correlation matrix not shown by default, as p = 24 > 12.
## Use print(x, correlation=TRUE)  or
##     vcov(x)        if you need it
```

```
anova(m1_lmm_PWA_tt_aat)
```

```
## Type III Analysis of Variance Table with Satterthwaite’s method
##                                       Sum Sq Mean Sq NumDF  DenDF F value
## PosOr.cont                           0.91237 0.91237     1   21.4 11.9861
## session                              2.58647 1.29323     2   19.7 16.9897
## TokenTest_c                          0.01012 0.01012     1   16.1  0.1330
## AAT_c                                0.36082 0.36082     1   16.4  4.7403
## PosOr.cont:session                   0.22815 0.11408     2 5433.5  1.4986
## TokenTest_c:AAT_c                    0.18520 0.18520     1   16.5  2.4331
## PosOr.cont:TokenTest_c               0.00204 0.00204     1   21.8  0.0268
## PosOr.cont:AAT_c                     0.03683 0.03683     1   33.5  0.4838
## session:TokenTest_c                  0.05496 0.02748     2   19.8  0.3610
## session:AAT_c                        0.06634 0.03317     2   29.6  0.4358
## PosOr.cont:TokenTest_c:AAT_c         0.11138 0.11138     1   39.1  1.4632
## session:TokenTest_c:AAT_c            0.27243 0.13622     2   36.2  1.7895
## PosOr.cont:session:TokenTest_c       0.12624 0.06312     2 5424.5  0.8292
## PosOr.cont:session:AAT_c             0.01157 0.00578     2 5418.3  0.0760
## PosOr.cont:session:TokenTest_c:AAT_c 0.04416 0.02208     2 5239.2  0.2901
##                                          Pr(>F)
## PosOr.cont                              0.00228 **
## session                              0.00005137 ***
## TokenTest_c                             0.72012
## AAT_c                                   0.04436 *
## PosOr.cont:session                      0.22352
## TokenTest_c:AAT_c                       0.13776
## PosOr.cont:TokenTest_c                  0.87137
## PosOr.cont:AAT_c                        0.49150
## session:TokenTest_c                     0.70148
## session:AAT_c                           0.65084
## PosOr.cont:TokenTest_c:AAT_c            0.23369
## session:TokenTest_c:AAT_c               0.18151
## PosOr.cont:session:TokenTest_c          0.43645
## PosOr.cont:session:AAT_c                0.92684
## PosOr.cont:session:TokenTest_c:AAT_c    0.74821
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

```
performance::model_performance(m1_lmm_PWA_tt_aat)
```

```
## # Indices of model performance
##
## AIC      |     AICc |      BIC | R2 (cond.) | R2 (marg.) |   ICC |  RMSE | Sigma
## ---------------------------------------------------------------------------------
## 1882.101 | 1882.587 | 2120.269 |      0.472 |      0.117 | 0.401 | 0.274 | 0.276
```

```
saveRDS(m1_lmm_PWA_tt_aat,  file = here::here("results", "tables",
 "CSI_online_aphasia_PWA_lmm_VOT_plus-TokenTest-plus-AachenAphhasie.RDS"))
tab_model(m1_lmm_PWA_tt_aat,transform = NULL,
          show.re.var = F, show.stat = T,show.r2 = F,show.icc = F,
          title = "LMM of VOTs Predicted by Ordinal Position and Session",
          df.method = "satterthwaite",
          dv.labels = "Vocal Onset Time (log-transformed)",
          #string.pred = "",
          string.stat = "t-Value",
          file = here::here(
            "results", "tables",
    "CSI_online_aphasia_PWA_control_lmm_VOT_TokenTestAachenAphase.html"))
```

LMM of VOTs Predicted by Ordinal Position and Session

Vocal Onset Time (log-transformed)

Predictors

Estimates

CI

t-Value

p

(Intercept)

7.19

7.08 – 7.31

127.36

<0.001

PosOr cont

0.02

0.01 – 0.03

3.46

0.002

session [2]

-0.07

-0.10 – -0.04

-5.16

<0.001

session [3]

-0.08

-0.12 – -0.04

-4.18

<0.001

TokenTest c

-0.03

-0.18 – 0.13

-0.36

0.720

AAT c

-0.20

-0.39 – -0.01

-2.18

0.044

PosOr cont × session [2]

0.02

-0.00 – 0.03

1.72

0.085

PosOr cont × session [3]

0.01

-0.01 – 0.03

1.04

0.299

TokenTest c × AAT c

-0.05

-0.11 – 0.02

-1.56

0.138

PosOr cont × TokenTest c

-0.00

-0.01 – 0.01

-0.16

0.871

PosOr cont × AAT c

0.01

-0.01 – 0.03

0.70

0.492

session [2] × TokenTest c

-0.01

-0.05 – 0.03

-0.75

0.465

session [3] × TokenTest c

0.00

-0.05 – 0.06

0.14

0.887

session [2] × AAT c

-0.03

-0.09 – 0.03

-0.93

0.359

session [3] × AAT c

-0.01

-0.08 – 0.06

-0.30

0.766

(PosOr cont × TokenTestc) × AAT c

0.00

-0.00 – 0.01

1.21

0.234

(session [2] × TokenTestc) × AAT c

-0.02

-0.04 – 0.00

-1.79

0.083

(session [3] × TokenTestc) × AAT c

-0.02

-0.04 – 0.01

-1.21

0.234

(PosOr cont × session[2]) × TokenTest c

-0.01

-0.04 – 0.01

-1.20

0.230

(PosOr cont × session[3]) × TokenTest c

-0.01

-0.04 – 0.01

-1.05

0.292

(PosOr cont × session[2]) × AAT c

-0.01

-0.04 – 0.03

-0.28

0.777

(PosOr cont × session[3]) × AAT c

-0.01

-0.04 – 0.03

-0.38

0.705

(PosOr cont × session [2]× TokenTest c) × AAT c

-0.00

-0.02 – 0.01

-0.62

0.536

(PosOr cont × session [3]× TokenTest c) × AAT c

-0.00

-0.02 – 0.01

-0.70

0.482

N subject

20

N category

24

Observations

5518

Main effect of Naming Test. Trend between Ordinal position and Session 2-1 x Token Test x Naming and Ordinal position x Session2-1

Does the model with Naming and Token test fit the data better? -> no (Trend)! Does the model with Naming and Token test fit the data better than the model only with Naming? -> no! => The variance explained by the token Test is also explained by the naming test Does the model with Naming test fit the data better than the main model? -> yes!

```
#summary(m1_lmm_PWA)
anova(m1_lmm_PWA, m1_lmm_PWA_tt_aat)
```

```
## refitting model(s) with ML (instead of REML)

## Data: df_RTs_PWA
## Models:
## m1_lmm_PWA: VOTlog ~ PosOr.cont * session + (PosOr.cont + session | subject) + (1 | category)
## m1_lmm_PWA_tt_aat: VOTlog ~ PosOr.cont * session * (TokenTest_c * AAT_c) + (PosOr.cont + session | su
##                     npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## m1_lmm_PWA          18 1697.5 1816.5 -830.73   1661.5
## m1_lmm_PWA_tt_aat   36 1706.5 1944.6 -817.23   1634.5 27.003 18    0.07893 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(m1_lmm_PWA_aat, m1_lmm_PWA_tt_aat)
```

```
## refitting model(s) with ML (instead of REML)

## Data: df_RTs_PWA
## Models:
## m1_lmm_PWA_aat: VOTlog ~ PosOr.cont * session * AAT_c + (PosOr.cont + session | subject) + (1 | categ
## m1_lmm_PWA_tt_aat: VOTlog ~ PosOr.cont * session * (TokenTest_c * AAT_c) + (PosOr.cont + session | su
##                     npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## m1_lmm_PWA_aat      24 1695.9 1854.6 -823.93   1647.9
## m1_lmm_PWA_tt_aat   36 1706.5 1944.6 -817.23   1634.5 13.394 12      0.341
```

```
anova(m1_lmm_PWA, m1_lmm_PWA_aat)
```

```
## refitting model(s) with ML (instead of REML)

## Data: df_RTs_PWA
## Models:
## m1_lmm_PWA: VOTlog ~ PosOr.cont * session + (PosOr.cont + session | subject) + (1 | category)
## m1_lmm_PWA_aat: VOTlog ~ PosOr.cont * session * AAT_c + (PosOr.cont + session | subject) + (1 | categ
##                  npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## m1_lmm_PWA       18 1697.5 1816.5 -830.73   1661.5
## m1_lmm_PWA_aat   24 1695.9 1854.6 -823.93   1647.9 13.609  6    0.03432 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
(means_final<- df_RTs_PWA %>%
   filter(category != "Filler") %>%
   Rmisc::summarySEwithin(.,"VOT",idvar = "category",
                          withinvars = c("session", "PosOr"),
                          betweenvars = c("subject", "AAT"), na.rm = T))
```

Plot Naming test

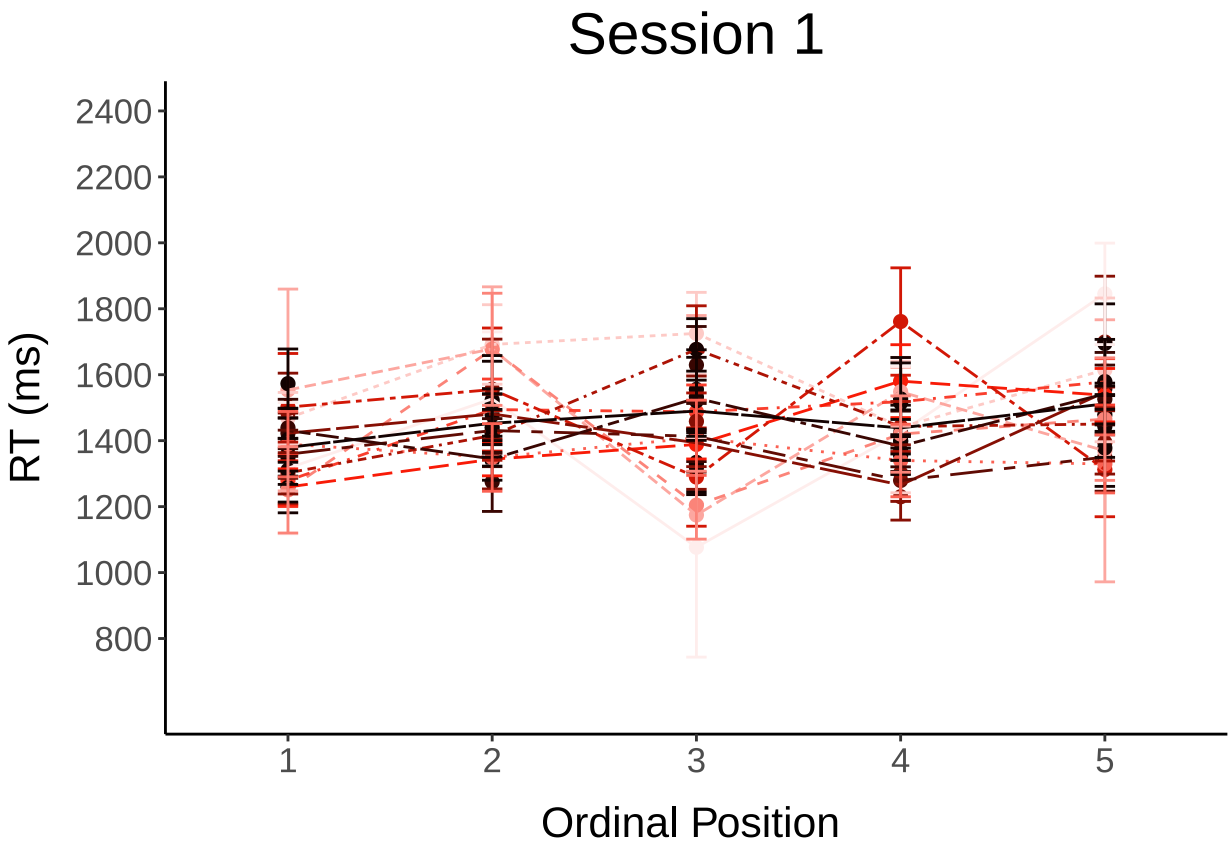
```
## Automatically converting the following non-factors to factors: AAT
```

```
## Warning in qt(conf.interval/2 + 0.5, datac$N - 1): NaNs produced
```

```
##     subject AAT session PosOr  N       VOT       sd        se        ci
## 1       101  98       1     1 21 1358.8342 334.4313  72.97888 152.23128
## 2       101  98       1     2 21 1430.7892 502.1845 109.58564 228.59165
## 3       101  98       1     3 23 1413.1229 519.2279 108.26651 224.53099
## 4       101  98       1     4 23 1279.2801 295.1891  61.55118 127.64933
## 5       101  98       1     5 23 1350.1123 495.6944 103.35943 214.35434
## 6       101  98       2     1 21 1236.0914 302.2888  65.96482 137.60020
## 7       101  98       2     2 24 1326.5282 394.2046  80.46668 166.45802
## 8       101  98       2     3 24 1264.7366 358.0604  73.08876 151.19563
## 9       101  98       2     4 24 1265.4449 309.4372  63.16360 130.66386
## 10      101  98       2     5 23 1350.5869 304.4667  63.48570 131.66127
## 11      101  98       3     1 22 1416.6396 352.1802  75.08507 156.14795
## 12      101  98       3     2 24 1383.0282 295.8206  60.38413 124.91408
## 13      101  98       3     3 24 1395.3616 291.3443  59.47040 123.02390
## 14      101  98       3     4 24 1375.4866 306.0538  62.47298 129.23520
## 15      101  98       3     5 24 1448.2782 295.9528  60.41111 124.96990
## 16      102  99       1     1 23 1422.7249 225.5446  47.02930  97.53280
## 17      102  99       1     2 23 1414.3154 251.7954  52.50297 108.88450
## 18      102  99       1     3 24 1428.2095 448.5557  91.56105 189.40846
## 19      102  99       1     4 24 1369.3762 238.9167  48.76867 100.88568
## 20      102  99       1     5 24 1557.6679 538.5309 109.92716 227.40166
## 21      102  99       2     1 23 1362.6814 320.0028  66.72519 138.37957
## 22      102  99       2     2 23 1352.6659 142.4267  29.69803  61.58994
## 23      102  99       2     3 23 1311.2218 176.2680  36.75441  76.22399
## 24      102  99       2     4 24 1331.5429 278.0828  56.76341 117.42406
## 25      102  99       2     5 24 1335.4179 226.0472  46.14170  95.45138
## 26      102  99       3     1 22 1306.8330 125.3060  26.71533  55.55758
## 27      102  99       3     2 24 1313.3345 177.1961  36.16999  74.82333
## 28      102  99       3     3 24 1241.7095 149.4035  30.49685  63.08755
## 29      102  99       3     4 23 1221.7000 107.7878  22.47530  46.61092
## 30      102  99       3     5 23 1317.8740 231.2996  48.22929 100.02143
## 31      103  97       1     1 21 1361.0269 562.2646 122.69620 255.93980
## 32      103  97       1     2 16 1405.8564 609.2341 152.30854 324.63796
## 33      103  97       1     3 14 1328.5140 284.6623  76.07919 164.35911
## 34      103  97       1     4 13 1229.9043 255.3610  70.82440 154.31311
## 35      103  97       1     5 14 1699.2952 746.5394 199.52104 431.03900
## 36      103  97       2     1 17 1059.3012 243.1891  58.98202 125.03629
## 37      103  97       2     2 20 1289.7725 450.1449 100.65546 210.67430
## 38      103  97       2     3 15 1522.8085 569.9557 147.16192 315.63093
## 39      103  97        2     4 17 1443.1595  717.8321 174.09986  369.07521
## 40      103  97        2     5 15 1386.7726  328.5558  84.83275  181.94816
## 41      103  97        3     1 18 1186.1687  468.7756 110.49148  233.11664
## 42      103  97        3     2 14 1099.7207  521.9427 139.49506  301.36076
## 43      103  97        3     3 17 1375.7002  458.5532 111.21548  235.76629
## 44      103  97        3     4 18 1384.9941  558.6514 131.67539  277.81080
## 45      103  97        3     5 16 1560.5688  611.2170 152.80426  325.69457
## 46      104  42        1     1 10 1468.3396  246.4016  77.91903  176.26509
## 47      104  42        1     2 11 1691.7417  399.8614 120.56276  268.63057
## 48      104  42        1     3 12 1725.2447  431.3717 124.52629  274.08051
## 49      104  42        1     4  7 1441.4886  527.8736 199.51745  488.20162
## 50      104  42        1     5  5 1614.7065  487.4739 218.00494  605.27875
## 51      104  42        2     1 12 1035.0665  388.5741 112.17169  246.88821
## 52      104  42        2     2 12 1322.7785  552.9101 159.61139  351.30229
## 53      104  42        2     3 10 1314.1556  275.2344  87.03676  196.89083
## 54      104  42        2     4 12 1356.2325  363.5779 104.95589  231.00635
## 55      104  42        2     5  8 1393.5998  349.2836 123.49041  292.00843
## 56      104  42        3     1  8  936.5166  178.8433  63.23066  149.51675
## 57      104  42        3     2 13 1045.0030  532.7783 147.76610  321.95468
## 58      104  42        3     3 15 1448.6131  631.5807 163.07342  349.75771
## 59      104  42        3     4 12 1363.5948  475.8995 137.38035  302.37210
## 60      104  42        3     5  7 1166.0978  296.0612 111.90063  273.81098
## 61      105  91        1     1 23 1278.9679  378.5265  78.92823  163.68714
## 62      105  91        1     2 22 1494.3695  433.7760  92.48136  192.32551
## 63      105  91        1     3 19 1487.5053  355.1332  81.47315  171.16875
## 64      105  91        1     4 23 1518.1817  505.4801 105.39990  218.58600
## 65      105  91        1     5 17 1578.3854  289.7805  70.28209  148.99138
## 66      105  91        2     1 22 1187.2017  250.4083  53.38723  111.02482
## 67      105  91        2     2 22 1120.1581  229.1538  48.85576  101.60113
## 68      105  91        2     3 24 1361.6607  373.0931  76.15732  157.54341
## 69      105  91        2     4 22 1462.1650  491.2600 104.73698  217.81248
## 70      105  91        2     5 20 1427.1733  435.0271  97.27502  203.59895
## 71      105  91        3     1 24 1216.7441  342.5561  69.92398  144.64876
## 72      105  91        3     2 23 1197.1853  286.6926  59.77954  123.97517
## 73      105  91        3     3 23 1303.6391  408.6771  85.21507  176.72524
## 74      105  91        3     4 21 1388.2115  528.1858 115.25959  240.42729
## 75      105  91        3     5 22 1367.0737  407.8340  86.95049  180.82345
## 76      106  94        1     1 14 1501.7248  608.7942 162.70709  351.50730
## 77      106  94        1     2 14 1555.0141  698.3146 186.63242  403.19484
## 78      106  94        1     3 11 1289.5354  494.3328 149.04695  332.09731
## 79      106  94        1     4 13 1761.3040  587.0137 162.80831  354.72883
## 80      106  94        1     5 12 1312.3281  495.3372 142.99154  314.72226
## 81      106  94        2     1 15 1429.7351  542.5796 140.09346  300.47059
## 82      106  94        2     2 17 1303.2390  788.9694 191.35319  405.65065
## 83      106  94        2     3 17 1155.0023  343.8604  83.39839  176.79669
## 84      106  94        2     4 17 1262.1727  462.2090 112.10214  237.64592
## 85      106  94        2     5 13 1740.9447  674.4934 187.07082  407.59231
## 86      106  94        3     1 21 1129.6410  305.2924  66.62027  138.96744
## 87      106  94        3     2 16 1151.1251  364.1948  91.04870  194.06571
## 88      106  94        3     3 21 1438.3866  619.6676 135.22255  282.06930
## 89      106  94        3     4 15 1119.3104  557.6370 143.98125  308.80907
## 90      106  94        3     5 17 1361.5606  623.4413 151.20673  320.54395
## 91      107  93        1     1 19 1259.4564  240.0093  55.06190  115.68076
## 92      107  93        1     2 23 1343.6728  241.3333  50.32147  104.36035
## 93       107  93       1     3 21 1388.6323  202.2287  44.12992   92.05340
## 94       107  93       1     4 23 1580.5209  530.0543 110.52396  229.21266
## 95       107  93       1     5 18 1537.8752  343.6372  80.99607  170.88677
## 96       107  93       2     1 23 1300.9721  320.2956  66.78624  138.50618
## 97       107  93       2     2 23 1271.6666  331.2548  69.07140  143.24531
## 98       107  93       2     3 24 1330.3263  438.0095  89.40832  184.95520
## 99       107  93       2     4 23 1465.8063  454.6168  94.79416  196.59106
## 100      107  93       2     5 22 1373.2215  370.4001  78.96957  164.22621
## 101      107  93       3     1 23 1211.1407  164.9986  34.40459   71.35074
## 102      107  93       3     2 24 1299.6179  380.6672  77.70338  160.74168
## 103      107  93       3     3 23 1291.7542  278.2545  58.02008  120.32627
## 104      107  93       3     4 23 1345.1355  316.6392  66.02383  136.92505
## 105      107  93       3     5 21 1322.7340  304.3807  66.42131  138.55242
## 106      108  99       1     1 22 1438.9657  405.9183  86.54208  179.97410
## 107      108  99       1     2 22 1275.0738  420.6970  89.69290  186.52659
## 108      108  99       1     3 22 1629.5510  547.5762 116.74364  242.78169
## 109      108  99       1     4 19 1398.0983  398.5282  91.42864  192.08444
## 110      108  99       1     5 17 1525.8691  426.0546 103.33342  219.05707
## 111      108  99       2     1 23 1237.0282  298.1722  62.17320  128.93932
## 112      108  99       2     2 23 1339.0127  297.7694  62.08921  128.76514
## 113      108  99       2     3 23 1373.2258  484.9160 101.11199  209.69342
## 114      108  99       2     4 23 1236.3170  342.8446  71.48803  148.25711
## 115      108  99       2     5 21 1521.5263  486.2544 106.10942  221.34036
## 116      108  99       3     1 24 1154.2816  187.9833  38.37194   79.37840
## 117      108  99       3     2 20 1223.6215  254.7277  56.95884  119.21621
## 118      108  99       3     3 22 1331.1782  408.2782  87.04520  181.02040
## 119      108  99       3     4 24 1310.7399  388.2077  79.24257  163.92574
## 120      108  99       3     5 19 1376.1429  391.7063  89.86360  188.79641
## 121      109  52       1     1  5 1553.7081  684.2034 305.98507  849.55074
## 122      109  52       1     2  4 1678.2281  376.5009 188.25047  599.09701
## 123      109  52       1     3  3 1174.4947 1047.8762 604.99161 2603.06879
## 124      109  52       1     4  6 1548.3169  255.5026 104.30850  268.13354
## 125      109  52       1     5  2 1369.1947  562.0209 397.40876 5049.55701
## 126      109  52       2     1  2 1274.4031  466.5656 329.91172 4191.92584
## 127      109  52       2     2  4 1086.4697  484.8184 242.40922  771.45432
## 128      109  52       2     3  4 1293.5114  393.9115 196.95576  626.80112
## 129      109  52       2     4  3 1312.8725  592.5290 342.09679 1471.92367
## 130      109  52       2     5  5 1406.8581  470.9505 210.61547  584.76230
## 131      109  52       3     1  5 1063.5614  655.2707 293.04598  813.62607
## 132      109  52       3     2  5 1439.2347  326.4862 146.00907  405.38617
## 133      109  52       3     3  5 1250.1781  689.2695 308.25070  855.84113
## 134      109  52       3     4  7 1310.9876  685.1531 258.96353  633.66094
## 135      109  52       3     5  7 1375.6471  456.2471 172.44521  421.95822
## 136      110 100       1     1 22 1257.7886  359.3965  76.62359  159.34748
## 137      110 100       1     2 19 1514.5425  550.3031 126.24819  265.23761
## 138      110 100       1     3 21 1334.1082  411.9154  89.88732  187.50166
## 139      110 100       1     4 17 1526.6584  450.0413 109.15105  231.38990
## 140      110 100       1     5 20 1694.7038  537.7702 120.24907  251.68420
## 141      110 100       2     1 23 1507.9951  583.6788 121.70544  252.40163
## 142      110 100       2     2 22 1311.4739  382.5842  81.56723  169.62833
## 143      110 100       2     3 23 1148.0501  171.1203  35.68106   73.99799
## 144      110 100       2     4 22 1276.1592  346.0953  73.78776  153.45006
## 145      110 100       2     5 22 1382.7185  476.3793 101.56440  211.21474
## 146      110 100       3     1 24 1243.3291  243.5664  49.71779  102.84908
## 147     110 100       3     2 24 1362.4957  531.6914 108.53104  224.51357
## 148     110 100       3     3 23 1294.4569  299.5477  62.46000  129.53412
## 149     110 100       3     4 23 1282.5385  286.9295  59.82893  124.07761
## 150     110 100       3     5 22 1265.8550  282.2211  60.16975  125.12985
## 151     111  96       1     1 21 1302.4626  428.0049  93.39832  194.82549
## 152     111  96       1     2 20 1415.6848  336.3217  75.20382  157.40339
## 153     111  96       1     3 18 1676.8457  560.6042 132.13568  278.78192
## 154     111  96       1     4 22 1442.8978  355.8347  75.86422  157.76829
## 155     111  96       1     5 19 1450.4708  488.6445 112.10273  235.51910
## 156     111  96       2     1 22 1200.5480  442.8777  94.42184  196.36096
## 157     111  96       2     2 19 1348.6471  452.2510 103.75349  217.97800
## 158     111  96       2     3 19 1277.7434  364.0641  83.52204  175.47330
## 159     111  96       2     4 24 1153.7975  269.9575  55.10485  113.99306
## 160     111  96       2     5 23 1202.3603  407.1926  84.90553  176.08329
## 161     111  96       3     1 21 1335.7529  486.6526 106.19630  221.52161
## 162     111  96       3     2 20 1275.1954  344.1587  76.95622  161.07122
## 163     111  96       3     3 18 1471.8820  433.2202 102.11099  215.43535
## 164     111  96       3     4 23 1351.5992  400.8776  83.58876  173.35247
## 165     111  96       3     5 22 1491.5915  511.9152 109.14070  226.97050
## 166     112  28       1     1  8 1316.5080  563.7412 199.31263  471.29948
## 167     112  28       1     2  5 1526.7209  454.1961 203.12268  563.95898
## 168     112  28       1     3  2 1076.8031  471.1768 333.17228 4233.35518
## 169     112  28       1     4  5 1423.9047  449.4000 200.97779  558.00381
## 170     112  28       1     5  2 1845.2852  217.5909 153.85997 1954.97634
## 171     112  28       2     1  7 1253.1491  350.4492 132.45733  324.11141
## 172     112  28       2     2  8 1492.6312  508.3645 179.73400  425.00338
## 173     112  28       2     3  4 1582.2843  425.4618 212.73088  677.00461
## 174     112  28       2     4  1 1352.9281        NA        NA         NaN
## 175     112  28       2     5  2 1714.9906  530.0052 374.77029 4761.90806
## 176     112  28       3     1 12 1138.1022  386.9141 111.69248  245.83349
## 177     112  28       3     2  7 1153.2813  321.1385 121.37893  297.00355
## 178     112  28       3     3  5 1428.7531  436.7453 195.31845  542.29097
## 179     112  28       3     4  4 1370.4906  223.2388 111.61939  355.22271
## 180     112  28       3     5  6 1369.6596  268.2209 109.50071  281.48054
## 181     113 100       1     1 21 1310.5205  443.3600  96.74909  201.81506
## 182     113 100       1     2 17 1439.9194  483.2373 117.20226  248.45769
## 183     113 100       1     3 17 1318.5734  339.0773  82.23831  174.33744
## 184     113 100       1     4 23 1399.5255  449.1382  93.65179  194.22193
## 185     113 100       1     5 20 1443.0444  418.9173  93.67275  196.05932
## 186     113 100       2     1 20 1201.2809  154.8110  34.61680   72.45379
## 187     113 100       2     2 23 1254.5001  384.4238  80.15790  166.23731
## 188     113 100       2     3 21 1244.8037  213.3766  46.56258   97.12784
## 189     113 100       2     4 19 1548.4409  454.7210 104.32014  219.16848
## 190     113 100       2     5 20 1512.8014  420.9777  94.13347  197.02362
## 191     113 100       3     1 18 1333.6522  250.8663  59.12976  124.75289
## 192     113 100       3     2 21 1243.2475  236.1940  51.54175  107.51421
## 193     113 100       3     3 22 1396.6814  445.0887  94.89323  197.34128
## 194     113 100       3     4 24 1209.4381  187.7392  38.32211   79.27532
## 195     113 100       3     5 19 1505.9964  548.9517 125.93816  264.58626
## 196     114  97       1     1 21 1484.1272  552.6513 120.59841  251.56388
## 197     114  97       1     2 15 1556.1974  587.5475 151.70412  325.37297
## 198     114  97       1     3 15 1458.5554  532.1168 137.39198  294.67648
## 199     114  97       1     4 17 1301.5413  353.8481  85.82077  181.93190
## 200     114  97       1     5 19 1394.6748  418.2269  95.94782  201.57889
## 201     114  97       2      1 20 1136.6087  245.9145  54.98815  115.09153
## 202     114  97       2      2 17 1215.5708  246.5963  59.80839  126.78812
## 203     114  97       2      3 19 1277.2423  253.1440  58.07522  122.01151
## 204     114  97       2      4 20 1466.7039  449.8238 100.58367  210.52404
## 205     114  97       2      5 18 1233.0174  291.5303  68.71434  144.97459
## 206     114  97       3      1 21 1133.5303  239.4278  52.24744  108.98625
## 207     114  97       3      2 22 1518.9333  667.0300 142.21127  295.74452
## 208     114  97       3      3 21 1272.4256  401.2322  87.55605  182.63872
## 209     114  97       3      4 22 1374.3754  373.6283  79.65783  165.65753
## 210     114  97       3      5 20 1495.3315  586.9165 131.23852  274.68538
## 211     115 100       1      1 21 1573.0226  481.1330 104.99184  219.00914
## 212     115 100       1      2 22 1477.9638  302.1047  64.40895  133.94573
## 213     115 100       1      3 23 1676.7618  447.1677  93.24092  193.36982
## 214     115 100       1      4 23 1354.8829  275.5391  57.45388  119.15206
## 215     115 100       1      5 19 1478.5661  267.1714  61.29332  128.77249
## 216     115 100       2      1 23 1335.1289  375.6216  78.32252  162.43096
## 217     115 100       2      2 21 1444.7999  552.2098 120.50207  251.36291
## 218     115 100       2      3 24 1376.3193  416.3729  84.99176  175.81885
## 219     115 100       2      4 22 1159.6730  174.2527  37.15079   77.25930
## 220     115 100       2      5 22 1259.2721  236.9885  50.52613  105.07483
## 221     115 100       3      1 22 1185.6920  409.7421  87.35730  181.66946
## 222     115 100       3      2 23 1263.2928  508.4072 106.01023  219.85177
## 223     115 100       3      3 24 1137.7360  232.3564  47.42956   98.11552
## 224     115 100       3      4 24 1383.9026  371.9953  75.93322  157.07984
## 225     115 100       3      5 22 1222.9151  319.3913  68.09446  141.61019
## 226     116 100       1      1 23 1349.4370  273.5274  57.03441  118.28212
## 227     116 100       1      2 22 1422.4348  339.1662  72.31048  150.37787
## 228     116 100       1      3 20 1520.7922  402.9043  90.09214  188.56503
## 229     116 100       1      4 22 1411.9017  245.3698  52.31302  108.79088
## 230     116 100       1      5 24 1496.7804  335.8464  68.55435  141.81548
## 231     116 100       2      1 24 1210.7387  256.9263  52.44486  108.49047
## 232     116 100       2      2 24 1276.5304  259.1567  52.90013  109.43226
## 233     116 100       2      3 24 1226.1971  272.0759  55.53726  114.88757
## 234     116 100       2      4 23 1371.8827  298.0578  62.14936  128.88988
## 235     116 100       2      5 24 1493.0721  515.0257 105.12917  217.47627
## 236     116 100       3      1 24 1262.7804  272.8139  55.68790  115.19919
## 237     116 100       3      2 24 1335.8637  236.6758  48.31124   99.93941
## 238     116 100       3      3 23 1334.7807  263.4028  54.92328  113.90390
## 239     116 100       3      4 23 1270.1552  209.5029  43.68437   90.59584
## 240     116 100       3      5 24 1345.6971  198.5101  40.52071   83.82348
## 241     117 100       1      1 19 1403.0974  415.2946  95.27512  200.16559
## 242     117 100       1      2 17 1339.9588  247.1947  59.95353  127.09581
## 243     117 100       1      3 21 1555.5001  550.5829 120.14704  250.62233
## 244     117 100       1      4 20 1423.8212  374.6269  83.76912  175.33078
## 245     117 100       1      5 21 1579.0680  586.8863 128.06908  267.14743
## 246     117 100       2      1 21 1291.3708  448.2158  97.80872  204.02541
## 247     117 100       2      2 23 1265.4818  369.6322  77.07365  159.84097
## 248     117 100       2      3 23 1346.1346  413.1105  86.13949  178.64238
## 249     117 100       2      4 22 1230.5466  261.3275  55.71521  115.86612
## 250     117 100       2      5 21 1384.1038  405.2388  88.43035  184.46248
## 251     117 100       3      1 23 1316.1883  469.1563  97.82584  202.87838
## 252     117 100       3      2 22 1219.4350  226.2870  48.24454  100.33002
## 253     117 100       3      3 24 1409.7018  433.5256  88.49305  183.06182
## 254     117 100       3      4 22 1259.2501  430.0137  91.67922  190.65738
## 255      117 100       3     5 24 1303.9518  327.6630  66.88394  138.35997
## 256      118 100       1     1 21 1382.3957  527.3595 115.07927  240.05116
## 257      118 100       1     2 21 1522.9463  619.2724 135.13632  281.88942
## 258      118 100       1     3 21 1532.9058  549.0874 119.82070  249.94161
## 259      118 100       1     4 21 1514.7571  629.7051 137.41293  286.63834
## 260      118 100       1     5 17 1378.0469  481.2649 116.72387  247.44356
## 261      118 100       2     1 21 1135.4206  474.2366 103.48690  215.86990
## 262      118 100       2     2 24 1220.0276  321.5133  65.62862  135.76315
## 263      118 100       2     3 23 1215.3544  335.6795  69.99401  145.15870
## 264      118 100       2     4 21 1293.3228  367.7129  80.24152  167.38088
## 265      118 100       2     5 23 1432.7240  572.8981 119.45750  247.73970
## 266      118 100       3     1 23 1273.8291  416.3052  86.80564  180.02387
## 267      118 100       3     2 24 1427.4026  616.6584 125.87487  260.39201
## 268      118 100       3     3 23 1236.0785  411.9566  85.89889  178.14340
## 269      118 100       3     4 21 1316.0199  467.8253 102.08786  212.95154
## 270      118 100       3     5 20 1454.8378  642.8706 143.75023  300.87270
## 271      119  83       1     1 19 1389.3999  431.9863  99.10445  208.21072
## 272      119  83       1     2 18 1349.1446  435.3157 102.60489  216.47740
## 273      119  83       1     3 18 1408.7464  483.9850 114.07636  240.68008
## 274      119  83       1     4 18 1339.8588  466.4117 109.93430  231.94110
## 275      119  83       1     5 17 1329.4041  362.4648  87.91062  186.36220
## 276      119  83       2     1 21 1543.9834  373.2356  81.44670  169.89483
## 277      119  83       2     2 22 1278.1155  484.0949 103.20939  214.63567
## 278      119  83       2     3 16 1424.1952  392.3671  98.09177  209.07766
## 279      119  83       2     4 21 1502.5358  418.0943  91.23567  190.31427
## 280      119  83       2     5 15 1654.3117  356.3750  92.01563  197.35391
## 281      119  83       3     1 21 1408.9437  439.2475  95.85166  199.94306
## 282      119  83       3     2 22 1097.6799  330.6858  70.50246  146.61789
## 283      119  83       3     3 21 1241.5090  330.0779  72.02890  150.24965
## 284      119  83       3     4 21 1210.6017  338.0194  73.76189  153.86460
## 285      119  83       3     5 22 1244.0312  502.9982 107.23957  223.01690
## 286      120  75       1     1  8 1250.6518  369.9371 130.79251  309.27514
## 287      120  75       1     2 13 1677.0523  612.9922 170.01344  370.42746
## 288      120  75       1     3 12 1204.3829  356.7538 102.98594  226.67053
## 289      120  75       1     4 18 1420.0321  490.9063 115.70773  244.12197
## 290      120  75       1     5 12 1465.3693  643.4496 185.74791  408.82840
## 291      120  75       2     1 15 1273.0214  503.6087 130.03121  278.88920
## 292      120  75       2     2 15 1185.5242  315.6730  81.50642  174.81388
## 293      120  75       2     3 10 1650.1187  759.4871 240.17092  543.30436
## 294      120  75       2     4 18 1317.0788  459.1972 108.23381  228.35337
## 295      120  75       2     5 14 1562.7185  586.0235 156.62137  338.35990
## 296      120  75       3     1 19 1175.8661  206.2909  47.32637   99.42901
## 297      120  75       3     2 14 1216.7220  378.7311 101.22013  218.67280
## 298      120  75       3     3 18 1156.6040  312.5545  73.66981  155.42971
## 299      120  75       3     4 10 1562.5556  557.6170 176.33399  398.89520
## 300      120  75       3     5 13 1428.6633  520.7045 144.41746  314.65860
```

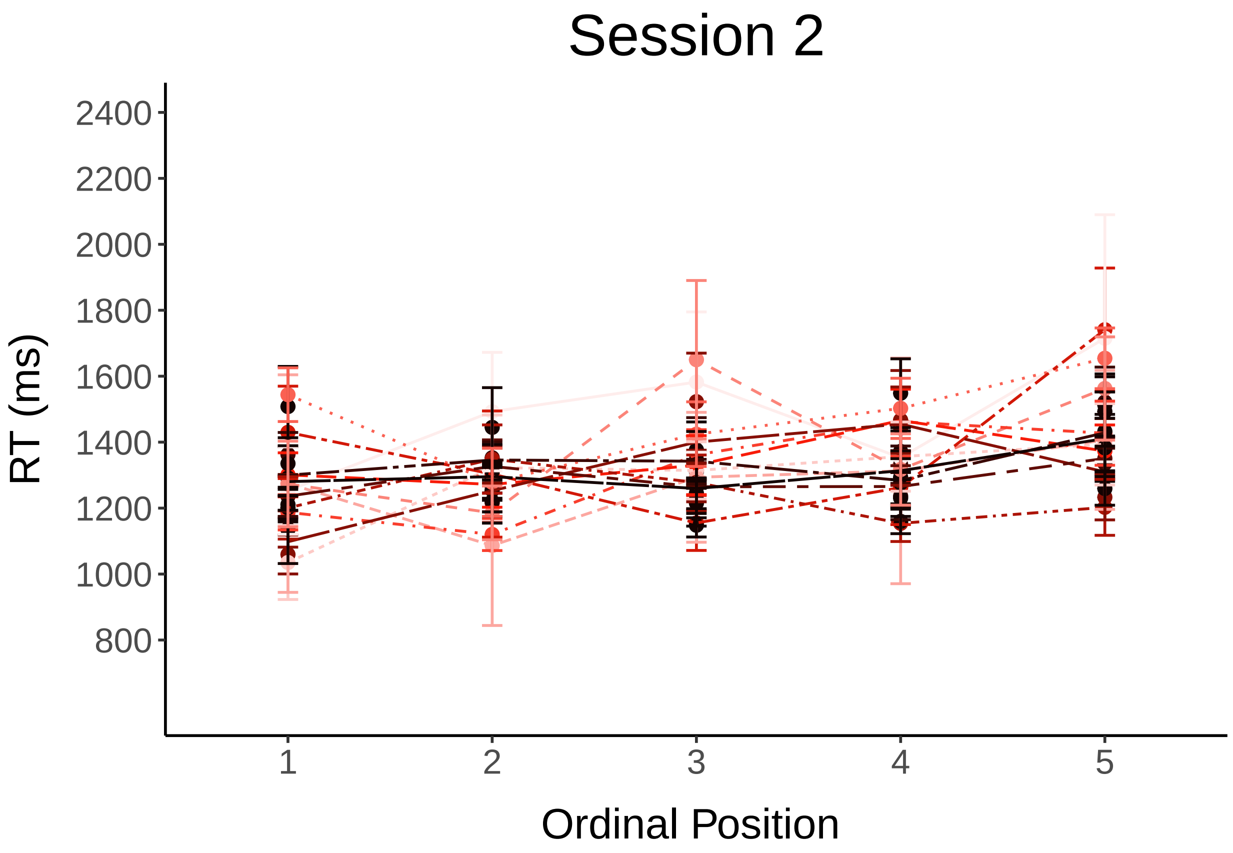
```r
override.linetype<-c("dashed")
(plot_rt_repetition_PWA_session1 <- means_final %>%
    filter(session=="1") %>%
    ggplot(., aes(x=PosOr, y=VOT, group=AAT, color = AAT)) +
    geom_point(size = 2)+
    stat_summary(aes(linetype=AAT, color=AAT),
                 fun=mean,  geom="line", size = 0.5) +
    scale_linetype_manual(values=c("dashed"))+
    scale_colour_manual(values=c("#feedec", "#fdcac6", "#fca7a0",
                                   "#fb8479", "#fa6153",
                                   "#f93e2d", "#f81b07",
                                   "#d21706", "#ac1305",
                                   "#860f04", "#5f0b03",
                                   "#390602", "#130201"))+
    geom_errorbar(aes(ymin=VOT-se, ymax=VOT+se, group = AAT), width =.1) +
    apatheme+
    scale_y_continuous(limits = c(600, 2400),
                       breaks =seq(800,2400, by = 200)) +
    labs(x="Ordinal Position ",y ="RT (ms)", colour="Session",
         linetype="Session",
         title = "Session 1") +
    theme(
    axis.title.y = element_text(margin = margin(0,10,0,0)),
    axis.title.x = element_text(margin = margin(10,0,0,0)),
    legend.key.width = unit(1, "cm"),
    legend.position="none")+
   guides(color=guide_legend(
     override.aes=list(linetype=override.linetype)))+
    scale_linetype(guide="none"))
```

```
## Scale for linetype is already present.
## Adding another scale for linetype, which will replace the existing scale.
```



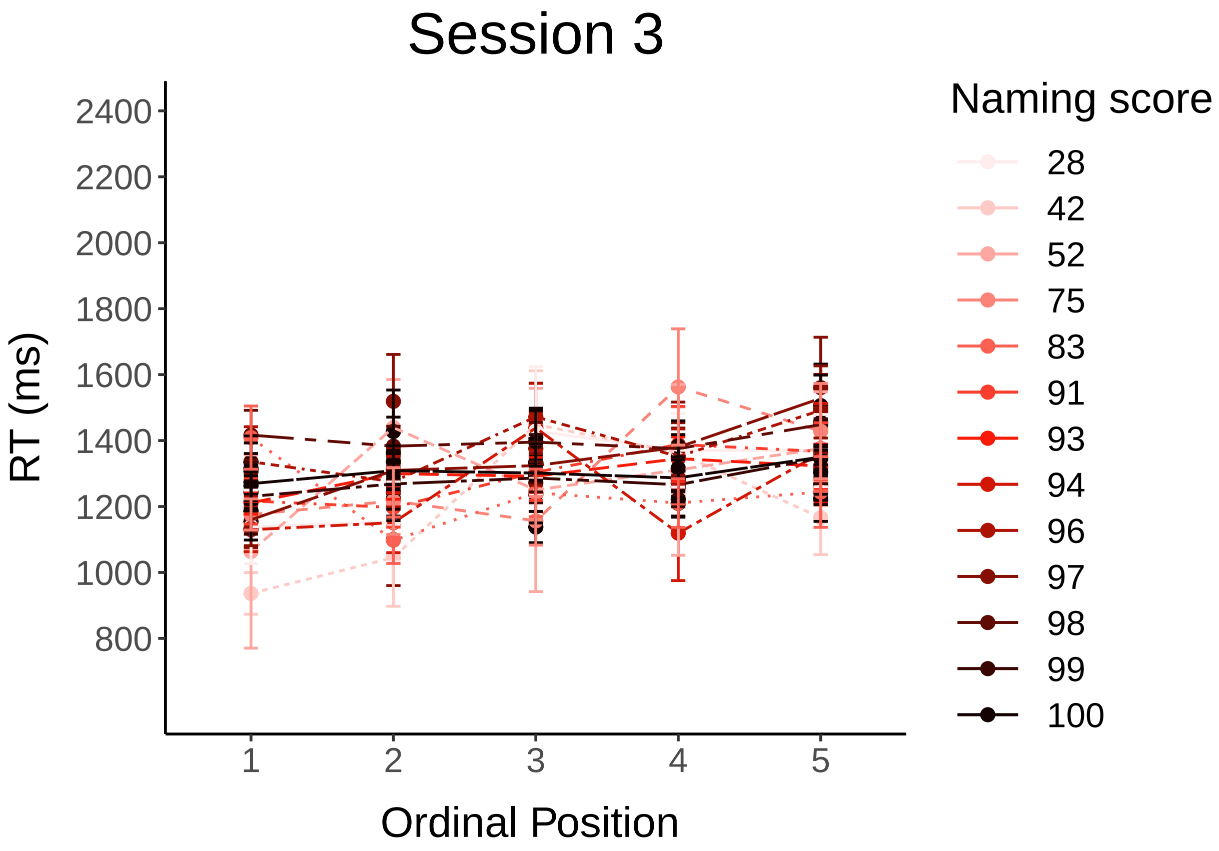
```r
(plot_rt_repetition_PWA_session2 <- means_final %>%
    filter(session=="2") %>%
    ggplot(., aes(x=PosOr, y=VOT, group=AAT, color = AAT)) +
    geom_point(size = 2)+
    stat_summary(aes(linetype=AAT, color=AAT),
                 fun=mean,  geom="line", size = 0.5) +
    scale_linetype_manual(values=c("dashed"))+
    scale_colour_manual(values=c("#feedec", "#fdcac6", "#fca7a0",
                                          "#fb8479", "#fa6153",
                                          "#f93e2d", "#f81b07",
                                          "#d21706", "#ac1305",
                                          "#860f04", "#5f0b03",
                                          "#390602", "#130201"))+
    geom_errorbar(aes(ymin=VOT-se, ymax=VOT+se, group = AAT), width =.1) +
    apatheme+
    scale_y_continuous(limits = c(600, 2400),
                       breaks =seq(800,2400, by = 200)) +
    labs(x="Ordinal Position ",y ="RT (ms)", colour="Session",
         linetype="Session",
         title = "Session 2") +
    theme(
    axis.title.y = element_text(margin = margin(0,10,0,0)),
    axis.title.x = element_text(margin = margin(10,0,0,0)),
    legend.key.width = unit(1, "cm"),
    legend.position="none")+
   guides(color=guide_legend(
     override.aes=list(linetype=override.linetype)))+
    scale_linetype(guide="none"))
```

```
## Scale for linetype is already present.
## Adding another scale for linetype, which will replace the existing scale.
```



```
(plot_rt_repetition_PWA_session3 <- means_final %>%
    filter(session=="3") %>%
    ggplot(., aes(x=PosOr, y=VOT, group=AAT, color = AAT)) +
    geom_point(size = 2)+
    stat_summary(aes(linetype=AAT, color=AAT),
                 fun=mean,  geom="line", size = 0.5) +
    scale_linetype_manual(values=c("dashed"))+
    scale_colour_manual(values=c("#feedec", "#fdcac6", "#fca7a0",
                                          "#fb8479", "#fa6153",
                                          "#f93e2d", "#f81b07",
                                          "#d21706", "#ac1305",
                                          "#860f04", "#5f0b03",
                                          "#390602", "#130201"))+
    geom_errorbar(aes(ymin=VOT-se, ymax=VOT+se, group = AAT), width =.1) +
    apatheme+
    scale_y_continuous(limits = c(600, 2400),
                       breaks =seq(800,2400, by = 200)) +
    labs(x="Ordinal Position ",y ="RT (ms)", colour="Naming score",
        # linetype="Session",
         title = "Session 3") +
    theme(
    axis.title.y = element_text(margin = margin(0,10,0,0)),
    axis.title.x = element_text(margin = margin(10,0,0,0)),
    legend.key.width = unit(1, "cm"),
    legend.position="right")+
    scale_linetype(guide="none"))
```

```
## Scale for linetype is already present.
## Adding another scale for linetype, which will replace the existing scale.
```



```
plots <- cowplot::plot_grid(plot_rt_repetition_PWA_session1,
                            plot_rt_repetition_PWA_session2,
                            plot_rt_repetition_PWA_session3,
          nrow = 1, ncol=3,  rel_widths = c(0.81,0.81,1),
          margin(1,1,1,1),
          labels = c("B", "C", "D"),label_size = 34,
                    label_fontfamily = "Helvetica", label_y = 1.01,
          label_x=-0.03)
```

```
## Warning in as_grob.default(plot): Cannot convert object of class
## marginsimpleUnitunitunit_v2 into a grob.
```

```
filename <- "CSI_online_aphasia_spoken_plot_PWA_AAT_scores.pdf"
ggsave(plots, filename =
         here::here("results", "figures", filename),
       width = 25, height = 13, units = "cm",
       dpi = 300, device = cairo_pdf)
#embedFonts(file = here::here("results", "figures", filename))
```

```
#### Trying to understand the interaction:
# median split of means final
median(as.numeric(as.character(means_final$AAT)))
```

```
## [1] 97
```

```
means_final %>%
  mutate(med_split=case_when(as.numeric(as.character(AAT)) >
                               median(as.numeric(as.character(
                                               means_final$AAT))) ~"high",
                              as.numeric(as.character(AAT)) <
                                median(as.numeric(as.character(
                              means_final$AAT))) ~ "low",
                               as.numeric(as.character(AAT)) ==
                               median(as.numeric(as.character(
                                               means_final$AAT))) ~
                               "median"),
                       mean_split=case_when(
                         as.numeric(as.character(AAT)) >
                           mean(as.numeric(as.character(
                             means_final$AAT))) ~ "high",
                         as.numeric(as.character(AAT)) <
                           mean(as.numeric(as.character(
                             means_final$AAT))) ~ "low",
                         as.numeric(as.character(AAT)) ==
                           mean(as.numeric(as.character(
                             means_final$AAT))) ~ "median")) ->
  means_final

means_final %>% group_by(med_split) %>% summarise(mean=mean(VOT),
                                                  sd = sd(VOT))
```

```
## # A tibble: 3 x 3
##   med_split  mean    sd
##   <chr>     <dbl> <dbl>
## 1 high      1356.  113.
## 2 low       1364.  169.
## 3 median    1355.  153.
```

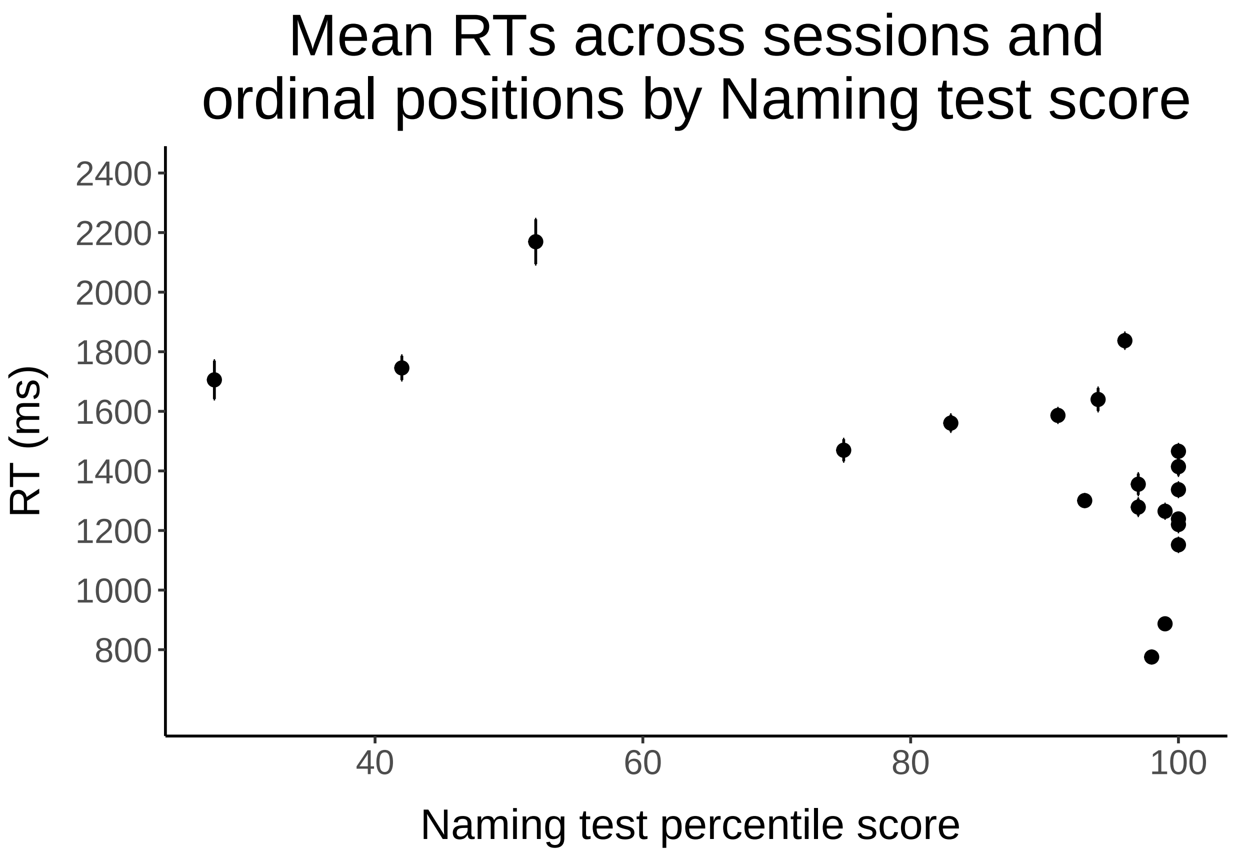
```
means_final %>% group_by(mean_split) %>% summarise(mean=mean(VOT),
                                                  sd = sd(VOT))
```

```
## # A tibble: 2 x 3
##   mean_split  mean    sd
##   <chr>      <dbl> <dbl>
## 1 high       1357.  128.
## 2 low        1366.  186.
```

```
(means_final<- df_RTs_PWA %>%
   filter(category != "Filler") %>%
   Rmisc::summarySE(.,measurevar="VOT",
                          groupvars = c("subject","AAT"), na.rm = T))
```

```
##    subject AAT   N       VOT       sd       se        ci
## 1      101  98 345  775.3710 380.0659 20.46206  40.24650
## 2      102  99 351  886.8604 292.5880 15.61719  30.71535
## 3      103  97 245 1355.3306 560.3643 35.80036  70.51719
## 4      104  42 154 1745.7143 508.6572 40.98876  80.97700
## 5      105  91 327 1586.3639 436.7228 24.15083  47.51114
## 6      106  94 233 1639.8584 592.6691 38.82705  76.49867
## 7      107  93 333 1300.3393 379.0231 20.77034  40.85806
## 8      108  99 324 1264.6759 434.0358 24.11310  47.43856
## 9      109  52  67 2169.3731 617.2171 75.40507 150.55108
## 10     110 100 327 1220.1957 449.9797 24.88394  48.95336
## 11     111  96 311 1837.2637 464.0219 26.31227  51.77323
## 12     112  28  78 1705.7179 569.9750 64.53696 128.50951
## 13     113 100 305 1151.9082 401.5656 22.99358  45.24672
## 14     114  97 287 1278.4530 486.1275 28.69520  56.48056
## 15     115 100 335 1465.9582 430.1200 23.49997  46.22659
## 16     116 100 348 1238.8707 350.2938 18.77772  36.93247
## 17     117 100 323 1336.9969 427.5771 23.79103  46.80548
## 18     118 100 324 1414.3981 539.9319 29.99621  59.01262
## 19     119  83 292 1560.5582 483.6372 28.30273  55.70400
## 20     120  75 209 1469.4641 537.0199 37.14644  73.23179
```

```
(plot_rt_PWA_AAT <- means_final %>%
    ggplot(., aes(x=AAT, y=VOT)) +
    geom_point(size = 2)+
    scale_colour_manual(values=c("#feedec", "#fdcac6", "#fca7a0",
                                          "#fb8479", "#fa6153",
                                          "#f93e2d", "#f81b07",
                                          "#d21706", "#ac1305",
                                          "#860f04", "#5f0b03",
                                          "#390602", "#130201"))+
    geom_errorbar(aes(ymin=VOT-se, ymax=VOT+se, group = AAT), width =.1) +
    apatheme+
    scale_y_continuous(limits = c(600, 2400),
                       breaks =seq(800,2400, by = 200)) +
    labs(x="Naming test percentile score ",y ="RT (ms)", colour="Naming score",
        # linetype="Session",
         title = "Mean RTs across sessions and\nordinal positions by Naming test score") +
    theme(
    axis.title.y = element_text(margin = margin(0,10,0,0)),
    axis.title.x = element_text(margin = margin(10,0,0,0)),
    legend.key.width = unit(1, "cm"),
    legend.position="none")+
    scale_linetype(guide="none"))
```



```
filename <- "CSI_online_aphasia_spoken_plot_PWA_AAT.pdf"
ggsave(plot_rt_PWA_AAT, filename =
         here::here("results", "figures", filename),
       width = 15, height = 10, units = "cm",
       dpi = 300, device = cairo_pdf)


plots2 <- cowplot::plot_grid(plot_rt_PWA_AAT, plots,
          nrow = 1, ncol=2,  rel_widths = c(0.81,0.81,1),
          rel_height = c(4,1),
          margin(1.5,1.5,1.5,1.5),
          labels = c("A", ""),label_size = 34,
                    #label_fontfamily = "Helvetica",
          label_y = 1.01,
          label_x=-0.03)
```

```
## Warning in as_grob.default(plot): Cannot convert object of class numeric into a
## grob.

## Warning in as_grob.default(plot): Cannot convert object of class
## marginsimpleUnitunitunit_v2 into a grob.
```

```
filename <- "CSI_online_aphasia_spoken_plot_PWA_AAT_summary_plot.pdf"
ggsave(plots2, filename =
         here::here("results", "figures", filename),
       width = 25, height = 20, units = "cm",
       dpi = 300, device = cairo_pdf)
```

Errors Add into converging model

```
cor.test(df_errors_PWA$TokenTest_c, df_errors_PWA$LHoverall_c)
```

```
##
##  Pearson's product-moment correlation
##
## data:  df_errors_PWA$TokenTest_c and df_errors_PWA$LHoverall_c
## t = -44.023, df = 7176, p-value < 0.0000000000000022
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.4791506 -0.4427161
## sample estimates:
##        cor
## -0.4611277
```

```
cor.test(df_errors_PWA$AAT_c, df_errors_PWA$LHoverall_c)
```

```
##
##  Pearson's product-moment correlation
##
## data:  df_errors_PWA$AAT_c and df_errors_PWA$LHoverall_c
## t = -55.609, df = 7176, p-value < 0.0000000000000022
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.5647379 -0.5323978
## sample estimates:
##        cor
## -0.5487731
```

```
cor.test(df_errors_PWA$AAT_c, df_errors_PWA$TokenTest_c)
```

```
##
##  Pearson's product-moment correlation
##
## data:  df_errors_PWA$AAT_c and df_errors_PWA$TokenTest_c
## t = 151.84, df = 7176, p-value < 0.0000000000000022
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.8676779 0.8786656
## sample estimates:
##       cor
## 0.8732828
```

```
cor.test(df_errors_PWA$AAT_spontansprache, df_errors_PWA$TokenTest_c)
```

```
##
##  Pearson's product-moment correlation
##
## data:  df_errors_PWA$AAT_spontansprache and df_errors_PWA$TokenTest_c
## t = 66.274, df = 7176, p-value < 0.00000000000000022
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.6016264 0.6303336
## sample estimates:
##       cor
## 0.6161846
```

```
# m1_glmm_PWA_all <- glmer(error_class ~
#                           PosOr.cont*session*TokenTest_c*AAT_c*LHoverall_c +
#                     (PosOr.cont |subject) +
#                     (1|category) ,
#                   data =df_errors_PWA, family = "binomial",
#                   control=glmerControl(optimizer = "bobyqa",
#                                   optCtrl = list(maxfun = 2e5)))
# m1_glmm_PWA_all <- afex::lmer_alt(error_class ~
#                           PosOr.cont*session*(TokenTest_c*AAT_c*
#                                               AAT_spontansprache_c*
#                                               LHoverall_c) +
#                     (PosOr.cont |subject) +
#                     (1|category) ,
#                   data =df_errors_PWA, family = "binomial",
#                   control=glmerControl(optimizer = "bobyqa",
#                                   optCtrl = list(maxfun = 2e5)))

m1_glmm_PWA_all <- afex::lmer_alt(error_class ~
                          PosOr.cont*session*(TokenTest_c+AAT_c+
                                              AAT_spontansprache_c+
                                              LHoverall_c) +
                    (PosOr.cont |subject) +
                    (1|category) ,
                  data =df_errors_PWA, family = "binomial",
                  control=glmerControl(optimizer = "bobyqa",
                                  optCtrl = list(maxfun = 2e5)))

# m1_glmm_PWA_all <- afex::lmer_alt(error_class ~
#                           PosOr.cont*session*(TokenTest_c*AAT_c*
#                                               LHoverall_c) +
#                     (PosOr.cont |subject) +
#                     (1|category) ,
#                   data =df_errors_PWA, family = "binomial",
#                   control=glmerControl(optimizer = "bobyqa",
#                                   optCtrl = list(maxfun = 2e5)))

didLmerConverge(m1_glmm_PWA_all)
```

```
##  The relative maximum gradient of 0.0000465 is less than our 0.001 criterion.
##  You can safely ignore any warnings about a claimed convergence failure.
## Warnings can be ignored

summary(m1_glmm_PWA_all)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula:
## error_class ~ PosOr.cont * session * (TokenTest_c + AAT_c + AAT_spontansprache_c +
##     LHoverall_c) + (1 + re1.PosOr.cont | subject) + (1 | category)
##    Data: data
## Control: glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 200000))
##
##      AIC      BIC   logLik deviance df.resid
##   5329.1   5563.0  -2630.5   5261.1     7144
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.9219 -0.3912 -0.2415 -0.1200  9.8488
##
## Random effects:
##  Groups   Name           Variance Std.Dev. Corr
##  category (Intercept)    0.349037 0.5908
##  subject  (Intercept)    0.661274 0.8132
##           re1.PosOr.cont 0.007276 0.0853   0.05
## Number of obs: 7178, groups:  category, 24; subject, 20
##
## Fixed effects:
##                                           Estimate Std. Error z value
## (Intercept)                              -1.868138   0.223130  -8.372
## PosOr.cont                                0.091736   0.036639   2.504
## session2                                 -0.444675   0.089118  -4.990
## session3                                 -0.736607   0.092820  -7.936
## TokenTest_c                              -0.078771   0.404053  -0.195
## AAT_c                                    -1.134324   0.402664  -2.817
## AAT_spontansprache_c                     -0.071116   0.239747  -0.297
## LHoverall_c                               0.002373   0.222665   0.011
## PosOr.cont:session2                      -0.049177   0.062922  -0.782
## PosOr.cont:session3                      -0.042042   0.065504  -0.642
## PosOr.cont:TokenTest_c                   -0.082566   0.063911  -1.292
## PosOr.cont:AAT_c                         -0.062287   0.062476  -0.997
## PosOr.cont:AAT_spontansprache_c           0.093650   0.047339   1.978
## PosOr.cont:LHoverall_c                   -0.054721   0.038568  -1.419
## session2:TokenTest_c                     -0.033723   0.150716  -0.224
## session3:TokenTest_c                     -0.198145   0.154573  -1.282
## session2:AAT_c                           -0.044075   0.155617  -0.283
## session3:AAT_c                            0.053915   0.156123   0.345
## session2:AAT_spontansprache_c            -0.017978   0.113045  -0.159
## session3:AAT_spontansprache_c             0.233157   0.121585   1.918
## session2:LHoverall_c                      0.127220   0.102052   1.247
## session3:LHoverall_c                      0.176587   0.102239   1.727
## PosOr.cont:session2:TokenTest_c          -0.085100   0.106578  -0.798
## PosOr.cont:session3:TokenTest_c          -0.218500   0.109560  -1.994
## PosOr.cont:session2:AAT_c                 0.154716   0.110082   1.405
## PosOr.cont:session3:AAT_c                 0.292621   0.110689   2.644
## PosOr.cont:session2:AAT_spontansprache_c -0.024067   0.079779  -0.302
## PosOr.cont:session3:AAT_spontansprache_c -0.070096   0.085723  -0.818
## PosOr.cont:session2:LHoverall_c           0.112933   0.072131   1.566
## PosOr.cont:session3:LHoverall_c           0.042705   0.072236   0.591
##                                                       Pr(>|z|)
## (Intercept)                              < 0.0000000000000002 ***
## PosOr.cont                                             0.01229 *
## session2                                  0.00000060468889586 ***
## session3                                  0.00000000000000209 ***
## TokenTest_c                                            0.84543
## AAT_c                                                  0.00485 **
## AAT_spontansprache_c                                   0.76675
## LHoverall_c                                            0.99150
## PosOr.cont:session2                                    0.43448
## PosOr.cont:session3                                    0.52098
## PosOr.cont:TokenTest_c                                 0.19640
## PosOr.cont:AAT_c                                       0.31878
## PosOr.cont:AAT_spontansprache_c                        0.04790 *
## PosOr.cont:LHoverall_c                                 0.15596
## session2:TokenTest_c                                   0.82295
## session3:TokenTest_c                                   0.19988
## session2:AAT_c                                         0.77700
## session3:AAT_c                                         0.72984
## session2:AAT_spontansprache_c                          0.87364
## session3:AAT_spontansprache_c                          0.05516 .
## session2:LHoverall_c                                   0.21254
## session3:LHoverall_c                                   0.08413 .
## PosOr.cont:session2:TokenTest_c                        0.42459
## PosOr.cont:session3:TokenTest_c                        0.04611 *
## PosOr.cont:session2:AAT_c                              0.15989
## PosOr.cont:session3:AAT_c                              0.00820 **
## PosOr.cont:session2:AAT_spontansprache_c               0.76290
## PosOr.cont:session3:AAT_spontansprache_c               0.41352
## PosOr.cont:session2:LHoverall_c                        0.11742
## PosOr.cont:session3:LHoverall_c                        0.55440
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

```
##
## Correlation matrix not shown by default, as p = 30 > 12.
## Use print(x, correlation=TRUE)  or
##     vcov(x)        if you need it
```

```
anova(m1_glmm_PWA_all)
```

```
## Analysis of Variance Table
##                                         npar Sum Sq Mean Sq F value
## PosOr.cont                                 1  8.624   8.624  8.6237
## session                                    2 62.788  31.394 31.3940
## TokenTest_c                                1 34.852  34.852 34.8517
## AAT_c                                      1  8.540   8.540  8.5397
## AAT_spontansprache_c                          1  0.207   0.207  0.2073
## LHoverall_c                                   1  0.001   0.001  0.0013
## PosOr.cont:session                            2  0.667   0.333  0.3333
## PosOr.cont:TokenTest_c                        1  2.824   2.824  2.8242
## PosOr.cont:AAT_c                              1  0.180   0.180  0.1801
## PosOr.cont:AAT_spontansprache_c               1  5.376   5.376  5.3759
## PosOr.cont:LHoverall_c                        1  2.205   2.205  2.2053
## session:TokenTest_c                           2  4.480   2.240  2.2402
## session:AAT_c                                 2  1.148   0.574  0.5740
## session:AAT_spontansprache_c                  2  4.964   2.482  2.4820
## session:LHoverall_c                           2  3.168   1.584  1.5840
## PosOr.cont:session:TokenTest_c                2  0.550   0.275  0.2752
## PosOr.cont:session:AAT_c                      2  6.627   3.314  3.3137
## PosOr.cont:session:AAT_spontansprache_c       2  0.666   0.333  0.3329
## PosOr.cont:session:LHoverall_c                2  2.469   1.234  1.2344
```

```
performance::model_performance(m1_glmm_PWA_all)
```

```
## # Indices of model performance
##
## AIC      |     AICc |      BIC | R2 (cond.) | R2 (marg.) |   ICC |  RMSE | Sigma | Log_loss | Score_l
## -----------------------------------------------------------------------------------------------------
## 5329.073 | 5329.406 | 5562.952 |      0.452 |      0.284 | 0.235 | 0.331 | 1.000 |    0.354 |      -
```

```
saveRDS(m1_glmm_PWA_all,  file = here::here("results", "tables", "CSI_online_aphasia_PWA_lmm_error_Toke
tab_model(m1_glmm_PWA_tt,transform = NULL,
          show.re.var = F, show.stat = T,show.r2 = F,show.icc = F,
          title =
            "GLMM of errors Predicted by Ordinal Position and Session",
          dv.labels = "Errors",
          #string.pred = "",
          string.stat = "z-Value",
          file = here::here(
            "results", "tables",
            "CSI_online_aphasia_PWA_control_glmm_error_TokenTest_Naming_Lesionsize.html"))
```

GLMM of errors Predicted by Ordinal Position and Session

Errors

Predictors

Log-Odds

CI

z-Value

p

(Intercept)

-1.87

-2.37 – -1.37

-7.33

<0.001

PosOr cont

0.09

0.01 – 0.17

2.14

0.032

session [2]

-0.43

-0.61 – -0.26

-4.96

<0.001

session [3]

-0.74

-0.92 – -0.56

-8.01

<0.001

TokenTest c

-1.12

-1.55 – -0.68

-5.03

<0.001

PosOr cont × session [2]

-0.05

-0.17 – 0.07

-0.77

0.441

PosOr cont × session [3]

-0.05

-0.18 – 0.08

-0.75

0.451

PosOr cont × TokenTest c

-0.05

-0.12 – 0.02

-1.45

0.146

session [2] × TokenTest c

-0.15

-0.30 – -0.01

-2.07

0.038

session [3] × TokenTest c

-0.12

-0.26 – 0.03

-1.55

0.120

(PosOr cont × session[2]) × TokenTest c

-0.03

-0.13 – 0.07

-0.59

0.558

(PosOr cont × session[3]) × TokenTest c

-0.04

-0.14 – 0.07

-0.71

0.478

N subject

20

N category

24

Observations

7178

```
# Odds Ratio:
x <- data.frame(summary(m1_glmm_PWA_all)$coefficients)
x$Odds_Ratio <- plogis(x$Estimate)
x %>% dplyr::select(Estimate, Odds_Ratio) %>%
  mutate(Estimate=round(Estimate,2),
         Odds_Ratio=round(Odds_Ratio,2))
```

```
##                                          Estimate Odds_Ratio
## (Intercept)                                 -1.87       0.13
## PosOr.cont                                   0.09       0.52
## session2                                    -0.44       0.39
## session3                                    -0.74       0.32
## TokenTest_c                                 -0.08       0.48
## AAT_c                                       -1.13       0.24
## AAT_spontansprache_c                        -0.07       0.48
## LHoverall_c                                  0.00       0.50
## PosOr.cont:session2                         -0.05       0.49
## PosOr.cont:session3                         -0.04       0.49
## PosOr.cont:TokenTest_c                      -0.08       0.48
## PosOr.cont:AAT_c                            -0.06       0.48
## PosOr.cont:AAT_spontansprache_c              0.09       0.52
## PosOr.cont:LHoverall_c                      -0.05       0.49
## session2:TokenTest_c                        -0.03       0.49
## session3:TokenTest_c                        -0.20       0.45
## session2:AAT_c                              -0.04       0.49
## session3:AAT_c                               0.05       0.51
## session2:AAT_spontansprache_c               -0.02       0.50
## session3:AAT_spontansprache_c                0.23       0.56
## session2:LHoverall_c                         0.13       0.53
## session3:LHoverall_c                         0.18       0.54
## PosOr.cont:session2:TokenTest_c             -0.09       0.48
## PosOr.cont:session3:TokenTest_c             -0.22       0.45
## PosOr.cont:session2:AAT_c                    0.15       0.54
## PosOr.cont:session3:AAT_c                    0.29       0.57
## PosOr.cont:session2:AAT_spontansprache_c    -0.02       0.49
## PosOr.cont:session3:AAT_spontansprache_c    -0.07       0.48
## PosOr.cont:session2:LHoverall_c              0.11       0.53
## PosOr.cont:session3:LHoverall_c              0.04       0.51
```

Plot errors and Tests

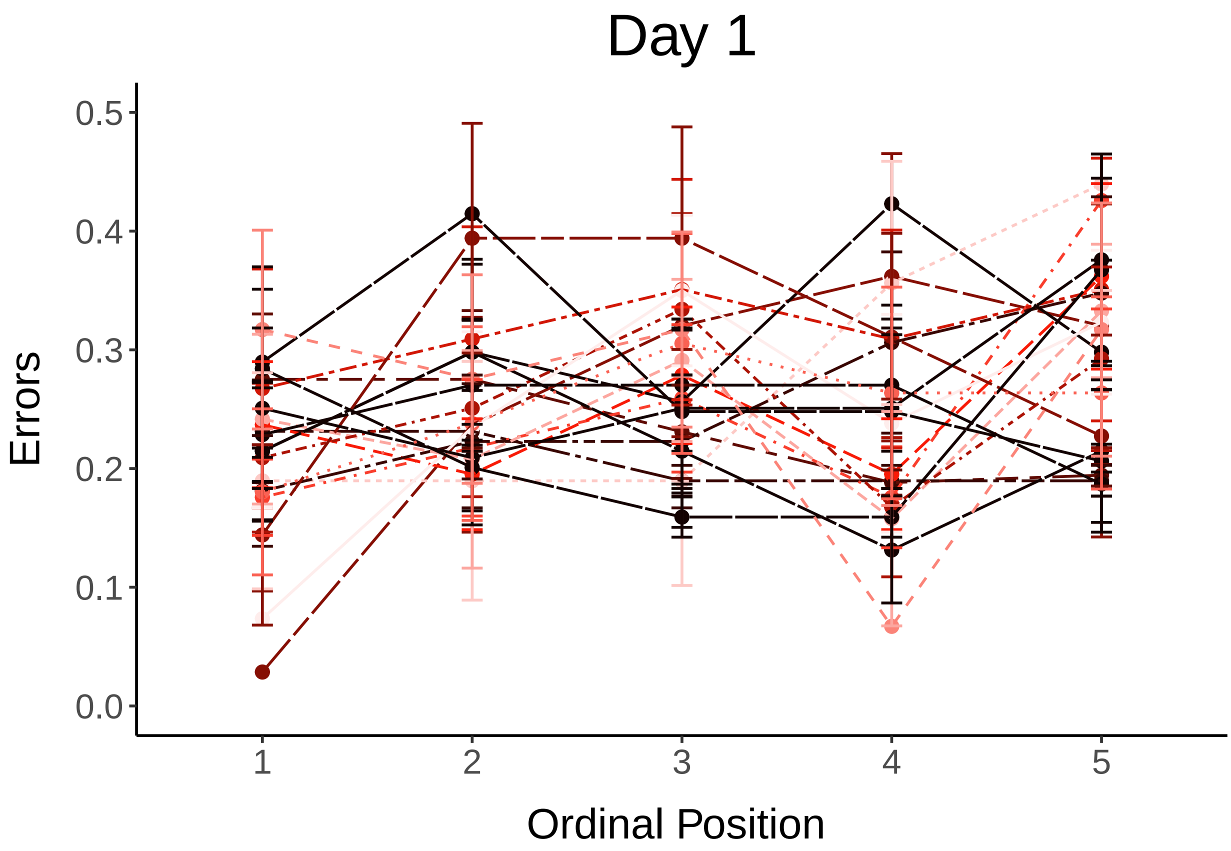
```
(means_final<- df_errors_PWA %>%
   filter(category != "Filler") %>%
   Rmisc::summarySEwithin(.,"error_class",idvar = "category",
                          withinvars = c("session", "PosOr"),
                          betweenvars = c("subject", "AAT", "TokenTest",
                                          "AAT_spontansprache"),
                          na.rm = T))
```

```
## Automatically converting the following non-factors to factors: AAT, TokenTest, AAT_spontansprache
```

```
##     subject AAT TokenTest AAT_spontansprache session PosOr  N  error_class
## 1       101  98       97                  3       1     1 23  0.275139242
## 2       101  98       97                  3       1     2 23  0.275139242
## 3       101  98       97                  3       1     3 24  0.230953597
## 4       101  98       97                  3       1     4 23  0.188182721
## 5       101  98       97                  3       1     5 23  0.194393901
## 6       101  98       97                  3       2     1 23  0.275139242
## 7       101  98       97                  3       2     2 24  0.189286931
## 8       101  98       97                  3       2     3 24  0.189286931
## 9       101  98       97                  3       2     4 24  0.189286931
## 10      101  98       97                  3       2     5 24  0.230953597
## 11      101  98       97                  3       3     1 24  0.230953597
## 12      101  98       97                  3       3     2 24  0.189286931
## 13      101  98       97                  3       3     3 24  0.189286931
## 14      101  98       97                  3       3     4 24  0.189286931
## 15      101  98        97                  3       3     5 24  0.189286931
## 16      102  99        99                  4       1     1 24  0.231350423
## 17      102  99        99                  4       1     2 24  0.231350423
## 18      102  99        99                  4       1     3 24  0.189683756
## 19      102  99        99                  4       1     4 24  0.189683756
## 20      102  99        99                  4       1     5 24  0.189683756
## 21      102  99        99                  4       2     1 24  0.231350423
## 22      102  99        99                  4       2     2 24  0.231350423
## 23      102  99        99                  4       2     3 24  0.231350423
## 24      102  99        99                  4       2     4 24  0.189683756
## 25      102  99        99                  4       2     5 24  0.189683756
## 26      102  99        99                  4       3     1 24  0.273017089
## 27      102  99        99                  4       3     2 24  0.189683756
## 28      102  99        99                  4       3     3 24  0.189683756
## 29      102  99        99                  4       3     4 24  0.231350423
## 30      102  99        99                  4       3     5 24  0.231350423
## 31      103  97        95                  4       1     1 24  0.028572645
## 32      103  97        95                  4       1     2 24  0.236905978
## 33      103  97        95                  4       1     3 24  0.320239311
## 34      103  97        95                  4       1     4 24  0.361905978
## 35      103  97        95                  4       1     5 24  0.320239311
## 36      103  97        95                  4       2     1 24  0.195239311
## 37      103  97        95                  4       2     2 24  0.070239311
## 38      103  97        95                  4       2     3 24  0.236905978
## 39      103  97        95                  4       2     4 24  0.195239311
## 40      103  97        95                  4       2     5 24  0.195239311
## 41      103  97        95                  4       3     1 24  0.153572645
## 42      103  97        95                  4       3     2 24  0.320239311
## 43      103  97        95                  4       3     3 24  0.195239311
## 44      103  97        95                  4       3     4 24  0.153572645
## 45      103  97        95                  4       3     5 24  0.236905978
## 46      104  42        39                  3       1     1 24  0.189683756
## 47      104  42        39                  3       1     2 24  0.189683756
## 48      104  42        39                  3       1     3 24  0.189683756
## 49      104  42        39                  3       1     4 24  0.356350423
## 50      104  42        39                  3       1     5 24  0.439683756
## 51      104  42        39                  3       2     1 24  0.189683756
## 52      104  42        39                  3       2     2 24  0.106350423
## 53      104  42        39                  3       2     3 24  0.273017089
## 54      104  42        39                  3       2     4 24  0.189683756
## 55      104  42        39                  3       2     5 24  0.314683756
## 56      104  42        39                  3       3     1 24  0.231350423
## 57      104  42        39                  3       3     2 24  0.064683756
## 58      104  42        39                  3       3     3 24  0.064683756
## 59      104  42        39                  3       3     4 24  0.189683756
## 60      104  42        39                  3       3     5 24  0.231350423
## 61      105  91        79                  4       1     1 24  0.175794867
## 62      105  91        79                  4       1     2 24  0.217461534
## 63      105  91        79                  4       1     3 24  0.259128200
## 64      105  91        79                  4       1     4 24  0.175794867
## 65      105  91        79                  4       1     5 24  0.425794867
## 66      105  91        79                  4       2     1 24  0.217461534
## 67      105  91        79                  4       2     2 24  0.217461534
## 68      105  91        79                  4       2     3 24  0.134128200
## 69      105  91        79                  4       2     4 24  0.217461534
## 70      105  91        79                  4       2     5 24  0.259128200
## 71      105  91        79                  4       3     1 24  0.134128200
## 72      105  91        79                  4       3     2 24  0.175794867
## 73      105  91        79                  4       3     3 24  0.175794867
## 74      105  91        79                  4       3     4 24  0.259128200
## 75      105  91        79                  4       3     5 24  0.175794867
## 76      106  94        86                  3       1     1 24  0.267461534
## 77      106  94        86                  3       1     2 24  0.309128200
## 78      106  94        86                  3       1     3 24  0.350794867
## 79      106  94        86                  3       1     4 24  0.309128200
## 80      106  94        86                  3       1     5 24  0.350794867
## 81      106  94        86                  3       2     1 24  0.267461534
## 82      106  94        86                  3       2     2 24  0.142461534
## 83      106  94        86                  3       2     3 24  0.184128200
## 84      106  94        86                  3       2     4 24  0.184128200
## 85      106  94        86                  3       2     5 24  0.309128200
## 86      106  94        86                  3       3     1 24  0.017461534
## 87      106  94        86                  3       3     2 24  0.184128200
## 88      106  94        86                  3       3     3 24  0.017461534
## 89      106  94        86                  3       3     4 24  0.142461534
## 90      106  94        86                  3       3     5 24  0.184128200
## 91      107  93        97                  4       1     1 24  0.236905978
## 92      107  93        97                  4       1     2 24  0.195239311
## 93      107  93        97                  4       1     3 24  0.278572645
## 94      107  93        97                  4       1     4 24  0.195239311
## 95      107  93        97                  4       1     5 24  0.361905978
## 96      107  93        97                  4       2     1 24  0.195239311
## 97      107  93        97                  4       2     2 24  0.195239311
## 98      107  93        97                  4       2     3 24  0.153572645
## 99      107  93        97                  4       2     4 24  0.195239311
## 100     107  93        97                  4       2     5 24  0.236905978
## 101     107  93        97                  4       3     1 24  0.153572645
## 102     107  93        97                  4       3     2 24  0.153572645
## 103     107  93        97                  4       3     3 24  0.195239311
## 104     107  93        97                  4       3     4 24  0.195239311
## 105     107  93        97                  4       3     5 24  0.278572645
## 106     108  99        93                  4       1     1 24  0.181152010
## 107     108  99        93                  4       1     2 24  0.222818677
## 108     108  99        93                  4       1     3 24  0.222818677
## 109     108  99        93                  4       1     4 24  0.306152010
## 110     108  99        93                  4       1     5 24  0.347818677
## 111     108  99        93                  4       2     1 24  0.181152010
## 112     108  99        93                  4       2     2 24  0.181152010
## 113     108  99        93                  4       2     3 24  0.181152010
## 114     108  99        93                  4       2     4 24  0.181152010
## 115     108  99        93                  4       2     5 23  0.226277959
## 116     108  99        93                  4       3     1 24  0.139485343
## 117     108  99        93                  4       3     2 24  0.222818677
## 118     108  99        93                  4       3     3 24  0.222818677
## 119     108  99        93                  4       3     4 24  0.139485343
## 120     108  99        93                  4       3     5 24  0.264485343
## 121     109  52        58                  3       1     1 23  0.241598870
## 122     109  52        58                  3       1     2 24  0.207739311
## 123     109  52        58                  3       1     3 24  0.291072645
## 124     109  52        58                  3       1     4 23  0.157747938
## 125     109  52        58                  3       1     5 24  0.332739311
## 126     109  52        58                  3       2     1 24  0.332739311
## 127     109  52        58                  3       2     2 24  0.249405978
## 128     109  52        58                  3       2     3 24  0.249405978
## 129     109  52        58                  3       2     4 24  0.249405978
## 130     109  52        58                  3       2     5 24  0.249405978
## 131     109  52        58                  3       3     1 24  0.166072645
## 132     109  52        58                  3       3     2 24  0.207739311
## 133     109  52        58                  3       3     3 23  0.095636137
## 134     109  52        58                  3       3     4 24  0.166072645
## 135     109  52        58                  3       3     5 23  0.008679615
## 136     110 100        97                  4       1     1 24  0.214683756
## 137     110 100        97                  4       1     2 24  0.298017089
## 138     110 100        97                  4       1     3 24  0.256350423
## 139     110 100        97                  4       1     4 24  0.423017089
## 140     110 100        97                  4       1     5 24  0.298017089
## 141     110 100        97                  4       2     1 24  0.173017089
## 142     110 100        97                  4       2     2 24  0.173017089
## 143     110 100        97                  4       2     3 24  0.173017089
## 144     110 100        97                  4       2     4 24  0.173017089
## 145     110 100        97                  4       2     5 24  0.214683756
## 146     110 100        97                  4       3     1 24  0.131350423
## 147     110 100        97                  4       3     2 24  0.131350423
## 148     110 100        97                  4       3     3 24  0.173017089
## 149     110 100        97                  4       3     4 24  0.173017089
## 150     110 100        97                  4       3     5 24  0.214683756
## 151     111  96        99                  4       1     1 24  0.209128200
## 152     111  96        99                  4       1     2 24  0.250794867
## 153     111  96        99                  4       1     3 24  0.334128200
## 154     111  96        99                  4       1     4 24  0.167461534
## 155     111  96        99                  4       1     5 24  0.292461534
## 156     111  96        99                  4       2     1 24  0.167461534
## 157     111  96        99                  4       2     2 24  0.292461534
## 158     111  96        99                  4       2     3 24  0.292461534
## 159     111  96        99                  4       2     4 24  0.084128200
## 160     111  96        99                  4       2     5 24  0.125794867
## 161     111  96        99                  4       3     1 24  0.209128200
## 162     111  96        99                  4       3     2 24  0.250794867
## 163     111  96        99                  4       3     3 24  0.292461534
## 164     111  96        99                  4       3     4 24  0.125794867
## 165     111  96        99                  4       3     5 24  0.125794867
## 166     112  28        15                  3       1     1 24  0.073474965
## 167     112  28        15                  3       1     2 23  0.234559532
## 168     112  28        15                  3       1     3 23  0.349944148
## 169     112  28        15                  3       1     4 23  0.238142905
## 170     112  28        15                  3       1     5 20  0.323255184
## 171     112  28        15                  3       2     1 24  0.115141631
## 172     112  28        15                  3       2     2 24  0.115141631
## 173     112  28        15                  3       2     3 24  0.240141631
## 174     112  28        15                  3       2     4 24  0.365141631
## 175     112  28        15                  3       2     5 24  0.365141631
## 176     112  28        15                  3       3     1 24 -0.051525035
## 177     112  28        15                  3       3     2 24  0.156808298
## 178     112  28        15                  3       3     3 24  0.240141631
## 179     112  28        15                  3       3     4 24  0.281808298
## 180     112  28        15                  3       3     5 24  0.198474965
## 181     113 100        99                  4       1     1 24  0.214683756
## 182     113 100        99                  4       1     2 24  0.298017089
## 183     113 100        99                  4       1     3 24  0.214683756
## 184     113 100        99                  4       1     4 24  0.131350423
## 185     113 100        99                  4       1     5 24  0.214683756
## 186     113 100        99                  4       2     1 24  0.214683756
## 187     113 100        99                  4       2     2 24  0.131350423
## 188     113 100        99                  4       2     3 24  0.214683756
## 189     113 100        99                  4       2     4 24  0.298017089
## 190     113 100        99                  4       2     5 24  0.256350423
## 191     113 100        99                  4       3     1 24  0.256350423
## 192     113 100        99                  4       3     2 24  0.214683756
## 193     113 100        99                  4       3     3 24  0.173017089
## 194     113 100        99                  4       3     4 24  0.089683756
## 195     113 100        99                  4       3     5 24  0.298017089
## 196     114  97        86                  3       1     1 24  0.144048835
## 197     114  97        86                  3       1     2 24  0.394048835
## 198     114  97        86                  3       1     3 24  0.394048835
## 199     114  97        86                  3       1     4 24  0.310715502
## 200     114  97        86                  3       1     5 24  0.227382169
## 201     114  97        86                  3       2     1 24  0.185715502
## 202     114  97        86                  3       2     2 24  0.310715502
## 203     114  97        86                  3       2     3 24  0.227382169
## 204     114  97        86                  3       2     4 24  0.185715502
## 205     114  97        86                  3       2     5 24  0.269048835
## 206     114  97        86                  3       3     1 23  0.097499491
## 207     114  97        86                  3       3     2 24  0.102382169
## 208     114  97        86                  3       3     3 24  0.144048835
## 209     114  97        86                  3       3     4 24  0.102382169
## 210     114  97        86                  3       3     5 22  0.111653453
## 211     115 100        83                  4       1     1 24  0.284128200
## 212     115 100        83                  4       1     2 24  0.200794867
## 213     115 100        83                  4       1     3 24  0.159128200
## 214     115 100        83                  4       1     4 24  0.159128200
## 215     115 100        83                  4       1     5 24  0.367461534
## 216     115 100        83                  4       2     1 24  0.200794867
## 217     115 100        83                  4       2     2 24  0.242461534
## 218     115 100        83                  4       2     3 24  0.159128200
## 219     115 100        83                  4       2     4 24  0.242461534
## 220     115 100        83                  4       2     5 24  0.200794867
## 221     115 100        83                  4       3     1 24  0.242461534
## 222     115 100        83                  4       3     2 24  0.200794867
## 223     115 100        83                  4       3     3 24  0.159128200
## 224     115 100        83                  4       3     4 24  0.159128200
## 225     115 100        83                  4       3     5 24  0.242461534
## 226     116 100        83                  3       1     1 24  0.228572645
## 227     116 100        83                  3       1     2 24  0.270239311
## 228     116 100        83                  3       1     3 24  0.270239311
## 229     116 100        83                  3       1     4 24  0.270239311
## 230     116 100        83                  3       1     5 24  0.186905978
## 231     116 100       83                  3       2     1 24  0.186905978
## 232     116 100       83                  3       2     2 24  0.186905978
## 233     116 100       83                  3       2     3 24  0.186905978
## 234     116 100       83                  3       2     4 24  0.228572645
## 235     116 100       83                  3       2     5 24  0.186905978
## 236     116 100       83                  3       3     1 24  0.186905978
## 237     116 100       83                  3       3     2 24  0.186905978
## 238     116 100       83                  3       3     3 24  0.228572645
## 239     116 100       83                  3       3     4 24  0.228572645
## 240     116 100       83                  3       3     5 24  0.186905978
## 241     117 100       99                  4       1     1 24  0.289683756
## 242     117 100       99                  4       1     2 24  0.414683756
## 243     117 100       99                  4       1     3 24  0.248017089
## 244     117 100       99                  4       1     4 24  0.248017089
## 245     117 100       99                  4       1     5 24  0.206350423
## 246     117 100       99                  4       2     1 24  0.206350423
## 247     117 100       99                  4       2     2 24  0.164683756
## 248     117 100       99                  4       2     3 24  0.164683756
## 249     117 100       99                  4       2     4 24  0.206350423
## 250     117 100       99                  4       2     5 24  0.248017089
## 251     117 100       99                  4       3     1 24  0.164683756
## 252     117 100       99                  4       3     2 24  0.206350423
## 253     117 100       99                  4       3     3 24  0.123017089
## 254     117 100       99                  4       3     4 24  0.206350423
## 255     117 100       99                  4       3     5 24  0.123017089
## 256     118 100       99                  4       1     1 24  0.250794867
## 257     118 100       99                  4       1     2 24  0.209128200
## 258     118 100       99                  4       1     3 24  0.250794867
## 259     118 100       99                  4       1     4 24  0.250794867
## 260     118 100       99                  4       1     5 24  0.375794867
## 261     118 100       99                  4       2     1 24  0.250794867
## 262     118 100       99                  4       2     2 24  0.125794867
## 263     118 100       99                  4       2     3 24  0.167461534
## 264     118 100       99                  4       2     4 24  0.250794867
## 265     118 100       99                  4       2     5 24  0.167461534
## 266     118 100       99                  4       3     1 24  0.167461534
## 267     118 100       99                  4       3     2 24  0.125794867
## 268     118 100       99                  4       3     3 24  0.167461534
## 269     118 100       99                  4       3     4 24  0.167461534
## 270     118 100       99                  4       3     5 24  0.292461534
## 271     119  83       74                  4       1     1 24  0.180358359
## 272     119  83       74                  4       1     2 23  0.237872162
## 273     119  83       74                  4       1     3 24  0.305358359
## 274     119  83       74                  4       1     4 24  0.263691692
## 275     119  83       74                  4       1     5 24  0.263691692
## 276     119  83       74                  4       2     1 24  0.138691692
## 277     119  83       74                  4       2     2 24  0.138691692
## 278     119  83       74                  4       2     3 24  0.305358359
## 279     119  83       74                  4       2     4 24  0.180358359
## 280     119  83       74                  4       2     5 24  0.388691692
## 281     119  83       74                  4       3     1 24  0.180358359
## 282     119  83       74                  4       3     2 24  0.138691692
## 283     119  83       74                  4       3     3 24  0.180358359
## 284     119  83       74                  4       3     4 24  0.180358359
## 285     119  83        74                  4       3      5 24  0.138691692
## 286     120  75        21                  3       1      1 24  0.317064708
## 287     120  75        21                  3       1      2 24  0.275398042
## 288     120  75        21                  3       1      3 24  0.317064708
## 289     120  75        21                  3       1      4 24  0.067064708
## 290     120  75        21                  3       1      5 24  0.317064708
## 291     120  75        21                  3       2      1 24  0.192064708
## 292     120  75        21                  3       2      2 24  0.150398042
## 293     120  75        21                  3       2      3 24  0.400398042
## 294     120  75        21                  3       2      4 24  0.067064708
## 295     120  75        21                  3       2      5 24  0.233731375
## 296     120  75        21                  3       3      1 24  0.025398042
## 297     120  75        21                  3       3      2 24  0.192064708
## 298     120  75        21                  3       3      3 23  0.023379408
## 299     120  75        21                  3       3      4 24  0.358731375
## 300     120  75        21                  3       3      5 24  0.275398042
##             sd          se         ci
## 1   0.26416256 0.055081702 0.11423246
## 2   0.27785430 0.057936626 0.12015321
## 3   0.19259716 0.039313732 0.08132665
## 4   0.05087420 0.010608003 0.02199965
## 5   0.04381574 0.009136214 0.01894735
## 6   0.27617995 0.057587501 0.11942917
## 7   0.05007002 0.010220500 0.02114272
## 8   0.05007002 0.010220500 0.02114272
## 9   0.05007002 0.010220500 0.02114272
## 10  0.19259716 0.039313732 0.08132665
## 11  0.20809817 0.042477861 0.08787215
## 12  0.05007002 0.010220500 0.02114272
## 13  0.05007002 0.010220500 0.02114272
## 14  0.05007002 0.010220500 0.02114272
## 15  0.05007002 0.010220500 0.02114272
## 16  0.21171683 0.043216517 0.08940018
## 17  0.19650149 0.040110699 0.08297530
## 18  0.06374282 0.013011448 0.02691623
## 19  0.06374282 0.013011448 0.02691623
## 20  0.06374282 0.013011448 0.02691623
## 21  0.21171683 0.043216517 0.08940018
## 22  0.21171683 0.043216517 0.08940018
## 23  0.16183472 0.033034374 0.06833681
## 24  0.06374282 0.013011448 0.02691623
## 25  0.06374282 0.013011448 0.02691623
## 26  0.23844953 0.048673306 0.10068841
## 27  0.06374282 0.013011448 0.02691623
## 28  0.06374282 0.013011448 0.02691623
## 29  0.16183472 0.033034374 0.06833681
## 30  0.16183472 0.033034374 0.06833681
## 31  0.33435403 0.068249730 0.14118532
## 32  0.44333886 0.090496166 0.18720558
## 33  0.45770001 0.093427624 0.19326977
## 34  0.50667559 0.103424721 0.21395034
## 35  0.50295881 0.102666037 0.21238088
## 36  0.43285329 0.088355809 0.18277792
## 37  0.38591488 0.078774546 0.16295756
## 38  0.43627762 0.089054797 0.18422388
## 39  0.43996949 0.089808396 0.18578282
## 40  0.49321658 0.100677413 0.20826710
## 41  0.43923360 0.089658183 0.18547208
## 42  0.48408060 0.098812538 0.20440931
## 43  0.49321658 0.100677413 0.20826710
## 44  0.45995619 0.093888164 0.19422247
## 45  0.42910020 0.087589712 0.18119312
## 46  0.44619973 0.091080139 0.18841362
## 47  0.49251646 0.100534501 0.20797146
## 48  0.43205537 0.088192933 0.18244098
## 49  0.50188568 0.102446985 0.21192774
## 50  0.43918450 0.089648160 0.18545135
## 51  0.38652912 0.078899927 0.16321693
## 52  0.49878216 0.101813482 0.21061723
## 53  0.42844632 0.087456240 0.18091702
## 54  0.47322185 0.096596007 0.19982406
## 55  0.47198975 0.096344505 0.19930379
## 56  0.49605066 0.101255916 0.20946382
## 57  0.45694548 0.093273606 0.19295115
## 58  0.42159610 0.086057943 0.17802442
## 59  0.41743201 0.085207952 0.17626608
## 60  0.47034193 0.096008145 0.19860798
## 61  0.15751258 0.032152121 0.06651173
## 62  0.28158025 0.057477329 0.11890091
## 63  0.30432811 0.062120715 0.12850649
## 64  0.20843165 0.042545933 0.08801297
## 65  0.44752213 0.091350071 0.18897202
## 66  0.28158025 0.057477329 0.11890091
## 67  0.30283626 0.061816193 0.12787654
## 68  0.09964568 0.020340089 0.04207668
## 69  0.28158025 0.057477329 0.11890091
## 70  0.28318410 0.057804713 0.11957816
## 71  0.09964568 0.020340089 0.04207668
## 72  0.15751258 0.032152121 0.06651173
## 73  0.22283387 0.045485774 0.09409449
## 74  0.30432811 0.062120715 0.12850649
## 75  0.23636014 0.048246811 0.09980613
## 76  0.49300665 0.100634561 0.20817845
## 77  0.46309637 0.094529151 0.19554845
## 78  0.45477871 0.092831315 0.19203621
## 79  0.44948402 0.091750542 0.18980046
## 80  0.54201494 0.110638336 0.22887284
## 81  0.47373201 0.096700141 0.20003948
## 82  0.42583090 0.086922368 0.17981262
## 83  0.44208201 0.090239613 0.18667486
## 84  0.38981822 0.079571312 0.16460580
## 85  0.47631986 0.097228384 0.20113224
## 86  0.36655137 0.074821984 0.15478107
## 87  0.43500038 0.088794080 0.18368455
## 88  0.31159721 0.063604515 0.13157596
## 89  0.44017532 0.089850411 0.18586974
## 90  0.35652984 0.072776349 0.15054935
## 91  0.26032840 0.053139312 0.10992704
## 92  0.22885001 0.046713812 0.09663488
## 93  0.28158025 0.057477329 0.11890091
## 94  0.22885001 0.046713812 0.09663488
## 95  0.38285144 0.078149223 0.16166398
## 96  0.18368121 0.037493769 0.07756177
## 97  0.22885001 0.046713812 0.09663488
## 98  0.09081336 0.018537200 0.03834712
## 99  0.22885001 0.046713812 0.09663488
## 100 0.22166943 0.045248083 0.09360279
## 101 0.09081336 0.018537200 0.03834712
## 102 0.09081336 0.018537200 0.03834712
## 103 0.16591445 0.033867146 0.07005953
## 104 0.22885001 0.046713812 0.09663488
## 105 0.31292328 0.063875197 0.13213591
## 106 0.22845068 0.046632300 0.09646626
## 107 0.27406555 0.055943396 0.11572773
## 108 0.27406555 0.055943396 0.11572773
## 109 0.37406234 0.076355155 0.15795267
## 110 0.39730809 0.081100174 0.16776849
## 111 0.19941754 0.040705934 0.08420664
## 112 0.21442606 0.043769536 0.09054418
## 113 0.18318344 0.037392164 0.07735158
## 114 0.21442606 0.043769536 0.09054418
## 115 0.27965050 0.058311161 0.12092995
## 116 0.08217039 0.016772961 0.03469751
## 117 0.29586194 0.060392565 0.12493154
## 118 0.26248952 0.053580448 0.11083960
## 119 0.08217039 0.016772961 0.03469751
## 120 0.31680172 0.064666881 0.13377363
## 121 0.34298663 0.071517656 0.14831854
## 122 0.44883802 0.091618677 0.18952767
## 123 0.33469683 0.068319705 0.14133008
## 124 0.43300244 0.090287251 0.18724430
## 125 0.27550136 0.056236480 0.11633402
## 126 0.36304493 0.074106236 0.15330043
## 127 0.39358441 0.080340082 0.16619612
## 128 0.39358441 0.080340082 0.16619612
## 129 0.37923251 0.077410512 0.16013584
## 130 0.37570648 0.076690765 0.15864693
## 131 0.42244457 0.086231137 0.17838270
## 132 0.43222093 0.088226727 0.18251089
## 133 0.45822238 0.095545972 0.19815022
## 134 0.40199564 0.082057017 0.16974787
## 135 0.44054150 0.091859252 0.19050443
## 136 0.28921254 0.059035262 0.12212374
## 137 0.36301589 0.074100309 0.15328817
## 138 0.34038859 0.069481529 0.14373349
## 139 0.41817532 0.085359680 0.17657995
## 140 0.37974057 0.077514219 0.16035038
## 141 0.20329837 0.041498106 0.08584537
## 142 0.21803992 0.044507213 0.09207018
## 143 0.18740077 0.038253021 0.07913240
## 144 0.21803992 0.044507213 0.09207018
## 145 0.25497128 0.052045794 0.10766493
## 146 0.08693064 0.017744642 0.03670759
## 147 0.08693064 0.017744642 0.03670759
## 148 0.21803992 0.044507213 0.09207018
## 149 0.20329837 0.041498106 0.08584537
## 150 0.28921254 0.059035262 0.12212374
## 151 0.30720585 0.062708131 0.12972165
## 152 0.36513656 0.074533188 0.15418365
## 153 0.39509370 0.080648164 0.16683344
## 154 0.28740228 0.058665744 0.12135934
## 155 0.37887633 0.077337807 0.15998544
## 156 0.22703341 0.046343000 0.09586780
## 157 0.37058885 0.075646132 0.15648595
## 158 0.36211174 0.073915750 0.15290638
## 159 0.12785727 0.026098756 0.05398939
## 160 0.24643770 0.050303884 0.10406151
## 161 0.37955879 0.077477113 0.16027362
## 162 0.38176834 0.077928136 0.16120663
## 163 0.37058885 0.075646132 0.15648595
## 164 0.25873291 0.052813634 0.10925333
## 165 0.24643770 0.050303884 0.10406151
## 166 0.45775172 0.093438178 0.19329160
## 167 0.38670665 0.080633911 0.16722450
## 168 0.30412939 0.063415362 0.13151541
## 169 0.43961919 0.091666939 0.19010560
## 170 0.27114954 0.060630879 0.12690189
## 171 0.42747381 0.087257726 0.18050636
## 172 0.44025589 0.089866858 0.18590376
## 173 0.35696806 0.072865801 0.15073439
## 174 0.35421482 0.072303798 0.14957180
## 175 0.32138039 0.065601498 0.13570704
## 176 0.45773774 0.093435325 0.19328570
## 177 0.39820525 0.081283306 0.16814733
## 178 0.36556447 0.074620534 0.15436434
## 179 0.38522945 0.078634632 0.16266813
## 180 0.39528600 0.080687417 0.16691464
## 181 0.28342770 0.057854438 0.11968102
## 182 0.38353807 0.078289380 0.16195392
## 183 0.31458672 0.064214745 0.13283832
## 184 0.21845495 0.044591931 0.09224544
## 185 0.29418097 0.060049439 0.12422173
## 186 0.28342770 0.057854438 0.11968102
## 187 0.24524463 0.050060350 0.10355772
## 188 0.29418097 0.060049439 0.12422173
## 189 0.35842420 0.073163034 0.15134927
## 190 0.40279176 0.082219523 0.17008404
## 191 0.37067031 0.075662761 0.15652035
## 192 0.28342770 0.057854438 0.11968102
## 193 0.28547480 0.058272299 0.12054543
## 194 0.14996549 0.030611577 0.06332487
## 195 0.36698650 0.074910805 0.15496481
## 196 0.37209081 0.075952719 0.15712017
## 197 0.47407026 0.096769186 0.20018231
## 198 0.45933570 0.093761506 0.19396045
## 199 0.42826240 0.087418697 0.18083935
## 200 0.41662251 0.085042714 0.17592426
## 201 0.35529221 0.072523718 0.15002674
## 202 0.44949107 0.091751981 0.18980343
## 203 0.44693400 0.091230021 0.18872368
## 204 0.34835663 0.071107999 0.14709810
## 205 0.42398940 0.086546474 0.17903502
## 206 0.28284247 0.058976733 0.12231026
## 207 0.30946437 0.063169149 0.13067534
## 208 0.31809510 0.064930890 0.13431978
## 209 0.30946437 0.063169149 0.13067534
## 210 0.27662304 0.058976230 0.12264778
## 211 0.32778772 0.066909388 0.13841262
## 212 0.17853692 0.036443696 0.07538953
## 213 0.08307413 0.016957436 0.03507913
## 214 0.08307413 0.016957436 0.03507913
## 215 0.37767907 0.077093417 0.15947988
## 216 0.21047041 0.042962092 0.08887386
## 217 0.24326682 0.049656631 0.10272257
## 218 0.08307413 0.016957436 0.03507913
## 219 0.26758383 0.054620320 0.11299074
## 220 0.22474201 0.045875271 0.09490023
## 221 0.26758383 0.054620320 0.11299074
## 222 0.19515791 0.039836442 0.08240796
## 223 0.08307413 0.016957436 0.03507913
## 224 0.08307413 0.016957436 0.03507913
## 225 0.26758383 0.054620320 0.11299074
## 226 0.19302449 0.039400959 0.08150709
## 227 0.27275655 0.055676198 0.11517499
## 228 0.27275655 0.055676198 0.11517499
## 229 0.27275655 0.055676198 0.11517499
## 230 0.04949713 0.010103559 0.02090080
## 231 0.04949713 0.010103559 0.02090080
## 232 0.04949713 0.010103559 0.02090080
## 233 0.04949713 0.010103559 0.02090080
## 234 0.20849373 0.042558604 0.08803918
## 235 0.04949713 0.010103559 0.02090080
## 236 0.04949713 0.010103559 0.02090080
## 237 0.04949713 0.010103559 0.02090080
## 238 0.19302449 0.039400959 0.08150709
## 239 0.19302449 0.039400959 0.08150709
## 240 0.04949713 0.010103559 0.02090080
## 241 0.39316758 0.080254996 0.16602011
## 242 0.44021452 0.089858412 0.18588629
## 243 0.33621939 0.068630496 0.14197300
## 244 0.31720832 0.064749877 0.13394533
## 245 0.29369672 0.059950592 0.12401725
## 246 0.26004658 0.053081786 0.10980804
## 247 0.20776839 0.042410544 0.08773289
## 248 0.19224079 0.039240987 0.08117617
## 249 0.26004658 0.053081786 0.10980804
## 250 0.33621939 0.068630496 0.14197300
## 251 0.20776839 0.042410544 0.08773289
## 252 0.27172671 0.055465982 0.11474013
## 253 0.09283325 0.018949508 0.03920004
## 254 0.27172671 0.055465982 0.11474013
## 255 0.09283325 0.018949508 0.03920004
## 256 0.33124345 0.067614786 0.13987184
## 257 0.27746014 0.056636313 0.11716114
## 258 0.33124345 0.067614786 0.13987184
## 259 0.33124345 0.067614786 0.13987184
## 260 0.43687042 0.089175801 0.18447420
## 261 0.31192926 0.063672293 0.13171617
## 262 0.07793044 0.015907485 0.03290714
## 263 0.20090805 0.041010184 0.08483603
## 264 0.34048994 0.069502219 0.14377629
## 265 0.20090805 0.041010184 0.08483603
## 266 0.20090805 0.041010184 0.08483603
## 267 0.07793044 0.015907485 0.03290714
## 268 0.22975292 0.046898118 0.09701615
## 269 0.22975292 0.046898118 0.09701615
## 270 0.38051217 0.077671721 0.16067620
## 271 0.34272581 0.069958612 0.14472042
## 272 0.39125102 0.081581476 0.16918963
## 273 0.45371729 0.092614654 0.19158801
## 274 0.43660875 0.089122388 0.18436371
## 275 0.39667748 0.080971451 0.16750221
## 276 0.30096982 0.061435208 0.12708841
## 277 0.30096982 0.061435208 0.12708841
## 278 0.41543375 0.084800059 0.17542229
## 279 0.31082014 0.063445895 0.13124783
## 280 0.45147088 0.092156108 0.19063943
## 281 0.31436831 0.064170164 0.13274610
## 282 0.29046799 0.059291531 0.12265388
## 283 0.33019921 0.067401632 0.13943090
## 284 0.33354130 0.068083832 0.14084214
## 285 0.27957195 0.057067385 0.11805288
## 286 0.41026926 0.083745863 0.17324152
## 287 0.43038964 0.087852918 0.18173761
## 288 0.40262847 0.082186193 0.17001509
## 289 0.49884304 0.101825910 0.21064294
## 290 0.52251662 0.106658259 0.22063942
## 291 0.38119054 0.077810193 0.16096265
## 292 0.46021639 0.093941278 0.19433234
## 293 0.50940369 0.103981593 0.21510231
## 294 0.43925364 0.089662274 0.18548055
## 295 0.45219333 0.092303578 0.19094450
## 296 0.36996861 0.075519525 0.15622404
## 297 0.37295449 0.076129015 0.15748487
## 298 0.44881004 0.093583363 0.19408002
## 299 0.41517373 0.084746983 0.17531249
## 300 0.43754591 0.089313685 0.18475943
```

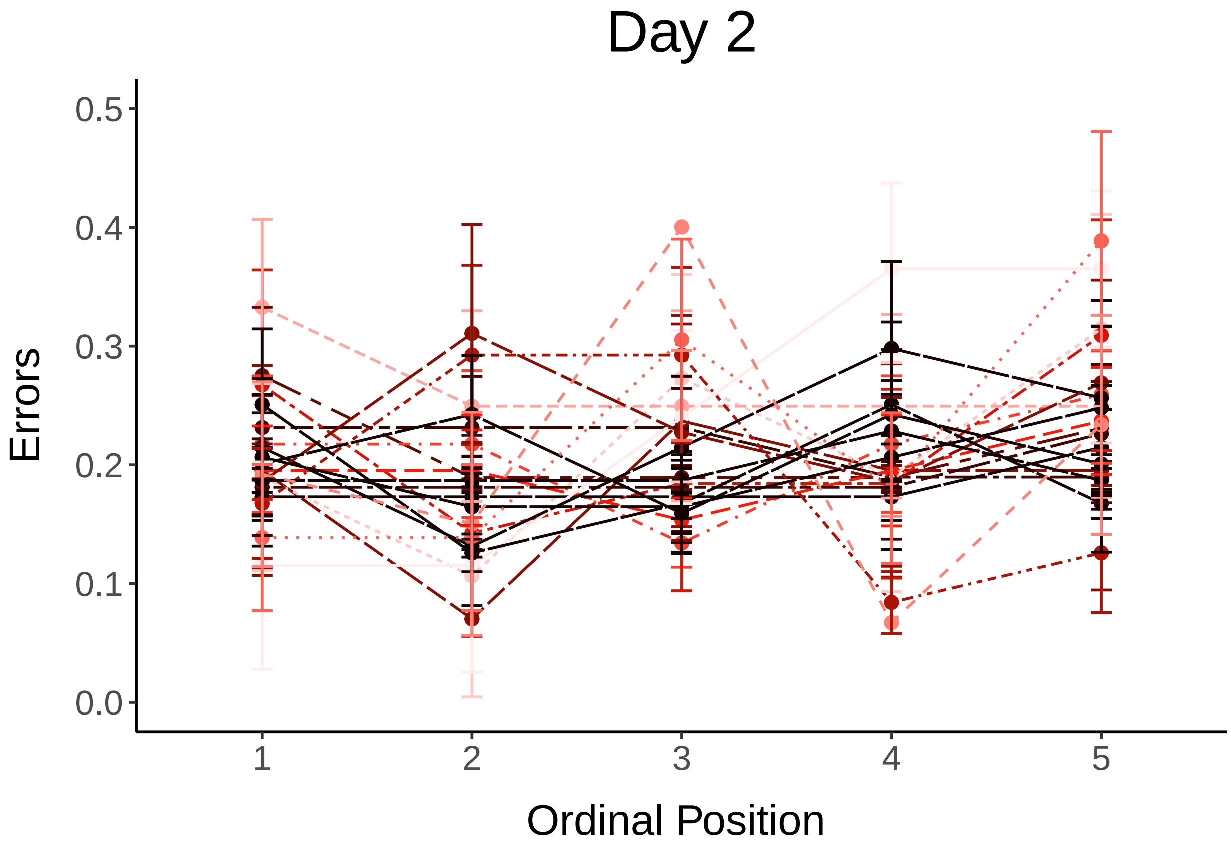
```
override.linetype<-c("dashed")
(plot_error_repetition_PWA_session1_Naming <-
    means_final %>% filter(session=="1") %>%
    ggplot(., aes(x=PosOr, y=error_class, group=subject, color = AAT)) +
    geom_point(size = 2)+
    stat_summary(aes(linetype=AAT, color=AAT),
                 fun=mean,  geom="line", size = 0.5) +
    scale_linetype_manual(values=c("dashed"))+
    scale_colour_manual(values=c("#feedec", "#fdcac6", "#fca7a0",
                                          "#fb8479", "#fa6153",
                                          "#f93e2d", "#f81b07",
                                          "#d21706", "#ac1305",
                                          "#860f04", "#5f0b03",
                                          "#390602", "#130201"))+
    geom_errorbar(
      aes(ymin=error_class-se, ymax=error_class+se, group = AAT),
                  width =.1) +
    apatheme+
    scale_y_continuous(limits = c(0, 0.5),
                        breaks =seq(0,0.5, by = 0.1)) +
    labs(x="Ordinal Position ",y ="Errors", colour="Session",
         linetype="Session",
         title = "Day 1") +
    theme(
    axis.title.y = element_text(margin = margin(0,10,0,0)),
    axis.title.x = element_text(margin = margin(10,0,0,0)),
    legend.key.width = unit(1, "cm"),
    legend.position="none")+
   guides(color=guide_legend(
     override.aes=list(linetype=override.linetype)))+
    scale_linetype(guide="none"))
```

```
## Scale for linetype is already present.
## Adding another scale for linetype, which will replace the existing scale.
```



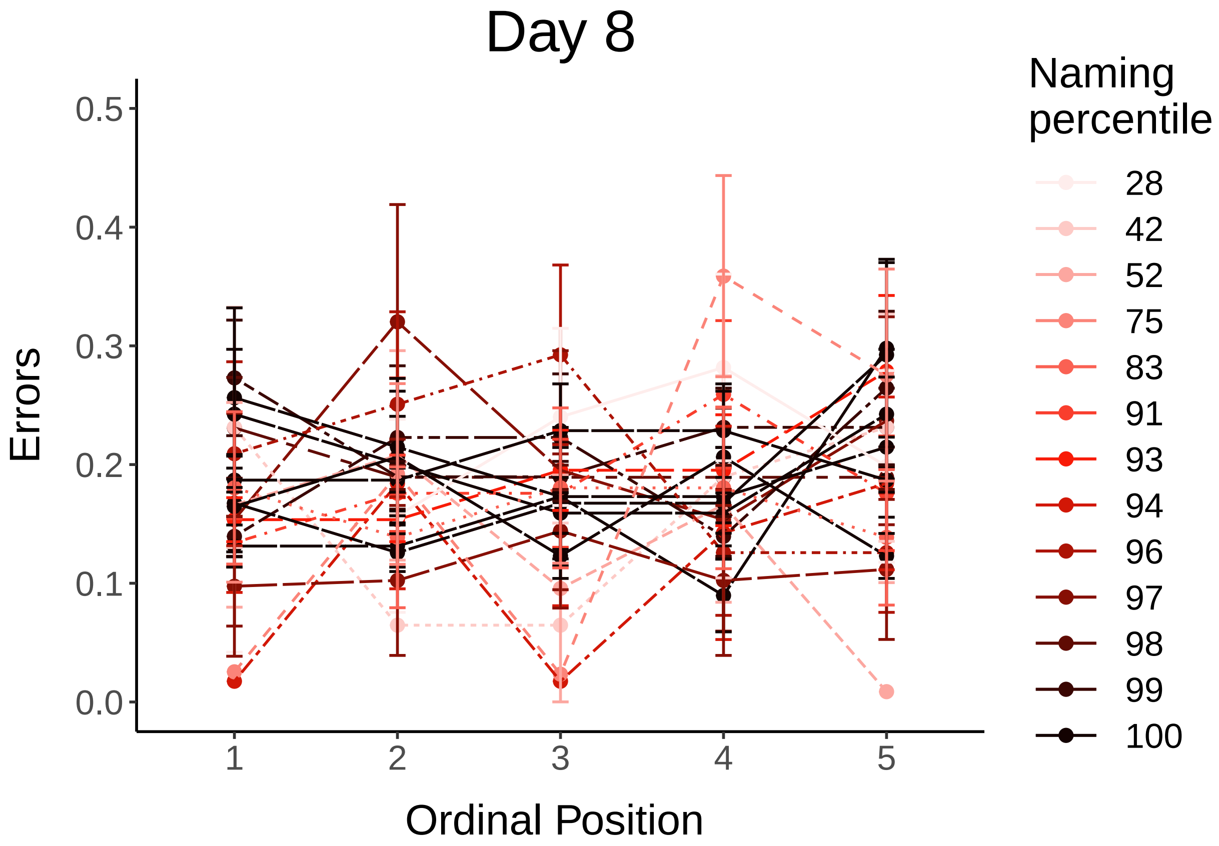
```
(plot_error_repetition_PWA_session2_Naming <- means_final %>%
    filter(session=="2") %>%
    ggplot(., aes(x=PosOr, y=error_class, group=subject, color = AAT)) +
    geom_point(size = 2)+
    stat_summary(aes(linetype=AAT, color=AAT),
                 fun=mean,  geom="line", size = 0.5) +
    scale_linetype_manual(values=c("dashed"))+
    scale_colour_manual(values=c("#feedec", "#fdcac6", "#fca7a0",
                                      "#fb8479", "#fa6153",
                                      "#f93e2d", "#f81b07",
                                      "#d21706", "#ac1305",
                                      "#860f04", "#5f0b03",
                                      "#390602", "#130201"))+
    geom_errorbar(aes(ymin=error_class-se, ymax=error_class+se,
                      group = AAT), width =.1) +
    apatheme+
    scale_y_continuous(limits = c(0, 0.5),
                       breaks =seq(0,0.5, by = 0.1)) +
    labs(x="Ordinal Position ",y ="Errors",
         colour="Session", linetype="Session",
         title = "Day 2") +
    theme(
    axis.title.y = element_text(margin = margin(0,10,0,0)),
    axis.title.x = element_text(margin = margin(10,0,0,0)),
    legend.key.width = unit(1, "cm"),
    legend.position="none")+
   guides(color=guide_legend(
     override.aes=list(linetype=override.linetype)))+
    scale_linetype(guide="none"))
```

```
## Scale for linetype is already present.
## Adding another scale for linetype, which will replace the existing scale.
```



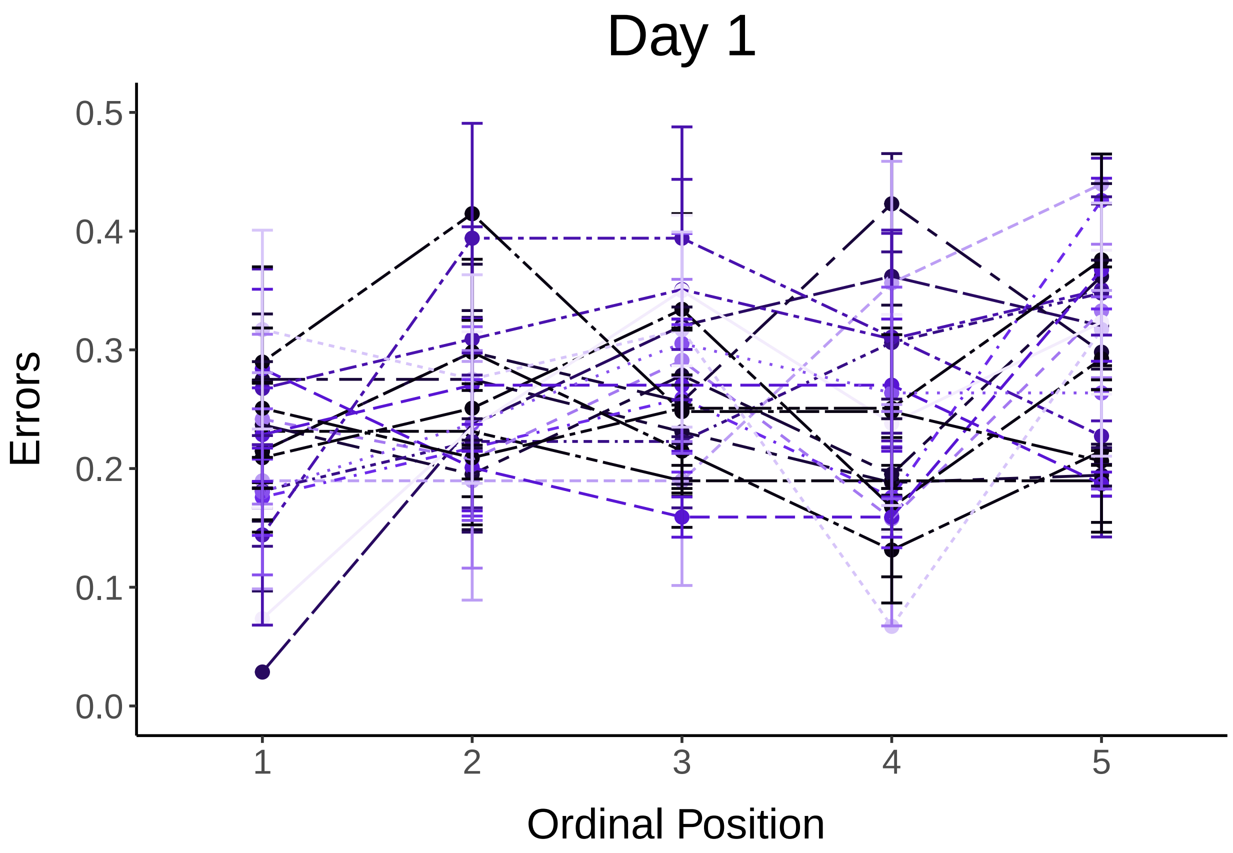
```
(plot_error_repetition_PWA_session3_Naming <- means_final %>%
    filter(session=="3") %>%
    ggplot(., aes(x=PosOr, y=error_class, group=subject, color = AAT)) +
    geom_point(size = 2)+
    stat_summary(aes(linetype=AAT, color=AAT),
                 fun=mean,  geom="line", size = 0.5) +
    scale_linetype_manual(values=c("dashed"))+
    scale_colour_manual(values=c("#feedec", "#fdcac6", "#fca7a0",
                                          "#fb8479", "#fa6153",
                                          "#f93e2d", "#f81b07",
                                          "#d21706", "#ac1305",
                                          "#860f04", "#5f0b03",
                                          "#390602", "#130201"))+
    geom_errorbar(aes(ymin=error_class-se, ymax=error_class+se,
                      group = AAT), width =.1) +
    apatheme+
    scale_y_continuous(limits = c(0, 0.5),
                         breaks =seq(0,0.5, by = 0.1)) +
    labs(x="Ordinal Position ",y ="Errors", colour="Naming\npercentile",
        # linetype="Session",
         title = "Day 8") +
    theme(
    axis.title.y = element_text(margin = margin(0,10,0,0)),
    axis.title.x = element_text(margin = margin(10,0,0,0)),
    legend.key.width = unit(1, "cm"),
    legend.position="right")+
    scale_linetype(guide="none"))
```

```
## Scale for linetype is already present.
## Adding another scale for linetype, which will replace the existing scale.

## Warning: Removed 1 rows containing non-finite values (‘stat_summary()‘).

## Warning: Removed 1 rows containing missing values (‘geom_point()‘).
```



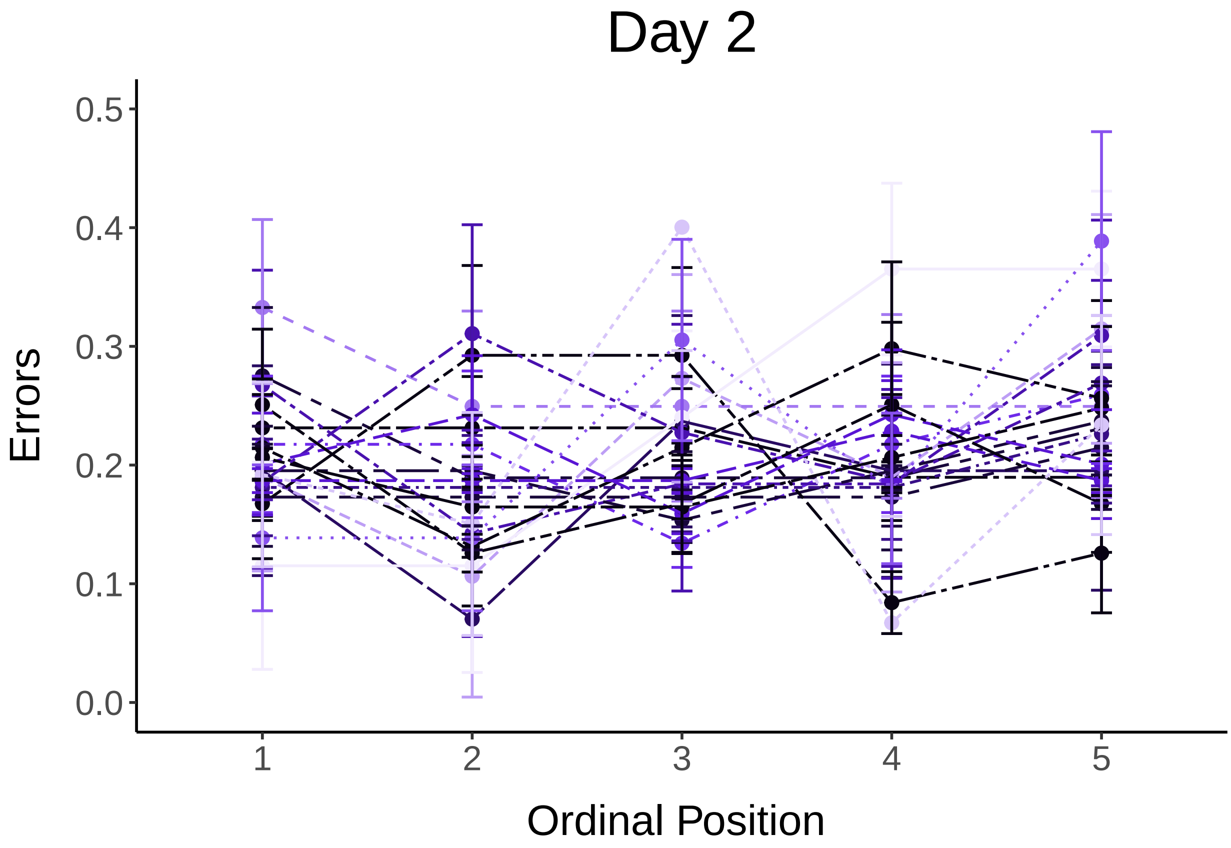
```
(plot_error_repetition_PWA_session1_Token <-
    means_final %>% filter(session=="1") %>%
    ggplot(., aes(x=PosOr, y=error_class, group=subject, color = TokenTest)) +
    geom_point(size = 2)+
    stat_summary(aes(linetype=TokenTest, color=TokenTest),
                 fun=mean,  geom="line", size = 0.5) +
    scale_linetype_manual(values=c("dashed"))+
    scale_colour_manual(values=c("#f2ecfd", "#d7c5f9", "#bd9ef5",
                                          "#a378f1", "#8851ee",
                                          "#6e2aea", "#5915d4",
                                          "#4911ae", "#380e87",
                                          "#280a61", "#18063a",
                                          "#080213"))+
    geom_errorbar(aes(ymin=error_class-se, ymax=error_class+se, group = TokenTest),
                  width =.1) +
    apatheme+
    scale_y_continuous(limits = c(0, 0.5),
                        breaks =seq(0,0.5, by = 0.1)) +
    labs(x="Ordinal Position ",y ="Errors", colour="Session",
         linetype="Session",
         title = "Day 1") +
    theme(
    axis.title.y = element_text(margin = margin(0,10,0,0)),
    axis.title.x = element_text(margin = margin(10,0,0,0)),
    legend.key.width = unit(1, "cm"),
    legend.position="none")+
   guides(color=guide_legend(
     override.aes=list(linetype=override.linetype))+
    scale_linetype(guide="none"))
```

```
## Scale for linetype is already present.
## Adding another scale for linetype, which will replace the existing scale.
```



```
(plot_error_repetition_PWA_session2_Token <- means_final %>%
    filter(session=="2") %>%
    ggplot(., aes(x=PosOr, y=error_class, group=subject, color = TokenTest)) +
    geom_point(size = 2)+
    stat_summary(aes(linetype=TokenTest, color=TokenTest),
                 fun=mean,  geom="line", size = 0.5) +
    scale_linetype_manual(values=c("dashed"))+
    scale_colour_manual(values=c("#f2ecfd", "#d7c5f9", "#bd9ef5",
                                        "#a378f1", "#8851ee",
                                        "#6e2aea", "#5915d4",
                                        "#4911ae", "#380e87",
                                        "#280a61", "#18063a",
                                        "#080213"))+
    geom_errorbar(aes(ymin=error_class-se, ymax=error_class+se,
                      group = TokenTest), width =.1) +
    apatheme+
    scale_y_continuous(limits = c(0, 0.5),
                       breaks =seq(0,0.5, by = 0.1)) +
    labs(x="Ordinal Position ",y ="Errors",
         colour="Session", linetype="Session",
         title = "Day 2") +
    theme(
    axis.title.y = element_text(margin = margin(0,10,0,0)),
    axis.title.x = element_text(margin = margin(10,0,0,0)),
    legend.key.width = unit(1, "cm"),
    legend.position="none")+
   guides(color=guide_legend(
     override.aes=list(linetype=override.linetype)))+
    scale_linetype(guide="none"))
```

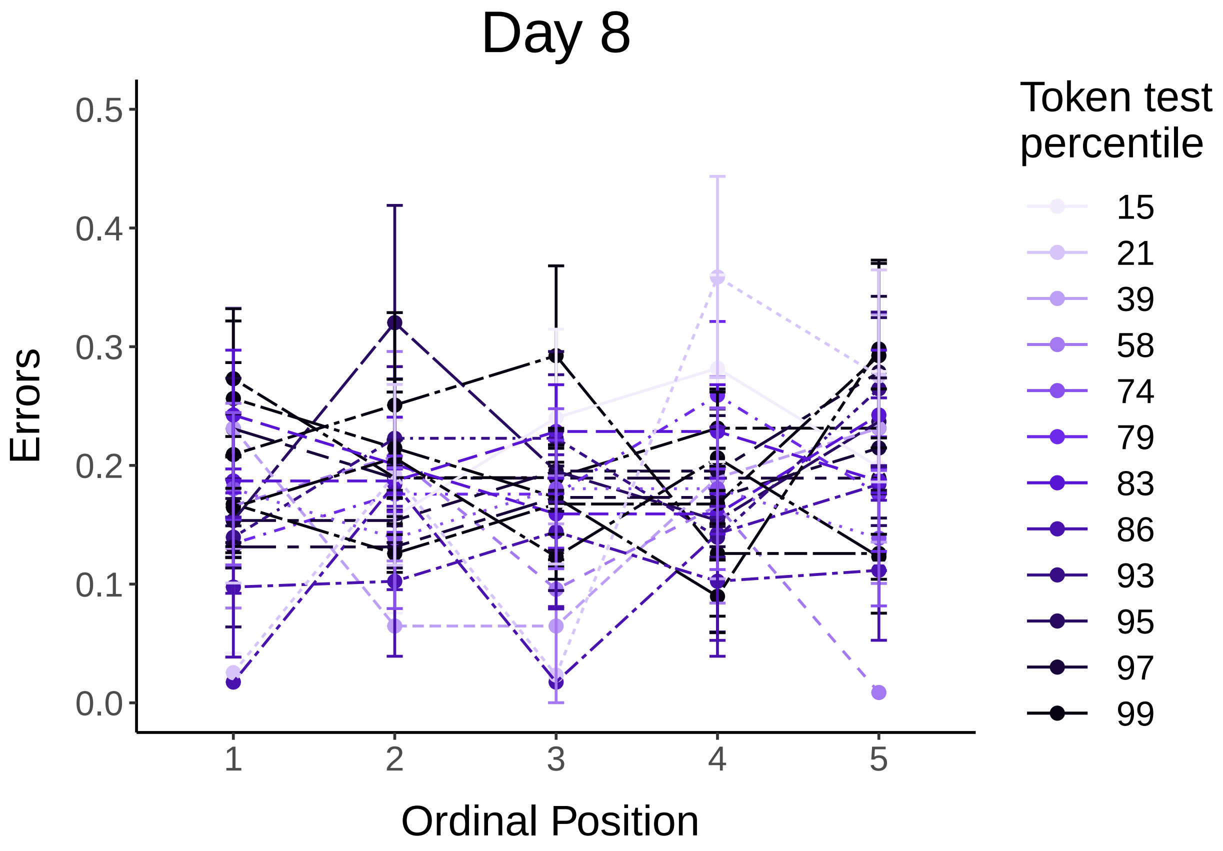
```
## Scale for linetype is already present.
## Adding another scale for linetype, which will replace the existing scale.
```



```
(plot_error_repetition_PWA_session3_Token <- means_final %>%
    filter(session=="3") %>%
    ggplot(., aes(x=PosOr, y=error_class, group=subject, color = TokenTest)) +
    geom_point(size = 2)+
    stat_summary(aes(linetype=TokenTest, color=TokenTest),
                 fun=mean,  geom="line", size = 0.5) +
    scale_linetype_manual(values=c("dashed"))+
    scale_colour_manual(values=c("#f2ecfd", "#d7c5f9", "#bd9ef5",
                                         "#a378f1", "#8851ee",
                                         "#6e2aea", "#5915d4",
                                         "#4911ae", "#380e87",
                                         "#280a61", "#18063a",
                                         "#080213"))+
    geom_errorbar(aes(ymin=error_class-se, ymax=error_class+se,
                      group = TokenTest), width =.1) +
    apatheme+
    scale_y_continuous(limits = c(0, 0.5),
                         breaks =seq(0,0.5, by = 0.1)) +
    labs(x="Ordinal Position ",y ="Errors", colour="Token test\npercentile",
        # linetype="Session",
         title = "Day 8") +
    theme(
    axis.title.y = element_text(margin = margin(0,10,0,0)),
    axis.title.x = element_text(margin = margin(10,0,0,0)),
    legend.key.width = unit(1, "cm"),
    legend.position="right")+
    scale_linetype(guide="none"))
```

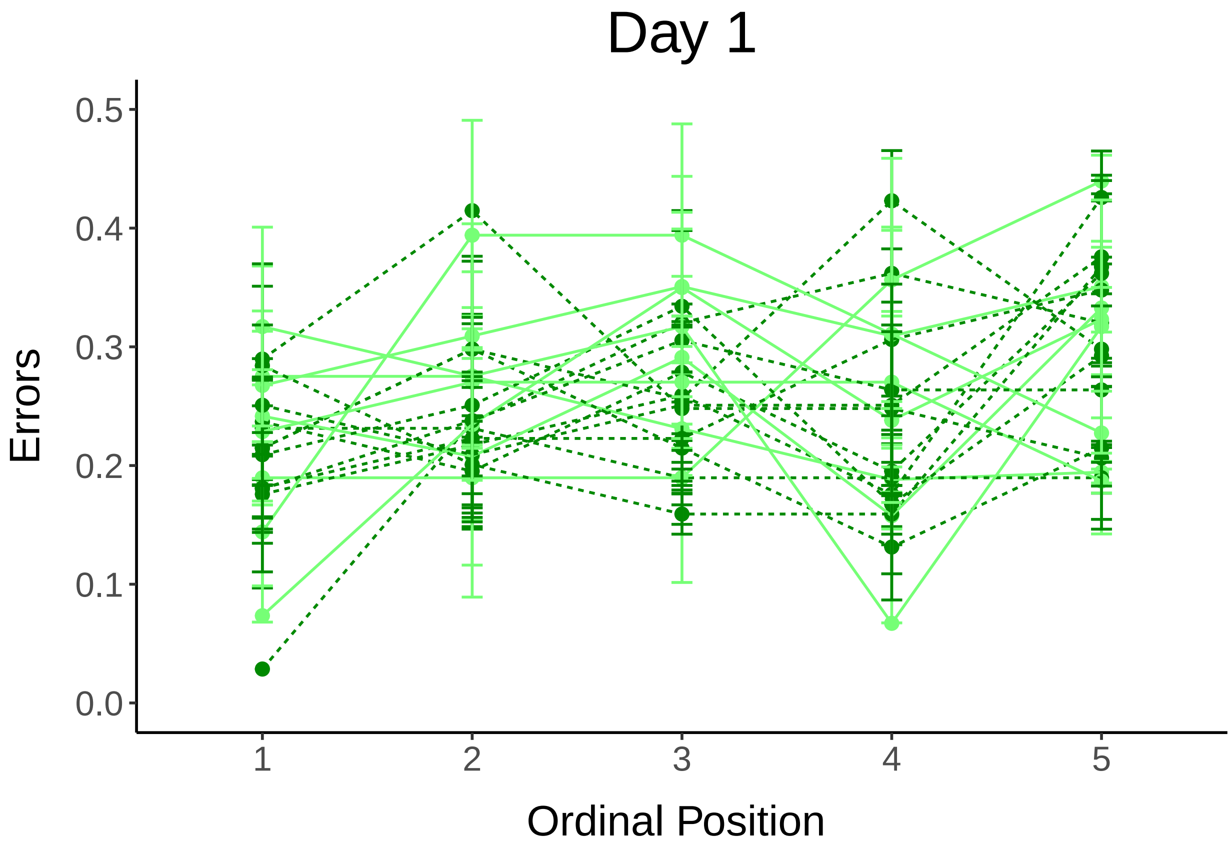
```
## Scale for linetype is already present.
## Adding another scale for linetype, which will replace the existing scale.
```

```
## Warning: Removed 1 rows containing non-finite values (‘stat_summary()‘).
## Removed 1 rows containing missing values (‘geom_point()‘).
```



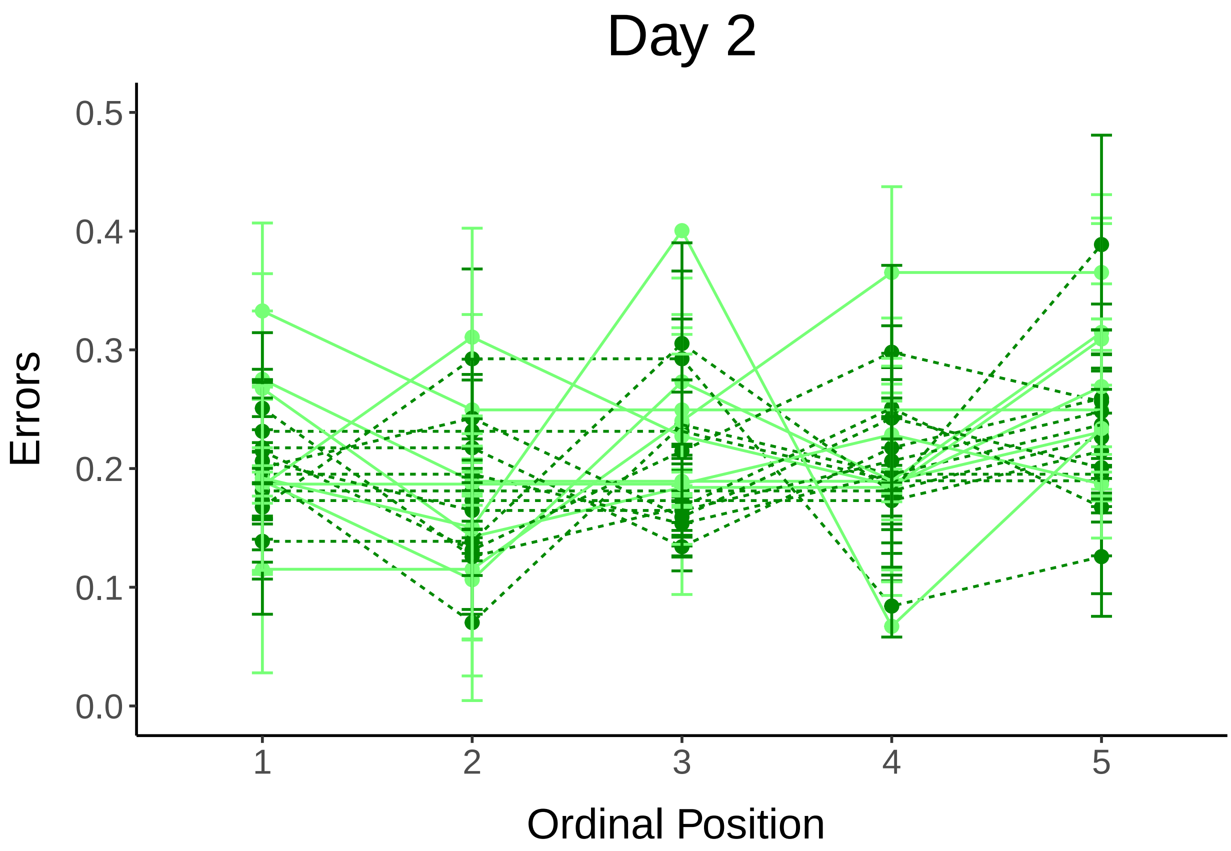
```
(plot_error_repetition_PWA_session1_Speech <-
    means_final %>% filter(session=="1") %>%
    ggplot(., aes(x=PosOr, y=error_class, group=subject,
                  color = AAT_spontansprache)) +
    geom_point(size = 2)+
    stat_summary(aes(linetype=AAT_spontansprache,
                     color=AAT_spontansprache),
                 fun=mean,  geom="line", size = 0.5) +
    scale_linetype_manual(values=c("dashed"))+
    scale_colour_manual(values=c("#76ff76", "#008900"))+
    geom_errorbar(aes(ymin=error_class-se, ymax=error_class+se,
                      group = AAT_spontansprache),
                  width =.1) +
    apatheme+
    scale_y_continuous(limits = c(0, 0.5),
                        breaks =seq(0,0.5, by = 0.1)) +
    labs(x="Ordinal Position ",y ="Errors", colour="Session",
         linetype="Session",
         title = "Day 1") +
    theme(
    axis.title.y = element_text(margin = margin(0,10,0,0)),
    axis.title.x = element_text(margin = margin(10,0,0,0)),
    legend.key.width = unit(1, "cm"),
    legend.position="none")+
   guides(color=guide_legend(
     override.aes=list(linetype=override.linetype)))+
    scale_linetype(guide="none"))
```

```
## Scale for linetype is already present.
## Adding another scale for linetype, which will replace the existing scale.
```



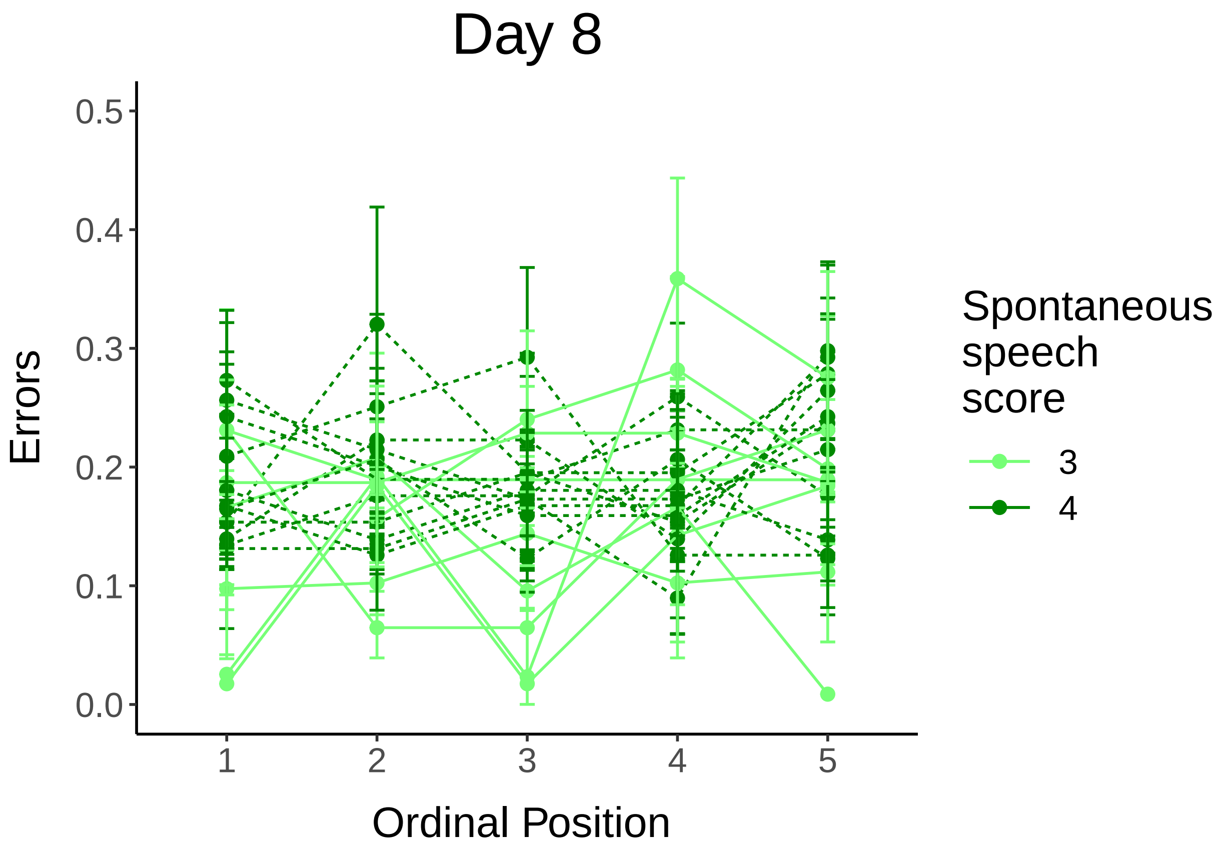
```
(plot_error_repetition_PWA_session2_Speech <- means_final %>%
    filter(session=="2") %>%
    ggplot(., aes(x=PosOr, y=error_class, group=subject, color = AAT_spontansprache)) +
    geom_point(size = 2)+
    stat_summary(aes(linetype=AAT_spontansprache, color=AAT_spontansprache),
                 fun=mean,  geom="line", size = 0.5) +
    scale_linetype_manual(values=c("dashed"))+
    scale_colour_manual(values=c("#76ff76", "#008900"))+
    geom_errorbar(aes(ymin=error_class-se, ymax=error_class+se,
                      group = AAT_spontansprache), width =.1) +
    apatheme+
    scale_y_continuous(limits = c(0, 0.5),
                        breaks =seq(0,0.5, by = 0.1)) +
    labs(x="Ordinal Position ",y ="Errors",
         colour="Session", linetype="Session",
         title = "Day 2") +
    theme(
    axis.title.y = element_text(margin = margin(0,10,0,0)),
    axis.title.x = element_text(margin = margin(10,0,0,0)),
    legend.key.width = unit(1, "cm"),
    legend.position="none")+
   guides(color=guide_legend(
     override.aes=list(linetype=override.linetype)))+
    scale_linetype(guide="none"))
```

```
## Scale for linetype is already present.
## Adding another scale for linetype, which will replace the existing scale.
```



```
(plot_error_repetition_PWA_session3_Speech <- means_final %>%
    filter(session=="3") %>%
    ggplot(., aes(x=PosOr, y=error_class, group=subject, color = AAT_spontansprache)) +
    geom_point(size = 2)+
    stat_summary(aes(linetype=AAT_spontansprache, color=AAT_spontansprache),
                 fun=mean,  geom="line", size = 0.5) +
    scale_linetype_manual(values=c("dashed"))+
    scale_colour_manual(values=c("#76ff76", "#008900"))+
    geom_errorbar(aes(ymin=error_class-se, ymax=error_class+se,
                      group = AAT_spontansprache), width =.1) +
    apatheme+
     scale_y_continuous(limits = c(0, 0.5),
                        breaks =seq(0,0.5, by = 0.1)) +
    labs(x="Ordinal Position ",y ="Errors",
         colour="Spontaneous\nspeech\nscore",
        # linetype="Session",
         title = "Day 8") +
    theme(
    axis.title.y = element_text(margin = margin(0,10,0,0)),
    axis.title.x = element_text(margin = margin(10,0,0,0)),
    legend.key.width = unit(1, "cm"),
    legend.position="right")+
    scale_linetype(guide="none"))
```

```
## Scale for linetype is already present.
## Adding another scale for linetype, which will replace the existing scale.
## Warning: Removed 1 rows containing non-finite values (‘stat_summary()‘).
## Removed 1 rows containing missing values (‘geom_point()‘).
```



```
plots <- cowplot::plot_grid(plot_error_repetition_PWA_session1_Naming,
                            plot_error_repetition_PWA_session2_Naming,
                            plot_error_repetition_PWA_session3_Naming,
                            plot_error_repetition_PWA_session1_Token,
                            plot_error_repetition_PWA_session2_Token,
                            plot_error_repetition_PWA_session3_Token,
                            plot_error_repetition_PWA_session1_Speech,
                            plot_error_repetition_PWA_session2_Speech,
                           plot_error_repetition_PWA_session3_Speech,
          nrow = 3, ncol=3,
          rel_widths = c(0.81,0.81,1,0.81,0.81,1,0.81,0.81,1),
          margin(1,1,1,1),
          labels = c("A", "", "", "B", "", "","C", "", ""),
          label_size = 34,
                    label_fontfamily = "Helvetica", label_y = 1.01,
          label_x=-0.03)
```

```
## Warning: Removed 1 rows containing non-finite values (‘stat_summary()‘).
## Removed 1 rows containing missing values (‘geom_point()‘).
```

```
## Warning: Removed 1 rows containing non-finite values (‘stat_summary()‘).
## Warning: Removed 1 rows containing missing values (`geom_point()`).

## Warning: Removed 1 rows containing non-finite values (`stat_summary()`).

## Warning: Removed 1 rows containing missing values (`geom_point()`).

## Warning in as_grob.default(plot): Cannot convert object of class
## marginsimpleUnitunitunit_v2 into a grob.
```

```
filename <- "CSI_online_aphasia_spoken_plot_PWA_Error_Test_scores.pdf"
ggsave(plots, filename =
         here::here("results", "figures", filename),
       width = 25, height = 39, units = "cm",
       dpi = 300, device = cairo_pdf)
filename_png <- "CSI_online_aphasia_spoken_plot_PWA_Error_Test_scores.png"
ggsave(plots, filename =
         here::here("results", "figures", filename_png),
       width = 25, height = 39, units = "cm",
       dpi = 300)
#embedFonts(file = here::here("results", "figures", filename))
#
# # Naming percenile scores: median split of means final
# median(as.numeric(as.character(means_final$AAT)))
# means_final %>%
#   mutate(med_split=case_when(as.numeric(as.character(AAT)) >
#   median(as.numeric(as.character(means_final$AAT))) ~ "high",
#   as.numeric(as.character(AAT)) <
#     median(as.numeric(as.character(means_final$AAT))) ~ "low",
#   as.numeric(as.character(AAT)) == median(as.numeric(as.character(
#     means_final$AAT))) ~
#     "median"),
#   mean_split=case_when(as.numeric(as.character(AAT)) >
#                        mean(as.numeric(as.character(means_final$AAT))) ~
#                        "high",
#   as.numeric(as.character(AAT)) < mean(as.numeric(as.character(
#     means_final$AAT))) ~ "low",
#   as.numeric(as.character(AAT)) == mean(as.numeric(as.character(
#     means_final$AAT))) ~ "median")) -> means_final
#
# means_final %>% group_by(session, med_split, PosOr) %>%
#            summarise(mean=mean(error_class), sd = sd(error_class))
# means_final %>% group_by(session, mean_split, PosOr)  %>%
#   summarise(mean=mean(error_class), sd = sd(error_class))
#
#
# # Token Test percenile scores: median split of means final
# median(as.numeric(as.character(means_final$TokenTest)))
# means_final %>% mutate(med_split=case_when(
#   as.numeric(as.character(TokenTest)) >
#     median(as.numeric(as.character(means_final$TokenTest))) ~ "high",
#   as.numeric(as.character(TokenTest)) <
#     median(as.numeric(as.character(means_final$TokenTest))) ~ "low",
#   as.numeric(as.character(TokenTest)) ==
#     median(as.numeric(as.character( means_final$TokenTest))) ~ "median"),
#   mean_split=case_when(
#     as.numeric(as.character(TokenTest)) >
#       mean(as.numeric(as.character(means_final$TokenTest))) ~ "high",
#     as.numeric(as.character(TokenTest)) <
#       mean(as.numeric(as.character( means_final$TokenTest))) ~ "low",
#     as.numeric(as.character(TokenTest)) ==
#       mean(as.numeric(as.character(means_final$TokenTest))) ~ "median")) ->
#   means_final
#
# means_final %>% group_by(session, med_split, PosOr) %>%
#            summarise(mean=mean(error_class), sd = sd(error_class))
# means_final %>% group_by(session, mean_split, PosOr)  %>%
#   summarise(mean=mean(error_class),sd = sd(error_class))
#
# # Spontaneous speech scores: median split of means final
# median(as.numeric(as.character(means_final$AAT_spontansprache)))
# means_final %>%
#   mutate(med_split=case_when(
#     as.numeric(as.character(AAT_spontansprache)) >
#       median(as.numeric(as.character(
#         means_final$AAT_spontansprache))) ~ "high",
#     as.numeric(as.character(AAT_spontansprache)) <
#       median(as.numeric(as.character(
#         means_final$AAT_spontansprache))) ~ "low",
#     as.numeric(as.character(AAT_spontansprache)) ==
#       median(as.numeric(as.character(
#         means_final$AAT_spontansprache))) ~ "median"),
#     mean_split=case_when(
#       as.numeric(as.character(AAT_spontansprache)) >
#         mean(as.numeric(as.character(
#           means_final$AAT_spontansprache))) ~ "high"
#       as.numeric(as.character(AAT_spontansprache)) <
#         mean(as.numeric(as.character(
#           means_final$AAT_spontansprache))) ~ "low",
#       as.numeric(as.character(AAT_spontansprache)) ==
#         mean(as.numeric(as.character(
#           means_final$AAT_spontansprache))) ~ "median")) -> means_final
#
# means_final %>% group_by(med_split, PosOr) %>%
#            summarise(mean=mean(error_class), sd = sd(error_class))
# means_final %>% group_by(mean_split, PosOr)  %>%
#   summarise(mean=mean(error_class), sd = sd(error_class))
#
# #### Nested models
# #
# # summary(afex::lmer_alt(error_class ~
# #                            session/(PosOr.cont*(TokenTest_c+AAT_c+
# #                                                 AAT_spontansprache_c+
# #                                                 LHoverall_c)) +
# #                      (PosOr.cont |subject) +
# #                      (1|category) ,
# #                    data =df_errors_PWA, family = "binomial",
# #                     control=glmerControl(optimizer = "bobyqa",
# #                                    optCtrl = list(maxfun = 2e5))))
```

```
# library(sjPlot)
# library(sjmisc)
# library(ggplot2)
# Change labels of sessions and re-compute model
override.linetypea<-c("longdash", "dashed")
override.linetypeb<-c("longdash", "dashed", "dotted")
df_errors_PWA <- df_errors_PWA %>%
  mutate(session_day = case_when(session=="1" ~ "Day 1",
                                 session=="2" ~ "Day 2",
                                 session=="3" ~ "Day 8")) %>%
  mutate(session_day=factor(session_day, levels=c("Day 1", "Day 2", "Day 8")))
c<-contr.treatment(3)
my.coding<-matrix(rep(1/3, 6), ncol=2)
my.simple<-c-my.coding
my.simple
```

Plot mean CSI effect by Session and test

```
##            2          3
## 1 -0.3333333 -0.3333333
## 2  0.6666667 -0.3333333
## 3 -0.3333333  0.6666667
```

```
contrasts(df_errors_PWA$session_day)<-my.simple
levels(df_errors_PWA$session_day)
```

```
## [1] "Day 1" "Day 2" "Day 8"
```

```
contrasts(df_errors_PWA$session_day)
```

```
##                2          3
## Day 1 -0.3333333 -0.3333333
## Day 2  0.6666667 -0.3333333
## Day 8 -0.3333333  0.6666667
```

```
m1_glmm_PWA_all <-
  afex::lmer_alt(error_class ~
  PosOr.cont*session_day*
    (TokenTest_c+AAT_c+AAT_spontansprache_c+LHoverall_c) +
  (PosOr.cont |subject) + (1|category),
  df_errors_PWA, family = "binomial",
  control=glmerControl(optimizer = "bobyqa",
                       optCtrl = list(maxfun = 2e5)))
# df_errors_PWA # data frame
# summary(m1_glmm_PWA_all)
```

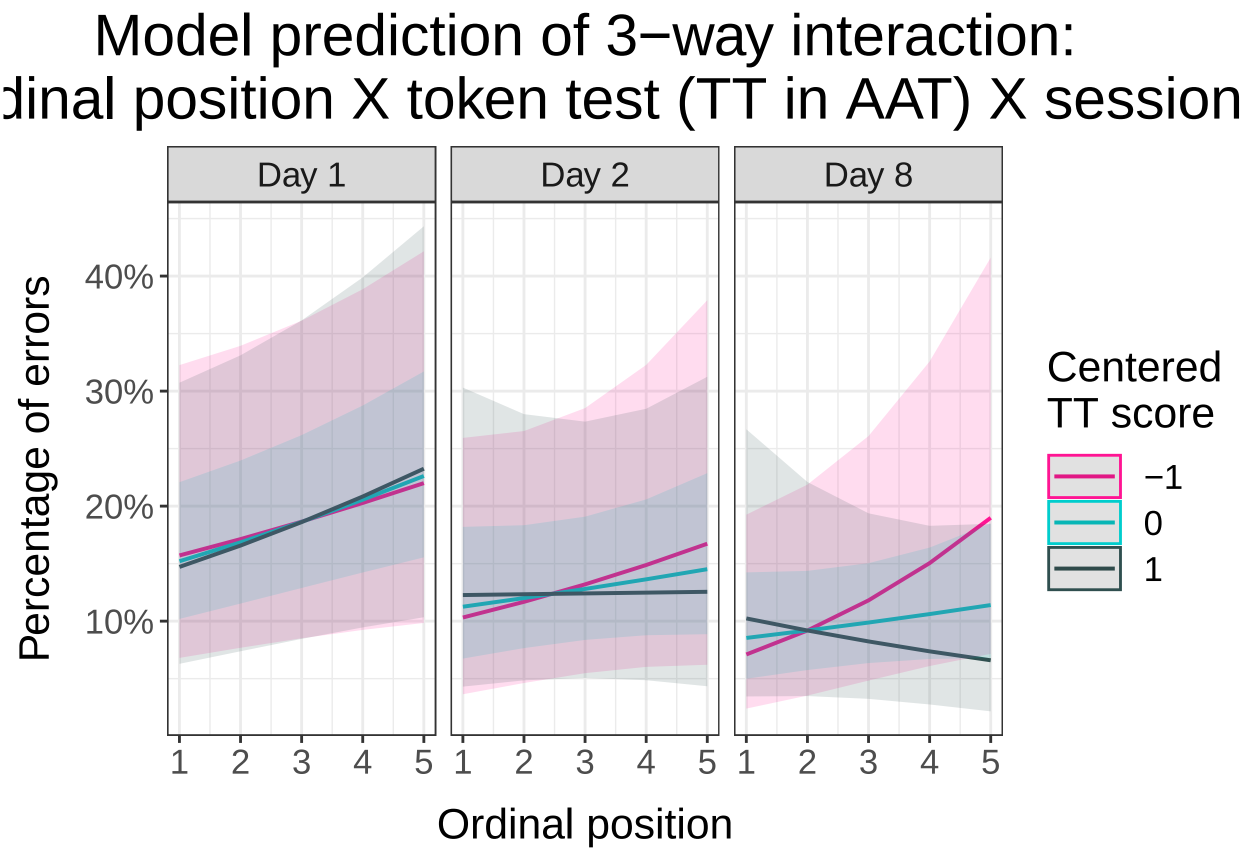
```
## Plot the significant interaction effects
# PosOr.cont:AAT_spontansprache_c
plot1 <- plot_model(m1_glmm_PWA_all,
           type = "pred",
           #transform=NULL,
           terms = c("PosOr.cont", "AAT_spontansprache_c"),
           mrt.values="all",
           title=c("Model prediction of 2-way interaction:\nOrdinal position X spontaneous speech (SEM
           linetype=override.linetypea) +
    scale_x_continuous(breaks = c(-1.99860685427696,
                                -0.998606854276957,
                                0.00139314572304272,
                                1.00139314572304,
                                2.00139314572304),
                     labels= c(1,2,3,4,5))  +
  xlab("Ordinal position") +
  ylab("Percentage of errors") +
  scale_colour_manual(name = "Centered\nSEM score",
                        labels=c("-1.23", " 0.81"),
                        values=c("#6495ed", "#8b0a50")) +

  scale_fill_manual(name = "Centered\nSEM score",
                        labels=c("-1.23", " 0.81"),
                        values=c("#6495ed", "#8b0a50")) +
   # guides(color=guide_legend(
   #   override.aes=list(linetype=override.linetypea))+
  theme_bw()+
  theme(plot.title=element_text(size=22,hjust = .5),
        # panel.grid.major=element_blank(),
        # panel.grid.minor=element_blank(),
        #panel.border=element_blank(),
        #axis.line=element_line(),
        text=element_text(size=16),
        axis.title.y = element_text(margin = margin(0,10,0,0)),
        axis.title.x = element_text(margin = margin(10,0,0,0)),
        legend.key.width = unit(1, "cm"),
        legend.position="right")
```

```
## Scale for colour is already present.
## Adding another scale for colour, which will replace the existing scale.
```

```
ggsave(plot1, filename =
         here::here("results", "figures",
                    "CSI_online_aphasia_spoken_plot_error_by_OrdPos_by_SEM.pdf"),
       width = 15,#25
       height = 13, units = "cm",
       dpi = 300, device = cairo_pdf)

# PosOr.cont:session_day3:TokenTest_c
plot2 <- plot_model(m1_glmm_PWA_all,
           type = "pred",
           #transform=NULL,
           terms = c("PosOr.cont", "TokenTest_c", "session_day"),
           mrt.values="quart",
           title=c("Model prediction of 3-way interaction:\nOrdinal position X token test (TT in AAT) X
    scale_x_continuous(breaks = c(-1.99860685427696,
                                -0.998606854276957,
                                0.00139314572304272,
                                1.00139314572304,
                                2.00139314572304),
                     labels= c(1,2,3,4,5))  +
  xlab("Ordinal position") +
  ylab("Percentage of errors") +
  scale_colour_manual(name = "Centered\nTT score",
                       # labels=c("-1.23", " 0.81"),
                        values=c("#ff1493", "#00cdcd",
                                 "#2f4f4f")) +
  scale_fill_manual(name = "Centered\nTT score",
                       # labels=c("-1.23", " 0.81"),
                        values=c("#ff1493", "#00cdcd",
                                 "#2f4f4f")) +
  theme_bw()+
  theme(plot.title=element_text(size=22,hjust = .5),
        # panel.grid.major=element_blank(),
        # panel.grid.minor=element_blank(),
        #panel.border=element_blank(),
        #axis.line=element_line(),
        text=element_text(size=16),
        axis.title.y = element_text(margin = margin(0,10,0,0)),
        axis.title.x = element_text(margin = margin(10,0,0,0)),
        legend.key.width = unit(1, "cm"),
        legend.position="right")
```

```
## Scale for colour is already present.
## Adding another scale for colour, which will replace the existing scale.
```

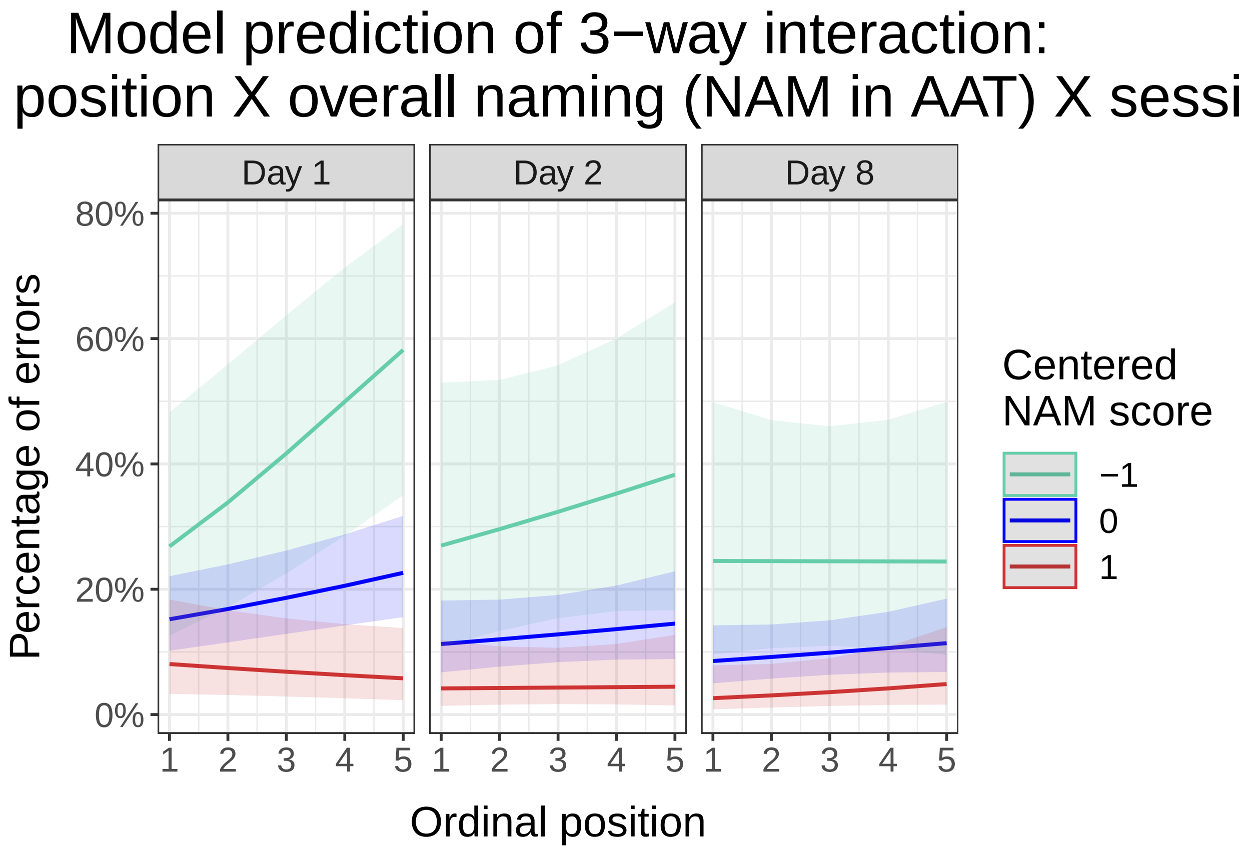
```
plot2
```



```
ggsave(plot2, filename =
         here::here("results", "figures",
                    "CSI_online_aphasia_spoken_plot_error_by_OrdPos_by_Session_by_TT.pdf"),
       width = 25,
       height = 13, units = "cm",
       dpi = 300, device = cairo_pdf)

# PosOr.cont:session_day3:AAT_c
plot3 <- plot_model(m1_glmm_PWA_all,
           type = "pred",
           #transform=NULL,
           terms = c("PosOr.cont", "AAT_c", "session_day"),
           mrt.values="quart", ci.lvl = .95,
           title=c("Model prediction of 3-way interaction:\nOrdinal position X overall naming (NAM in A
    scale_x_continuous(breaks = c(-1.99860685427696,
                                -0.998606854276957,
                                0.00139314572304272,
                                1.00139314572304,
                                2.00139314572304),
                     labels= c(1,2,3,4,5))  +
  xlab("Ordinal position") +
  ylab("Percentage of errors") +
  scale_colour_manual(name = "Centered\nNAM score",
                       # labels=c("-1.23", " 0.81"),
                        values=c("#66cdaa", "#0000ff",
                                 "#cd3333")) +
  scale_fill_manual(name = "Centered\nTT score",
                      # labels=c("-1.23", " 0.81"),
                       values=c("#66cdaa", "#0000ff",
                                "#cd3333")) +
  theme_bw()+
  theme(plot.title=element_text(size=22,hjust = .5),
        # panel.grid.major=element_blank(),
        # panel.grid.minor=element_blank(),
        #panel.border=element_blank(),
        #axis.line=element_line(),
        text=element_text(size=16),
        axis.title.y = element_text(margin = margin(0,10,0,0)),
        axis.title.x = element_text(margin = margin(10,0,0,0)),
        legend.key.width = unit(1, "cm"),
        legend.position="right")
```

```
## Scale for colour is already present.
## Adding another scale for colour, which will replace the existing scale.
```

```
plot3
```



```
ggsave(plot3, filename =
         here::here("results", "figures",
                    "CSI_online_aphasia_spoken_plot_error_by_OrdPos_by_Session_by_NAM.pdf"),
       width = 25,
       height = 13, units = "cm",
       dpi = 300, device = cairo_pdf)

plotgrid <- cowplot::plot_grid(plot1, plot3, plot2,
          nrow = 3, ncol=1,  rel_widths = c(0.3,1,1),
          margin(1,1,1,1),
          labels = c("A", "B", "C"),label_size = 34,
          label_y = 1.01,
          label_x=-0.01)
```

```
## Warning in as_grob.default(plot): Cannot convert object of class
## marginsimpleUnitunitunit_v2 into a grob.
```

```
#plotgrid
ggsave(plotgrid, filename =
         here::here("results", "figures",
                    "CSI_online_aphasia_spoken_plot_error_by_covariates_plotgrid.pdf"),
       width = 25,
       height = 25, units = "cm",
       dpi = 300, device = cairo_pdf)
ggsave(plotgrid, filename =
         here::here("results", "figures",
                    "CSI_online_aphasia_spoken_plot_error_by_covariates_plotgrid.png"),
       width = 25,
       height = 25, units = "cm",
       dpi = 300)
```

Additional (pre-planned) analyses

Preregistered analyses with GLMM

We deviated from our preregistered analyses because SEs seemed suspiciously small.

Check fit of normal vs gamma distribution in histograms, q-q-plots and using objective criteria:
1) Fit normal and gamma distributions to the reaction time data

```
library(fitdistrplus)
```

```
## Loading required package: MASS

##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select

## Loading required package: survival
## Warning: package 'survival' was built under R version 4.3.2
```

```
fit.normal<- fitdist(df_RTs$VOT, distr = "norm", method = "mle")
summary(fit.normal)
```

```
## Fitting of the distribution ' norm ' by maximum likelihood
## Parameters :
##       estimate Std. Error
## mean 1256.4520   4.029886
## sd    449.7363   2.849560
## Loglikelihood:  -93756.26   AIC:  187516.5   BIC:  187531.4
## Correlation matrix:
##      mean sd
## mean    1  0
## sd      0  1
```

```
#plot(fit.normal)

fit.normal_PWA<- fitdist(df_RTs_PWA$VOT, distr = "norm", method = "mle")
summary(fit.normal_PWA)
```

```
## Fitting of the distribution ' norm ' by maximum likelihood
## Parameters :
##      estimate Std. Error
## mean 1352.928   7.160560
## sd    531.897   5.063163
## Loglikelihood:  -42463.15   AIC:  84930.31   BIC:  84943.54
## Correlation matrix:
##      mean sd
## mean    1  0
## sd      0  1
```

```
#plot(fit.normal_PWA)
```

```
fit.gamma <- fitdist(df_RTs$VOT, distr = "gamma", method = "mle")
summary(fit.gamma)
```

```
## Fitting of the distribution ' gamma ' by maximum likelihood
## Parameters :
##          estimate    Std. Error
## shape 9.303155467 0.09618813487
## rate  0.007404132 0.00007718889
## Loglikelihood:  -92203.61   AIC:  184411.2   BIC:  184426.1
## Correlation matrix:
##           shape      rate
## shape 1.0000000 0.9610023
## rate  0.9610023 1.0000000
```

```
#plot(fit.gamma)

fit.gamma_PWA <- fitdist(df_RTs_PWA$VOT, distr = "gamma", method = "mle")
summary(fit.gamma_PWA)
```

```
## Fitting of the distribution ' gamma ' by maximum likelihood
## Parameters :
##          estimate    Std. Error
## shape 7.228017511 0.10366210322
## rate  0.005343025 0.00007644925
## Loglikelihood:  -41893.43   AIC:  83790.86   BIC:  83804.09
## Correlation matrix:
##           shape      rate
## shape 1.0000000 0.9414491
## rate  0.9414491 1.0000000
```

```
#plot(fit.gamma_PWA)
```

```
# library(actuar)
# fit.invgamma <- fitdist(df_RTs$VOT, distr = "invgamma",  method = "mle")
# summary(fit.invgauss)
# #plot(fit.invgauss)
#
# fit.invgamma_PWA <- fitdist(df_RTs_PWA$VOT, distr = "invgamma", method = "mle")
# summary(fit.invgamma_PWA)
# #plot(fit.invgauss_PWA)
```

```
library(actuar)
```

```
##
## Attaching package: 'actuar'

## The following objects are masked from 'package:stats':
##
##     sd, var

## The following object is masked from 'package:grDevices':
##
##     cm
```

```
fit.invgauss <- fitdist(
  df_RTs$VOT, distr = "invgauss", start = list(mean = 5, shape = 1),
  method = "mle")
```

```
## Warning in checkparamlist(arg_startfix$start.arg, arg_startfix$fix.arg, : Some
## parameter names have no starting/fixed value but have a default value:
## dispersion.
```

```
summary(fit.invgauss)
```

```
## Fitting of the distribution ' invgauss ' by maximum likelihood
## Parameters :
##        estimate Std. Error
## mean   1256.267   3.737018
## shape 11394.058 151.349229
## Loglikelihood:  -91804.31   AIC:  183612.6   BIC:  183627.5
## Correlation matrix:
##              mean       shape
## mean  1.000000000 0.002057613
## shape 0.002057613 1.000000000
```

```
#plot(fit.invgauss)

fit.invgauss_PWA <- fitdist(df_RTs_PWA$VOT, distr = "invgauss",
                            start = list(mean = 5, shape = 1),
                            method = "mle")
```

```
## Warning in checkparamlist(arg_startfix$start.arg, arg_startfix$fix.arg, : Some
## parameter names have no starting/fixed value but have a default value:
## dispersion.
```

```
## Warning in sqrt(diag(varcovar)): NaNs produced
```

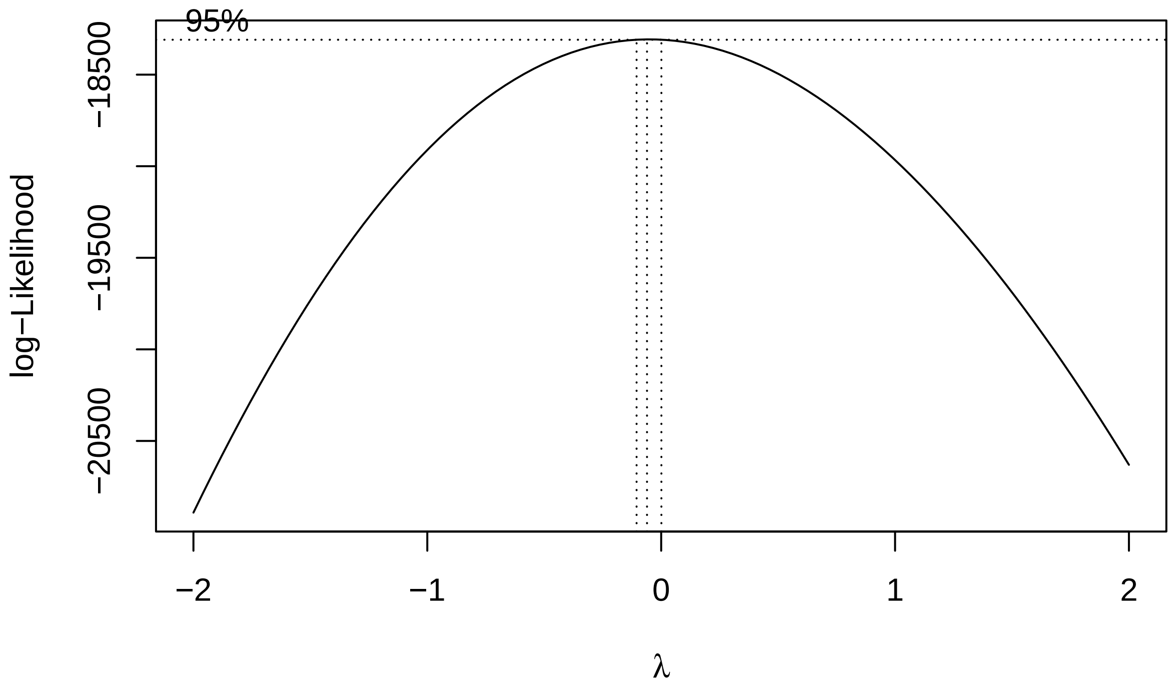
```
## Warning in sqrt(1/diag(V)): NaNs produced
```

```
## Warning in cov2cor(varcovar): diag(.) had 0 or NA entries; non-finite result is
## doubtful
```

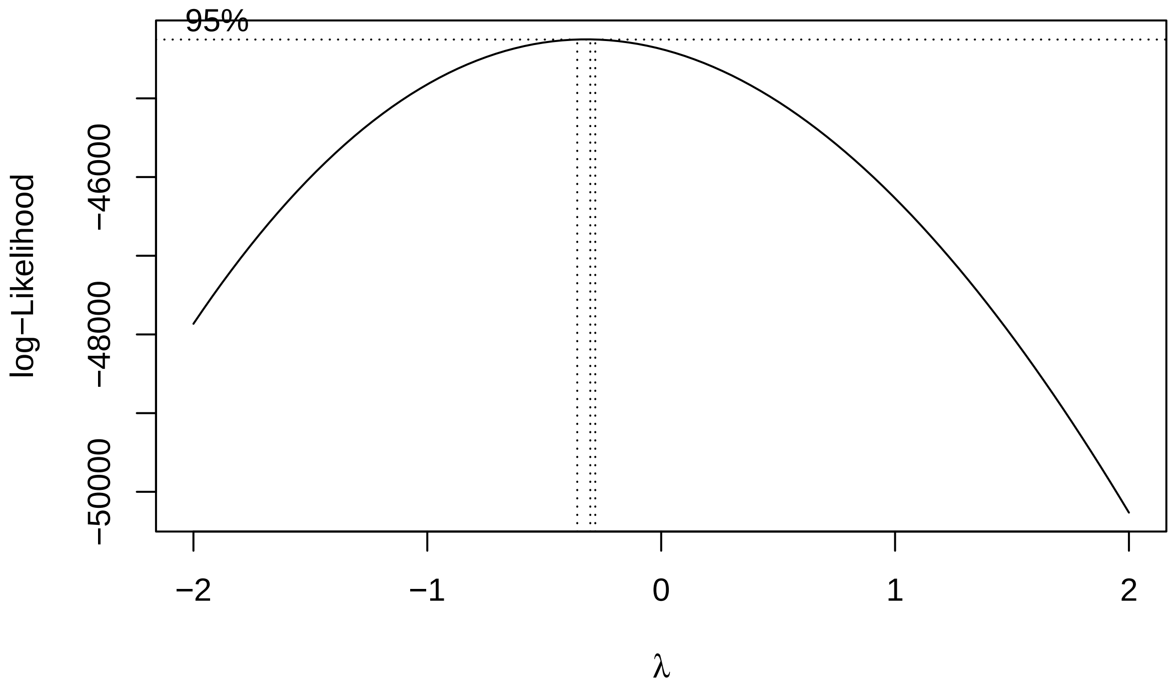
```
summary(fit.invgauss_PWA)
```

```
## Fitting of the distribution ’ invgauss ’ by maximum likelihood
## Parameters :
##        estimate Std. Error
## mean  99734.213        NaN
## shape  1176.467   22.41566
## Loglikelihood:  -47351.68   AIC:  94707.36   BIC:  94720.59
## Correlation matrix:
##       mean shape
## mean     1   NaN
## shape  NaN     1
```

```
#plot(fit.invgauss_PWA)

MASS::boxcox(df_RTs_PWA$VOT~df_RTs_PWA$PosOr_cont*df_RTs_PWA$session)
```



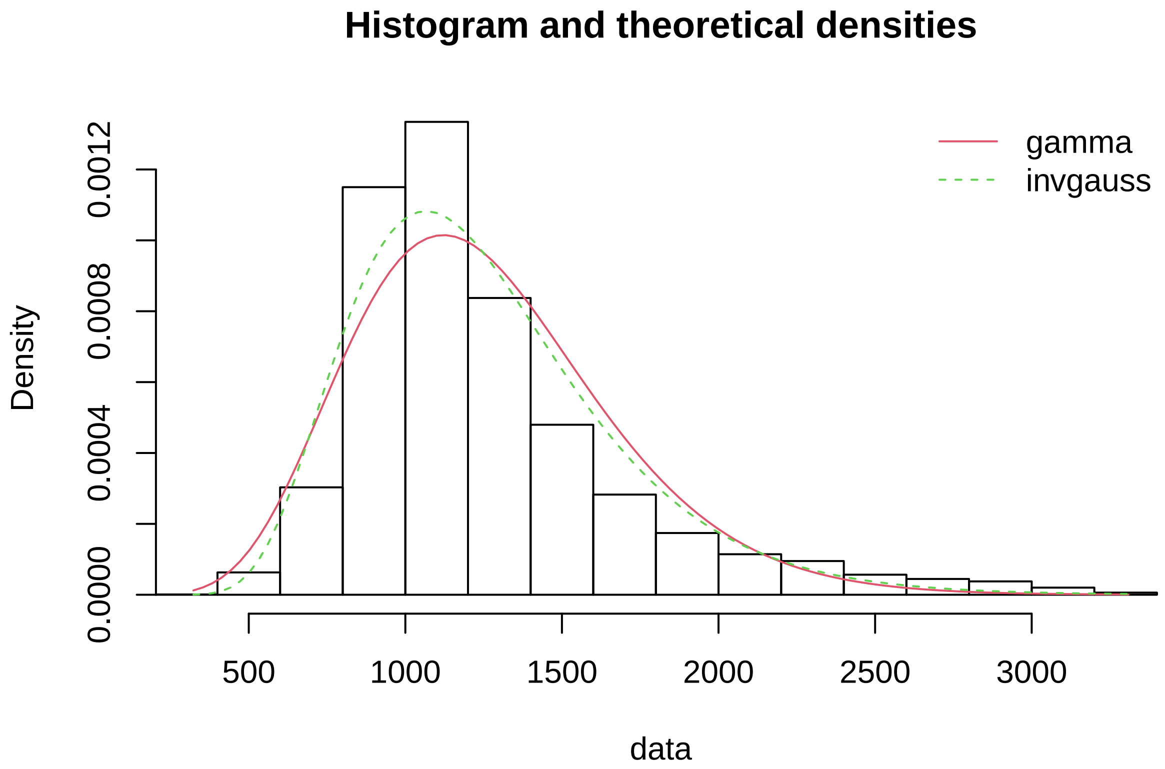
```
MASS::boxcox(df_RTs$VOT~df_RTs$PosOr_cont*df_RTs$session*df_RTs$group)
```



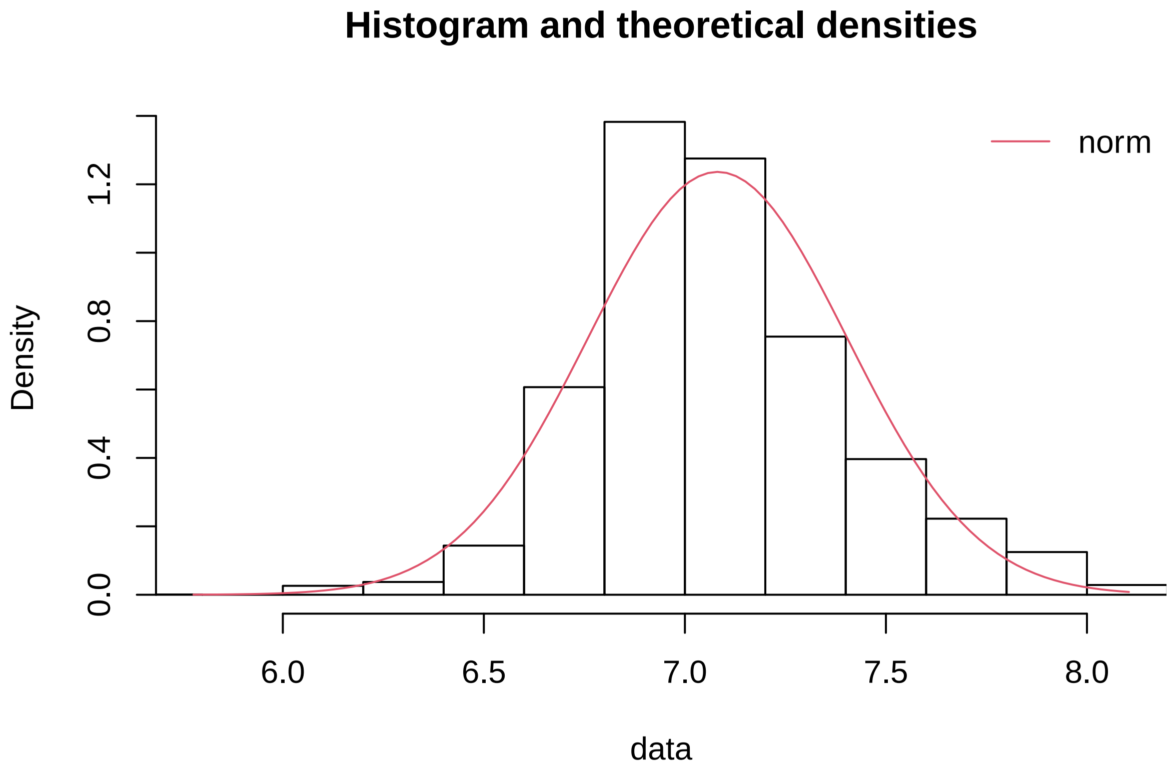
```
fit.transf <- fitdist(log(df_RTs$VOT), distr = "norm",method = "mle")
fit.transf_PWA <- fitdist(log(df_RTs_PWA$VOT),
                          distr = "norm",method = "mle")
```

2) Compare the fit of the two distributions
   Visually compare fit of both distributions in histogram

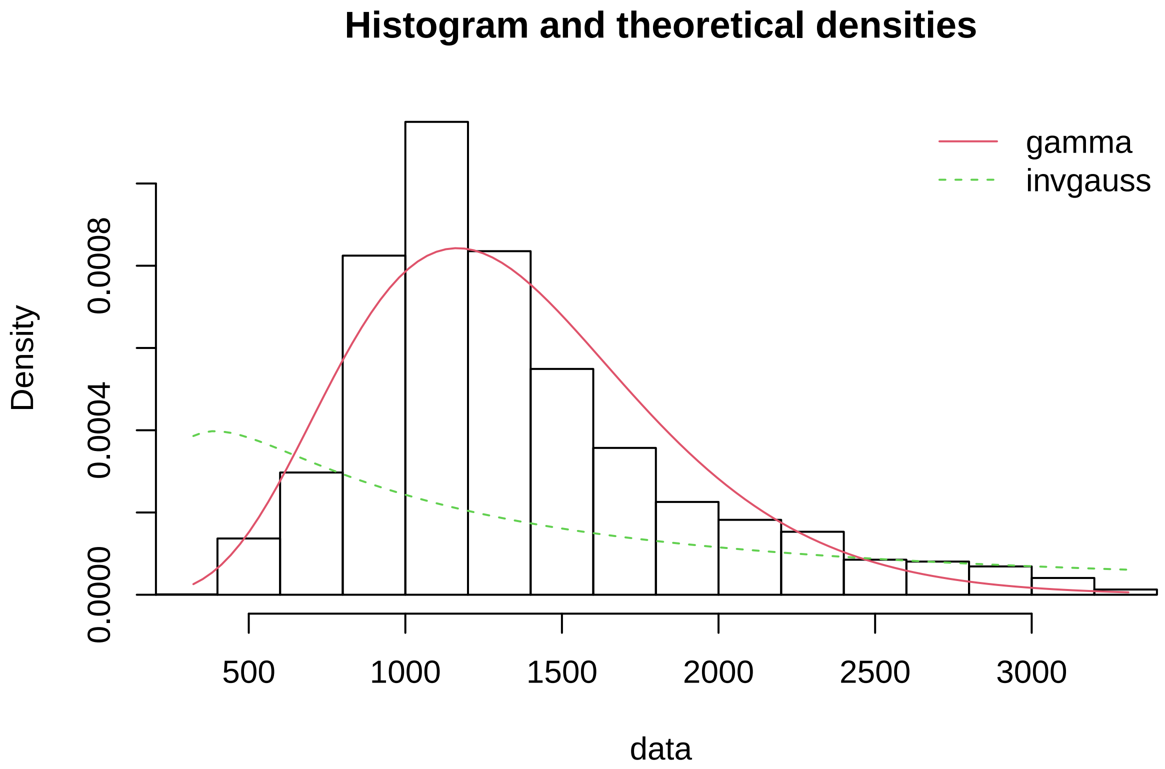
```
denscomp(list(fit.gamma, fit.invgauss))
```



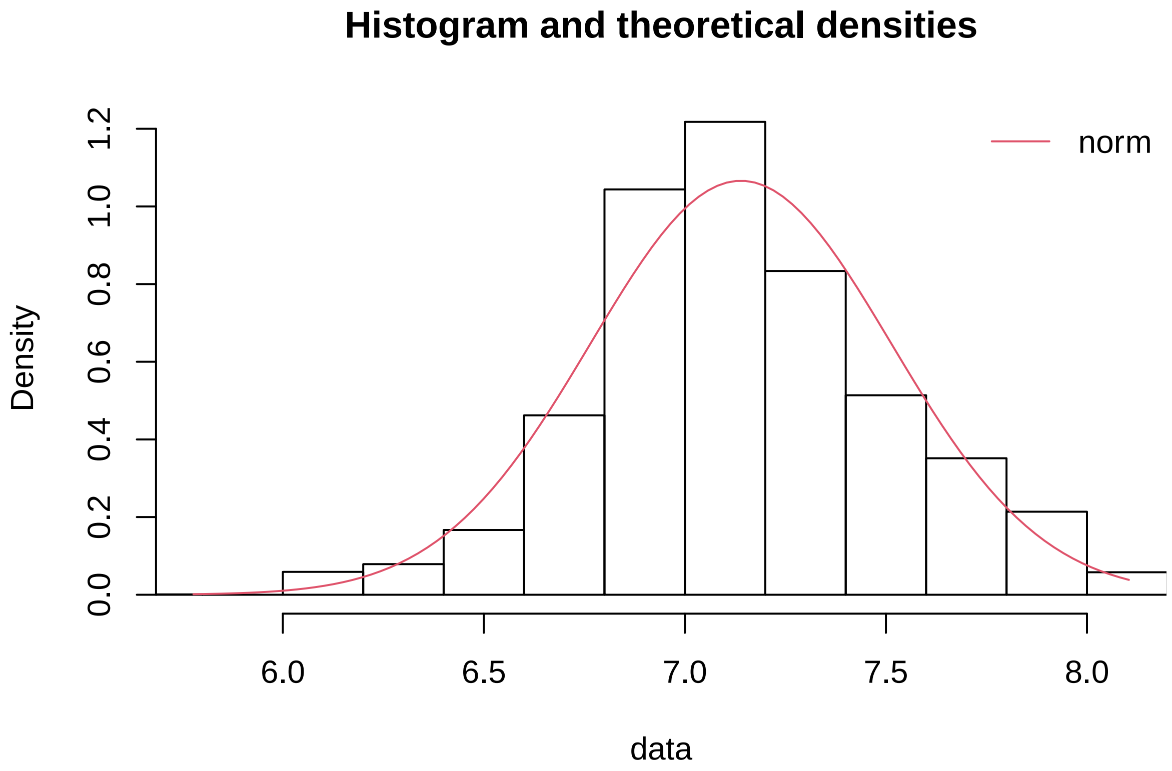
```
denscomp(list(fit.transf))
```



```
denscomp(list(fit.gamma_PWA, fit.invgauss_PWA))
```

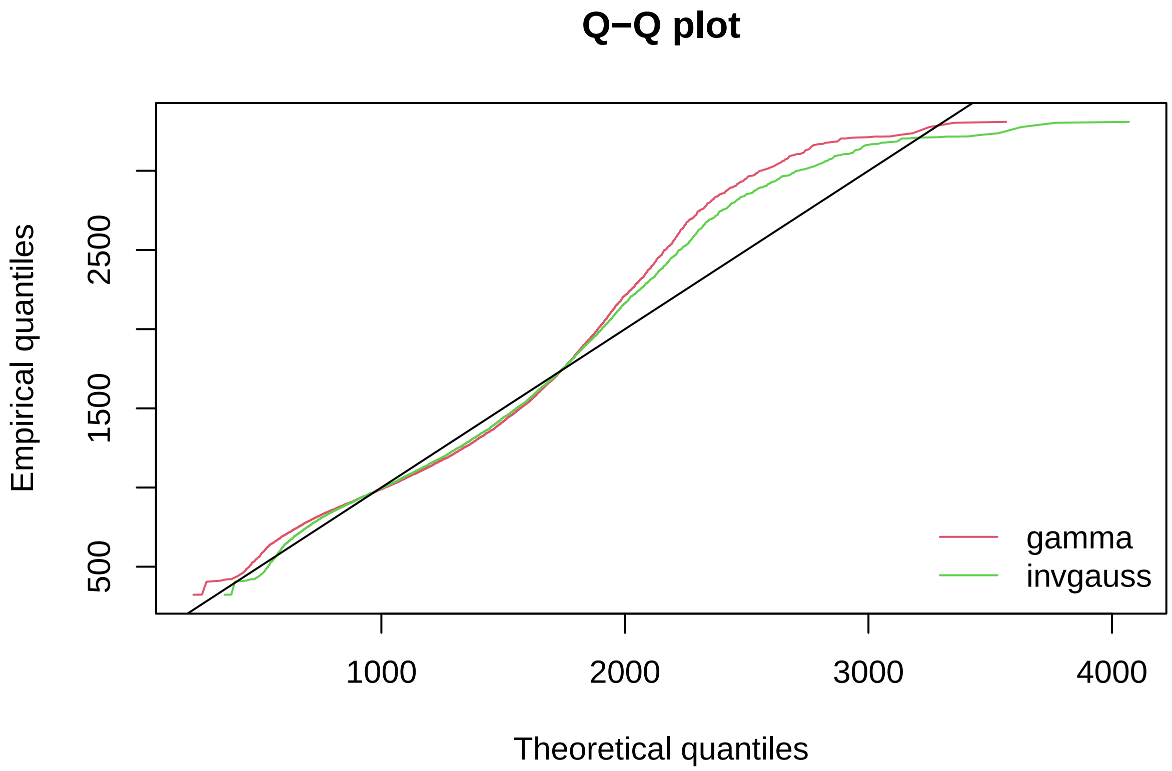


```
denscomp(list(fit.transf_PWA))
```

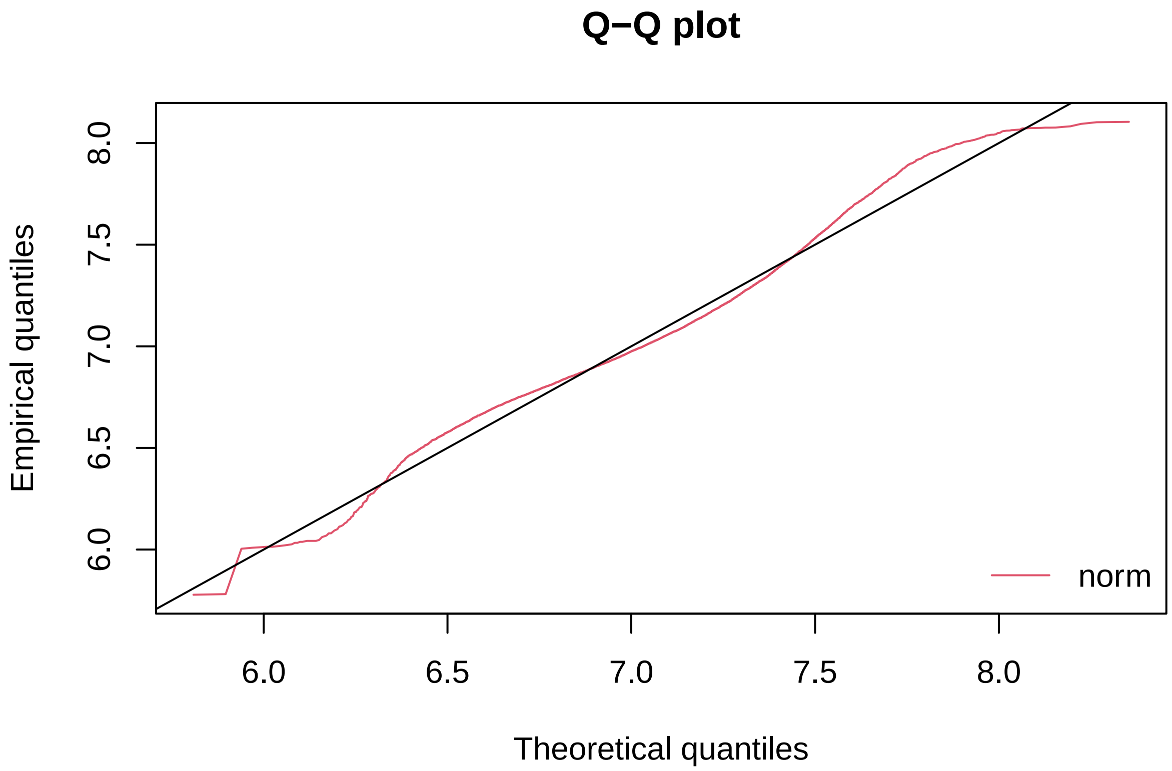


Visually compare fit of both distributions in Q-Q-plots

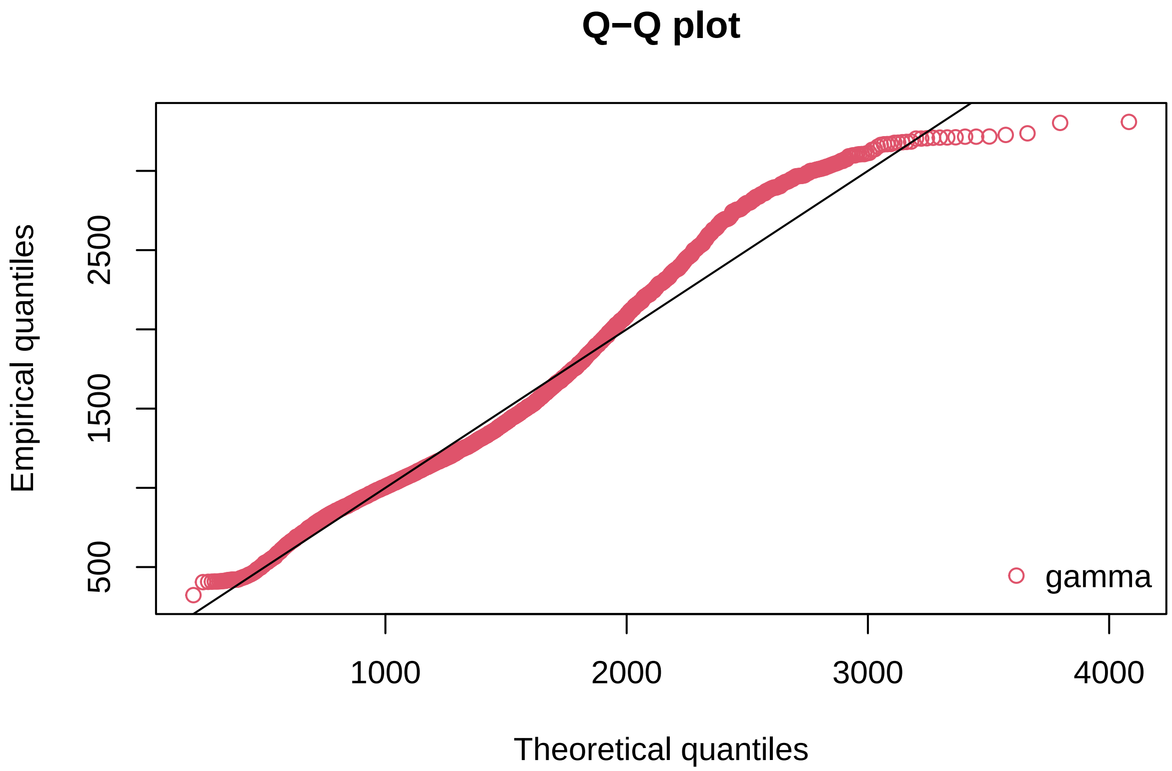
```
qqcomp(list(fit.gamma, fit.invgauss))
```



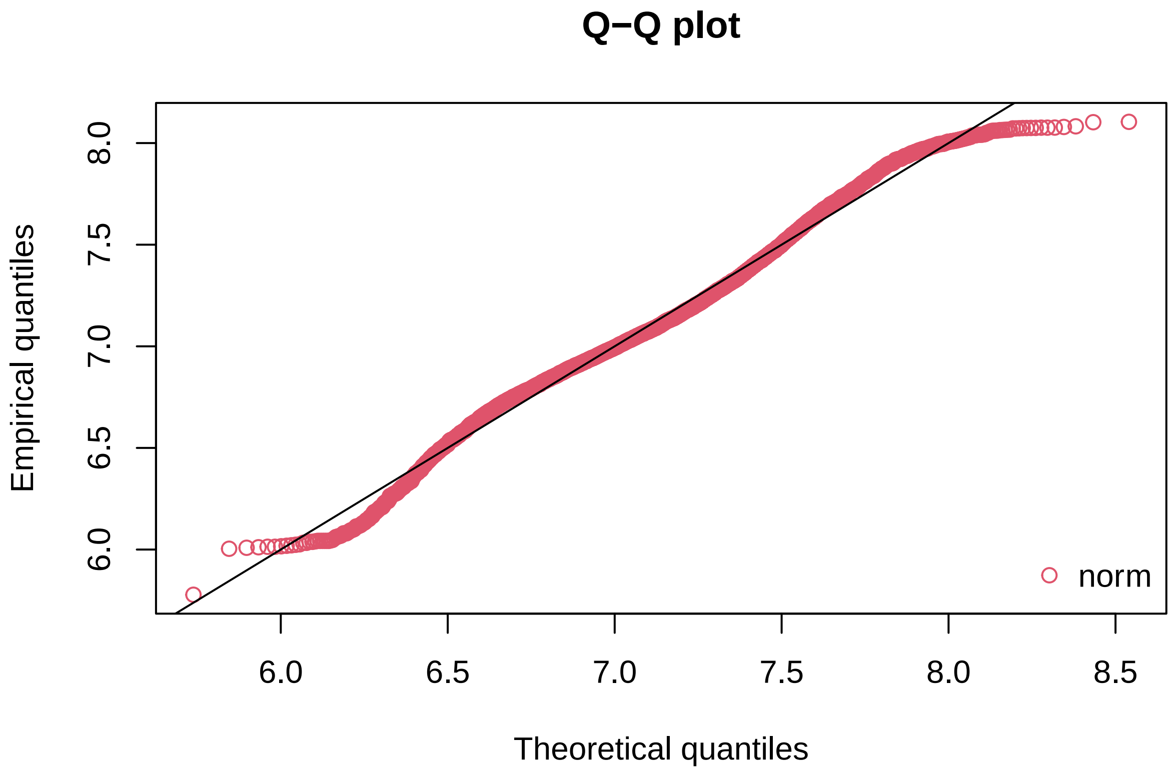
```
qqcomp(list(fit.transf))
```



```
qqcomp(list(fit.gamma_PWA))#, fit.invgauss_PWA))
```



```
qqcomp(list(fit.transf_PWA))
```



Compare information criteria

```
gofstat(list(fit.gamma, fit.invgauss, fit.normal),
        fitnames = c("Gamma", "Inverse Gaussian", "Normal"))
```

```
## Goodness-of-fit statistics
##                                      Gamma Inverse Gaussian     Normal
## Kolmogorov-Smirnov statistic   0.08343887       0.06232548   0.127563
## Cramer-von Mises statistic    30.32364092      16.58123861  75.033011
## Anderson-Darling statistic   176.46689389      96.39448742 433.186766
##
## Goodness-of-fit criteria
##                                    Gamma Inverse Gaussian   Normal
## Akaike's Information Criterion 184411.2         183612.6 187516.5
## Bayesian Information Criterion 184426.1         183627.5 187531.4
```

```
gofstat(list(fit.gamma_PWA, fit.invgauss_PWA, fit.normal_PWA),
        fitnames = c("Gamma", "Inverse Gaussian", "Normal" ))
```

```
## Goodness-of-fit statistics
##                                     Gamma Inverse Gaussian      Normal
## Kolmogorov-Smirnov statistic  0.06892153        0.4539719   0.1180601
## Cramer-von Mises statistic    8.06338088      375.0566303  25.8912637
## Anderson-Darling statistic   45.03214204     1728.3004897 147.8033420
##
```

```
## Goodness-of-fit criteria
##                                   Gamma Inverse Gaussian   Normal
## Akaike's Information Criterion 83790.86         94707.36 84930.31
## Bayesian Information Criterion 83804.09         94720.59 84943.54
```

Conclusion: . Overall, (inverse) gamma fits the data better than a normal model with uncontrolled data and an inverse Gaussian distribution for both the entire data set and the PWA group only. The inverse Gamma is not yet implemented in glmer. Therefore we will use the Gamma distribution.

```
# m1 <- glmer(VOT ~ PosOr.cont*session +
#                 (PosOr.cont*session|subject) +
#                (PosOr.cont*session|category),
#              data = df_RTs_PWA,
#             family =Gamma(link ="identity"),
#             control=glmerControl(optimizer = "bobyqa"))
```

PWA only - Ordinal position x session Model fails to converge –> reduce

```
# 1) Increase optimizer iterations
# m1 <- glmer(VOT ~ PosOr.cont*session +
#                 (PosOr.cont*session|subject) +
#                (PosOr.cont*session|category),
#              data = df_RTs_PWA,
#             family =Gamma(link ="identity"),
#             control=glmerControl(optimizer = "bobyqa",
#                                  optCtrl = list(maxfun = 2e5)))
# 2) Set correlation parameters to zero
m1 <- afex::lmer_alt(VOT ~ PosOr.cont*session +
               (PosOr.cont*session||subject) +
              (PosOr.cont*session||category),
             data = df_RTs_PWA,
            family =Gamma(link ="identity"),
            control=glmerControl(optimizer = "bobyqa",
                                 optCtrl = list(maxfun = 2e5)))
didLmerConverge(m1)
```

```
##  The relative maximum gradient of 0.0000625 is less than our 0.001 criterion.
##  You can safely ignore any warnings about a claimed convergence failure.
```

```
## The warnings can be safely ignored

# inspect model
summary(m1)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: Gamma  ( identity )
## Formula: VOT ~ PosOr.cont * session + (1 + re1.PosOr.cont + re1.session2 +
##     re1.session3 + re1.PosOr.cont_by_session2 + re1.PosOr.cont_by_session3 ||
##     subject) + (1 + re2.PosOr.cont + re2.session2 + re2.session3 +
##     re2.PosOr.cont_by_session2 + re2.PosOr.cont_by_session3 ||      category)
##    Data: data
## Control: glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 200000))
##
##      AIC      BIC   logLik deviance df.resid
##  80877.2  81002.9 -40419.6  80839.2     5499
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.9676 -0.6347 -0.2620  0.3412  6.9889
##
## Random effects:
##  Groups     Name                       Variance   Std.Dev.
##  category   re2.PosOr.cont_by_session3  1388.0844  37.2570
##  category.1 re2.PosOr.cont_by_session2  1746.8469  41.7953
##  category.2 re2.session3                2080.6296  45.6139
##  category.3 re2.session2                1856.3737  43.0857
##  category.4 re2.PosOr.cont               234.6941  15.3197
##  category.5 (Intercept)                 5374.8827  73.3136
##  subject    re1.PosOr.cont_by_session3   600.7064  24.5093
##  subject.1  re1.PosOr.cont_by_session2   965.2678  31.0688
##  subject.2  re1.session3                3742.7243  61.1778
##  subject.3  re1.session2                2955.8420  54.3677
##  subject.4  re1.PosOr.cont               361.3693  19.0097
##  subject.5  (Intercept)                20917.7899 144.6298
##  Residual                                  0.1001   0.3164
## Number of obs: 5518, groups:  category, 24; subject, 20
##
## Fixed effects:
##                     Estimate Std. Error t value             Pr(>|z|)
## (Intercept)         1545.112      4.381 352.700 < 0.0000000000000002 ***
## PosOr.cont            31.556      5.918   5.333         0.0000000968 ***
## session2            -112.882      4.345 -25.979 < 0.0000000000000002 ***
## session3            -144.043      5.425 -26.552 < 0.0000000000000002 ***
## PosOr.cont:session2   14.551      3.782   3.847             0.000119 ***
## PosOr.cont:session3    9.690      6.903   1.404             0.160386
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) PsOr.c sessn2 sessn3 PsO.:2
## PosOr.cont  -0.237
## session2    -0.090  0.058
## session3     0.083 -0.109 -0.029
## PsOr.cnt:s2 -0.065  0.057  0.015 -0.028
## PsOr.cnt:s3  0.018 -0.287  0.125  0.106  0.084
```

```
anova(m1)
```

```
## Analysis of Variance Table
##                    npar Sum Sq Mean Sq F value
## PosOr.cont            1 2.2286 2.22856 22.2599
## session               2 5.7970 2.89852 28.9518
## PosOr.cont:session    2 0.1159 0.05794  0.5788
```

```r
# save model output
saveRDS(m1, file = here::here("results", "tables",
                              "CSI_online_aphasia_PWA_glmm_cont.RDS"))
tab_model(m1,transform = NULL,
          show.re.var = F, show.stat = T,show.r2 = F,show.icc = F,
          title = "GLMM (Gamma distribution) of VOTs Predicted by Ordinal Position and Session, PWA onl
          dv.labels = "Vocal Onset Time",
          #string.pred = "",
          string.stat = "t-Value",
          file = here::here(
            "results", "tables", "CSI_online_aphasia_PWA_glmm_cont.html"))
```

GLMM (Gamma distribution) of VOTs Predicted by Ordinal Position and Session, PWA only

Vocal Onset Time

Predictors

Estimates

CI

t-Value

p

(Intercept)

1545.11

1536.52 – 1553.70

352.70

<0.001

PosOr cont

31.56

19.96 – 43.16

5.33

<0.001

session [2]

-112.88

-121.40 – -104.36

-25.98

<0.001

session [3]

-144.04

-154.68 – -133.41

-26.55

<0.001

PosOr cont × session [2]

14.55

7.14 – 21.97

3.85

<0.001

PosOr cont × session [3]

9.69

-3.84 – 23.22

1.40

0.160

N subject

20

N category

24

Observations

5518

Analyses with Ordinal position x Session x Group

Make sure contrasts are correctly defined

```
contrasts(df_RTs$session)
```

```
##            2          3
## 1 -0.3333333 -0.3333333
## 2  0.6666667 -0.3333333
## 3 -0.3333333  0.6666667
```

```
levels(df_RTs$session)
```

```
## [1] "1" "2" "3"
```

```
## Define contrast of group
contrasts(df_RTs$group)
```

```
##          2-1
## control -0.5
## PWA      0.5
```

```
levels(df_RTs$group)
```

```
## [1] "control" "PWA"
```

Compute model

```
# m2 <- glmer(VOT ~ PosOr.cont*session*group +
#                (PosOr.cont*session|subject) +
#               (PosOr.cont*session*group|category),
#              data = df_RTs,
#             family =Gamma(link ="identity"),
#             control=glmerControl(optimizer = "bobyqa"))
```

Model fails to converge –> reduce

```
# 1) Increase optimizer iterations
# m2 <- glmer(VOT ~ PosOr.cont*session*group +
#                (PosOr.cont*session|subject) +
#               (PosOr.cont*session*group|category),
#              data = df_RTs,
#             family =Gamma(link ="identity"),
#             control=glmerControl(optimizer = "bobyqa",
#                                  optCtrl = list(maxfun = 2e5)))
# 2) Set correlation parameters to zero
m2 <- afex::lmer_alt(VOT ~ PosOr.cont*session*group +
               (PosOr.cont*session||subject) +
              (PosOr.cont*session*group||category),
             data = df_RTs,
            family =Gamma(link ="identity"),
            control=glmerControl(optimizer = "bobyqa",
                                 optCtrl = list(maxfun = 2e5)))
didLmerConverge(m2)
```

```
##  The relative maximum gradient of 0.000238 is less than our 0.001 criterion.
##  You can safely ignore any warnings about a claimed convergence failure.
```

```
## The warnings can be safely ignored

# inspect model
summary(m2)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: Gamma  ( identity )
## Formula:
## VOT ~ PosOr.cont * session * group + (1 + re1.PosOr.cont + re1.session2 +
##     re1.session3 + re1.PosOr.cont_by_session2 + re1.PosOr.cont_by_session3 ||
##     subject) + (1 + re2.PosOr.cont + re2.session2 + re2.session3 +
##     re2.group2.1 + re2.PosOr.cont_by_session2 + re2.PosOr.cont_by_session3 +
##     re2.PosOr.cont_by_group2.1 + re2.session2_by_group2.1 + re2.session3_by_group2.1 +
##     re2.PosOr.cont_by_session2_by_group2.1 + re2.PosOr.cont_by_session3_by_group2.1 ||
##     category)
##    Data: data
## Control: glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 200000))
##
##      AIC      BIC   logLik deviance df.resid
## 178422.0 178652.4 -89180.0 178360.0    12424
##
```

```
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.5240 -0.6220 -0.2386  0.3365  8.0705
##
## Random effects:
##  Groups      Name                                   Variance    Std.Dev.
##  subject     (Intercept)                            11534.92084 107.4007
##  subject.1   re1.PosOr.cont                           230.18085  15.1717
##  subject.2   re1.session2                            3987.89997  63.1498
##  subject.3   re1.session3                            3759.80311  61.3172
##  subject.4   re1.PosOr.cont_by_session2               799.75422  28.2799
##  subject.5   re1.PosOr.cont_by_session3               337.48878  18.3709
##  category    (Intercept)                             3218.47862  56.7316
##  category.1  re2.PosOr.cont                           107.70350  10.3780
##  category.2  re2.session2                             675.10353  25.9828
##  category.3  re2.session3                             671.62800  25.9158
##  category.4  re2.group2.1                            1350.46946  36.7487
##  category.5  re2.PosOr.cont_by_session2               201.22400  14.1853
##  category.6  re2.PosOr.cont_by_session3               294.15346  17.1509
##  category.7  re2.PosOr.cont_by_group2.1               180.28554  13.4270
##  category.8  re2.session2_by_group2.1                3093.25604  55.6170
##  category.9  re2.session3_by_group2.1                2663.70627  51.6111
##  category.10 re2.PosOr.cont_by_session2_by_group2.1  2197.76058  46.8803
##  category.11 re2.PosOr.cont_by_session3_by_group2.1  1097.63575  33.1306
##  Residual                                               0.08145   0.2854
## Number of obs: 12455, groups:  subject, 40; category, 24
##
## Fixed effects:
##                              Estimate Std. Error t value             Pr(>|z|)
## (Intercept)                  1390.793      2.531 549.448 < 0.0000000000000002
## PosOr.cont                     22.805      3.054   7.466   0.0000000000000824
## session2                      -98.052      3.804 -25.779 < 0.0000000000000002
## session3                     -125.369      3.869 -32.401 < 0.0000000000000002
## group2-1                      338.070      2.909 116.195 < 0.0000000000000002
## PosOr.cont:session2             6.167      2.323   2.655              0.00794
## PosOr.cont:session3             5.634      6.493   0.868              0.38552
## PosOr.cont:group2-1            15.687      2.325   6.747   0.0000000000151152
## session2:group2-1             -30.708      3.306  -9.287 < 0.0000000000000002
## session3:group2-1             -42.597      3.231 -13.186 < 0.0000000000000002
## PosOr.cont:session2:group2-1   17.956      2.722   6.598   0.0000000000417505
## PosOr.cont:session3:group2-1    6.852      2.409   2.844              0.00446
##
## (Intercept)                  ***
## PosOr.cont                   ***
## session2                     ***
## session3                     ***
## group2-1                     ***
## PosOr.cont:session2          **
## PosOr.cont:session3
## PosOr.cont:group2-1          ***
## session2:group2-1            ***
## session3:group2-1            ***
## PosOr.cont:session2:group2-1 ***
## PosOr.cont:session3:group2-1 **
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Correlation of Fixed Effects:
##             (Intr) PsOr.c sessn2 sessn3 grp2-1 PsO.:2 PsO.:3 PO.:2- s2:2-1
## PosOr.cont  -0.091
## session2     0.116 -0.162
## session3    -0.047  0.306 -0.507
## group2-1    -0.019 -0.172 -0.081 -0.052
## PsOr.cnt:s2 -0.044 -0.099 -0.046  0.027  0.199
## PsOr.cnt:s3  0.112 -0.423  0.321 -0.571  0.252  0.030
## PsOr.cn:2-1 -0.252  0.050 -0.354  0.259 -0.015  0.094 -0.275
## sssn2:gr2-1 -0.049  0.094 -0.425  0.381  0.098  0.032 -0.286  0.058
## sssn3:gr2-1 -0.085  0.023 -0.432  0.218  0.047  0.055 -0.093  0.262  0.119
## PsOr.:2:2-1  0.209 -0.124 -0.180  0.107  0.166  0.064  0.183 -0.064  0.176
## PsOr.:3:2-1  0.135 -0.085  0.033  0.048  0.104 -0.011  0.011 -0.181  0.159
##             s3:2-1 PO.:2:
## PosOr.cont
## session2
## session3
## group2-1
## PsOr.cnt:s2
## PsOr.cnt:s3
## PsOr.cn:2-1
## sssn2:gr2-1
## sssn3:gr2-1
## PsOr.:2:2-1  0.140
## PsOr.:3:2-1 -0.157  0.195
```

```
anova(m2)
```

```
## Analysis of Variance Table
##                          npar Sum Sq Mean Sq F value
## PosOr.cont                  1 2.4993  2.4993 30.6836
## session                     2 8.7004  4.3502 53.4068
## group                       1 7.2716  7.2716 89.2723
## PosOr.cont:session          2 0.0633  0.0317  0.3886
## PosOr.cont:group            1 0.3947  0.3947  4.8462
## session:group               2 0.2507  0.1254  1.5390
## PosOr.cont:session:group    2 0.0888  0.0444  0.5449
```

```
# save model output
saveRDS(m2, file = here::here("results", "tables",
                          "CSI_online_aphasia_SessionxGroup_glmm_cont.RDS"))
tab_model(m2,transform = NULL,
          show.re.var = F, show.stat = T,show.r2 = F,show.icc = F,
          title = "GLMM (Gamma distribution) of VOTs Predicted by Ordinal Position, Session and Group",
          dv.labels = "Vocal Onset Time",
          #string.pred = "",
          string.stat = "t-Value",
          file = here::here(
            "results", "tables",
            "CSI_online_aphasia_SessionxGroup_glmm_cont.html"))
```

GLMM (Gamma distribution) of VOTs Predicted by Ordinal Position, Session and Group

Vocal Onset Time

Predictors

Estimates

CI

t-Value

p

(Intercept)

1390.79

1385.83 – 1395.76

549.45

<0.001

PosOr cont

22.80

16.82 – 28.79

7.47

<0.001

session [2]

-98.05

-105.51 – -90.60

-25.78

<0.001

session [3]

-125.37

-132.95 – -117.78

-32.40

<0.001

group2-1

338.07

332.37 – 343.77

116.20

<0.001

PosOr cont × session [2]

6.17

1.61 – 10.72

2.65

0.008

PosOr cont × session [3]

5.63

-7.09 – 18.36

0.87

0.386

PosOr.cont:group2-1

15.69

11.13 – 20.25

6.75

<0.001

session2:group2-1

-30.71

-37.19 – -24.23

-9.29

<0.001

session3:group2-1

-42.60

-48.93 – -36.26

-13.19

<0.001

PosOr.cont:session2:group2-1

17.96

12.62 – 23.29

6.60

<0.001

PosOr.cont:session3:group2-1

6.85

2.13 – 11.57

2.84

0.004

N subject

40

N category

24

Observations

12455

Control analyses without PWA 1 and 17 The procedure deviated slightly in two participants.PWA1 had the same array in sessions 1 and 2 and was tested a day too late in session 2, PWA17 was tested a day too late in session 3. Here, we conduct control analyses without these two participants to test whether the data are affected.

RTs

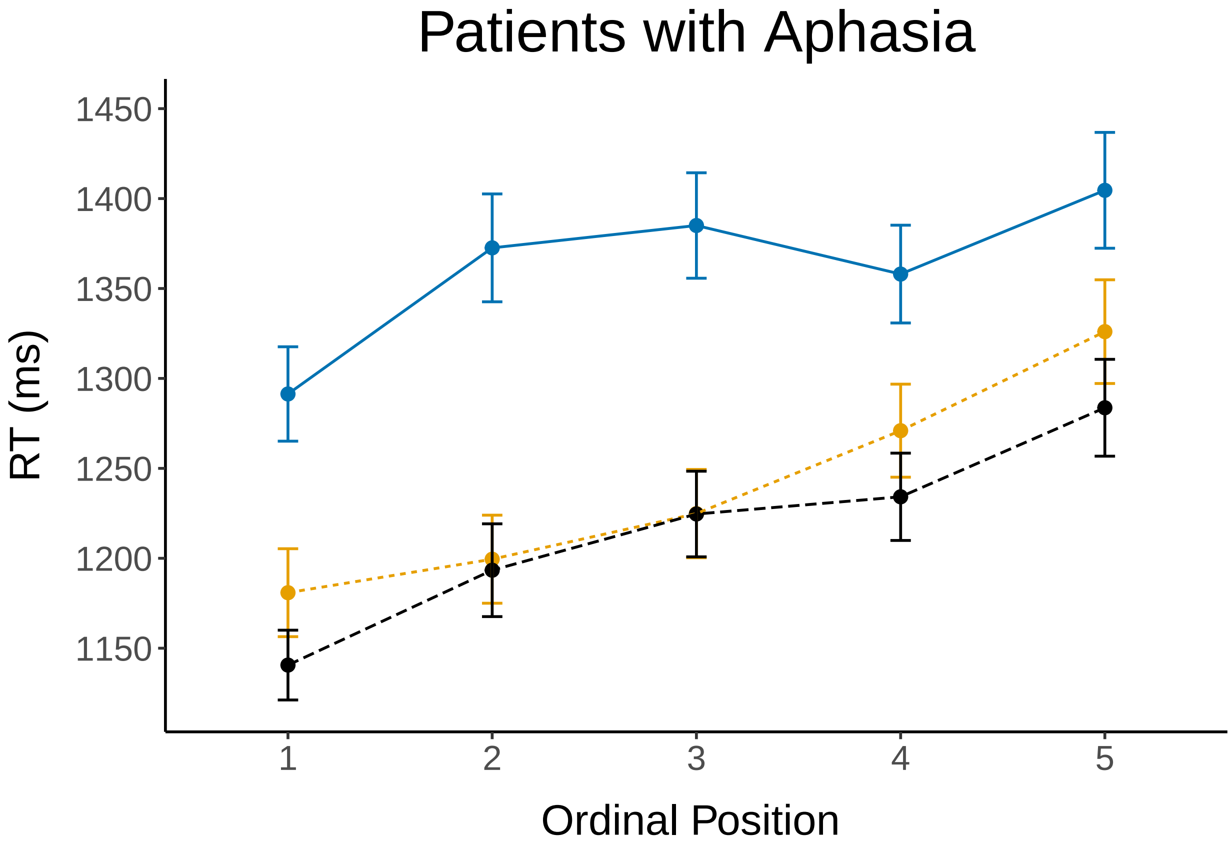
Plotting

```
(means_final<- df_RTs %>%
   filter(category != "Filler") %>%
   filter(subject != 101 & subject != 117) %>%
   Rmisc::summarySEwithin(.,"VOT",idvar = "subject",
                          withinvars = c("session", "PosOr"),
                          betweenvars = "group", na.rm = T))
```

```
##      group session PosOr   N      VOT       sd       se       ci
## 1  control       1     1 461 1309.126 386.6087 18.00616 35.38453
## 2  control       1     2 456 1289.949 368.6706 17.26458 33.92821
## 3  control       1     3 446 1364.816 419.3224 19.85549 39.02218
## 4  control       1     4 457 1341.318 375.5740 17.56861 34.52548
## 5  control       1     5 456 1359.540 395.4900 18.52052 36.39636
## 6  control       2     1 466 1216.815 327.2233 15.15832 29.78730
## 7  control       2     2 462 1256.861 364.0399 16.93667 33.28264
## 8  control       2     3 464 1225.331 289.9285 13.45959 26.44945
## 9  control       2     4 466 1256.116 296.8703 13.75225 27.02425
## 10 control       2     5 454 1278.405 329.4120 15.46007 30.38236
## 11 control       3     1 469 1211.518 323.1567 14.92199 29.32239
## 12 control       3     2 471 1184.996 282.9187 13.03621 25.61646
## 13 control       3     3 467 1220.181 305.2131 14.12358 27.75378
## 14 control       3     4 470 1244.625 306.1228 14.12039 27.74706
## 15 control       3     5 472 1275.156 349.0940 16.06835 31.57453
## 16     PWA       1     1 322 1291.359 470.8797 26.24110 51.62626
## 17     PWA       1     2 307 1372.601 525.3676 29.98431 59.00153
## 18     PWA       1     3 293 1385.028 501.7716 29.31381 57.69314
## 19     PWA       1     4 314 1358.021 481.5093 27.17314 53.46511
## 20     PWA       1     5 278 1404.586 536.9389 32.20346 63.39460
## 21     PWA       2     1 333 1180.871 446.1987 24.45154 48.09948
## 22     PWA       2     2 339 1199.500 450.7117 24.47930 48.15096
## 23     PWA       2     3 323 1224.846 440.9998 24.53789 48.27482
## 24     PWA       2     4 332 1270.946 471.2779 25.86473 50.87999
## 25     PWA       2     5 311 1326.022 508.4508 28.83160 56.73038
## 26     PWA       3     1 350 1140.621 363.0123 19.40383 38.16314
## 27     PWA       3     2 340 1193.366 475.6980 25.79835 50.74500
## 28     PWA       3     3 348 1224.618 444.0864 23.80553 46.82129
## 29     PWA       3     4 338 1234.189 446.2534 24.27299 47.74565
## 30     PWA       3     5 322 1283.698 483.1446 26.92460 52.97096
```

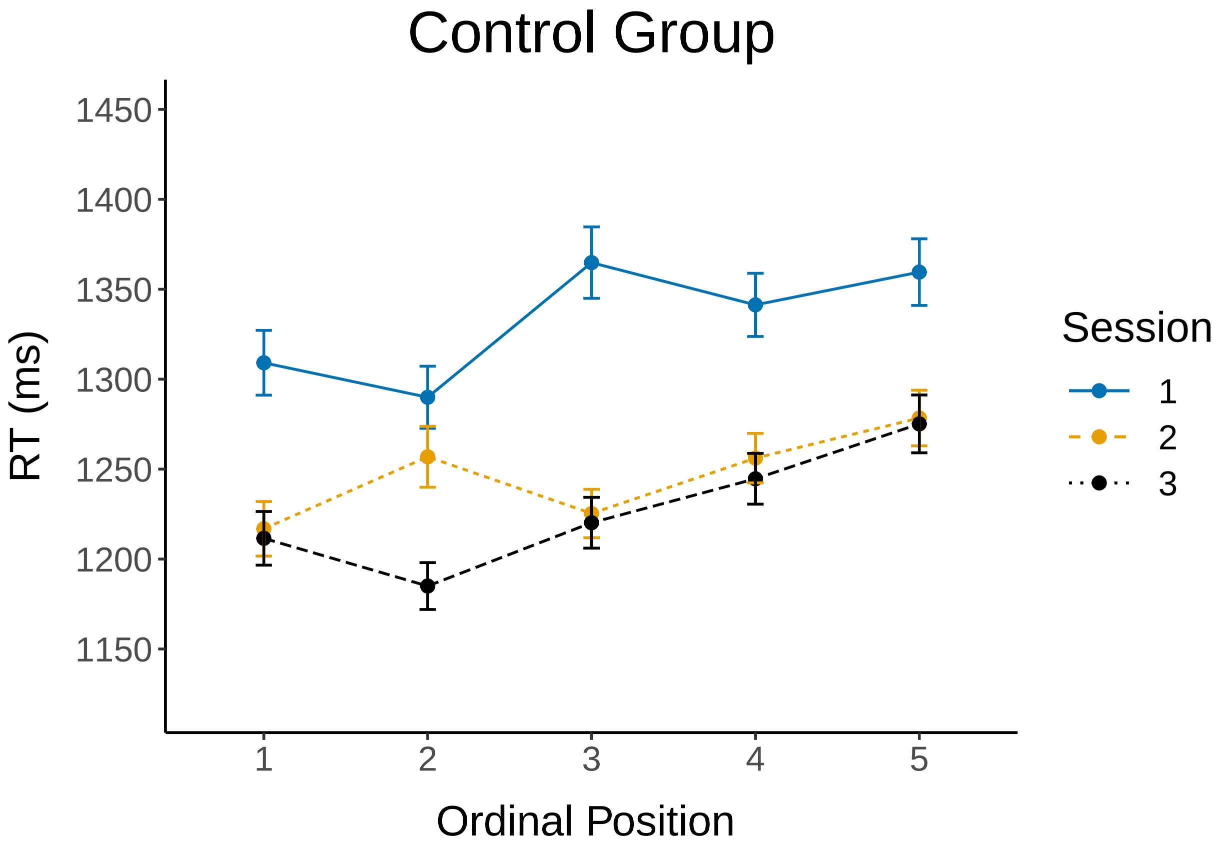
```
override.linetype<-c("solid", "dashed", "dotted")
(plot_rt_repetition_PWA <- means_final %>% filter(group=="PWA") %>%
    ggplot(., aes(x=PosOr, y=VOT, group=session, color = session)) +
    geom_point(size = 2)+
    stat_summary(aes(linetype=session),
                 fun=mean,  geom="line", size = 0.5) +
    scale_linetype_manual(values=c("solid", "dashed", "dotted"))+
    scale_color_manual(values=c("#0072B2", "#E69F00", "#000000"))+
    geom_errorbar(aes(ymin=VOT-se, ymax=VOT+se, group = session),
                  width =.1) +
    apatheme+
    scale_y_continuous(limits = c(1120, 1450),
                       breaks =seq(1150,1450, by = 50)) +
    labs(x="Ordinal Position ",y ="RT (ms)", colour="Session",
         linetype="Session",
         title = "Patients with Aphasia") +
    theme(
    axis.title.y = element_text(margin = margin(0,10,0,0)),
    axis.title.x = element_text(margin = margin(10,0,0,0)),
    legend.key.width = unit(1, "cm"),
    legend.position="none")+
   guides(color=guide_legend(
     override.aes=list(linetype=override.linetype)))+
    scale_linetype(guide="none"))
```

```
## Scale for linetype is already present.
## Adding another scale for linetype, which will replace the existing scale.
```



```
(plot_rt_repetition_control <- means_final %>%
    filter(group=="control") %>%
    ggplot(., aes(x=PosOr, y=VOT, group=session, color = session)) +
    geom_point(size = 2)+
    stat_summary(aes(linetype=session),fun=mean,  geom="line", size = 0.5) +
    scale_linetype_manual(values=c("solid", "dashed", "dotted"))+
    scale_color_manual(values=c("#0072B2", "#E69F00", "#000000"))+
    geom_errorbar(aes(ymin=VOT-se, ymax=VOT+se, group = session),
                  width =.1) +
    apatheme+
    scale_y_continuous(limits = c(1120, 1450),
                       breaks =seq(1150,1450, by = 50)) +
    labs(x="Ordinal Position ",y ="RT (ms)", colour="Session",
         linetype="Session",
         title = "Control Group") +
    theme(
    axis.title.y = element_text(margin = margin(0,10,0,0)),
    axis.title.x = element_text(margin = margin(10,0,0,0)),
    legend.key.width = unit(1, "cm"))+
   guides(color=guide_legend(
     override.aes=list(linetype=override.linetype)))+
    scale_linetype(guide="none"))
```

```
## Scale for linetype is already present.
## Adding another scale for linetype, which will replace the existing scale.
```



```
plots <- cowplot::plot_grid(plot_rt_repetition_PWA,
                            plot_rt_repetition_control,
          nrow = 1, ncol=2,  rel_widths = c(0.81,1),
          margin(1,1,1,1),
          labels = c("A", "B"),label_size = 34,
          label_y = 1.01,
          label_x=-0.03)
```

```
## Warning in as_grob.default(plot): Cannot convert object of class
## marginsimpleUnitunitunit_v2 into a grob.
```

```
filename <-
  "CSI_online_aphasia_spoken_plot_rt_by_repetition_without_101-117.pdf"
ggsave(plots, filename =
         here::here("results", "figures", filename),
       width = 25, height = 13, units = "cm",
       dpi = 300, device = cairo_pdf)
#embedFonts(file = here::here("results", "figures", filename))
```

Statistical analyses

```
df_RTs_red <- df_RTs %>% filter(subject != "101" & subject != "117") %>%
  droplevels()
df_RTs_red$PosOr.cont <- scale(as.numeric(as.character(df_RTs_red$PosOr)),
                                                    center = T, scale = F)
# m2_f <- afex::lmer_alt(lVOT ~ PosOr.cont*session*group +
#                 (PosOr.cont*session||subject) +
#                (PosOr.cont*session*group||category),
#               data = df_RTs_red,
#              control=lmerControl(optimizer = "bobyqa",
#                                    optCtrl = list(maxfun = 2e5)))
# m2_f <- afex::lmer_alt(lVOT ~ PosOr.cont*session*group +
#                 (PosOr.cont*session||subject) +
#                (PosOr.cont+session*group-session||category),
#               data = df_RTs_red,
#              control=lmerControl(optimizer = "bobyqa",
#                                    optCtrl = list(maxfun = 2e5)))
m2_f <- afex::lmer_alt(lVOT ~ PosOr.cont*session*group +
               (PosOr.cont+session||subject) +
              (PosOr.cont+group||category),
             data = df_RTs_red,
            control=lmerControl(optimizer = "bobyqa",
                                  optCtrl = list(maxfun = 2e5)))
# m2_f <- lmer(lVOT ~ PosOr.cont*session*group +
#                 (PosOr.cont+session|subject) +
#                (PosOr.cont+group|category),
#               data = df_RTs_red,
#              control=lmerControl(optimizer = "bobyqa",
#                                    optCtrl = list(maxfun = 2e5)))
summary(m2_f)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula:
## lVOT ~ PosOr.cont * session * group + (1 + re1.PosOr.cont + re1.session2 +
##     re1.session3 || subject) + (1 + re2.PosOr.cont + re2.groupPWA ||
##     category)
##    Data: data
## Control: lmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 200000))
##
## REML criterion at convergence: 566
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -5.0588 -0.6700 -0.1747  0.4856  4.7417
##
## Random effects:
##  Groups     Name           Variance   Std.Dev.
##  subject    (Intercept)    0.02712626 0.164701
##  subject.1  re1.PosOr.cont 0.00011070 0.010521
##  subject.2  re1.session2   0.00278164 0.052741
##  subject.3  re1.session3   0.00224735 0.047406
##  category   (Intercept)    0.00879305 0.093771
##  category.1 re2.PosOr.cont 0.00002853 0.005341
##  category.2 re2.groupPWA   0.00104258 0.032289
##  Residual                  0.05875025 0.242385
## Number of obs: 11787, groups:  subject, 38; category, 24
##
## Fixed effects:
##                                  Estimate   Std. Error           df t value
## (Intercept)                      7.085116     0.041816    51.242816 169.436
## PosOr.cont                       0.013363     0.004422   104.743190   3.022
## session2                        -0.063178     0.013800    35.126977  -4.578
## session3                        -0.079132     0.012778    33.206693  -6.193
## groupPWA                         0.210482     0.054569    37.088668   3.857
## PosOr.cont:session2             -0.001952     0.005059 11557.848779  -0.386
## PosOr.cont:session3              0.002122     0.005032 11552.244382   0.422
## PosOr.cont:groupPWA              0.003169     0.006681   138.528159   0.474
## session2:groupPWA               -0.021305     0.020707    38.666337  -1.029
## session3:groupPWA               -0.026097     0.019192    36.812204  -1.360
## PosOr.cont:session2:groupPWA     0.012872     0.007959 11587.162798   1.617
## PosOr.cont:session3:groupPWA     0.002883     0.007903 11588.134367   0.365
##                                          Pr(>|t|)
## (Intercept)                  < 0.0000000000000002 ***
## PosOr.cont                               0.003160 **
## session2                              0.000056588 ***
## session3                              0.000000534 ***
## groupPWA                                 0.000442 ***
## PosOr.cont:session2                      0.699649
## PosOr.cont:session3                      0.673249
## PosOr.cont:groupPWA                      0.635994
## session2:groupPWA                        0.309918
## session3:groupPWA                        0.182169
## PosOr.cont:session2:groupPWA             0.105839
## PosOr.cont:session3:groupPWA             0.715293
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) PsOr.c sessn2 sessn3 grpPWA PsO.:2 PsO.:3 PO.:PW s2:PWA
## PosOr.cont  -0.001
## session2    -0.045  0.003
## session3    -0.048  0.003  0.147
## groupPWA    -0.606  0.001  0.034  0.037
## PsOr.cnt:s2  0.001 -0.573 -0.004 -0.003 -0.001
## PsOr.cnt:s3  0.001 -0.577 -0.003 -0.007 -0.001  0.504
## PsOr.cn:PWA  0.001 -0.622 -0.002 -0.002  0.003  0.380  0.382
## sssn2:grPWA  0.030 -0.002 -0.666 -0.098 -0.060  0.003  0.002 -0.005
## sssn3:grPWA  0.032 -0.002 -0.098 -0.666 -0.066  0.002  0.004 -0.006  0.169
## PsOr.:2:PWA -0.001  0.365  0.002  0.002 -0.002 -0.636 -0.320 -0.609  0.006
## PsOr.:3:PWA -0.001  0.367  0.002  0.004 -0.002 -0.321 -0.637 -0.616  0.004
##             s3:PWA PO.:2:
## PosOr.cont
## session2
## session3
## groupPWA
## PsOr.cnt:s2
## PsOr.cnt:s3
## PsOr.cn:PWA
## sssn2:grPWA
## sssn3:grPWA
## PsOr.:2:PWA  0.004
## PsOr.:3:PWA  0.005  0.515
```

```
didLmerConverge(m2_f)
```

```
##  The relative maximum gradient of 0.000000349 is less than our 0.001 criterion.
##  You can safely ignore any warnings about a claimed convergence failure.
```

Error rates

```
df_errors_red <- df_errors %>%
  filter(subject != "101" & subject != "117") %>%
  filter(group=="PWA") %>%
  droplevels()
df_errors_red$PosOr.cont <-
  scale(as.numeric(as.character(df_errors_red$PosOr)),
        center = T, scale = F)
# m2_f <- afex::lmer_alt(error_class ~ PosOr.cont*session +
#                 (PosOr.cont*session||subject) +
#                (PosOr.cont*session||category),
#               data = df_errors_red,
#               family="binomial",
#              control=glmerControl(optimizer = "bobyqa",
#                                    optCtrl = list(maxfun = 2e5)))
# m2_f <- afex::lmer_alt(error_class ~ PosOr.cont*session*group +
#                 (PosOr.cont||subject) +
#                (PosOr.cont+group||category),
#               data = df_errors_red,
#              family="binomial",
#             control=glmerControl(optimizer = "bobyqa",
#                                  optCtrl = list(maxfun = 2e5)))
# m2_f <- afex::lmer_alt(lVOT ~ PosOr.cont*session*group +
#                (PosOr.cont*session||subject) +
#               (PosOr.cont+session*group-session||category),
#              data = df_RTs_red,
#             control=lmerControl(optimizer = "bobyqa",
#                                  optCtrl = list(maxfun = 2e5)))
m2_f <- afex::lmer_alt(lVOT ~ PosOr.cont*session*group +
               (PosOr.cont+session||subject) +
              (PosOr.cont+group||category),
             data = df_RTs_red,
            control=lmerControl(optimizer = "bobyqa",
                                 optCtrl = list(maxfun = 2e5)))
# m2_f <- lmer(lVOT ~ PosOr.cont*session*group +
#                (PosOr.cont+session|subject) +
#               (PosOr.cont+group|category),
#              data = df_RTs_red,
#             control=lmerControl(optimizer = "bobyqa",
#                                  optCtrl = list(maxfun = 2e5)))
summary(m2_f)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula:
## lVOT ~ PosOr.cont * session * group + (1 + re1.PosOr.cont + re1.session2 +
##     re1.session3 || subject) + (1 + re2.PosOr.cont + re2.groupPWA ||
##     category)
##    Data: data
## Control: lmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 200000))
##
## REML criterion at convergence: 566
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -5.0588 -0.6700 -0.1747  0.4856  4.7417
##
## Random effects:
##  Groups     Name           Variance   Std.Dev.
##  subject    (Intercept)    0.02712626 0.164701
##  subject.1  re1.PosOr.cont 0.00011070 0.010521
##  subject.2  re1.session2   0.00278164 0.052741
##  subject.3  re1.session3   0.00224735 0.047406
##  category   (Intercept)    0.00879305 0.093771
##  category.1 re2.PosOr.cont 0.00002853 0.005341
##  category.2 re2.groupPWA   0.00104258 0.032289
##  Residual                  0.05875025 0.242385
## Number of obs: 11787, groups:  subject, 38; category, 24
##
## Fixed effects:
##                                   Estimate   Std. Error           df t value
## (Intercept)                       7.085116     0.041816    51.242816 169.436
## PosOr.cont                       0.013363     0.004422   104.743190   3.022
## session2                        -0.063178     0.013800    35.126977  -4.578
## session3                        -0.079132     0.012778    33.206693  -6.193
## groupPWA                         0.210482     0.054569    37.088668   3.857
## PosOr.cont:session2             -0.001952     0.005059 11557.848779  -0.386
## PosOr.cont:session3              0.002122     0.005032 11552.244382   0.422
## PosOr.cont:groupPWA              0.003169     0.006681   138.528159   0.474
## session2:groupPWA               -0.021305     0.020707    38.666337  -1.029
## session3:groupPWA               -0.026097     0.019192    36.812204  -1.360
## PosOr.cont:session2:groupPWA     0.012872     0.007959 11587.162798   1.617
## PosOr.cont:session3:groupPWA     0.002883     0.007903 11588.134367   0.365
##                                           Pr(>|t|)
## (Intercept)                  < 0.0000000000000002 ***
## PosOr.cont                               0.003160 **
## session2                              0.000056588 ***
## session3                              0.000000534 ***
## groupPWA                                 0.000442 ***
## PosOr.cont:session2                      0.699649
## PosOr.cont:session3                      0.673249
## PosOr.cont:groupPWA                      0.635994
## session2:groupPWA                        0.309918
## session3:groupPWA                        0.182169
## PosOr.cont:session2:groupPWA             0.105839
## PosOr.cont:session3:groupPWA             0.715293
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) PsOr.c sessn2 sessn3 grpPWA PsO.:2 PsO.:3 PO.:PW s2:PWA
## PosOr.cont  -0.001
## session2    -0.045  0.003
## session3    -0.048  0.003  0.147
## groupPWA    -0.606  0.001  0.034  0.037
## PsOr.cnt:s2  0.001 -0.573 -0.004 -0.003 -0.001
## PsOr.cnt:s3  0.001 -0.577 -0.003 -0.007 -0.001  0.504
## PsOr.cn:PWA  0.001 -0.622 -0.002 -0.002  0.003  0.380  0.382
## sssn2:grPWA  0.030 -0.002 -0.666 -0.098 -0.060  0.003  0.002 -0.005
## sssn3:grPWA  0.032 -0.002 -0.098 -0.666 -0.066  0.002  0.004 -0.006  0.169
## PsOr.:2:PWA -0.001  0.365  0.002  0.002 -0.002 -0.636 -0.320 -0.609  0.006
## PsOr.:3:PWA -0.001  0.367  0.002  0.004 -0.002 -0.321 -0.637 -0.616  0.004
##             s3:PWA PO.:2:
## PosOr.cont
## session2
## session3
## groupPWA
## PsOr.cnt:s2
## PsOr.cnt:s3
## PsOr.cn:PWA
## sssn2:grPWA
## sssn3:grPWA
## PsOr.:2:PWA  0.004
## PsOr.:3:PWA  0.005  0.515
```

```
didLmerConverge(m2_f)
```

```
##  The relative maximum gradient of 0.000000349 is less than our 0.001 criterion.
##  You can safely ignore any warnings about a claimed convergence failure.
```

Comparison to verbal CSI with young participants Load data

Load data from both the verbal online CSI experiment (Stark et al., 2022)

```
load(here::here("data", "verbal_CSI_young_starketal2022",
                "CSI_online_verbal_df_full.RData"))
df_young <- df_full
```

Combine both data frames into one

1) Subset relevant columns and give identical names

```
df_young <- df_young %>%
  dplyr::select(VP, Item, subcat, VOT, correct, Pos) %>%
  dplyr::rename(subject = VP, item = Item,
                category = subcat, PosOr=Pos) %>%
  mutate(group="young") %>%
  mutate(session="young group")

x <- df_RTs %>%
  dplyr::select(subject, group, session, item, category, VOT, PosOr)
```

2) Give subjects from both experiments different names

```
df_young <- df_young %>% mutate(subject = subject + 300)
```

3) Put columns into correct format

```
df_young <- df_young %>%
  mutate(subject = as.factor(subject)) %>%
  mutate(item = as.character(item)) %>%
  mutate(category = as.factor(category)) %>%
  mutate(VOT = as.numeric(VOT)) %>%
  mutate(PosOr = as.factor(PosOr)) %>%
  filter(!is.na(correct) & correct != 0) %>%
  dplyr::select(-correct) %>%
  droplevels()
```

4) Bind both data frames into one

```
df_combi <- bind_rows(x, df_young)
```

5) Give identical category names in both experiments

```r
df_combi <- df_combi %>% dplyr::mutate(category = case_when(
  category == "Buero" ~ "Büro",
  category == "Gebaeude" ~ "Gebäude",
  category == "Gemuese" ~ "Gemüse",
  category ==  "Koerperteile" ~ "Körperteile",
  category == "Kueche" ~ "Küche",
  category == "Suessigkeiten" ~ "Süssigkeiten",
  category == "Trinkgefaesse" ~ "Trinkgefässe",
  category == "Voegel" ~ "Vögel",
  TRUE ~ as.character(category))) %>%
  mutate(category == as.factor(category)) %>% droplevels()
table(df_combi$category)
```

```
##
## Aufbewahrung    Bauernhof       Blumen         Büro      Filler1      Filler2
##          652          681          567          657          586          581
##       Fische      Gebäude       Gemüse   Heimwerker     Huftiere     Insekten
##          669          628          678          658          693          657
##  Instrumente       Jacken       Kochen  Körperteile        Küche         Obst
##          651          595          652          696          649          709
##    Raubtiere      Schmuck       Sitzen      Strasse Süssigkeiten Trinkgefässe
##          629          610          652          692          687          652
##        Vögel       Wasser
##          684          621
```

5) Drop filler trials

```r
 df_combi <- df_combi %>% filter(category != "Filler" &
                     category != "Filler1" & category != "Filler2") %>%
  droplevels()
```

6) Export combined data frame for post-hoc power plot

```r
write.csv(df_combi, here::here("data", "CSI_online_young_PWA_old_combined.csv"))
```

Descriptives

```r
(descriptives <- df_combi %>%
   Rmisc::summarySEwithin(.,"VOT",idvar = "subject",
                          withinvars = c("PosOr", "session"),
                          betweenvars = "group",
                          na.rm = T))
```

```
## Automatically converting the following non-factors to factors: group, session
```

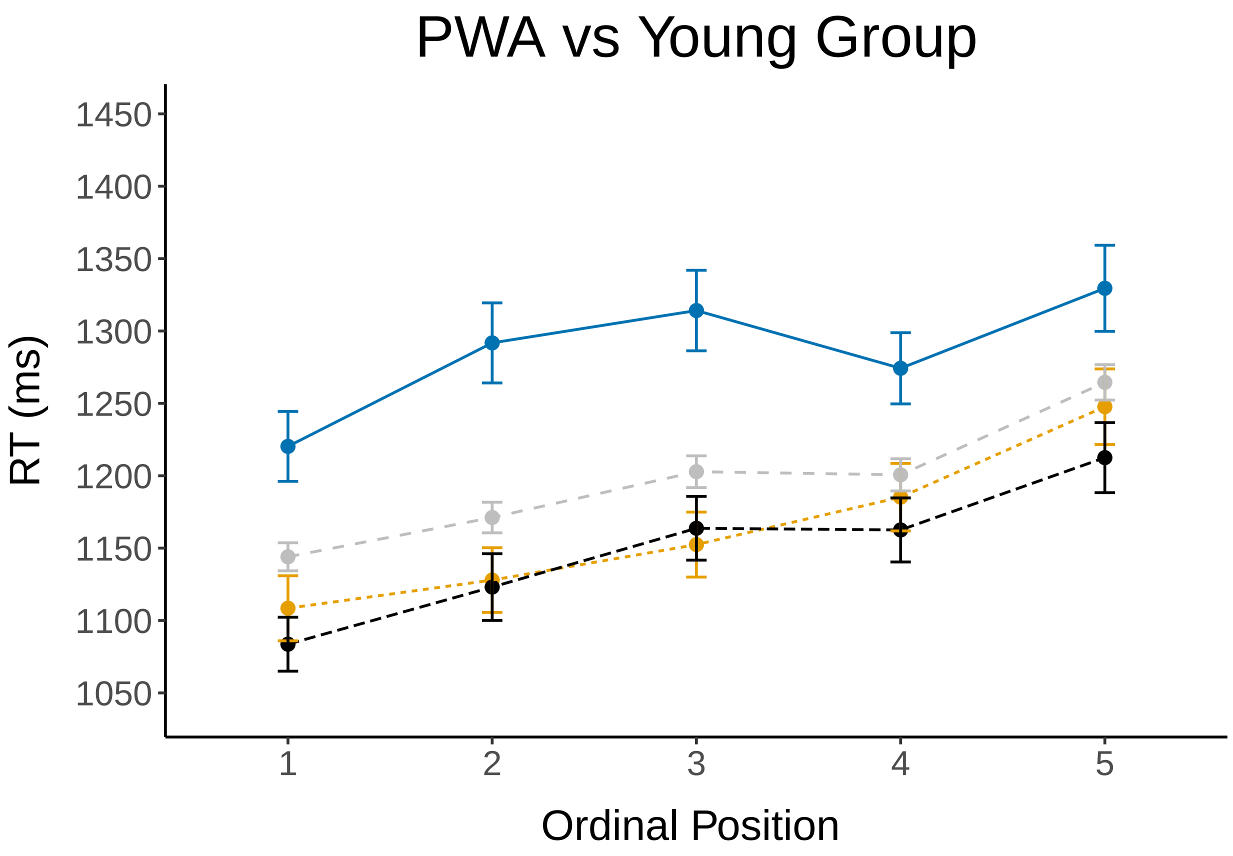
```
##      group PosOr     session   N      VOT       sd        se        ci
## 1  control     1           1 461 1236.721 383.2024 17.847516 35.07277
## 2  control     1           2 466 1144.410 324.3402 15.024768 29.52485
## 3  control     1           3 469 1139.114 320.3094 14.790516 29.06404
## 4  control     2           1 456 1217.544 365.4223 17.112469 33.62928
## 5  control     2           2 462 1184.456 360.8324 16.787443 32.98939
## 6  control     2           3 471 1112.591 280.4260 12.921349 25.39076
## 7  control     3           1 446 1292.412 415.6279 19.680549 38.67837
## 8  control     3           2 464 1152.927 287.3740 13.341002 26.21641
## 9  control     3           3 467 1147.777 302.5239 13.999138 27.50925
## 10 control     4           1 457 1268.913 372.2649 17.413816 34.22128
## 11 control     4           2 466 1183.711 294.2546 13.631082 26.78615
## 12 control     4           3 470 1172.220 303.4256 13.995981 27.50259
## 13 control     5           1 456 1287.135 392.0054 18.357336 36.07568
## 14 control     5           2 454 1206.000 326.5096 15.323856 30.11466
## 15 control     5           3 472 1202.752 346.0182 15.926780 31.29634
## 16     PWA     1           1 362 1220.274 459.3182 24.141226 47.47510
## 17     PWA     1           2 375 1108.467 435.1172 22.469357 44.18211
## 18     PWA     1           3 395 1083.626 370.4159 18.637649 36.64168
## 19     PWA     2           1 345 1291.769 513.4978 27.645791 54.37606
## 20     PWA     2           2 386 1127.932 439.4589 22.367867 43.97847
## 21     PWA     2           3 386 1123.099 453.0649 23.060394 45.34007
## 22     PWA     3           1 337 1314.130 510.4865 27.807971 54.69965
## 23     PWA     3           2 370 1152.440 431.8690 22.451801 44.14953
## 24     PWA     3           3 396 1163.716 438.2590 22.023345 43.29763
## 25     PWA     4           1 357 1274.240 464.6651 24.592687 48.36521
## 26     PWA     4           2 378 1185.207 453.1751 23.308811 45.83156
## 27     PWA     4           3 384 1162.566 433.5719 22.125623 43.50289
## 28     PWA     5           1 322 1329.476 533.6370 29.738428 58.50684
## 29     PWA     5           2 355 1247.699 492.1252 26.119292 51.36849
## 30     PWA     5           3 370 1212.513 465.4573 24.197971 47.58322
## 31   young     1 young group 670 1144.019 250.5367  9.679078 19.00503
## 32   young     2 young group 651 1171.140 269.5845 10.565851 20.74732
## 33   young     3 young group 662 1202.806 282.1272 10.965186 21.53079
## 34   young     4 young group 650 1200.640 283.4120 11.116334 21.82832
## 35   young     5 young group 631 1264.514 307.4940 12.241145 24.03838
```

Plotting

Plot RTs by Session and ordinal position for both experiments

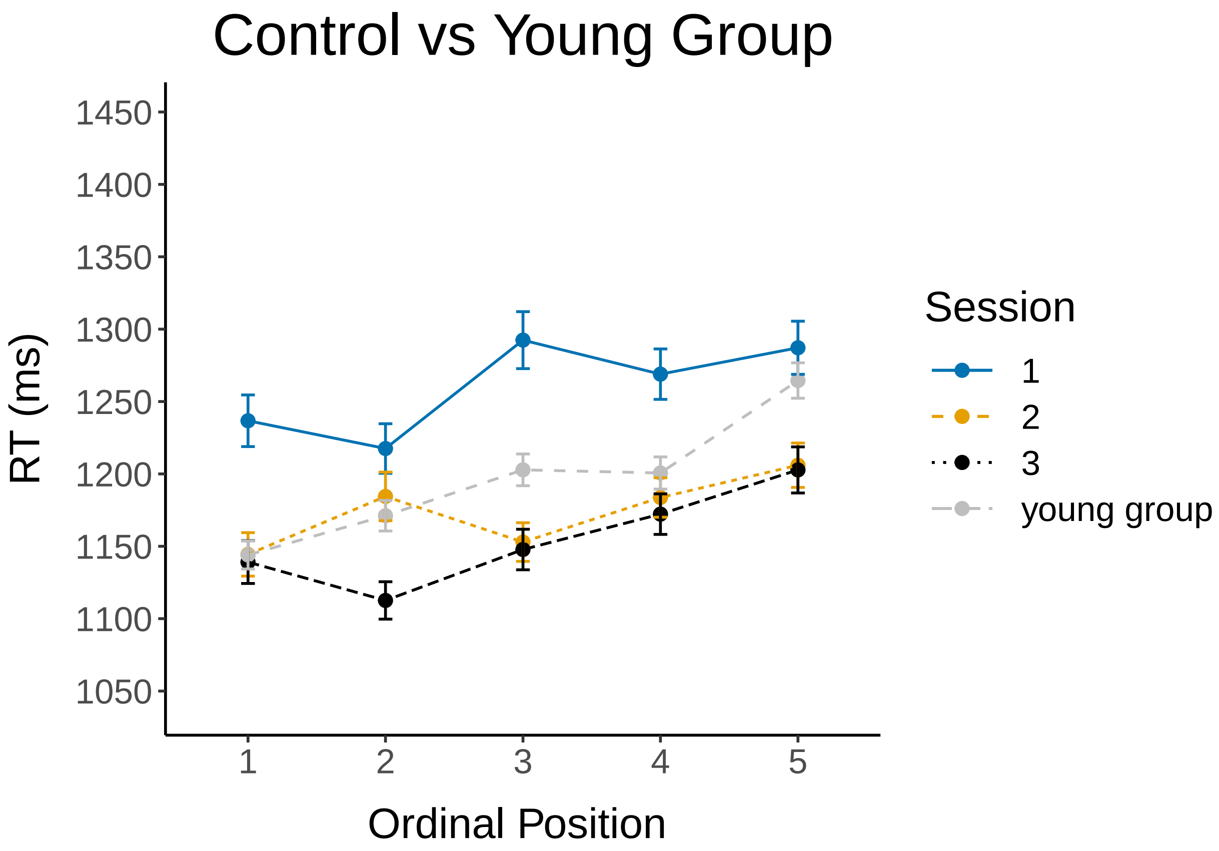
```
override.linetype<-c("solid", "dashed", "dotted", "longdash")
(plot_rt_repetition_PWA <- descriptives %>%
    filter(group=="PWA" | group=="young") %>%
    ggplot(., aes(x=PosOr, y=VOT, group=session, color = session)) +
    geom_point(size = 2)+
    stat_summary(aes(linetype=session),fun=mean,
                 geom="line", size = 0.5) +
    scale_linetype_manual(values=c("solid", "dashed",
                                   "dotted", "longdash"))+
    scale_color_manual(values=c("#0072B2", "#E69F00", "#000000", "gray"))+
    geom_errorbar(aes(ymin=VOT-se, ymax=VOT+se, group = session),
                  width =.1) +
    apatheme  +
    scale_y_continuous(limits = c(1040, 1450),
                       breaks =seq(1050,1450, by = 50)) +
    labs(x="Ordinal Position ",y ="RT (ms)", colour="Session",
         linetype="Session",
         title = "PWA vs Young Group") +
    theme(
    axis.title.y = element_text(margin = margin(0,10,0,0)),
    axis.title.x = element_text(margin = margin(10,0,0,0)),
    legend.key.width = unit(1, "cm"),
    legend.position="none")+
   guides(color=guide_legend(
     override.aes=list(linetype=override.linetype)))+
    scale_linetype(guide="none"))
```

```
## Scale for linetype is already present.
## Adding another scale for linetype, which will replace the existing scale.
```



```
(plot_rt_repetition_control <- descriptives %>%
    filter(group=="control" | group=="young") %>%
    ggplot(., aes(x=PosOr, y=VOT, group=session, color = session)) +
    geom_point(size = 2)+
    stat_summary(aes(linetype=session),fun=mean,
                 geom="line", size = 0.5) +
    scale_linetype_manual(values=c("solid", "dashed", "dotted", "longdash"))+
    scale_color_manual(values=c("#0072B2", "#E69F00", "#000000", "gray"))+
    geom_errorbar(aes(ymin=VOT-se, ymax=VOT+se, group = session),
                  width =.1) +
    apatheme+
    scale_y_continuous(limits = c(1040, 1450),
                       breaks =seq(1050,1450, by = 50)) +
    labs(x="Ordinal Position ",y ="RT (ms)", colour="Session",
         linetype="Session",
         title = "Control vs Young Group") +
    theme(
    axis.title.y = element_text(margin = margin(0,10,0,0)),
    axis.title.x = element_text(margin = margin(10,0,0,0)),
    legend.key.width = unit(1, "cm"))+
   guides(color=guide_legend(
     override.aes=list(linetype=override.linetype)))+
    scale_linetype(guide="none"))
```

```
## Scale for linetype is already present.
## Adding another scale for linetype, which will replace the existing scale.
```



```
plots <- cowplot::plot_grid(
  plot_rt_repetition_PWA,plot_rt_repetition_control,
          nrow = 1, ncol=2,  rel_widths = c(0.7,1), #rel_height = c(1,1),
          margin(1,1,1,1),
          labels = c("A", "B"),label_size = 34,
                       label_fontfamily = "Helvetica", label_y = 1.01,
  label_x=-0.03)
```

```
## Warning in as_grob.default(plot): Cannot convert object of class
## marginsimpleUnitunitunit_v2 into a grob.
```

```
filename <- "CSI_online_aphasia_spoken_comparison-to-young.pdf"
ggsave(plots, filename =
         here::here("results", "figures", filename),
       width = 25, height = 13, units = "cm",
       dpi = 300, device = cairo_pdf)
#embedFonts(file = here::here("results", "figures", filename))
```

Appendix

List of stimuli

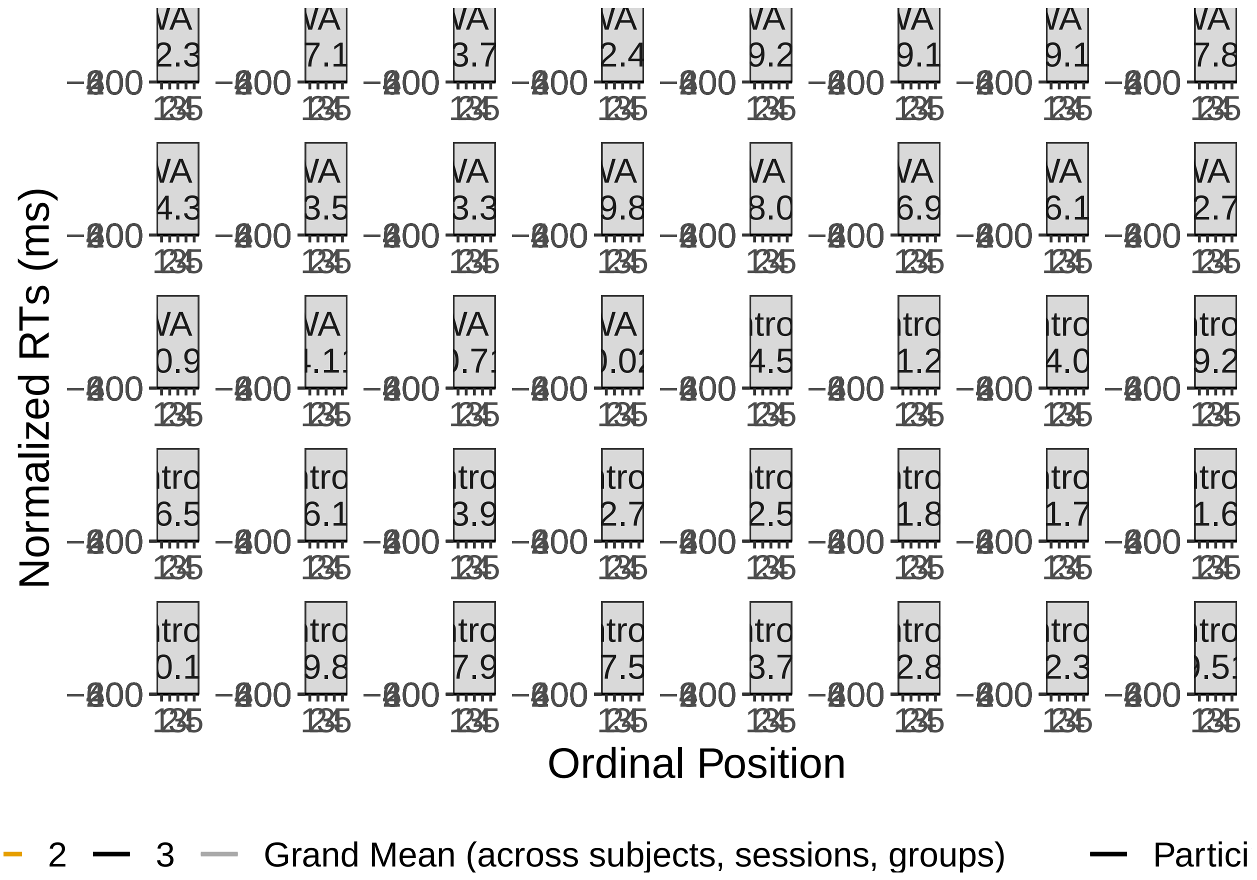
```
df %>% arrange(category) %>%
  group_by(category, item, correct, AR) %>% count()
```

```
## # A tibble: 1,657 x 5
## # Groups:   category, item, correct, AR [1,657]
##    category     item           correct AR                           n
##    <chr>        <chr>          <chr>   <chr>                    <int>
##  1 Aufbewahrung kleiderschrank 0       Br [e.o.r.]                  1
##  2 Aufbewahrung kleiderschrank 0       Greu [e.o.r.]                1
##  3 Aufbewahrung kleiderschrank 0       Kreider [e.o.r.]             1
##  4 Aufbewahrung kleiderschrank 0       NR                           9
##  5 Aufbewahrung kleiderschrank 0       Schra e Schra e              1
##  6 Aufbewahrung kleiderschrank 1       <NA>                        11
##  7 Aufbewahrung kleiderschrank 1.1     Holzschrank                  1
##  8 Aufbewahrung kleiderschrank 1.1     Schrank                     92
##  9 Aufbewahrung kleiderschrank 1.1     Schrank Gleicher Schrank     1
## 10 Aufbewahrung kleiderschrank 1.2     Schran Schank                1
## # i 1,647 more rows
```

Response times and error rates by participant and category

RTs by subject Line graph for each participant:

```
modeloutput <- coef(m2)$subject
means_final_subject <- df_RTs %>%
   summarySEwithin(.,"VOT",withinvars = c("subject","PosOr", "session"),
                   betweenvars="group")
means_final<- df_RTs %>%
   Rmisc::summarySEwithin(.,"VOT",idvar = "subject",
                          withinvars = c("PosOr"),
                          na.rm = T)

for(i in 1:nrow(means_final_subject)) {
  means_final_subject$grandmean[i] <-
    means_final$VOT[means_final$PosOr == means_final_subject$PosOr[i]] -
    means_final$VOT[means_final$PosOr== 1]
  means_final_subject$normalizedRT[i] <-
    means_final_subject$VOT[i] -
    means_final_subject$VOT[means_final_subject$subject ==
                              means_final_subject$subject[i] &
                              means_final_subject$PosOr == 1 &
                                    means_final_subject$session == 1]
  # prepare for ordering
  means_final_subject$effect[i] <-
    round(modeloutput$PosOr.cont[means_final_subject$subject[i]] +
    modeloutput$re1.PosOr.cont[means_final_subject$subject[i]] +
    modeloutput$re2.PosOr.cont[means_final_subject$subject[i]],2)
}


means_final_subject <- means_final_subject[
  order(desc(means_final_subject$group),
        desc(means_final_subject$effect)),]
means_final_subject$effect <-
  as.factor(round(means_final_subject$effect, 2))
means_final_subject$effect <-
  factor(means_final_subject$effect,
         levels=rev(levels(means_final_subject$effect )))


# add participant number
means_final_subject <- means_final_subject %>%
  mutate(subject_en = case_when(
    group == "PWA" ~ paste0("PWA ",
                            substr(as.character(
                              means_final_subject$subject), 2,3),
                            "\n(",effect,")",sep=''),
    group == "control" ~ paste0("Control ",
                            substr(as.character(means_final_subject$subject), 2,3),
                            "\n(",effect,")",sep='')))  %>%
  mutate(subject_en = case_when(subject_en=="PWA 04\n(29.1)" ~
                                    "PWA 04\n(29.10)",
                                  subject_en=="PWA 16\n(24.3)" ~
                                    "PWA 16\n(24.30)",
                                  subject_en=="Participant 12\n(38.3)" ~
                                    "Participant 12\n(38.30)",
                                    subject_en=="Control 12\n(17.5)" ~
                                      "Control 12\n(17.50)",
                                  TRUE~subject_en)) %>%
  mutate(subject_en=factor(subject_en,levels=c(
     "PWA 03\n(42.36)","PWA 05\n(37.11)","PWA 20\n(33.71)",
     "PWA 13\n(32.42)","PWA 07\n(29.28)","PWA 08\n(29.13)",
     "PWA 04\n(29.10)","PWA 12\n(27.84)","PWA 16\n(24.30)",
     "PWA 18\n(23.59)","PWA 06\n(23.31)", "PWA 09\n(19.86)",
     "PWA 14\n(18.04)","PWA 11\n(16.91)","PWA 17\n(16.18)",
     "PWA 10\n(12.79)","PWA 19\n(10.94)","PWA 02\n(4.11)",
     "PWA 01\n(0.71)","PWA 15\n(0.02)","Control 09\n(44.58)",
     "Control 17\n(41.22)","Control 10\n(34.06)","Control 20\n(29.22)",
     "Control 02\n(26.54)","Control 07\n(26.11)","Control 01\n(23.97)",
     "Control 05\n(22.77)","Control 03\n(22.53)","Control 15\n(21.83)",
     "Control 19\n(21.75)", "Control 13\n(21.66)","Control 14\n(20.13)",
     "Control 04\n(19.86)","Control 08\n(17.97)","Control 12\n(17.50)",
     "Control 11\n(13.73)","Control 06\n(12.86)","Control 16\n(12.39)",
     "Control 18\n(9.51)" )))

# Plotting
(plot_rt_subject <- means_final_subject %>%
    ggplot(., aes(x=PosOr,y=normalizedRT,
                  color=session, group=session, na.rm=T)) +
    geom_point(size =1, color = 'black') +
    geom_line(aes(x=PosOr,y=normalizedRT, color=session, linetype="c"),
              size = 0.5) +
    geom_line(aes(x=PosOr,y=grandmean, color="b", linetype="d"),
              group = 1,size = 0.8)+
    geom_errorbar(aes(ymin=normalizedRT-se, ymax=normalizedRT+se),
                  width =.1) +
    scale_color_manual(name="Session",
                       values=c(
                         "#0072B2", "#E69F00", "#000000", "dark gray"),
                       labels=c(
                         "1", "2", "3",
                      "Grand Mean (across subjects, sessions, groups)")) +
    scale_linetype_manual(name="",values=c("c"="solid","d"="dashed"),
                          labels=c("Participant mean",
                                   "Grand Mean"))+
    apatheme+
    labs(x="Ordinal Position",y ="Normalized RTs (ms)") +
    facet_wrap(means_final_subject$subject_en, scales='free', ncol=8)+
    scale_y_continuous(limits = c(-800, 800),
                       breaks = c(-600,-400,-200,0,200,400,600)) +
    scale_x_discrete(breaks=c(1,2,3,4,5))+
    theme(legend.position = "bottom"))
```



```
filename <- "CSI_online_aphasia_effect_by_participant.pdf"
ggsave(plot_rt_subject, filename =
         here::here("results", "figures", filename),
       width = 34, height = 26, units = "cm",
       dpi = 300, device = cairo_pdf)
#embedFonts(file = here::here("results", "figures", filename))

## save for exploratory analyses:
means_final_subject_RTs <- means_final_subject
```

Get an idea of the variance of the ordinal position effect between groups

```
# # sd of ordinal position effects based on model estimate
# means_final_subject %>% group_by(group, session) %>%
#   summarise(mean_effect=mean(effect), sd_effect=sd(effect))
# means_final_subject %>%
#    summarySEwithin(.,"effect",withinvars = c("PosOr", "session"),
#                    betweenvars="group")
# # sd of raw ordinal position effect
# means_final_subject <- means_final_subject %>% arrange(subject, group, session, PosOr)
# for(i in 1:nrow(means_final_subject)) {
#   if(means_final_subject$PosOr[i]==1){
#     means_final_subject$raw_ordpos[i] <- NA
#   } else {
#     means_final_subject$raw_ordpos[i] <-
#       means_final_subject$VOT[i] - means_final_subject$VOT[i-1]
#   }
# }
# means_final_subject %>%
#    summarySEwithin(.,"raw_ordpos",withinvars = c("session"),
#                    betweenvars="group", na.rm=T)
```

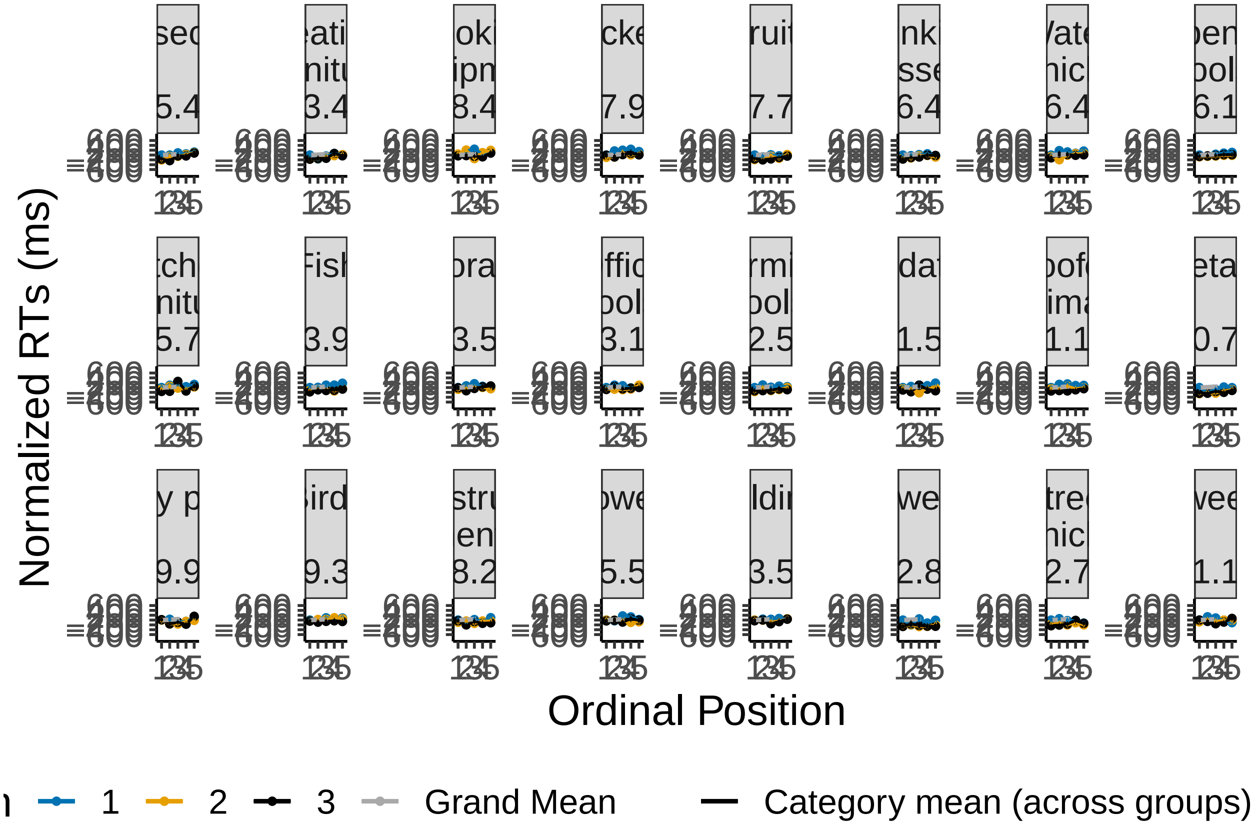
RTs by category Line graph for each category:

```
modeloutput <- coef(m2)$category
means_final_category <- df_RTs %>%
   summarySEwithin(.,"VOT",withinvars =
                     c("category","PosOr", "session"))
```

```
## Automatically converting the following non-factors to factors: category
```

```
means_final<- df_RTs %>%
   Rmisc::summarySEwithin(.,"VOT",idvar = "category",
                          withinvars = c("PosOr"),
                          na.rm = T)

for(i in 1:nrow(means_final_category)) {
  means_final_category$grandmean[i] <-
    means_final$VOT[means_final$PosOr == means_final_category$PosOr[i]] -
    means_final$VOT[means_final$PosOr== 1]
  means_final_category$normalizedRT[i] <-
    means_final_category$VOT[i] -
    means_final_category$VOT[means_final_category$category == means_final_category$category[i] & means_
                                    means_final_category$session == 1]
  # prepare for ordering
  means_final_category$effect[i] <-
    modeloutput$PosOr.cont[means_final_category$category[i]] +
    modeloutput$re2.PosOr.cont[means_final_category$category[i]]
}

means_final_category <- means_final_category[
  order(desc(means_final_category$effect)),]
means_final_category$effect <-
  as.factor(round(means_final_category$effect, 2))
means_final_category$effect <-
  factor(means_final_category$effect,
         levels=rev(levels(means_final_category$effect )))
means_final_category$category <- factor(
  means_final_category$category, levels=c(
       "Insekten",     "Sitzen",       "Kochen",       "Jacken",
       "Obst",         "Trinkgefässe", "Wasser",       "Heimwerker",
       "Küche",        "Fische",       "Aufbewahrung",
       "Büro", "Bauernhof", "Raubtiere", "Huftiere", "Gemüse",
       "Körperteile", "Vögel", "Instrumente", "Blumen",
       "Gebäude", "Schmuck", "Strasse", "Süssigkeiten"))

# order category levels by effect size
```

```
means_final_category$category <- factor(
  means_final_category$category, levels=c(
    "Gebäude","Schmuck","Raubtiere","Sitzen","Jacken",
    "Blumen","Huftiere","Wasser","Trinkgefässe","Küche",
    "Insekten","Büro","Bauernhof","Strasse","Kochen",
    "Gemüse","Körperteile","Fische","Heimwerker","Aufbewahrung",
    "Obst","Vögel","Instrumente","Süssigkeiten"))
# give categories English names and combine with effect size
means_final_category <- means_final_category %>%
  mutate(category_en = case_when(
    category == "Aufbewahrung" ~ paste0(
      "Storage\n\n(", effect, ")", sep=''),
    category == "Bauernhof" ~ paste0(
      "Farming\ntools\n(", effect, ")", sep=''),
    category == "Blumen" ~ paste0(
      "Flowers\n\n(", effect, ")", sep=''),
    category == "Büro" ~ paste0(
      "Office\ntools\n(", effect, ")", sep=''),
    category == "Fische" ~ paste0(
      "Fish\n\n(", effect, ")", sep=''),
    category == "Gebäude" ~ paste0(
      "Buildings\n\n(", effect, ")", sep=''),
    category == "Gemüse" ~ paste0(
      "Vegetables\n\n(", effect, ")", sep=''),
    category == "Heimwerker" ~ paste0(
      "Carpenter.s\ntools\n(", effect, ")", sep=''),
    category == "Huftiere" ~ paste0(
      "Hoofed\nanimals\n(", effect, ")", sep=''),
    category == "Insekten" ~ paste0(
      "Insects\n\n(", effect, ")", sep=''),
    category == "Instrumente" ~ paste0(
      "Instru-\nments\n(", effect, ")", sep=''),
    category == "Jacken" ~ paste0(
      "Jackets\n\n(", effect, ")", sep=''),
    category == "Kochen" ~ paste0(
      "Cooking\nequipment\n(", effect, ")", sep=''),
    category == "Körperteile" ~ paste0(
      "Body parts\n\n(", effect, ")", sep=''),
    category == "Küche" ~ paste0(
      "Kitchen\nfurniture\n(", effect, ")", sep=''),
    category == "Obst" ~ paste0(
      "Fruits\n\n(", effect, ")", sep=''),
    category == "Raubtiere" ~ paste0(
      "Predators\n\n(", effect, ")", sep=''),
    category == "Schmuck" ~ paste0(
      "Jewelry\n\n(", effect, ")", sep=''),
    category == "Sitzen" ~ paste0(
      "Seating\nfurniture\n(", effect, ")", sep=''),
    category == "Strasse" ~ paste0(
      "Street\nvehicles\n(", effect, ")", sep=''),
    category == "Süssigkeiten" ~ paste0(
      "Sweets\n\n(", effect, ")", sep=''),
    category == "Trinkgefässe" ~ paste0(
      "Drinking\nvessels\n(", effect, ")", sep=''),
    category == "Vögel" ~ paste0(
      "Birds\n\n(", effect, ")", sep=''),
    category == "Wasser" ~ paste0(
      "Water\nvehicles\n(", effect, ")", sep='')))  %>%
  mutate(category_en = case_when(category_en=="Insects\n\n(35.4)" ~
                                    "Insects\n\n(35.40)",
                                  category_en=="Jackets\n\n(27.9)"  ~
                                    "Jackets\n\n(27.90)" ,
                                  TRUE~category_en)) %>%
  mutate(category_en=factor(category_en,levels=c(
     "Insects\n\n(35.40)", "Seating\nfurniture\n(33.45)",
     "Cooking\nequipment\n(28.45)", "Jackets\n\n(27.90)",
     "Fruits\n\n(27.77)", "Drinking\nvessels\n(26.47)",
     "Water\nvehicles\n(26.44)","Carpenter.s\ntools\n(26.18)",
     "Kitchen\nfurniture\n(25.76)", "Fish\n\n(23.97)",
     "Storage\n\n(23.56)","Office\ntools\n(23.13)",
     "Farming\ntools\n(22.55)","Predators\n\n(21.51)",
     "Hoofed\nanimals\n(21.18)","Vegetables\n\n(20.73)" ,
     "Body parts\n\n(19.93)","Birds\n\n(19.33)",
     "Instru-\nments\n(18.27)","Flowers\n\n(15.53)",
     "Buildings\n\n(13.56)","Jewelry\n\n(12.87)",
     "Street\nvehicles\n(12.76)","Sweets\n\n(11.11)")))

# Plotting
(plot_rt_category <- means_final_category %>%
    ggplot(., aes(x=PosOr,y=normalizedRT, color=session,
                  group=session, na.rm=T)) +
    geom_point(size =1) +
    geom_line(aes(x=PosOr,y=normalizedRT, color=session, linetype="c"),
              size = 0.5) +
    geom_line(aes(x=PosOr,y=grandmean, color="b", linetype="d"),
              group = 1,size = 0.8)+
    geom_errorbar(aes(ymin=normalizedRT-se, ymax=normalizedRT+se),
                  width =.1) +
    scale_color_manual(name="Session",
                       values=c("#0072B2", "#E69F00", "#000000",
                                         "dark gray"),
                       labels=c("1", "2", "3", "Grand Mean")) +
    scale_linetype_manual(name="",values=c("c"="solid","d"="dashed"),
                          labels=c("Category mean (across groups)",
                                   "Grand Mean"))+
    apatheme+
    labs(x="Ordinal Position",y ="Normalized RTs (ms)") +
    facet_wrap(means_final_category$category_en, scales='free', ncol=8)+
    scale_y_continuous(limits = c(-800, 800),
                       breaks = c(-600,-400,-200,0,200,400,600)) +
    scale_x_discrete(breaks=c(1,2,3,4,5))+
    theme(legend.position = "bottom"))
```



```
filename <- "CSI_online_aphasia_effect_by_category.pdf"
ggsave(plot_rt_category, filename =
         here::here("results", "figures", filename),
       width = 26, height = 20,units = "cm",
       dpi = 300, device = cairo_pdf)
#embedFonts(file = here::here("results", "figures", filename))
```

Combine both

```
plots <- cowplot::plot_grid(plot_rt_subject,plot_rt_category,
          nrow = 2, ncol=1,  #rel_widths = c(1,1),
          rel_heights = c(1,0.6),
          margin(1,1,1,1),
          labels = c("A", "B"),label_size = 34,
                    label_fontfamily = "Helvetica", label_y = 1.01,
          label_x=0)
```

```
## Warning: Removed 1 rows containing missing values (‘geom_point()‘).
```

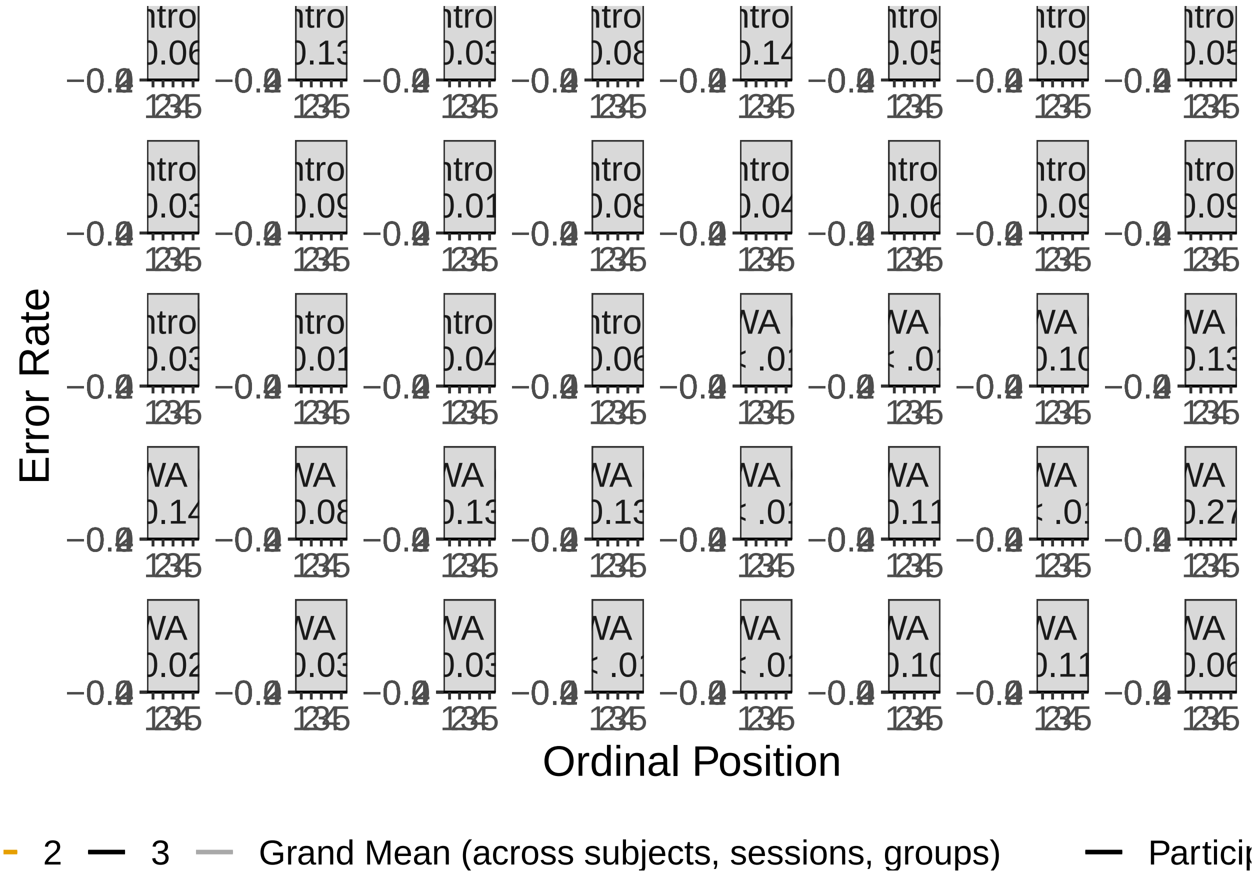
```
## Warning in as_grob.default(plot): Cannot convert object of class
## marginsimpleUnitunitunit_v2 into a grob.
```

```
filename <- "CSI_online_aphasia_spoken_RTs-by-category-and-subject.pdf"
ggsave(plots, filename =
         here::here("results", "figures", filename),
       width = 30, height = 50, units = "cm",
       dpi = 300, device = cairo_pdf)
```

Errors by subject Line graph for each participant:

```
m2_error <- readRDS(here::here(
  "results", "tables",
  "CSI_online_aphasia_SessionxGroup_glmm_errors.RDS"))
modeloutput <- coef(m2_error)$subject
means_final_subject <- df_errors %>%
   summarySEwithin(
     .,"error_class",withinvars = c("subject","PosOr", "session"),
                   betweenvars="group")
means_final<- df_errors %>%
   Rmisc::summarySEwithin(.,"error_class",idvar = "subject",
                          withinvars = c("PosOr"),
                          na.rm = T)

for(i in 1:nrow(means_final_subject)) {
  means_final_subject$grandmean[i] <-
    means_final$error_class[means_final$PosOr ==
                                means_final_subject$PosOr[i]] -
    means_final$error_class[means_final$PosOr== 1]
  means_final_subject$normalizedRT[i] <-
    means_final_subject$error_class[i] -
    means_final_subject$error_class[means_final_subject$subject ==
                                        means_final_subject$subject[i] &
                                        means_final_subject$PosOr == 1 &
                                      means_final_subject$session == 1]
  # prepare for ordering
  means_final_subject$effect[i] <-
    modeloutput$PosOr.cont[means_final_subject$subject[i]] #+
   # modeloutput$re1.PosOr.cont[means_final_subject$subject[i]]
}

means_final_subject <- means_final_subject[order(desc(means_final_subject$group), desc(means_final_subj
means_final_subject$effect <-
  as.factor(round(means_final_subject$effect, 2))
means_final_subject$effect <-
  factor(means_final_subject$effect,
         levels=rev(levels(means_final_subject$effect )))


# add participant number
means_final_subject <- means_final_subject %>%
  mutate(effect=round(as.numeric(as.character(effect)),2)) %>%
  mutate(subject_en = case_when(
    group == "PWA" & as.numeric(as.character(effect)) == 0.10~
      paste0("PWA ",
                              substr(as.character(means_final_subject$subject), 2,3),
                              "\n(",effect,"0)",sep=''),
    group == "control" & as.numeric(as.character(effect)) == 0.10~
```

```
      paste0("Control ",
             substr(as.character(means_final_subject$subject), 2,3),
                            "\n(",effect,"0)",sep=''),
     group == "PWA" & as.numeric(as.character(effect)) >= 0.01~
      paste0("PWA ",
             substr(as.character(means_final_subject$subject), 2,3),
                            "\n(",effect,")",sep=''),
    group == "control"& as.numeric(as.character(effect)) >= 0.01 ~
      paste0("Control ",
                            substr(as.character(means_final_subject$subject), 2,3),
                            "\n(",effect,")",sep=''),
    group == "PWA" & as.numeric(as.character(effect)) < 0.01~
      paste0("PWA ",
             substr(as.character(means_final_subject$subject), 2,3),
                            "\n(< .01)",sep=''),
    group == "control"& as.numeric(as.character(effect)) < 0.01 ~
      paste0("Control ",
             substr(as.character(means_final_subject$subject), 2,3),
                            "\n(<.01)",sep='')))  %>%
  mutate(subject_en=factor(subject_en))
  # mutate(subject_en=factor(subject_en,levels=c(
  #     "PWA 12\n(0.24)","PWA 07\n(0.16)","PWA 05\n(0.16)","PWA 08\n(0.16)",
  #     "PWA 10\n(0.13)","PWA 18\n(0.12)","PWA 04\n(0.12)","PWA 19\n(0.12)",
  #     "PWA 03\n(0.10)","PWA 06\n(0.07)","PWA 15\n(0.06)","PWA 20\n(0.05)",
  #     "PWA 16\n(0.05)","PWA 14\n(0.03)","PWA 13\n(0.03)","PWA 02\n(< .01)",
  #     "PWA 17\n(< .01)","PWA 09\n(< .01)","PWA 01\n(< .01)","PWA 11\n(< .01)",
  #     "Control 02\n(0.13)","Control 05\n(0.13)","Control 07\n(0.11)",
  #     "Control 15\n(0.10)","Control 10\n(0.10)","Control 16\n(0.09)",
  #     "Control 01\n(0.09)","Control 04\n(0.08)","Control 20\n(0.08)",
  #     "Control 09\n(0.07)","Control 19\n(0.07)","Control 12\n(0.07)",
  #     "Control 06\n(0.07)","Control 14\n(0.06)","Control 08\n(0.05)",
  #     "Control 13\n(0.04)","Control 11\n(0.03)","Control 17\n(0.03)",
  #     "Control 03\n(0.02)","Control 18\n(0.01)")))

# Plotting
(plot_error_subject <- means_final_subject %>%
    mutate(session=case_when(session=="1" ~ "day 1",
                             session=="2" ~ "day 2",
                             session=="3" ~ "day 8")) %>%
    ggplot(., aes(x=PosOr,y=normalizedRT, color=session,
                  group=session, na.rm=T)) +
    geom_point(size =1, color = 'black') +
    geom_line(aes(x=PosOr,y=normalizedRT, color=session, linetype="c"),
              size = 0.5) +
    geom_line(aes(x=PosOr,y=grandmean, color="b", linetype="d"),
              group = 1,size = 0.8)+
    geom_errorbar(aes(ymin=normalizedRT-se, ymax=normalizedRT+se), width =.1) +
    scale_color_manual(name="Session",
                       values=c("#0072B2", "#E69F00", "#000000",
                                         "dark gray"),
                       labels=c("1", "2", "3",
                      "Grand Mean (across subjects, sessions, groups)")) +
    scale_linetype_manual(name="",values=c("c"="solid","d"="dashed"),
                          labels=c("Participant mean",
                                   "Grand Mean"))+
    apatheme+
    labs(x="Ordinal Position",y ="Error Rate") +
    facet_wrap(means_final_subject$subject_en, scales='free', ncol=8)+
    scale_y_continuous(limits = c(-0.5, 0.5),
                       breaks = c(-0.4,-0.2,0,0.2,0.4)) +
    scale_x_discrete(breaks=c(1,2,3,4,5))+
    theme(legend.position = "bottom"))
```



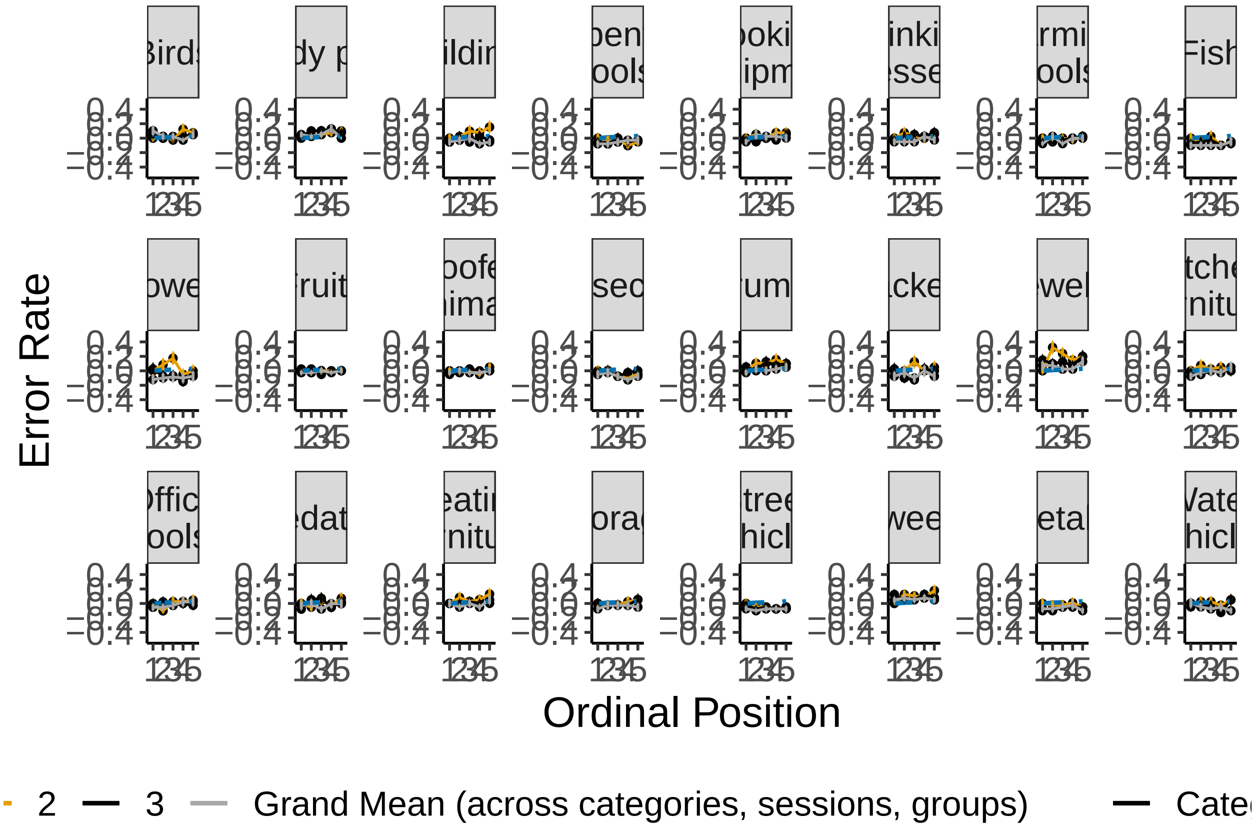
```
filename <- "CSI_online_aphasia_errors_by_participant.pdf"
ggsave(plot_error_subject, filename =
         here::here("results", "figures", filename),
       width = 34, height = 26, units = "cm",
       dpi = 300, device = cairo_pdf)
#embedFonts(file = here::here("results", "figures", filename))

## save for exploratory analyses:
means_final_subject_errors <- means_final_subject
```

Errors by category Line graph for each participant:

```
m2_error <- readRDS(here::here(
  "results", "tables",
  "CSI_online_aphasia_SessionxGroup_glmm_errors.RDS"))
modeloutput <- coef(m2_error)$category
means_final_category <- df_errors %>%
   summarySEwithin(.,"error_class",
                   withinvars = c("category","PosOr", "session"))
```

```
## Automatically converting the following non-factors to factors: category
```

```
means_final<- df_errors %>%
   Rmisc::summarySEwithin(.,"error_class",idvar = "category",
                          withinvars = c("PosOr"),#, "session"),
                          #betweenvars="group",
                          na.rm = T)

for(i in 1:nrow(means_final_category)) {
  means_final_category$grandmean[i] <-
    means_final$error_class[means_final$PosOr ==
                              means_final_category$PosOr[i]] -
    means_final$error_class[means_final$PosOr== 1]
  means_final_category$normalized_error[i] <-
    means_final_category$error_class[i] -
    means_final_category$error_class[
      means_final_category$category == means_final_category$category[i] &
                                       means_final_category$PosOr == 1 &
                                    means_final_category$session == 1]
  # prepare for ordering
  means_final_category$effect[i] <-
    modeloutput$PosOr.cont[means_final_category$category[i]]# +
    # modeloutput$re1.PosOr.cont+
    # modeloutput$re2.group2.1[means_final_category$category[i]]
}

means_final_category <-
  means_final_category[order(desc(means_final_category$effect)),]
means_final_category$effect <-
  as.factor(round(means_final_category$effect, 2))
means_final_category$effect <-
  factor(means_final_category$effect,
         levels=rev(levels(means_final_category$effect )))

means_final_category$category <- factor(
  means_final_category$category, levels=c(
       "Insekten",     "Sitzen",       "Kochen",       "Jacken",
       "Obst",         "Trinkgefässe", "Wasser",       "Heimwerker",
       "Küche",        "Fische",       "Aufbewahrung",
       "Büro", "Bauernhof", "Raubtiere", "Huftiere", "Gemüse",
       "Körperteile", "Vögel", "Instrumente", "Blumen",
       "Gebäude", "Schmuck", "Strasse", "Süssigkeiten"))

# give categories English names and combine with effect size
means_final_category <- means_final_category %>%
  mutate(category_en =
         case_when(
  category == "Aufbewahrung" ~ "Storage",
  category == "Bauernhof" ~"Farming\ntools",
  category == "Blumen" ~ "Flowers",
  category == "Büro" ~"Office\ntools",
  category == "Fische" ~ "Fish",
  category == "Gebäude" ~ "Buildings",
  category == "Gemüse" ~"Vegetables",
  category == "Heimwerker" ~ "Carpenter.s\ntools",
  category == "Huftiere" ~ "Hoofed\nanimals",
  category == "Insekten" ~ "Insects",
  category == "Instrumente" ~  "Instruments",
  category == "Jacken" ~ "Jackets",
  category == "Kochen" ~ "Cooking\nequipment",
  category == "Körperteile" ~ "Body part",
  category == "Küche" ~ "Kitchen\nfurniture",
  category == "Obst" ~ "Fruits",
  category == "Raubtiere" ~"Predators",
  category == "Schmuck" ~ "Jewelry",
  category == "Sitzen" ~"Seating\nfurniture",
  category == "Strasse" ~"Street\nvehicles",
  category == "Süssigkeiten" ~ "Sweets",
  category == "Trinkgefässe" ~  "Drinking\nvessels",
  category == "Vögel" ~  "Birds",
  category == "Wasser" ~ "Water\nvehicles")) %>%
  # mutate(category_en = case_when(
  #   as.numeric(as.character(effect)) == 0.10~ paste0(category_en, " ",
  #                           "\n(",effect,"0)",sep=''),
  #    as.numeric(as.character(effect)) >= 0.01~ paste0(category_en, " ",
  #                           "\n(",effect,")",sep=''),
  #   as.numeric(as.character(effect)) < 0.01~ paste0(category_en, " ",
  #                           "\n(< .01)",sep=''),
  #   TRUE ~ paste0(category_en, " ",
  #                           "\n(",effect,")",sep='')))  %>%
  mutate(category_en=factor(category_en))

# Plotting
(plot_error_category <- means_final_category %>%
        mutate(session=case_when(session=="1" ~ "day 1",
                             session=="2" ~ "day 2",
                             session=="3" ~ "day 8")) %>%
    ggplot(., aes(x=PosOr,y=normalized_error, color=session, group=session, na.rm=T)) +
    geom_point(size =1, color = 'black') +
    geom_line(aes(x=PosOr,y=normalized_error, color=session,
                  linetype="c"),
              size = 0.5) +
    geom_line(aes(x=PosOr,y=grandmean, color="b", linetype="d"),
              group = 1,size = 0.8)+
    geom_errorbar(aes(ymin=normalized_error-se, ymax=normalized_error+se),
                  width =.1) +
    scale_color_manual(name="Session",
                       values=c("#0072B2", "#E69F00", "#000000",
                                         "dark gray"),
                       labels=c("1", "2", "3",
                    "Grand Mean (across categories, sessions, groups)")) +
    scale_linetype_manual(name="",values=c("c"="solid","d"="dashed"),
                          labels=c("Category mean",
                                   "Grand Mean"))+
    apatheme+
    labs(x="Ordinal Position",y ="Error Rate") +
     facet_wrap(means_final_category$category_en, scales='free', ncol=8)+
    scale_y_continuous(limits = c(-0.5, 0.5),
                       breaks = c(-0.4,-0.2,0,0.2,0.4)) +
    scale_x_discrete(breaks=c(1,2,3,4,5))+
    theme(legend.position = "bottom"))
```



```
filename <- "CSI_online_aphasia_errors_by_category.pdf"
ggsave(plot_error_category, filename =
         here::here("results", "figures", filename),
       width = 26, height = 20, units = "cm",
       dpi = 300, device = cairo_pdf)
#embedFonts(file = here::here("results", "figures", filename))
```

Combine both

```
plots <- cowplot::plot_grid(plot_error_subject,plot_error_category,
          nrow = 2, ncol=1,
          rel_heights = c(1,0.6),
          margin(1,1,1,1),
          labels = c("A", "B"),label_size = 34,
                    label_fontfamily = "Helvetica", label_y = 1.01,
          label_x=0)
```

```
## Warning in as_grob.default(plot): Cannot convert object of class
## marginsimpleUnitunitunit_v2 into a grob.
```

```
filename <- "CSI_online_aphasia_spoken_errors-by-category-and-subject.pdf"
ggsave(plots, filename =
         here::here("results", "figures", filename),
       width = 30, height = 50, units = "cm",
       dpi = 300, device = cairo_pdf)
```

Exploratory nested model with group and ordinal position nested into session

Take the same random structure as in the main model

```
# m2_lmm_n <- lmer(lVOT ~ group/session/PosOr.cont +
#                 (PosOr.cont+session|subject) +
#                (PosOr.cont+group|category),
#               data = df_RTs,
#              control=lmerControl(optimizer = "bobyqa",
#                                  optCtrl = list(maxfun = 2e5)))
# summary(m2_lmm_n)

m2_lmm_n <- lmer(lVOT ~ session/(group*PosOr.cont) +
               (PosOr.cont+session|subject) +
              (PosOr.cont+group|category),
             data = df_RTs,
            control=lmerControl(optimizer = "bobyqa",
                                optCtrl = list(maxfun = 2e5)))
summary(m2_lmm_n)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: lVOT ~ session/(group * PosOr.cont) + (PosOr.cont + session |
##     subject) + (PosOr.cont + group | category)
##    Data: df_RTs
## Control: lmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 200000))
##
## REML criterion at convergence: 1158.2
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.8980 -0.6647 -0.1747  0.4804  5.6443
##
## Random effects:
##  Groups   Name        Variance   Std.Dev. Corr
##  subject  (Intercept) 0.03572288 0.189005
##           PosOr.cont  0.00010406 0.010201  0.18
##           session2    0.00339912 0.058302 -0.13  0.24
##           session3    0.00406332 0.063744 -0.49  0.26  0.64
##  category (Intercept) 0.00884984 0.094074
##           PosOr.cont  0.00002358 0.004856 0.07
##           group2-1    0.00090450 0.030075 0.08  0.16
##  Residual             0.06154884 0.248090
## Number of obs: 12455, groups:  subject, 40; category, 24
##
## Fixed effects:
##                                 Estimate Std. Error         df t value
## (Intercept)                     7.116562   0.035604  58.495842 199.884
## session2                       -0.076082   0.010856  38.393028  -7.009
## session3                       -0.089211   0.011561  36.616625  -7.717
## session1:group2-1               0.172863   0.065639  38.187285   2.634
## session2:group2-1               0.147257   0.062950  38.261949   2.339
## session3:group2-1               0.152784   0.055622  38.529375   2.747
## session1:PosOr.cont             0.014161   0.003397 104.345986   4.168
## session2:PosOr.cont             0.018796   0.003349  98.553423   5.612
## session3:PosOr.cont             0.017810   0.003310  94.964792   5.380
## session1:group2-1:PosOr.cont    0.001525   0.006499 151.024051   0.235
## session2:group2-1:PosOr.cont    0.014830   0.006398 141.844633   2.318
## session3:group2-1:PosOr.cont    0.004620   0.006317 136.892022   0.731
##                                           Pr(>|t|)
## (Intercept)                  < 0.0000000000000002 ***
## session2                            0.00000002267 ***
## session3                            0.00000000345 ***
## session1:group2-1                          0.0121 *
## session2:group2-1                          0.0246 *
## session3:group2-1                          0.0091 **
## session1:PosOr.cont                 0.00006349491 ***
## session2:PosOr.cont                 0.00000018378 ***
## session3:PosOr.cont                 0.00000053431 ***
## session1:group2-1:PosOr.cont               0.8148
## session2:group2-1:PosOr.cont               0.0219 *
## session3:group2-1:PosOr.cont               0.4658
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Correlation of Fixed Effects:
##             (Intr) sessn2 sessn3 ss1:2-1 ss2:2-1 ss3:2-1 s1:PO. s2:PO. s3:PO.
## session2    -0.096
## session3    -0.361  0.612
## sssn1:gr2-1  0.006 -0.009 -0.009
## sssn2:gr2-1  0.006  0.008  0.000  0.944
## sssn3:gr2-1  0.006  0.000  0.006  0.941   0.952
## sssn1:PsOr.  0.084  0.095  0.108  0.006   0.005   0.006
## sssn2:PsOr.  0.085  0.102  0.113  0.004   0.006   0.006   0.323
## sssn3:PsOr.  0.085  0.101  0.114  0.004   0.005   0.007   0.326  0.331
## ss1:2-1:PO.  0.001 -0.004 -0.003  0.057   0.094   0.115   0.112  0.009  0.007
## ss2:2-1:PO.  0.001  0.005  0.002  0.057   0.096   0.117   0.009  0.088  0.008
## ss3:2-1:PO.  0.001  0.002  0.005  0.058   0.097   0.118   0.007  0.008  0.079
##             s1:2-1: s2:2-1:
## session2
## session3
## sssn1:gr2-1
## sssn2:gr2-1
## sssn3:gr2-1
## sssn1:PsOr.
## sssn2:PsOr.
## sssn3:PsOr.
## ss1:2-1:PO.
## ss2:2-1:PO.  0.259
## ss3:2-1:PO.  0.261   0.266
```

Exploratory analyses

We conducted several exploratory analyses to assess the stability of our effects, especially because we deviated from the pre-planned GLMMs for RT analyses.

Here, we report additional analyses that were suggested by reviewers

Inspecting participants with long or short time since onset of lesion

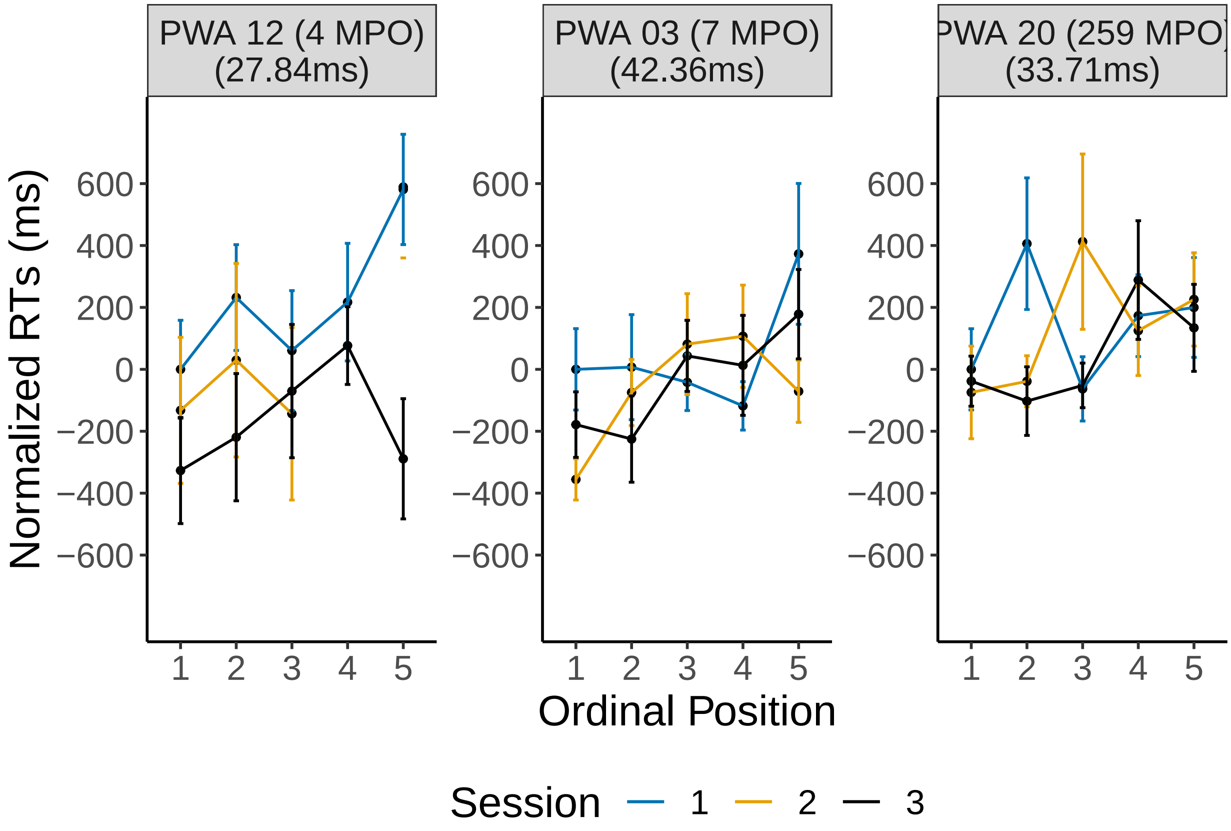
Subject 112 was tested 4 months after onset, subject 103 was tested 7 months after onset, and subject 120 wast tested 259 months after onset

```
means_final_subject_RTs_r <-
  means_final_subject_RTs %>% filter(subject %in%
                                       c(112, 103, 120)) %>%
  droplevels() %>%
  mutate(subject_en_mpo=case_when(
     subject_en=="PWA 03\n(42.36)"~"PWA 03 (7 MPO)\n(42.36ms)",
     subject_en=="PWA 20\n(33.71)"~"PWA 20 (259 MPO)\n(33.71ms)",
     subject_en=="PWA 12\n(27.84)"~"PWA 12 (4 MPO)\n(27.84ms)",
  )) %>%
  mutate(subject_en_mpo=factor(subject_en_mpo, levels =
                   c("PWA 12 (4 MPO)\n(27.84ms)",
                     "PWA 03 (7 MPO)\n(42.36ms)",
                     "PWA 20 (259 MPO)\n(33.71ms)")))

means_final_subject_errors_r <-
  means_final_subject_errors %>%
  filter(subject %in% c(112, 103, 120)) %>% droplevels() %>%
  mutate(subject_en_mpo=case_when(
     subject_en=="PWA 03\n(0.10)"~"PWA 03 (7 MPO)\n(0.10)",
     subject_en=="PWA 20\n(0.06)"~"PWA 20 (259 MPO)\n(0.06)",
     subject_en=="PWA 12\n(0.27)"~"PWA 12 (4 MPO)\n(0.27)",
  )) %>%
  mutate(subject_en_mpo=factor(subject_en_mpo, levels =
                   c("PWA 12 (4 MPO)\n(0.27)",
                     "PWA 03 (7 MPO)\n(0.10)",
                     "PWA 20 (259 MPO)\n(0.06)")))
```

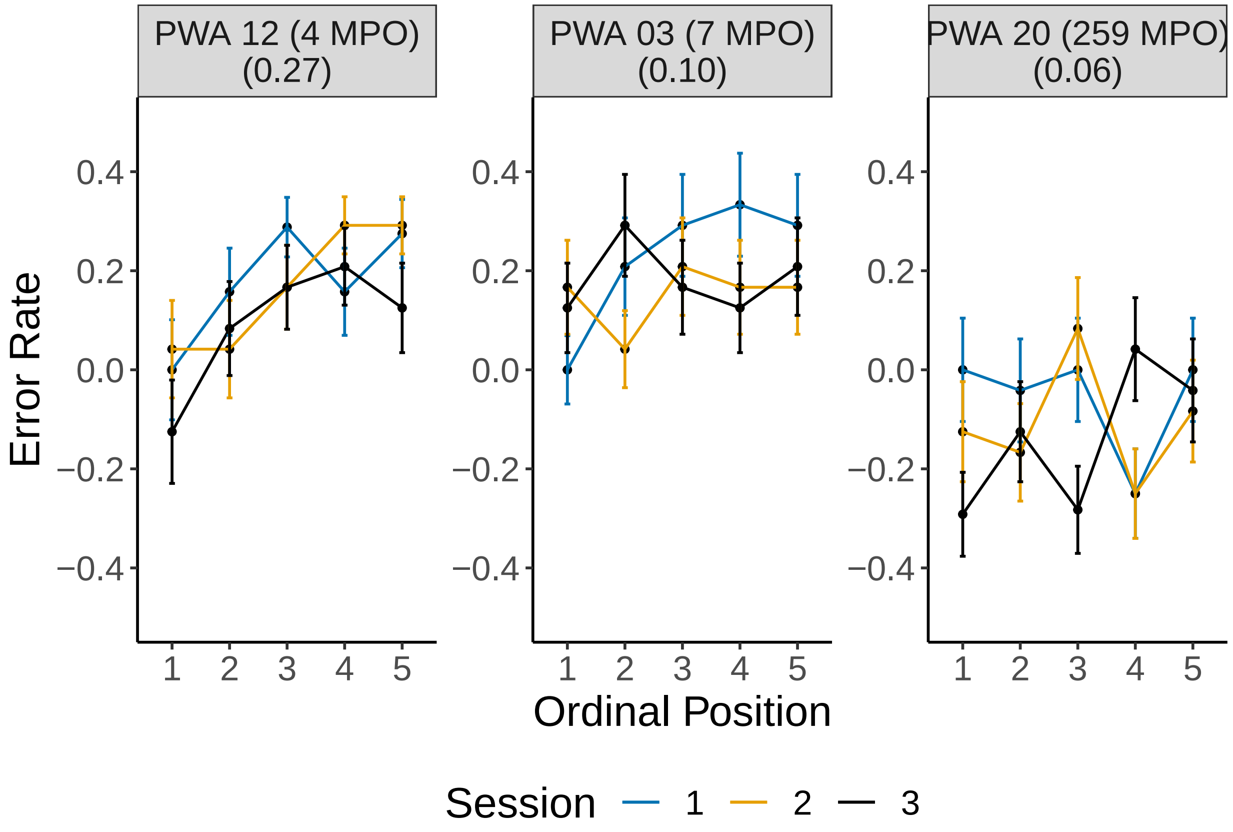
```
### change grand mean to mean group effect
modeloutput <- coef(m2)$subject

## extract plots
(plot_rt_subject_r <- means_final_subject_RTs_r %>%
    ggplot(., aes(x=PosOr,y=normalizedRT,
                  color=session, group=session, na.rm=T)) +
    geom_point(size =1, color = 'black') +
    geom_line(aes(x=PosOr,y=normalizedRT, color=session,
                  linetype="c"),
              size = 0.5) +
    # geom_line(aes(x=PosOr,y=grandmean, color="b", linetype="d"),
    #           group = 1,size = 0.8)+
    geom_errorbar(aes(ymin=normalizedRT-se, ymax=normalizedRT+se),
                  width =.1) +
    guides(linetype="none") +
    scale_color_manual(name="Session",
                       values=c(
                         "#0072B2", "#E69F00", "#000000",
                         "dark gray"),
                       labels=c(
                         "1", "2", "3")) + #,
                   #   "Grand Mean (across subjects, sessions, groups)")) +
    # scale_linetype_manual(name="",values=c("c"="solid","d"="dashed"),
    #                        labels=c("Participant mean",
    #                                 "Grand Mean"))+
    apatheme+
    labs(x="Ordinal Position",y ="Normalized RTs (ms)") +
    facet_wrap(means_final_subject_RTs_r$subject_en_mpo,
               scales='free', ncol=3)+
    scale_y_continuous(limits = c(-800, 800),
                       breaks = c(-600,-400,-200,0,200,400,600)) +
    scale_x_discrete(breaks=c(1,2,3,4,5))+
    theme(legend.position = "bottom"))
```

Extract by participant plots for these participants and plot

```
## Warning: Removed 1 rows containing missing values (`geom_point()`).
```



```
(plot_error_subject_r <- means_final_subject_errors_r %>%
    mutate(session=case_when(session=="1" ~ "day 1",
                             session=="2" ~ "day 2",
                             session=="3" ~ "day 8")) %>%
    ggplot(., aes(x=PosOr,y=normalizedRT, color=session,
                  group=session, na.rm=T)) +
    geom_point(size =1, color = 'black') +
    geom_line(aes(x=PosOr,y=normalizedRT, color=session, linetype="c"),
              size = 0.5) +
    geom_errorbar(aes(ymin=normalizedRT-se, ymax=normalizedRT+se), width =.1) +
    # geom_line(aes(x=PosOr,y=grandmean, color="b", linetype="d"),
    #           group = 1,size = 0.8)+
    scale_color_manual(name="Session",
                       breaks=c("day 1", "day 2", "day 8"), #, "b"),
                       values=c("#0072B2", "#E69F00", "#000000"),#,
                                                #  "dark gray"),
                        labels=c("1", "2", "3")) + #,
                    #   "Grand Mean (across subjects, sessions, groups)")) +
    #scale_linetype_manual(name="",
                          #values=c("c"="solid","d"="dashed"),
                          #labels=c("Participant mean",
                                 #  "Grand Mean"))+
    apatheme+
    labs(x="Ordinal Position",y ="Error Rate") +
    facet_wrap(means_final_subject_errors_r$subject_en_mpo, scales='free', ncol=3)+
    scale_y_continuous(limits = c(-0.5, 0.5),
                        breaks = c(-0.4,-0.2,0,0.2,0.4)) +
    scale_x_discrete(breaks=c(1,2,3,4,5))+
    guides(linetype="none") +
    theme(legend.position = "bottom"))
```



```
## combine both
plots <- cowplot::plot_grid(plot_rt_subject_r, plot_error_subject_r,
          nrow = 2, ncol=1,
          rel_heights = c(1,1),
          margin(1,1,1,1),
          labels = c("A", "B"),label_size = 34,
                    label_fontfamily = "Helvetica", label_y = 1.01,
          label_x=0)
```

```
## Warning: Removed 1 rows containing missing values (‘geom_point()‘).

## Warning in as_grob.default(plot): Cannot convert object of class
## marginsimpleUnitunitunit_v2 into a grob.
```

```
filename <- "CSI_online_aphasia_spoken_errors-by-category-and-subject_MPO.pdf"
ggsave(plots, filename =
         here::here("results", "figures", filename),
       width = 25, height = 20, units = "cm",
       dpi = 300, device = cairo_pdf)
```

```
mean(as.numeric(as.character(
  means_final_subject_RTs$effect[
    means_final_subject_RTs$group=="PWA"])), na.rm=T)
```

Get descriptives based on model data

```
## [1] 21.5855
```

```
sd(as.numeric(as.character(
  means_final_subject_RTs$effect[
    means_final_subject_RTs$group=="PWA"])), na.rm=T)
```

```
## [1] 11.47912
```

```
mean(as.numeric(as.character(
  means_final_subject_errors$effect[
    means_final_subject_RTs$group=="PWA"
  ])), na.rm=T)
```

```
## [1] 0.06
```

```
sd(as.numeric(as.character(
  means_final_subject_errors$effect[means_final_subject_RTs$group=="PWA"])), na.rm=T)
```

```
## [1] 0.08493555
```

Absence of CSI effect for fillers?

```
# load preprocessed data
input = "aphasia_final.csv"
d <- read.csv2(here::here("data","transient_data_files", input), sep=",") #%>% select(-"X")

# select only fillers
d <- d %>% filter(category=="Filler")

## Add filler subcategories
# "The 24 categories were distributed across eight blocks of
# three categories each. To each block, five filler items
# were added, resulting in eight blocks of 20 items."
d <- d %>%
  mutate(filler_block = case_when(
    trial <= 20 ~ "block1",
    trial > 20 & trial <= 40  ~ "block2",
    trial > 40 & trial <= 60  ~ "block3",
    trial > 60 & trial <= 80  ~ "block4",
    trial > 80 & trial <= 100  ~ "block5",
    trial > 100 & trial <= 120  ~ "block6",
    trial > 120 & trial <= 140  ~ "block7",
    trial > 140 & trial <= 160  ~ "block8"))
table(d$filler_block)
```

Preparation

```
##
## block1 block2 block3 block4 block5 block6 block7 block8
##    600    600    600    600    600    600    600    600
```

```
## Add ordinal position
d <- d %>% group_by(subject, session,filler_block) %>%
      add_count() %>%
      dplyr::mutate(PosOr = seq(1:n)) %>%
  dplyr::select(-n)
```

```
## Warning: There were 960 warnings in ‘dplyr::mutate()‘.
## The first warning was:
## i In argument: ‘PosOr = seq(1:n)‘.
## i In group 1: ‘subject = 101‘, ‘session = 1‘, ‘filler_block = "block1"‘.
## Caused by warning in ‘1:n‘:
## ! numerical expression has 5 elements: only the first used
## i Run ‘dplyr::last_dplyr_warnings()‘ to see the 959 remaining warnings.
```

```
table(df$PosOr)
```

```
##
##    1    2    3    4    5
## 2880 2880 2880 2880 2880
```

```
# d %>% group_by(subject, session,filler_block) %>% count(PosOr)
# table(df$PosOr,df$session, df$subject)

# factorize columns
d$VOT <- as.numeric(as.character(d$VOT))
is.numeric(d$VOT)
```

```
## [1] TRUE
```

```
d$PosOr <- as.factor(d$PosOr)
d$group <- as.factor(d$type)
d$subject <- as.factor(d$subject)
d$session <- as.factor(d$session)

# define contrasts of session: compare 1 to 2 and 1 to 3, intercept is the grand mean => simple coding
c<-contr.treatment(3)
my.coding<-matrix(rep(1/3, 6), ncol=2)
my.simple<-c-my.coding
my.simple
```

```
##            2          3
## 1 -0.3333333 -0.3333333
## 2  0.6666667 -0.3333333
## 3 -0.3333333  0.6666667
```

```
contrasts(d$session)<-my.simple
levels(d$session)
```

```
## [1] "1" "2" "3"
```

```
## Define contrast of group
contrasts(d$group) <- MASS::contr.sdif(2)
levels(d$group)
```

```
## [1] "control" "PWA"
```

```
## Define Ordinal position as continuous predictor variable
d$PosOr_cont <- as.numeric(scale(as.numeric(
  as.character(d$PosOr)),
  center = T, scale = F))

## Errors and correct responses
# Correct responses start with 1.  We will consider all responses in 1.1-1.3 for analyses of response t
d$error[is.na(d$VOT)] <- "99"
d$correct[is.na(d$VOT)] <- "0"

## Two trials were forgotten to be classified, but AR == 99 --> technical error?
d$error[is.na(d$correct) & d$AR == "99"] <- "99"
d$correct[is.na(d$correct) & d$AR == "99"] <- "0"
sum(is.na(d$correct))
```

```
## [1] 0
```

```
## NR and 0.4 are the same -> replace this
d$error[d$error=="NR"] <- "4"

## Rename broken names
# unique(df$error)
d$error <- stringr::str_replace(d$error, ";;;;;;", "")
d$error <- stringr::str_replace(d$error, "ok", "") # subject 113, session 2, trial 121 (Couch)
df$error <- gsub("?",NA,df$error, fixed = TRUE)
# unique(df$error)

# unique(df$correct)
d$correct <- stringr::str_replace(d$correct, ";;;;;;", "")
# unique(df$correct)

d$error[d$correct == 0 & is.na(d$error)] <- 1 # phonet. paraphrasia > 25 %

## Subset data for reaction time and error analyses and delete fillers
# As correct reaction times will be considered:
# 1 - correct.
# 1.1 - correct with alternative response.
# 1.2 - correct with phonematic paraphasia (<=25% of the word).
# 1.3 - correct with correct article [*].
d %>% mutate(correct_class = case_when(
  correct == 1 ~ 1,
  correct ==1.1 ~ 1,
  correct == 1.2 ~ 1,
  correct == 1.3 ~ 1,
  correct == 1.4 ~ 0,
  correct == 0 ~ 0)) -> d

### Save data frame for RT analyses: Only correct responses
d_RTs <- d %>% filter(correct_class == 1)
print(paste0("Amount of trials that went into RT analyses of Filler trials: ",
             nrow(d_RTs)))
```

```
## [1] "Amount of trials that went into RT analyses of Filler trials: 4218"
```

```
table(d_RTs$group)
```

```
##
## control     PWA
##    2348    1870
```

```
# Subset data frame for error analyses
# As errors on the participant side will be considered:
# 1 - wrong with phonematic paraphrasia (> 25 % of the word).
# 2 - wrong: semantic paraphrasia (word in the experiment).
# 3 - wrong: semantic paraphrasia (word not in the experiment).
# 4 - wrong: null reaction.
# 5 - wrong: replacement without connection to the word (word in the experiment).
# 6 - wrong: replacement without connection to the word (word not in the experiment).
# 7 - superordinate word.
# 8 - neologism.
# 9 - etc.

d %>% mutate(error_class = case_when(
  error ==1 | error == 2 |error == 3 |
    error==4 | error==5 | error == 6 | error == 7 |
    error == 8 | error == 9 ~ 1,
  error == 99 | is.na(error) ~ 0)) -> d

#### Subset data for error analyses: All trials excluding technical errors, invalid RTs and fillers
d_errors <- d %>% filter((error != "99" | is.na(error)))

print(paste0("Amount of trials that went into error analyses: ",
             nrow(d_errors)))
```

```
## [1] "Amount of trials that went into error analyses: 4771"
```

```r
print(paste0("Amount of these trials with errors analyses: ",
             nrow(d_errors[d_errors$error_class==1,])))
```

```
## [1] "Amount of these trials with errors analyses: 491"
```

```r
table(d_errors$group)
```

```
##
## control     PWA
##    2372    2399
```

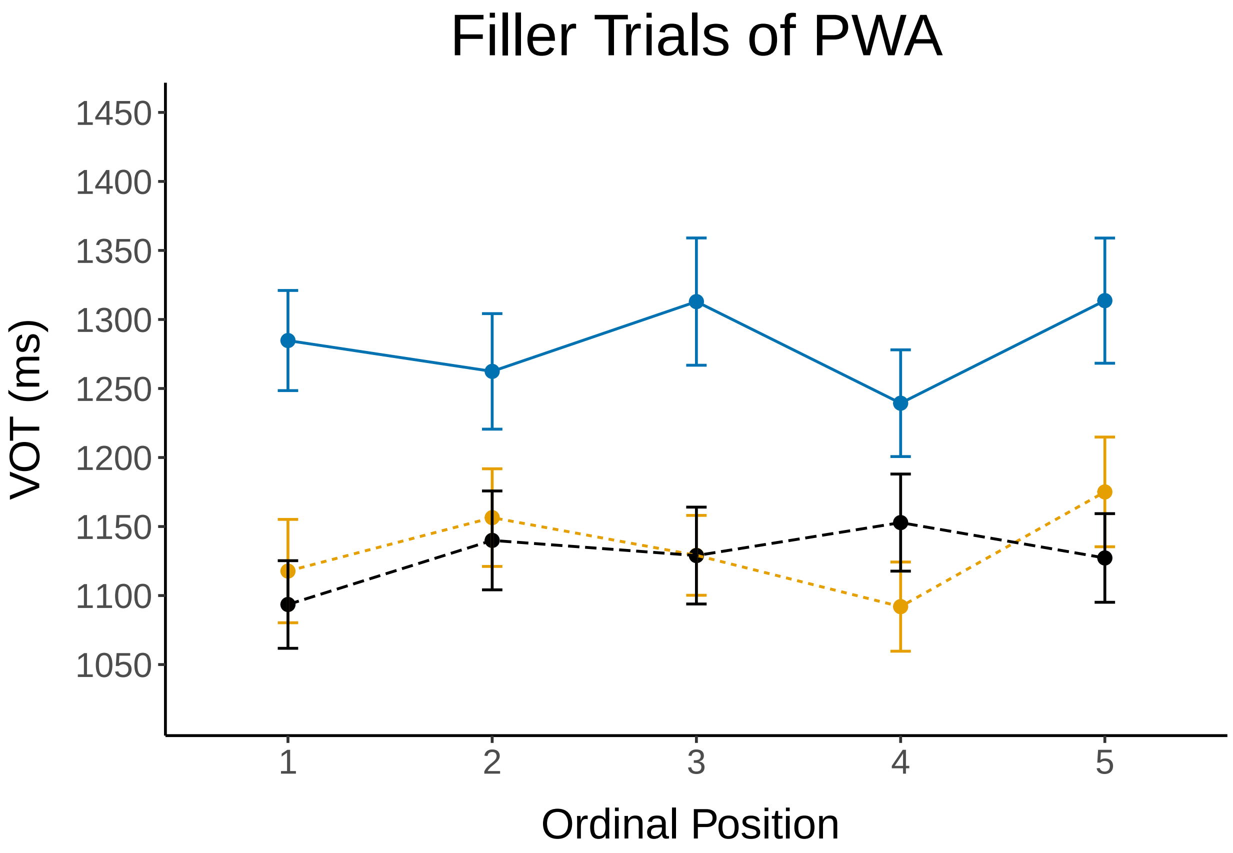
```r


(means_final<-
   d %>% filter(correct_class == 1) %>%
   filter(category == "Filler") %>%
   Rmisc::summarySEwithin(.,"VOT",idvar = "subject",
                          withinvars = c("session", "PosOr"),
                          betweenvars = "group", na.rm = T))
```

Plotting of VOTs

```
##      group session PosOr   N      VOT       sd       se       ci
## 1  control       1     1 157 1237.344 298.7901 23.84605 47.10280
## 2  control       1     2 158 1229.314 320.2176 25.47515 50.31823
## 3  control       1     3 155 1217.212 251.7419 20.22040 39.94515
## 4  control       1     4 154 1224.877 270.7734 21.81954 43.10647
## 5  control       1     5 156 1213.521 268.7514 21.51733 42.50507
## 6  control       2     1 155 1124.055 210.6816 16.92235 33.42991
## 7  control       2     2 158 1154.887 222.5629 17.70615 34.97300
## 8  control       2     3 156 1168.960 239.1948 19.15091 37.83046
## 9  control       2     4 158 1150.262 200.6234 15.96074 31.52548
## 10 control       2     5 154 1215.952 294.2336 23.71002 46.84128
## 11 control       3     1 154 1152.081 232.6809 18.74996 37.04224
## 12 control       3     2 156 1171.249 242.4956 19.41518 38.35251
## 13 control       3     3 159 1131.397 226.2960 17.94644 35.44587
## 14 control       3     4 159 1133.064 184.0503 14.59614 28.82871
## 15 control       3     5 159 1170.278 257.3952 20.41277 40.31709
## 16     PWA       1     1 126 1284.737 407.1925 36.27560 71.79391
## 17     PWA       1     2 122 1262.389 462.3727 41.86126 82.87541
## 18     PWA       1     3 115 1312.944 494.5014 46.11249 91.34850
## 19     PWA       1     4 115 1239.336 414.7010 38.67107 76.60709
## 20     PWA       1     5 118 1313.645 492.6735 45.35427 89.82176
## 21     PWA       2     1 118 1117.702 406.7237 37.44195 74.15182
## 22     PWA       2     2 132 1156.443 406.4393 35.37600 69.98216
## 23     PWA       2     3 126 1129.120 324.7708 28.93289 57.26177
## 24     PWA       2     4 128 1091.962 365.5339 32.30894 63.93355
## 25     PWA       2     5 124 1175.063 442.5136 39.73889 78.66070
## 26     PWA       3     1 135 1093.505 368.7910 31.74048 62.77713
## 27     PWA       3     2 127 1139.945 403.6176 35.81527 70.87736
## 28     PWA       3     3 128 1128.958 396.9119 35.08238 69.42171
## 29     PWA       3     4 135 1152.840 408.6211 35.16850 69.55717
## 30     PWA       3     5 121 1127.213 353.1359 32.10326 63.56222
```

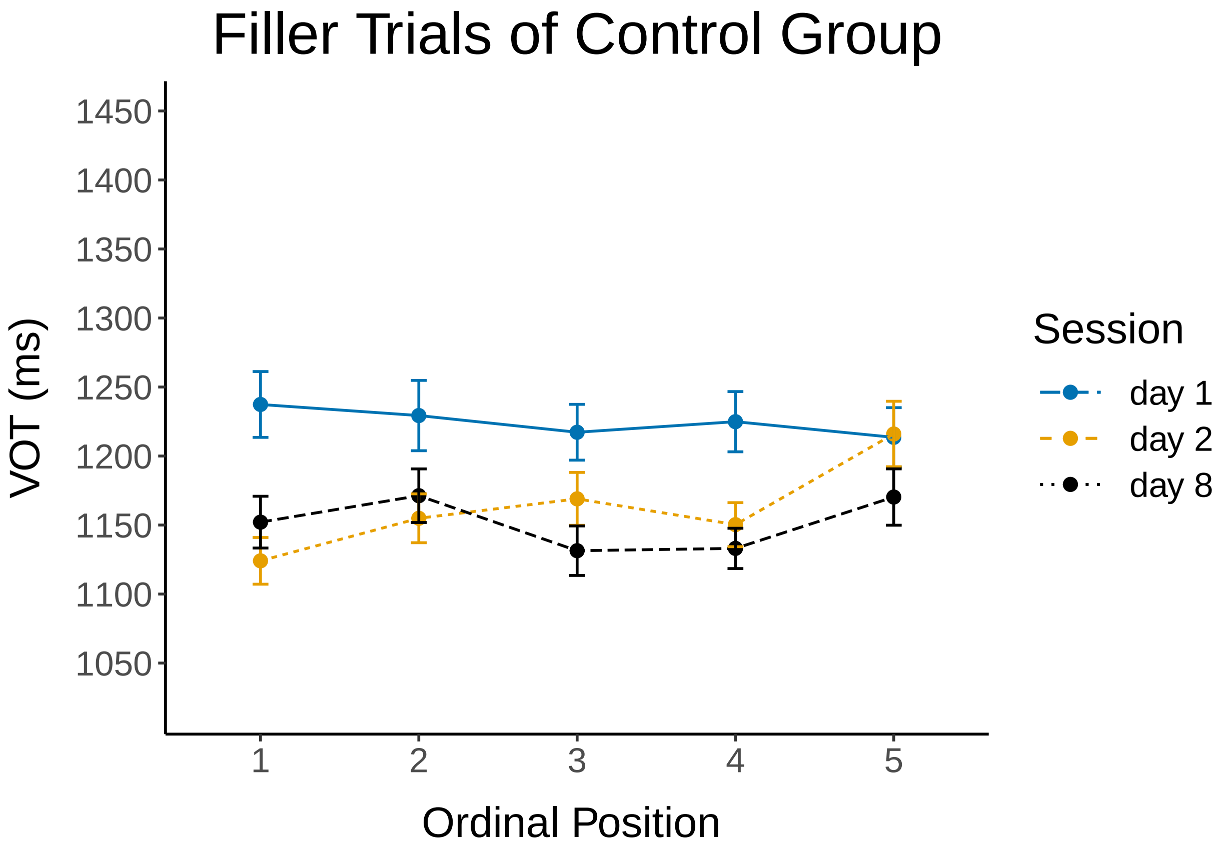
```r
override.linetype<-c("longdash", "dashed", "dotted")
(plot_rt_repetition_PWA <- means_final %>% filter(group=="PWA") %>%
    mutate(session=case_when(session == "1" ~ "day 1",
                                 session == "2" ~ "day 2",
                                 session == "3" ~ "day 8")) %>%
    ggplot(., aes(x=PosOr, y=VOT, group=session, color = session)) +
    geom_point(size = 2)+
    stat_summary(aes(linetype=session),fun=mean,
                 geom="line", size = 0.5) +
    scale_linetype_manual(values=c("longdash", "dashed", "dotted"))+
    scale_color_manual(values=c("#0072B2", "#E69F00", "#000000"))+
    geom_errorbar(aes(ymin=VOT-se, ymax=VOT+se, group = session),
                  width =.1) +
    apatheme+
    scale_y_continuous(limits = c(1020, 1450),
                       breaks =seq(1050,1450, by = 50)) +
    labs(x="Ordinal Position ",y ="VOT (ms)", colour="Session",
         linetype="Session",
         title = "Filler Trials of PWA") +
    theme(
    axis.title.y = element_text(margin = margin(0,10,0,0)),
    axis.title.x = element_text(margin = margin(10,0,0,0)),
    legend.key.width = unit(1, "cm"),
    legend.position="none")+
   guides(color=guide_legend(
     override.aes=list(linetype=override.linetype)))+
    scale_linetype(guide="none"))
```

```
## Scale for linetype is already present.
## Adding another scale for linetype, which will replace the existing scale.
```



```
(plot_rt_repetition_control <- means_final %>%
    filter(group=="control") %>%
    mutate(session=case_when(session == "1" ~ "day 1",
                             session == "2" ~ "day 2",
                             session == "3" ~ "day 8")) %>%
    ggplot(., aes(x=PosOr, y=VOT, group=session, color = session)) +
    geom_point(size = 2)+
    stat_summary(aes(linetype=session),fun=mean,
                 geom="line", size = 0.5) +
    scale_linetype_manual(values=c("longdash", "dashed", "dotted"))+
    scale_color_manual(values=c("#0072B2", "#E69F00", "#000000"))+
    geom_errorbar(aes(ymin=VOT-se, ymax=VOT+se, group = session),
                  width =.1) +
    apatheme+
    scale_y_continuous(limits = c(1020, 1450),
                       breaks =seq(1050,1450, by = 50)) +
    labs(x="Ordinal Position ",y ="VOT (ms)", colour="Session", linetype="Session",
         title = "Filler Trials of Control Group") +
    theme(
    axis.title.y = element_text(margin = margin(0,10,0,0)),
    axis.title.x = element_text(margin = margin(10,0,0,0)),
    legend.key.width = unit(1, "cm"))+
   guides(color=guide_legend(
    override.aes=list(linetype=override.linetype)))+
    scale_linetype(guide="none"))
```

```
## Scale for linetype is already present.
## Adding another scale for linetype, which will replace the existing scale.
```



```
plots <- cowplot::plot_grid(
  plot_rt_repetition_PWA,plot_rt_repetition_control,
  nrow = 1, ncol=2,  rel_widths = c(0.81,1), #rel_height = c(1,1),
  margin(1,1,1,1),
  labels = c("A", "B"),label_size = 34,
  label_fontfamily = "Helvetica", label_y = 1.01,
  label_x=-0.03)
```

```
## Warning in as_grob.default(plot): Cannot convert object of class
## marginsimpleUnitunitunit_v2 into a grob.
```

```
filename <- "CSI_online_aphasia_spoken_plot_rt_by_repetition_FILLERS.pdf"
ggsave(plots, filename =
         here::here("results", "figures", filename),
       width = 25, height = 13, units = "cm",
       dpi = 300, device = cairo_pdf)
#embedFonts(file = here::here("results", "figures", filename))
```

Inferential statistics of VOTs Preparation

```r
d_RTs$PosOr.cont <-
  scale(as.numeric(as.character(d_RTs$PosOr)),
        center = T, scale = F)
# *Compute further contrasts*
# define contrasts of session: compare 1 to 2 and 1 to 3, intercept is the grand mean => simple coding
c<-contr.treatment(3)
my.coding<-matrix(rep(1/3, 6), ncol=2)
my.simple<-c-my.coding
my.simple
```

```
##            2          3
## 1 -0.3333333 -0.3333333
## 2  0.6666667 -0.3333333
## 3 -0.3333333  0.6666667
```

```r
contrasts(d_RTs$session)<-my.simple
levels(d_RTs$session)
```

```
## [1] "1" "2" "3"
```

```r
## Define contrast of group
contrasts(d_RTs$group) <- MASS::contr.sdif(2)
levels(d_RTs$group)
```

```
## [1] "control" "PWA"
```

```r
# *didLMERconverge function*
## This function provides a better convergence check for lme4 v>1.0 models, which have a nasty habit of
didLmerConverge = function(lmerModel){
  relativeMaxGradient=signif(max(abs(with(
    lmerModel@optinfo$derivs,solve(Hessian,gradient)))),3)
  if (relativeMaxGradient < 0.001) {
    cat(sprintf("\tThe relative maximum gradient of %s is less than our 0.001 criterion.\n\tYou can safe
  }
  else {
    cat(sprintf("The relative maximum gradient of %s exceeds our 0.001 criterion.\nThis looks like a rea
  }
}

#### Transform RTs: log transformation as in main analyses
d_RTs$lVOT <- log(d_RTs$VOT)
```

Compute full model, then compute step-wise reduction

```r
library(lmerTest)
# m2_lmm <- lmer(lVOT ~ PosOr.cont*session*group +
#                 (PosOr.cont*session|subject),
#               data = d_RTs,
#              control=lmerControl(optimizer = "bobyqa"))
# didLmerConverge(m2_lmm)
# Model fails to converge --> Reduce
# 1) Increase optimizer iterations
# m2_lmm <- lmer(lVOT ~ PosOr.cont*session*group +
#                 (PosOr.cont*session|subject),
#               data = d_RTs,
#              control=lmerControl(optimizer = "bobyqa",
#                                  optCtrl = list(maxfun = 2e5)))
# didLmerConverge(m2_lmm)

# 2) Omit correlation parameters as model still fails to converge
# m2_lmm <- afex::lmer_alt(lVOT ~ PosOr.cont*session*group +
#                 (PosOr.cont*session||subject),
#               data = d_RTs,
#              control=lmerControl(optimizer = "bobyqa",
#               optCtrl = list(maxfun = 2e5)))

# 3) Model fit is still singular -> Further reduce the model
# m2_lmm <- afex::lmer_alt(lVOT ~ PosOr.cont*session*group +
#                 (PosOr.cont+session||subject),
#               data = d_RTs,
#              control=lmerControl(optimizer = "bobyqa",
#               optCtrl = list(maxfun = 2e5)))
# m2_lmm <- afex::lmer_alt(lVOT ~ PosOr.cont*session*group +
#                 (session||subject),
#               data = d_RTs,
#              control=lmerControl(optimizer = "bobyqa",
#               optCtrl = list(maxfun = 2e5)))
# 4) Does the model also converge when correlation parameters are included - yes!
m2_lmm <- lmer(lVOT ~ PosOr.cont*session*group +
               (session|subject),
             data = d_RTs,
            control=lmerControl(optimizer = "bobyqa",
                       optCtrl = list(maxfun = 2e5)))
# rePCA(m2_lmm)
didLmerConverge(m2_lmm)
```

```
##  The relative maximum gradient of 0.000000727 is less than our 0.001 criterion.
##  You can safely ignore any warnings about a claimed convergence failure.
```

```
## Warnings can be ignored

summary(m2_lmm)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: lVOT ~ PosOr.cont * session * group + (session | subject)
##    Data: d_RTs
## Control: lmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 200000))
##
## REML criterion at convergence: -391.4
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.7368 -0.6288 -0.1654  0.4322  5.1883
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  subject  (Intercept) 0.038296 0.19569
##           session2    0.003026 0.05500  -0.06
##           session3    0.005456 0.07386  -0.48  0.60
##  Residual             0.049623 0.22276
## Number of obs: 4218, groups:  subject, 40
##
## Fixed effects:
##                                 Estimate  Std. Error          df t value
## (Intercept)                     7.057860    0.031172   37.669431 226.417
## PosOr.cont                      0.003775    0.002449 4104.169425   1.542
## session2                       -0.079458    0.012316   37.786634  -6.451
## session3                       -0.084154    0.014550   37.884349  -5.784
## group2-1                        0.196398    0.062344   37.669431   3.150
## PosOr.cont:session2             0.008929    0.006029 4104.476679   1.481
## PosOr.cont:session3             0.003490    0.005986 4105.352910   0.583
## PosOr.cont:group2-1            -0.001560    0.004897 4104.169426  -0.318
## session2:group2-1              -0.055915    0.024633   37.786634  -2.270
## session3:group2-1              -0.046108    0.029101   37.884349  -1.584
## PosOr.cont:session2:group2-1   -0.015678    0.012058 4104.476680  -1.300
## PosOr.cont:session3:group2-1    0.003927    0.011973 4105.352910   0.328
##                                          Pr(>|t|)
## (Intercept)                  < 0.0000000000000002 ***
## PosOr.cont                                0.12325
## session2                              0.000000141 ***
## session3                              0.000001139 ***
## group2-1                                  0.00319 **
## PosOr.cont:session2                       0.13866
## PosOr.cont:session3                       0.55990
## PosOr.cont:group2-1                       0.75014
## session2:group2-1                         0.02900 *
## session3:group2-1                         0.12141
## PosOr.cont:session2:group2-1              0.19359
## PosOr.cont:session3:group2-1              0.74291
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Correlation of Fixed Effects:
##             (Intr) PsOr.c sessn2 sessn3 grp2-1 PsO.:2 PsO.:3 PO.:2- s2:2-1
## PosOr.cont   0.001
## session2    -0.039 -0.004
## session3    -0.389 -0.002  0.555
## group2-1     0.004  0.002  0.001 -0.002
## PsOr.cnt:s2 -0.001 -0.002  0.001  0.003  0.000
## PsOr.cnt:s3  0.000 -0.014  0.003  0.004  0.001  0.505
## PsOr.cn:2-1  0.002  0.116 -0.003  0.002  0.001 -0.005 -0.013
## sssn2:gr2-1  0.001 -0.003  0.081  0.038 -0.039  0.002  0.004 -0.004
## sssn3:gr2-1 -0.002  0.002  0.038  0.056 -0.389  0.003  0.008 -0.002  0.555
## PsOr.:2:2-1  0.000 -0.005  0.002  0.003 -0.001  0.123  0.065 -0.002  0.001
## PsOr.:3:2-1  0.001 -0.013  0.004  0.008  0.000  0.065  0.115 -0.014  0.003
##             s3:2-1 PO.:2:
```

```
## PosOr.cont
## session2
## session3
## group2-1
## PsOr.cnt:s2
## PsOr.cnt:s3
## PsOr.cn:2-1
## sssn2:gr2-1
## sssn3:gr2-1
## PsOr.:2:2-1  0.003
## PsOr.:3:2-1  0.004  0.505
```

```
anova(m2_lmm)
```

```
## Type III Analysis of Variance Table with Satterthwaite’s method
##                           Sum Sq Mean Sq NumDF  DenDF F value      Pr(>F)
## PosOr.cont               0.11793 0.11793     1 4104.2  2.3764    0.123255
## session                  2.41377 1.20688     2   38.2 24.3210 0.000000154 ***
## group                    0.49246 0.49246     1   37.7  9.9240    0.003191 **
## PosOr.cont:session       0.11067 0.05534     2 4104.4  1.1151    0.327971
## PosOr.cont:group         0.00503 0.00503     1 4104.2  0.1014    0.750137
## session:group            0.26327 0.13164     2   38.2  2.6527    0.083409 .
## PosOr.cont:session:group 0.14854 0.07427     2 4104.4  1.4967    0.224000
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

```
saveRDS(m2_lmm,  file = here::here("results", "tables", "CSI_online_aphasia_PosOrSessionxGroup_lmm_VOT_F
tab_model(m2_lmm,transform = NULL,
          show.re.var = F, show.stat = T,show.r2 = F,show.icc = F,
          title = "LMM of VOTs of Filler trials Predicted by Ordinal Position and Session",
          dv.labels = "Vocal Onset Time (log-transformed)",
          #string.pred = "",
          df.method = "satterthwaite",
          string.stat = "t-Value",
          file = here::here(
            "results", "tables",
            "CSI_online_aphasia_spoken_SessionxGroup_lmm_VOT_FILLERS.html"))
```

LMM of VOTs of Filler trials Predicted by Ordinal Position and Session

Vocal Onset Time (log-transformed)

Predictors

Estimates

CI

t-Value

p

(Intercept)

7.06

6.99 – 7.12

226.42

<0.001

PosOr cont

0.00

-0.00 – 0.01

1.54

0.123

session [2]

-0.08

-0.10 – -0.05

-6.45

<0.001

session [3]

-0.08

-0.11 – -0.05

-5.78

<0.001

group2-1

0.20

0.07 – 0.32

3.15

0.003

PosOr cont × session [2]

0.01

-0.00 – 0.02

1.48

0.139

PosOr cont × session [3]

0.00

-0.01 – 0.02

0.58

0.560

PosOr.cont:group2-1

-0.00

-0.01 – 0.01

-0.32

0.750

session2:group2-1

-0.06

-0.11 – -0.01

-2.27

0.029

session3:group2-1

-0.05

-0.11 – 0.01

-1.58

0.121

PosOr.cont:session2:group2-1

-0.02

-0.04 – 0.01

-1.30

0.194

PosOr.cont:session3:group2-1

0.00

-0.02 – 0.03

0.33

0.743

N subject

40

Observations

4218

```r
## Check model
# performance::check_model(m2_lmm)
# ggResidpanel::resid_panel(m2_lmm, smoother = TRUE,
                          # qqbands = TRUE, type = "pearson")
```

Main model with category block as covariate Add “block” variable for trials of individual participants

```r
df_RTs <- df_RTs %>%
  mutate(filler_block = case_when(
    trial <= 20 ~ 1,
    trial > 20 & trial <= 40  ~ 2,
    trial > 40 & trial <= 60  ~ 3,
    trial > 60 & trial <= 80  ~ 4,
    trial > 80 & trial <= 100  ~ 5,
    trial > 100 & trial <= 120  ~ 6,
    trial > 120 & trial <= 140  ~ 7,
    trial > 140 & trial <= 160  ~ 8))
# use as centered continuous variable
df_RTs$filler_block_c <-
  scale(df_RTs$filler_block, center=T, scale=F)
```

Add to converging model

```
m2_lmm_control <- lmer(lVOT ~
                 PosOr.cont*session*group*filler_block_c +
               (PosOr.cont+session|subject) +
              (PosOr.cont+group|category),
             data = df_RTs,
            control=lmerControl(optimizer = "bobyqa",
                                 optCtrl = list(
                                   maxfun = 2e5)))
summary(m2_lmm_control)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: lVOT ~ PosOr.cont * session * group * filler_block_c + (PosOr.cont +
##     session | subject) + (PosOr.cont + group | category)
##    Data: df_RTs
## Control: lmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 200000))
##
## REML criterion at convergence: 1267.8
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.9413 -0.6604 -0.1718  0.4775  5.6569
##
## Random effects:
##  Groups   Name        Variance   Std.Dev. Corr
##  subject  (Intercept) 0.03577866 0.189152
##           PosOr.cont  0.00010267 0.010133  0.18
##           session2    0.00339497 0.058266 -0.13  0.25
##           session3    0.00404501 0.063600 -0.49  0.27  0.64
##  category (Intercept) 0.00874730 0.093527
##           PosOr.cont  0.00002298 0.004794 0.10
##           group2-1    0.00090366 0.030061 0.06  0.21
##  Residual             0.06152928 0.248051
## Number of obs: 12455, groups:  subject, 40; category, 24
##
## Fixed effects:
##                                                   Estimate    Std. Error
## (Intercept)                                      7.1166377     0.0355631
## PosOr.cont                                       0.0168851     0.0024766
## session2                                        -0.0758537     0.0108508
## session3                                        -0.0891813     0.0115410
## group2-1                                         0.1578366     0.0603215
## filler_block_c                                   0.0028800     0.0011690
## PosOr.cont:session2                              0.0046453     0.0039247
## PosOr.cont:session3                              0.0037132     0.0038953
## PosOr.cont:group2-1                              0.0069304     0.0045497
## session2:group2-1                               -0.0251090     0.0217017
## session3:group2-1                               -0.0198596     0.0230820
## PosOr.cont:filler_block_c                       -0.0005891     0.0007235
## session2:filler_block_c                          0.0054085     0.0024334
## session3:filler_block_c                          0.0026637     0.0024132
## group2-1:filler_block_c                          0.0011232     0.0021901
## PosOr.cont:session2:group2-1                     0.0132400     0.0078495
## PosOr.cont:session3:group2-1                     0.0029669     0.0077908
## PosOr.cont:session2:filler_block_c              -0.0020722     0.0017197
## PosOr.cont:session3:filler_block_c              -0.0029060     0.0017016
## PosOr.cont:group2-1:filler_block_c              -0.0006622     0.0013896
## session2:group2-1:filler_block_c                 0.0057401     0.0048733
## session3:group2-1:filler_block_c                 0.0015736     0.0048220
## PosOr.cont:session2:group2-1:filler_block_c      0.0003111     0.0034432
## PosOr.cont:session3:group2-1:filler_block_c      0.0013833     0.0034003
##                                                         df t value
## (Intercept)                                     58.3818764 200.113
## PosOr.cont                                      29.9245813   6.818
## session2                                        38.4343494  -6.991
## session3                                        36.6227844  -7.727
## group2-1                                        38.5604588   2.617
## filler_block_c                               11895.6821369   2.464
## PosOr.cont:session2                          12249.6910925   1.184
## PosOr.cont:session3                          12250.8770133   0.953
## PosOr.cont:group2-1                             37.1498889   1.523
## session2:group2-1                               38.4339969  -1.157
## session3:group2-1                               36.6227637  -0.860
## PosOr.cont:filler_block_c                      415.4699244  -0.814
## session2:filler_block_c                      12234.0714453   2.223
## session3:filler_block_c                      12250.7940892   1.104
## group2-1:filler_block_c                       1213.4440408   0.513
## PosOr.cont:session2:group2-1                 12247.7180439   1.687
## PosOr.cont:session3:group2-1                 12250.2037653   0.381
## PosOr.cont:session2:filler_block_c           12255.4780239  -1.205
## PosOr.cont:session3:filler_block_c           12138.4041122  -1.708
## PosOr.cont:group2-1:filler_block_c           12241.3921513  -0.477
## session2:group2-1:filler_block_c             12248.8057815   1.178
## session3:group2-1:filler_block_c             12253.0730201   0.326
## PosOr.cont:session2:group2-1:filler_block_c  11992.8602949   0.090
## PosOr.cont:session3:group2-1:filler_block_c  12253.2741177   0.407
##                                                          Pr(>|t|)
## (Intercept)                                  < 0.0000000000000002 ***
## PosOr.cont                                          0.00000014742 ***
## session2                                            0.00000002385 ***
## session3                                            0.00000000333 ***
## group2-1                                                   0.0126 *
## filler_block_c                                             0.0138 *
## PosOr.cont:session2                                        0.2366
## PosOr.cont:session3                                        0.3405
## PosOr.cont:group2-1                                        0.1362
## session2:group2-1                                          0.2544
## session3:group2-1                                          0.3952
## PosOr.cont:filler_block_c                                 0.4160
## session2:filler_block_c                                   0.0263 *
## session3:filler_block_c                                   0.2697
## group2-1:filler_block_c                                   0.6082
## PosOr.cont:session2:group2-1                              0.0917 .
## PosOr.cont:session3:group2-1                              0.7033
## PosOr.cont:session2:filler_block_c                        0.2282
## PosOr.cont:session3:filler_block_c                        0.0877 .
## PosOr.cont:group2-1:filler_block_c                        0.6337
## session2:group2-1:filler_block_c                          0.2389
## session3:group2-1:filler_block_c                          0.7442
## PosOr.cont:session2:group2-1:filler_block_c               0.9280
## PosOr.cont:session3:group2-1:filler_block_c               0.6841
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

```
##
## Correlation matrix not shown by default, as p = 24 > 12.
## Use print(x, correlation=TRUE)  or
##     vcov(x)        if you need it
```

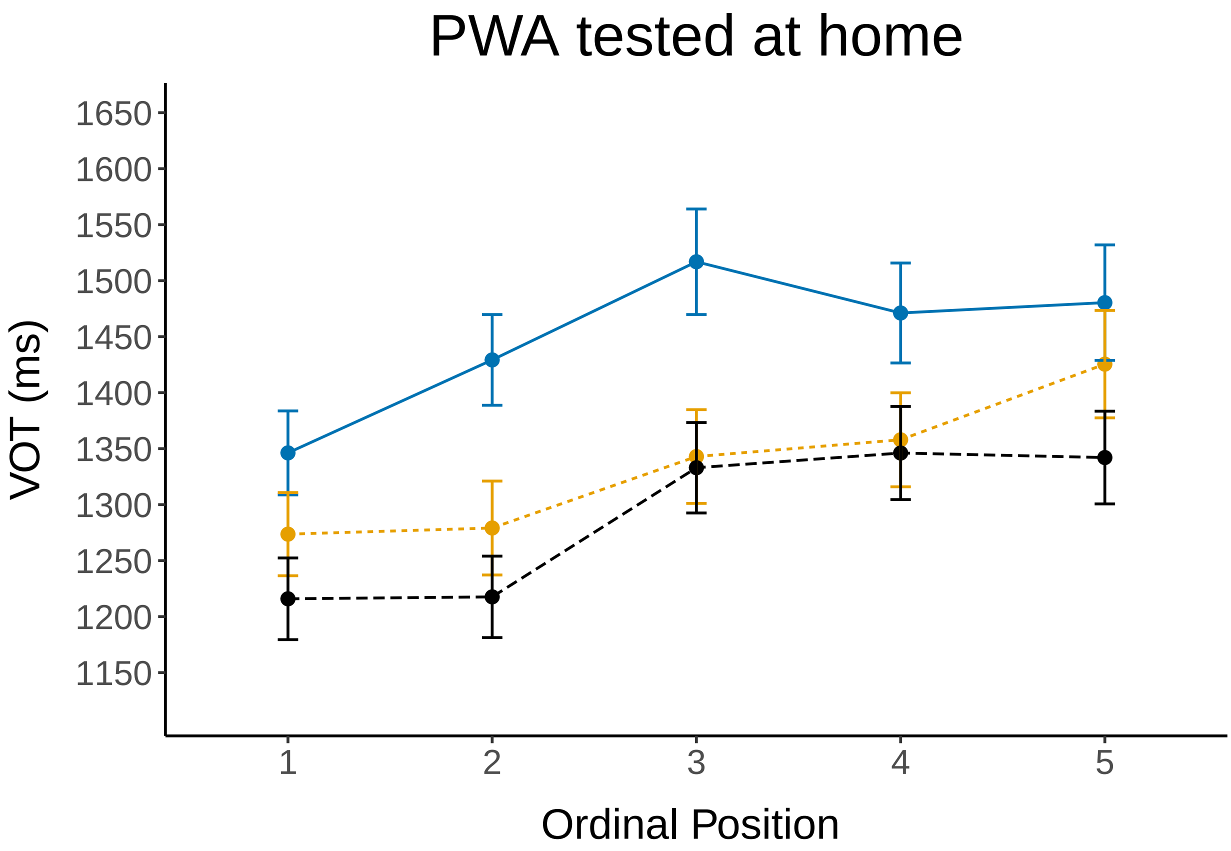
Comparing PWA tested in the clinic and at home

```
read.csv2(here::here("data", "additional_data",
                     "questionnaire-test_data.csv")) -> tests
df_RTs_PWA$test_location <- NA
for(i in 1:length(unique(df_RTs_PWA$subject))){
  df_RTs_PWA$test_location[
    df_RTs_PWA$subject==unique(df_RTs_PWA$subject)[i]] <-
    tests$Ort.zu.Hause..1...in.Klinik..2.[
      tests$Proband.in==unique(df_RTs_PWA$subject)[i]]
}

# 1 = home testes, 2 = tested in clinic
df_RTs_PWA <- df_RTs_PWA %>% mutate(test_location=case_when(
  test_location==1~"home",
  test_location==2~"clinic")) %>%
  mutate(test_location=factor(test_location, levels=c("home", "clinic")))


# descriptives
(means_final<- df_RTs_PWA %>%
   filter(category != "Filler") %>%
   Rmisc::summarySEwithin(.,"VOT",idvar = "subject",
                          withinvars = c("PosOr","session"),
                          betweenvars = "test_location", na.rm = T))
```

```
##    test_location PosOr session   N      VOT       sd       se        ci
## 1           home     1       1 127 1346.199 422.3752 37.47974  74.17130
## 2           home     1       2 132 1273.638 426.7381 37.14278  73.47729
## 3           home     1       3 138 1215.905 428.6614 36.49009  72.15663
## 4           home     2       1 125 1429.225 453.0344 40.52063  80.20168
## 5          home       2       2 133 1279.128 483.4542 41.92079  82.92347
## 6          home       2       3 134 1217.631 421.1310 36.38017  71.95857
## 7          home       3       1 115 1516.877 505.2941 47.11892  93.34221
## 8          home       3       2 124 1342.945 466.1062 41.85757  82.85449
## 9          home       3       3 132 1332.938 463.9254 40.37953  79.88033
## 10         home       4       1 123 1471.126 494.8829 44.62208  88.33387
## 11         home       4       2 129 1357.914 476.7573 41.97614  83.05696
## 12         home       4       3 135 1346.080 483.0407 41.57352  82.22519
## 13         home       5       1  97 1480.381 507.7860 51.55785 102.34153
## 14         home       5       2 116 1425.451 516.4443 47.95065  94.98100
## 15         home       5       3 126 1342.039 464.5974 41.38963  81.91521
## 16       clinic       1       1 235 1394.079 484.1078 31.57971  62.21689
## 17       clinic       1       2 243 1261.039 446.2997 28.63014  56.39608
## 18       clinic       1       3 257 1253.911 340.6824 21.25118  41.84940
## 19       clinic       2       1 220 1459.885 552.2705 37.23407  73.38296
## 20       clinic       2       2 253 1287.994 421.7362 26.51433  52.21791
## 21       clinic       2       3 252 1313.327 472.3141 29.75299  58.59734
## 22       clinic       3       1 222 1447.442 519.4234 34.86141  68.70335
## 23       clinic       3       2 246 1292.562 419.4500 26.74316  52.67583
## 24       clinic       3       3 264 1314.615 431.6229 26.56455  52.30626
## 25       clinic       4       1 234 1410.285 454.0671 29.68329  58.48196
## 26       clinic       4       2 249 1334.080 447.4857 28.35825  55.85372
## 27       clinic       4       3 249 1305.202 410.8462 26.03632  51.28050
## 28       clinic       5       1 225 1489.114 552.1233 36.80822  72.53468
## 29       clinic       5       2 239 1394.637 487.2906 31.52021  62.09422
## 30       clinic       5       3 244 1383.710 472.4320 30.24436  59.57456
```

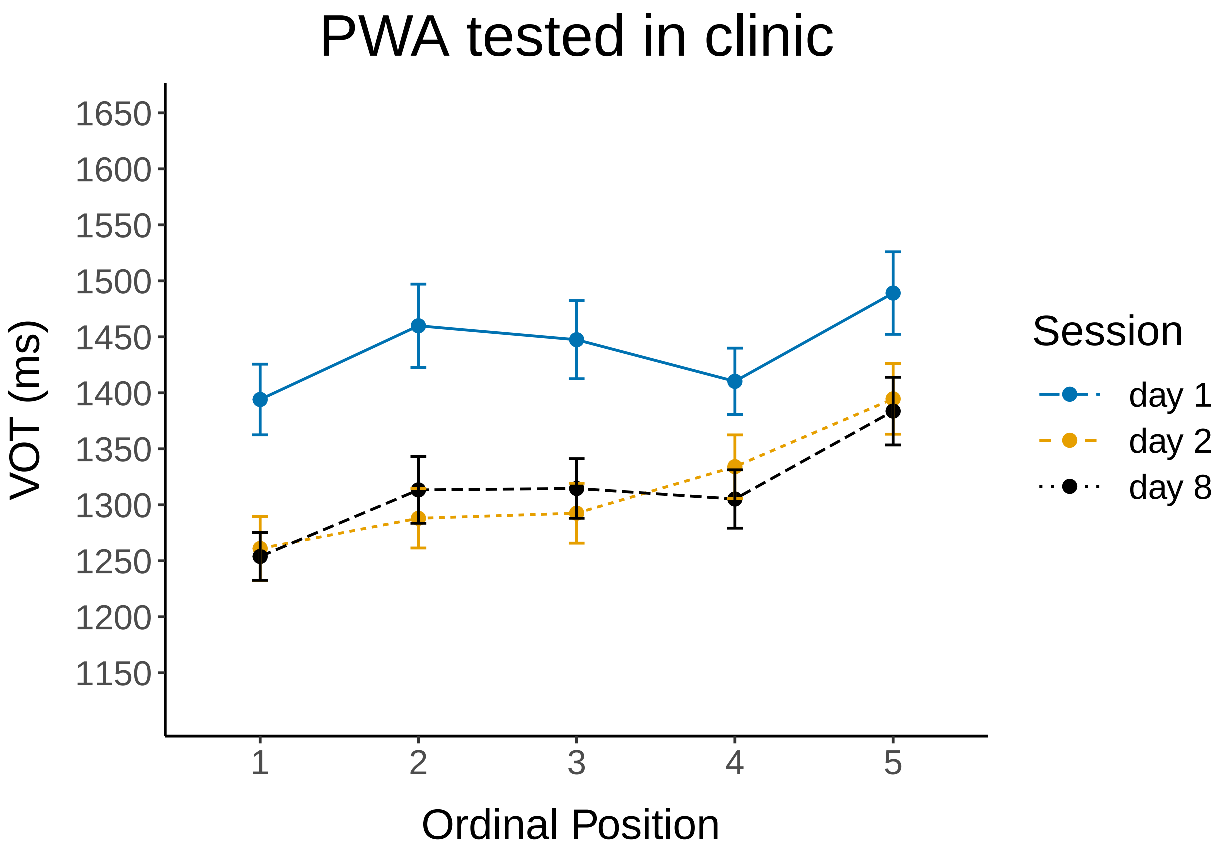
```
# plot
override.linetype<-c("longdash", "dashed", "dotted")
(plot_rt_repetition_home <- means_final %>% filter(test_location=="home") %>%
    mutate(session=case_when(session == "1" ~ "day 1",
                                 session == "2" ~ "day 2",
                                 session == "3" ~ "day 8")) %>%
    ggplot(., aes(x=PosOr, y=VOT, group=session, color = session)) +
    geom_point(size = 2)+
    stat_summary(aes(linetype=session),fun=mean,
                 geom="line", size = 0.5) +
    scale_linetype_manual(values=c("longdash", "dashed", "dotted"))+
    scale_color_manual(values=c("#0072B2", "#E69F00", "#000000"))+
    geom_errorbar(aes(ymin=VOT-se, ymax=VOT+se, group = session),
                  width =.1) +
    apatheme+
    scale_y_continuous(limits = c(1120, 1650),
                       breaks =seq(1150,1650, by = 50)) +
    labs(x="Ordinal Position ",y ="VOT (ms)", colour="Session",
         linetype="Session",
         title = "PWA tested at home") +
    theme(
    axis.title.y = element_text(margin = margin(0,10,0,0)),
    axis.title.x = element_text(margin = margin(10,0,0,0)),
    legend.key.width = unit(1, "cm"),
    legend.position="none")+
   guides(color=guide_legend(
     override.aes=list(linetype=override.linetype)))+
    scale_linetype(guide="none"))
```

```
## Scale for linetype is already present.
## Adding another scale for linetype, which will replace the existing scale.
```



```
(plot_rt_repetition_clinic <- means_final %>%
    filter(test_location=="clinic") %>%
    mutate(session=case_when(session == "1" ~ "day 1",
                             session == "2" ~ "day 2",
                             session == "3" ~ "day 8")) %>%
    ggplot(., aes(x=PosOr, y=VOT, group=session, color = session)) +
    geom_point(size = 2)+
    stat_summary(aes(linetype=session),fun=mean,
                 geom="line", size = 0.5) +
    scale_linetype_manual(values=c("longdash", "dashed", "dotted"))+
    scale_color_manual(values=c("#0072B2", "#E69F00", "#000000"))+
    geom_errorbar(aes(ymin=VOT-se, ymax=VOT+se, group = session),
                  width =.1) +
    apatheme+
    scale_y_continuous(limits = c(1120, 1650),
                       breaks =seq(1150,1650, by = 50)) +
    labs(x="Ordinal Position ",y ="VOT (ms)", colour="Session", linetype="Session",
         title = "PWA tested in clinic") +
    theme(
    axis.title.y = element_text(margin = margin(0,10,0,0)),
    axis.title.x = element_text(margin = margin(10,0,0,0)),
    legend.key.width = unit(1, "cm"))+
  guides(color=guide_legend(
    override.aes=list(linetype=override.linetype)))+
    scale_linetype(guide="none"))
```

```
## Scale for linetype is already present.
## Adding another scale for linetype, which will replace the existing scale.
```



```
plots <- cowplot::plot_grid(
  plot_rt_repetition_home,plot_rt_repetition_clinic,
  nrow = 1, ncol=2,  rel_widths = c(0.81,1), #rel_height = c(1,1),
  margin(1,1,1,1),
  labels = c("A", "B"),label_size = 34,
  label_fontfamily = "Helvetica", label_y = 1.01,
  label_x=-0.03)
```

```
## Warning in as_grob.default(plot): Cannot convert object of class
## marginsimpleUnitunitunit_v2 into a grob.
```

```
filename <- "CSI_online_aphasia_spoken_plot_rt_homevsclinic.pdf"
ggsave(plots, filename =
         here::here("results", "figures", filename),
       width = 25, height = 13, units = "cm",
       dpi = 300, device = cairo_pdf)
#embedFonts(file = here::here("results", "figures", filename))

## model
contrasts(df_RTs_PWA$test_location) <- MASS::contr.sdif(2)
levels(df_RTs_PWA$test_location)
```

```
## [1] "home"   "clinic"
```

```
# use converging model and add test_location as predictor
## fixed structure
m1_lmm_PWA_location <- lmer(VOTlog ~ test_location*(PosOr.cont*session) +
               (PosOr.cont+session|subject) +
              (1|category),
             data = df_RTs_PWA,
            control=lmerControl(optimizer = "bobyqa",
                                 optCtrl = list(maxfun = 2e5)))
summary(m1_lmm_PWA_location)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: VOTlog ~ test_location * (PosOr.cont * session) + (PosOr.cont +
##     session | subject) + (1 | category)
##    Data: df_RTs_PWA
## Control: lmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 200000))
##
## REML criterion at convergence: 1731.4
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.4937 -0.6783 -0.1854  0.5093  5.1042
##
## Random effects:
##  Groups   Name        Variance  Std.Dev. Corr
##  category (Intercept) 0.0092411 0.09613
##  subject  (Intercept) 0.0432058 0.20786
##           PosOr.cont  0.0001311 0.01145   0.22
##           session2    0.0009489 0.03080  -0.44 -0.01
##           session3    0.0033828 0.05816  -0.63 -0.07  0.28
##  Residual             0.0761010 0.27586
## Number of obs: 5518, groups:  category, 24; subject, 20
##
## Fixed effects:
##                                         Estimate  Std. Error          df
## (Intercept)                              7.224581    0.051538   24.098573
## test_location2-1                        -0.287603    0.095310   18.047629
## PosOr.cont                               0.021490    0.003917   17.802761
## session2                                -0.086762    0.012219   15.325272
## session3                                -0.103111    0.016666   18.352910
## PosOr.cont:session2                      0.009888    0.006943 5437.311688
## PosOr.cont:session3                      0.004646    0.006869 5437.400089
## test_location2-1:PosOr.cont             -0.007095    0.007833   17.798044
## test_location2-1:session2              -0.012742    0.024436   15.321425
## test_location2-1:session3               0.037982    0.033326   18.341805
## test_location2-1:PosOr.cont:session2    0.009872    0.013887 5437.175137
## test_location2-1:PosOr.cont:session3    0.004143    0.013734 5437.249047
##                                      t value             Pr(>|t|)    
## (Intercept)                          140.178 < 0.0000000000000002 ***
## test_location2-1                      -3.018              0.00738 ** 
## PosOr.cont                             5.486           0.00003412 ***
## session2                              -7.101           0.00000321 ***
## session3                              -6.187           0.00000706 ***
## PosOr.cont:session2                    1.424              0.15446    
## PosOr.cont:session3                    0.676              0.49885    
## test_location2-1:PosOr.cont           -0.906              0.37715    
## test_location2-1:session2             -0.521              0.60952    
## test_location2-1:session3              1.140              0.26908    
## test_location2-1:PosOr.cont:session2   0.711              0.47717    
## test_location2-1:PosOr.cont:session3   0.302              0.76292    
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
## 
## Correlation of Fixed Effects:
##              (Intr) ts_2-1 PsOr.c sessn2 sessn3 PsO.:2 PsO.:3 ts_2-1:PO. t_2-1:2
## tst_lctn2-1 -0.187                                                               
## PosOr.cont   0.138 -0.033                                                        
## session2    -0.235  0.051 -0.007                                                 
## session3    -0.466  0.103 -0.044  0.375                                          
## PsOr.cnt:s2 -0.001  0.001 -0.024  0.019  0.010                                   
## PsOr.cnt:s3 -0.002  0.002 -0.042  0.014  0.013  0.526                            
## tst_2-1:PO. -0.030  0.149 -0.295  0.005  0.017  0.011  0.024                     
## tst_lc2-1:2  0.047 -0.254  0.005 -0.299 -0.107 -0.027 -0.019 -0.007              
## tst_lc2-1:3  0.095 -0.504  0.017 -0.107 -0.260 -0.014 -0.017 -0.044      0.375  
## t_2-1:PO.:2  0.001 -0.001  0.011 -0.027 -0.014 -0.330 -0.177 -0.024      0.018  
## t_2-1:PO.:3  0.002 -0.002  0.024 -0.019 -0.017 -0.177 -0.324 -0.042      0.014  
##              t_2-1:3 t_2-1:PO.:2
## tst_lctn2-1                     
## PosOr.cont                      
## session2                        
## session3                        
## PsOr.cnt:s2                     
## PsOr.cnt:s3                     
## tst_2-1:PO.                     
## tst_lc2-1:2                     
## tst_lc2-1:3                     
## t_2-1:PO.:2  0.010              
## t_2-1:PO.:3  0.013   0.526
```

```
## random structure
m1_lmm_PWA_location2 <- lmer(VOTlog ~ (PosOr.cont*session) +
               (PosOr.cont+session|subject) +
              (test_location|category),
             data = df_RTs_PWA,
            control=lmerControl(optimizer = "bobyqa",
                                 optCtrl = list(maxfun = 2e5)))
```

```
## boundary (singular) fit: see help(’isSingular’)
```

```
summary(m1_lmm_PWA_location2) # model does not converge and test_location explains least variance
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite’s method [
## lmerModLmerTest]
## Formula: VOTlog ~ (PosOr.cont * session) + (PosOr.cont + session | subject) +
##     (test_location | category)
##    Data: df_RTs_PWA
## Control: lmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 200000))
##
## REML criterion at convergence: 1698.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.5191 -0.6861 -0.1837  0.5152  5.0874
##
## Random effects:
##  Groups   Name             Variance  Std.Dev. Corr
##  category (Intercept)      0.0086220 0.09285
##           test_location2-1 0.0004396 0.02097   1.00
##  subject  (Intercept)      0.0612978 0.24758
##           PosOr.cont       0.0001267 0.01125   0.35
##           session2         0.0009234 0.03039  -0.25  0.09
##           session3         0.0034611 0.05883  -0.67 -0.14  0.19
##  Residual                  0.0759800 0.27564
## Number of obs: 5518, groups:  category, 24; subject, 20
##
## Fixed effects:
##                       Estimate  Std. Error          df t value
## (Intercept)            7.197088    0.058800   23.491270 122.400
## PosOr.cont             0.020659    0.003705   17.705800   5.576
## session2              -0.087833    0.011606   16.110144  -7.568
## session3              -0.098606    0.016190   17.686389  -6.090
## PosOr.cont:session2    0.011482    0.006547 5438.018353   1.754
## PosOr.cont:session3    0.005429    0.006491 5440.059343   0.836
##                                  Pr(>|t|)
## (Intercept)         < 0.0000000000000002 ***
## PosOr.cont                    0.00002885 ***
## session2                      0.00000108 ***
## session3                      0.00001011 ***
## PosOr.cont:session2               0.0796 .
## PosOr.cont:session3               0.4030
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Correlation of Fixed Effects:
##             (Intr) PsOr.c sessn2 sessn3 PsO.:2
## PosOr.cont   0.223
## session2    -0.138  0.036
## session3    -0.515 -0.077  0.332
## PsOr.cnt:s2  0.000 -0.021  0.004  0.003
## PsOr.cnt:s3  0.000 -0.035  0.003  0.003  0.522
## optimizer (bobyqa) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
```

Assessing the variance of the ordinal position effect between groups

VOTs Compute a nested model on the VOT model

```
m1_VOTs <- readRDS(file =
                     here::here(
                       "results", "tables",
                       "CSI_online_aphasia_SessionxGroup_control_lmm_VOT.RDS"))
#summary(m1_VOTs)
#m1_VOTs@call

## Use this structure for a nested model:
m1_VOTs_nested <- lmer(lVOT ~ group/session/PosOr.cont +
                         (PosOr.cont+session|subject) +
                         (PosOr.cont+group|category),
                       data = df_RTs,
                       control=lmerControl(optimizer = "bobyqa",
                                           optCtrl = list(maxfun = 2e5)))
summary(m1_VOTs_nested)
```

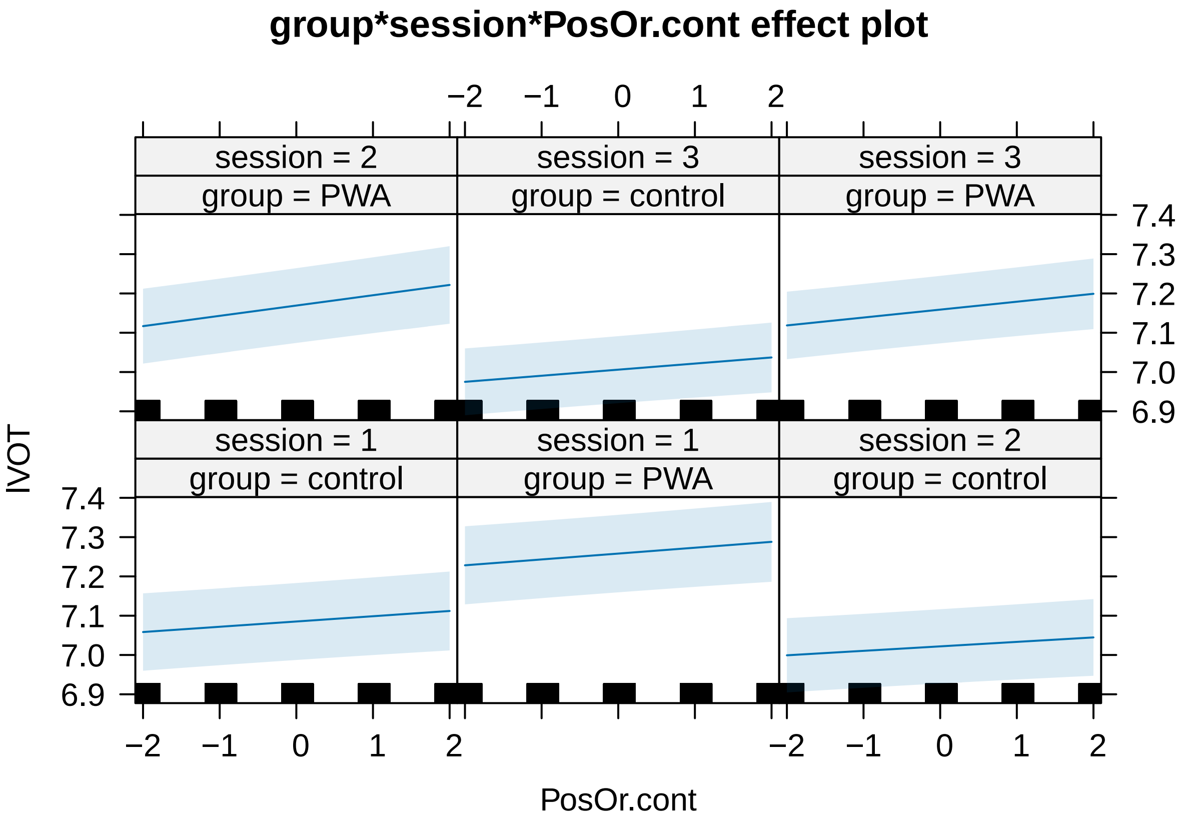
```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: lVOT ~ group/session/PosOr.cont + (PosOr.cont + session | subject) +
##     (PosOr.cont + group | category)
##    Data: df_RTs
## Control: lmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 200000))
##
## REML criterion at convergence: 1158.2
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.8980 -0.6647 -0.1747  0.4804  5.6443
##
## Random effects:
##  Groups   Name        Variance   Std.Dev. Corr
##  subject  (Intercept) 0.03572329 0.189006
##           PosOr.cont  0.00010406 0.010201  0.18
##           session2    0.00339921 0.058303 -0.13  0.24
##           session3    0.00406340 0.063745 -0.49  0.26  0.64
##  category (Intercept) 0.00884966 0.094073
##           PosOr.cont  0.00002358 0.004856 0.07
##           group2-1    0.00090450 0.030075 0.08  0.16
##  Residual             0.06154884 0.248090
## Number of obs: 12455, groups:  subject, 40; category, 24
##
## Fixed effects:
##                                   Estimate Std. Error         df t value
## (Intercept)                       7.116562   0.035604  58.495156 199.883
## group2-1                          0.157635   0.060276  38.561678   2.615
## groupcontrol:session2            -0.063279   0.014959  35.588407  -4.230
## groupPWA:session2                -0.088885   0.015736  41.218683  -5.649
## groupcontrol:session3             -0.079172   0.016016  34.253327  -4.943
## groupPWA:session3                 -0.099251   0.016677  39.015066  -5.951
## groupcontrol:session1:PosOr.cont   0.013399   0.004431 115.483152   3.024
## groupPWA:session1:PosOr.cont       0.014924   0.004957 164.783105   3.011
## groupcontrol:session2:PosOr.cont   0.011381   0.004423 114.610795   2.573
## groupPWA:session2:PosOr.cont       0.026211   0.004831 148.762370   5.426
## groupcontrol:session3:PosOr.cont   0.015500   0.004390 111.385092   3.531
## groupPWA:session3:PosOr.cont       0.020120   0.004753 142.968524   4.233
##                                              Pr(>|t|)    
## (Intercept)                      < 0.0000000000000002 ***
## group2-1                                     0.012659 *  
## groupcontrol:session2                        0.000156 ***
## groupPWA:session2                         0.000001345 ***
## groupcontrol:session3                     0.000019994 ***
## groupPWA:session3                         0.000000604 ***
## groupcontrol:session1:PosOr.cont             0.003074 ** 
## groupPWA:session1:PosOr.cont                 0.003016 ** 
## groupcontrol:session2:PosOr.cont             0.011351 *  
## groupPWA:session2:PosOr.cont              0.000000229 ***
## groupcontrol:session3:PosOr.cont             0.000604 ***
## groupPWA:session3:PosOr.cont              0.000041084 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Correlation of Fixed Effects:
##             (Intr) grp2-1 grpc:2 grPWA:2 grpc:3 grPWA:3 g:1:PO gPWA:1 g:2:PO
## group2-1     0.006                                                          
## grpcntrl:s2 -0.070  0.082                                                   
## grpPWA:sss2 -0.067 -0.079  0.000                                            
## grpcntrl:s3 -0.259  0.307  0.613  0.000                                     
## grpPWA:sss3 -0.251 -0.296  0.000  0.612   0.000                             
## grpcn:1:PO.  0.063 -0.061  0.110  0.000   0.123  0.000                      
## grPWA:1:PO.  0.058  0.062  0.000  0.086   0.000  0.100   0.045              
## grpcn:2:PO.  0.064 -0.061  0.107  0.000   0.121  0.000   0.316  0.045       
## grPWA:2:PO.  0.059  0.063  0.000  0.102   0.000  0.111   0.046  0.274  0.046
## grpcn:3:PO.  0.064 -0.061  0.109  0.000   0.119  0.000   0.318  0.045  0.319
## grPWA:3:PO.  0.060  0.064  0.000  0.099   0.000  0.116   0.047  0.276  0.047
##             gPWA:2: g:3:PO
## group2-1                  
## grpcntrl:s2               
## grpPWA:sss2               
## grpcntrl:s3               
## grpPWA:sss3               
## grpcn:1:PO.               
## grPWA:1:PO.               
## grpcn:2:PO.               
## grPWA:2:PO.               
## grpcn:3:PO.  0.046        
## grPWA:3:PO.  0.284   0.047
```

```
as.data.frame(effects::allEffects(m1_VOTs_nested))
```

```
## Warning in Analyze.model(focal.predictors, mod, xlevels, default.levels, : the
## predictor PosOr.cont is a one-column matrix that was converted to a vector
## $‘group:session:PosOr.cont‘
##      group session PosOr.cont      fit         se    lower    upper
## 1  control       1      -2.00 7.058431 0.05023269 6.959967 7.156895
## 2      PWA       1      -2.00 7.228244 0.05070769 7.128849 7.327638
## 3  control       2      -2.00 6.999187 0.04819351 6.904720 7.093653
## 4      PWA       2      -2.00 7.116784 0.04862002 7.021481 7.212086
## 5  control       3      -2.00 6.975058 0.04340011 6.889987 7.060128
## 6      PWA       3      -2.00 7.118601 0.04380863 7.032729 7.204473
## 7  control       1      -1.00 7.071830 0.04989989 6.974018 7.169641
## 8      PWA       1      -1.00 7.243167 0.05026643 7.144637 7.341697
## 9  control       2      -1.00 7.010568 0.04799961 6.916481 7.104655
## 10     PWA       2      -1.00 7.142995 0.04834359 7.048234 7.237756
## 11 control       3      -1.00 6.990557 0.04322436 6.905831 7.075284
## 12     PWA       3      -1.00 7.138721 0.04355954 7.053337 7.224104
## 13 control       1       0.02 7.085496 0.04996453 6.987558 7.183435
## 14     PWA       1       0.02 7.258390 0.05031788 7.159759 7.357020
## 15 control       2       0.02 7.022177 0.04822077 6.927657 7.116697
## 16     PWA       2       0.02 7.169731 0.04855766 7.074550 7.264911
## 17 control       3       0.02 7.006367 0.04350313 6.921094 7.091640
## 18     PWA       3       0.02 7.159243 0.04383823 7.073313 7.245172
## 19 control       1       1.00 7.098627 0.05040966 6.999816 7.197438
## 20     PWA       1       1.00 7.273015 0.05084307 7.173355 7.372675
## 21 control       2       1.00 7.033330 0.04882653 6.937623 7.129038
## 22     PWA       2       1.00 7.195418 0.04922923 7.098921 7.291915
## 23 control       3       1.00 7.021556 0.04419867 6.934920 7.108193
## 24     PWA       3       1.00 7.178960 0.04460355 7.091530 7.266390
## 25 control       1       2.00 7.112025 0.05124054 7.011586 7.212465
## 26     PWA       1       2.00 7.287939 0.05184473 7.186315 7.389562
## 27 control       2       2.00 7.044712 0.04982719 6.947043 7.142381
## 28     PWA       2       2.00 7.221629 0.05036602 7.122904 7.320354
## 29 control       3       2.00 7.037056 0.04532024 6.948221 7.125891
## 30     PWA       3       2.00 7.199080 0.04586156 7.109184 7.288975
```

```
plot(effects::allEffects(m1_VOTs_nested))
```

```
## Warning in Analyze.model(focal.predictors, mod, xlevels, default.levels, : the
## predictor PosOr.cont is a one-column matrix that was converted to a vector
```



```
## Check whether the by-group slope causes the SEs being similar between groups
m1_VOTs_nested2 <- lmer(lVOT ~ group/session/PosOr.cont +
                         (PosOr.cont+session|subject) +
                         (PosOr.cont|category),
                       data = df_RTs,
                       control=lmerControl(optimizer = "bobyqa",
                                           optCtrl = list(maxfun = 2e5)))
summary(m1_VOTs_nested2)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: lVOT ~ group/session/PosOr.cont + (PosOr.cont + session | subject) +
##     (PosOr.cont | category)
##    Data: df_RTs
## Control: lmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 200000))
##
## REML criterion at convergence: 1176.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.8547 -0.6655 -0.1737  0.4808  5.6868
##
## Random effects:
##  Groups   Name        Variance   Std.Dev. Corr
##  subject  (Intercept) 0.03561229 0.188712
##           PosOr.cont  0.00010428 0.010212  0.17
##           session2    0.00340190 0.058326 -0.13  0.24
##           session3    0.00406562 0.063762 -0.49  0.27  0.65
##  category (Intercept) 0.00882438 0.093938
##           PosOr.cont  0.00002319 0.004816 0.07
##  Residual             0.06176389 0.248523
## Number of obs: 12455, groups:  subject, 40; category, 24
##
## Fixed effects:
##                                    Estimate Std. Error         df t value
## (Intercept)                        7.116181   0.035550  58.503544 200.175
## group2-1                           0.156914   0.059870  37.785294   2.621
## groupcontrol:session2             -0.063272   0.014969  35.590650  -4.227
## groupPWA:session2                 -0.088634   0.015747  41.222784  -5.628
## groupcontrol:session3             -0.079181   0.016025  34.288283  -4.941
## groupPWA:session3                 -0.098842   0.016687  39.053913  -5.923
## groupcontrol:session1:PosOr.cont   0.013395   0.004435 115.666590   3.020
## groupPWA:session1:PosOr.cont       0.014846   0.004962 165.013007   2.992
## groupcontrol:session2:PosOr.cont   0.011378   0.004428 114.792897   2.570
## groupPWA:session2:PosOr.cont       0.026096   0.004836 149.004889   5.396
## groupcontrol:session3:PosOr.cont   0.015543   0.004394 111.556603   3.537
## groupPWA:session3:PosOr.cont       0.019961   0.004759 143.192069   4.195
##                                              Pr(>|t|)
## (Intercept)                      < 0.0000000000000002 ***
## group2-1                                     0.012563 *
## groupcontrol:session2                        0.000157 ***
## groupPWA:session2                         0.000001435 ***
## groupcontrol:session3                     0.000020083 ***
## groupPWA:session3                         0.000000659 ***
## groupcontrol:session1:PosOr.cont             0.003110 **
## groupPWA:session1:PosOr.cont                 0.003196 **
## groupcontrol:session2:PosOr.cont             0.011459 *
## groupPWA:session2:PosOr.cont              0.000000263 ***
## groupcontrol:session3:PosOr.cont             0.000591 ***
## groupPWA:session3:PosOr.cont              0.000047661 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Correlation of Fixed Effects:
##             (Intr) grp2-1 grpc:2 grPWA:2 grpc:3 grPWA:3 g:1:PO gPWA:1 g:2:PO
## group2-1     0.001
## grpcntrl:s2 -0.069  0.082
## grpPWA:sss2 -0.066 -0.079  0.000
## grpcntrl:s3 -0.258  0.307  0.614  0.000
## grpPWA:sss3 -0.250 -0.297  0.000  0.613   0.000
## grpcn:1:PO.  0.061 -0.063  0.109  0.000   0.124  0.000
## grPWA:1:PO.  0.056  0.058  0.000  0.085   0.000  0.100   0.044
## grpcn:2:PO.  0.061 -0.063  0.106  0.000   0.122  0.000   0.315  0.044
## grPWA:2:PO.  0.057  0.059  0.000  0.101   0.000  0.112   0.045  0.273  0.045
## grpcn:3:PO.  0.061 -0.063  0.108  0.000   0.120  0.000   0.317  0.044  0.318
## grPWA:3:PO.  0.057  0.059  0.000  0.098   0.000  0.116   0.046  0.275  0.046
##             gPWA:2: g:3:PO
## group2-1
## grpcntrl:s2
## grpPWA:sss2
## grpcntrl:s3
## grpPWA:sss3
## grpcn:1:PO.
## grPWA:1:PO.
## grpcn:2:PO.
## grPWA:2:PO.
## grpcn:3:PO.  0.046
## grPWA:3:PO.  0.283   0.046
```

```r
anova(m1_VOTs_nested, m1_VOTs_nested2)
```

```
## refitting model(s) with ML (instead of REML)
```

```
## Data: df_RTs
## Models:
## m1_VOTs_nested2: lVOT ~ group/session/PosOr.cont + (PosOr.cont + session | subject) + (PosOr.cont | c
## m1_VOTs_nested: lVOT ~ group/session/PosOr.cont + (PosOr.cont + session | subject) + (PosOr.cont + gr
##                 npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## m1_VOTs_nested2   26 1138.9 1332.0 -543.43   1086.9
## m1_VOTs_nested    29 1126.1 1341.6 -534.06   1068.1 18.736  3    0.00031 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

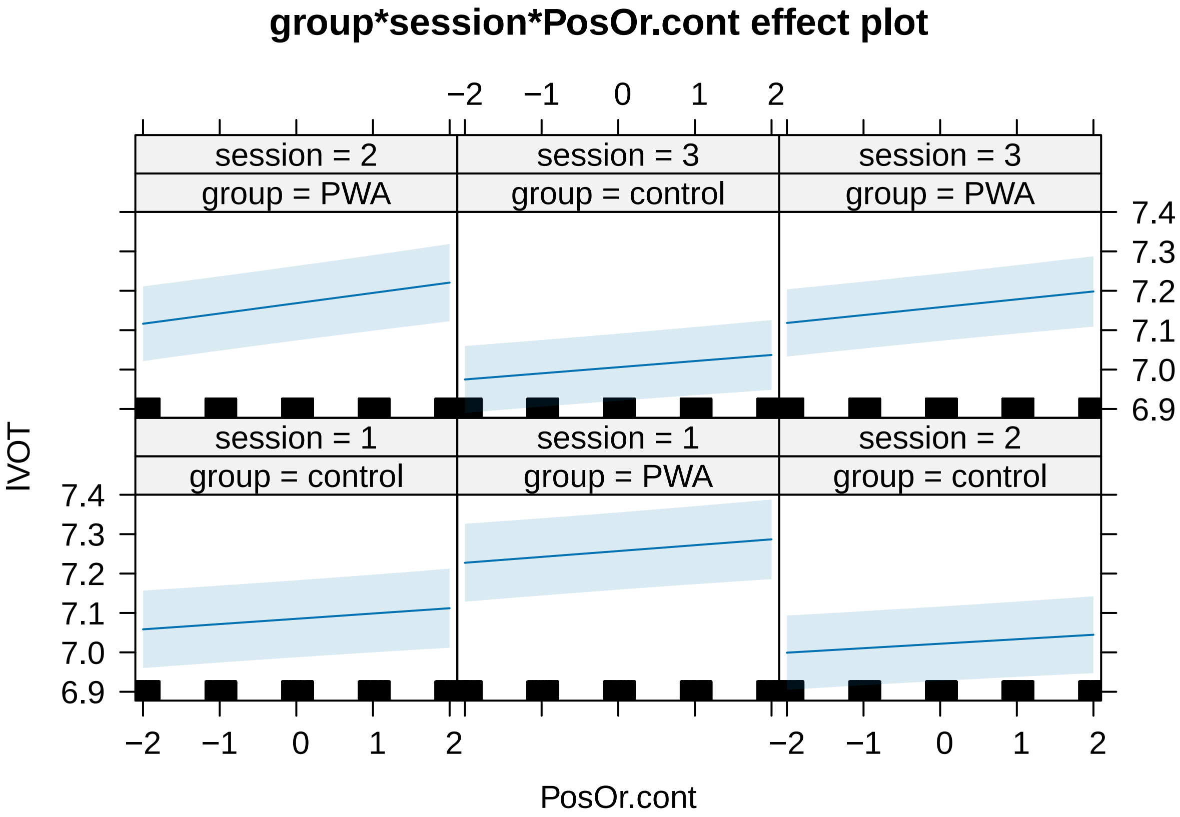
```r
as.data.frame(effects::allEffects(m1_VOTs_nested2))
```

```
## Warning in Analyze.model(focal.predictors, mod, xlevels, default.levels, : the
## predictor PosOr.cont is a one-column matrix that was converted to a vector
```

```
## $‘group:session:PosOr.cont‘
##      group session PosOr.cont      fit         se    lower    upper
## 1  control       1      -2.00 7.058418 0.05017104 6.960075 7.156761
## 2      PWA       1      -2.00 7.227437 0.05048740 7.128474 7.326400
## 3  control       2      -2.00 6.999181 0.04814300 6.904814 7.093549
## 4      PWA       2      -2.00 7.116304 0.04840454 7.021424 7.211184
## 5  control       3      -2.00 6.974940 0.04335975 6.889948 7.059932
## 6      PWA       3      -2.00 7.118366 0.04358522 7.032932 7.203800
## 7  control       1      -1.00 7.071813 0.04983158 6.974135 7.169490
## 8      PWA       1      -1.00 7.242284 0.05001849 7.144240 7.340328
## 9  control       2      -1.00 7.010559 0.04794163 6.916586 7.104532
## 10     PWA       2      -1.00 7.142400 0.04809915 7.048118 7.236682
## 11 control       3      -1.00 6.990484 0.04317804 6.905848 7.075119
## 12     PWA       3      -1.00 7.138327 0.04330637 7.053440 7.223214
## 13 control       1       0.02 7.085476 0.04989082 6.987682 7.183270
## 14     PWA       1       0.02 7.257427 0.05004502 7.159331 7.355523
## 15 control       2       0.02 7.022164 0.04815657 6.927770 7.116558
## 16     PWA       2       0.02 7.169018 0.04828726 7.074367 7.263668
## 17 control       3       0.02 7.006338 0.04345219 6.921165 7.091511
## 18     PWA       3       0.02 7.158687 0.04355896 7.073305 7.244069
## 19 control       1       1.00 7.098603 0.05033216 6.999944 7.197262
## 20     PWA       1       1.00 7.271976 0.05055007 7.172890 7.371062
## 21 control       2       1.00 7.033314 0.04875779 6.937741 7.128887
## 22     PWA       2       1.00 7.194591 0.04893795 7.098666 7.290517
## 23 control       3       1.00 7.021571 0.04414478 6.935040 7.108101
## 24     PWA       3       1.00 7.178249 0.04430401 7.091406 7.265092
## 25 control       1       2.00 7.111998 0.05116065 7.011715 7.212281
## 26     PWA       1       2.00 7.286823 0.05153541 7.185805 7.387840
## 27 control       2       2.00 7.044692 0.04975533 6.947163 7.142220
## 28     PWA       2       2.00 7.220687 0.05005785 7.122566 7.318808
## 29 control       3       2.00 7.037114 0.04526483 6.948388 7.125840
## 30     PWA       3       2.00 7.198210 0.04554644 7.108932 7.287488
```

```
plot(effects::allEffects(m1_VOTs_nested2))
```

```
## Warning in Analyze.model(focal.predictors, mod, xlevels, default.levels, : the
## predictor PosOr.cont is a one-column matrix that was converted to a vector
```



```
### Analyse with bayesian approach:
# bayes_VOTs_nested <- brms::brm(lVOT ~ group/session/PosOr.cont +
#                           (PosOr.cont+session|subject) +
#                           (PosOr.cont+group|category) , data=df_RTs,
#     prior= c(prior(normal(0,100), class = Intercept),
#              prior(normal(0,100), class = sd)),
#     control=list(adapt_delta=.999, max_treedepth=12),
#     warmup = 100, #2000
#     chains=8, iter=100, # 20000
#     save_pars=save_pars(all=TRUE)
#     )
```

```r
#
# save(fit_freq, file="fit_freq.Rdata")
```

Other code

Session info

```r
sessionInfo()
```

```
## R version 4.3.1 (2023-06-16)
## Platform: x86_64-apple-darwin20 (64-bit)
## Running under: macOS Sonoma 14.4
## 
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/lib/libRblas.0.dylib 
## LAPACK: /Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/lib/libRlapack.dylib;  LAPACK v
## 
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
## 
## time zone: Europe/Berlin
## tzcode source: internal
## 
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base     
## 
## other attached packages:
##  [1] actuar_3.3-2        fitdistrplus_1.1-11 survival_3.5-8     
##  [4] MASS_7.3-60.0.1     flextable_0.9.2     dplyr_1.1.2        
##  [7] sjPlot_2.8.14       ggplot2_3.4.2       Cairo_1.6-0        
## [10] Rmisc_1.5.1         plyr_1.8.8          lattice_0.22-6     
## [13] lmerTest_3.1-3      lme4_1.1-35.1       Matrix_1.6-5       
## [16] tidyr_1.3.0        
## 
## loaded via a namespace (and not attached):
##   [1] RColorBrewer_1.1-3      rstudioapi_0.16.0       jsonlite_1.8.7         
##   [4] datawizard_0.8.0        magrittr_2.0.3          estimability_1.4.1     
##   [7] farver_2.1.1            nloptr_2.0.3            rmarkdown_2.23         
##  [10] ragg_1.2.5              vctrs_0.6.3             minqa_1.2.5            
##  [13] askpass_1.1             effectsize_0.8.3        forcats_1.0.0          
##  [16] htmltools_0.5.8.1       haven_2.5.4             curl_5.2.1             
##  [19] survey_4.4-2            broom_1.0.5             sjmisc_2.8.9           
##  [22] huxtable_5.5.2          pracma_2.4.4            htmlwidgets_1.6.4      
##  [25] emmeans_1.8.7           plotly_4.10.4           uuid_1.1-0             
##  [28] mime_0.12               lifecycle_1.0.3         iterators_1.0.14       
##  [31] pkgconfig_2.0.3         sjlabelled_1.2.0        R6_2.5.1               
##  [34] fastmap_1.1.1           shiny_1.7.4.1           digest_0.6.33          
##  [37] numDeriv_2016.8-1.1     colorspace_2.1-0        rprojroot_2.0.3        
##  [40] textshaping_0.3.6       labeling_0.4.2          effects_4.2-2
##  [43] fansi_1.0.4             abind_1.4-5             httr_1.4.7
##  [46] mgcv_1.9-1              compiler_4.3.1          here_1.0.1
##  [49] fontquiver_0.2.1        withr_2.5.0             doParallel_1.0.17
##  [52] backports_1.4.1         DBI_1.2.3               carData_3.0-5
##  [55] performance_0.10.4      highr_0.10              openssl_2.1.0
##  [58] sjstats_0.18.2          gfonts_0.2.0            caTools_1.18.2
##  [61] tools_4.3.1             zip_2.3.0               httpuv_1.6.11
##  [64] nnet_7.3-19             qqconf_1.3.2            glue_1.6.2
##  [67] nlme_3.1-164            promises_1.2.0.1        grid_4.3.1
##  [70] reshape2_1.4.4          generics_0.1.3          qqplotr_0.0.6
##  [73] gtable_0.3.3            ggResidpanel_0.3.0      hms_1.1.3
##  [76] data.table_1.14.8       car_3.1-2               xml2_1.3.5
##  [79] utf8_1.2.3              foreach_1.5.2           pillar_1.9.0
##  [82] stringr_1.5.0           later_1.3.1             mitools_2.4
##  [85] robustbase_0.99-3       splines_4.3.1           tidyselect_1.2.0
##  [88] fontLiberation_0.1.0    expint_0.1-8            knitr_1.43
##  [91] fontBitstreamVera_0.1.1 crul_1.4.0              xfun_0.39
##  [94] DEoptimR_1.1-3          stringi_1.7.12          lazyeval_0.2.2
##  [97] yaml_2.3.7              boot_1.3-30             evaluate_0.21
## [100] codetools_0.2-19        httpcode_0.3.0          officer_0.6.2
## [103] twosamples_2.0.1        gdtools_0.3.3           tibble_3.2.1
## [106] cli_3.6.1               xtable_1.8-4            parameters_0.21.1
## [109] systemfonts_1.0.4       pbmcapply_1.5.1         afex_1.3-0
## [112] munsell_0.5.0           modelr_0.1.11           Rcpp_1.0.11
## [115] ggeffects_1.2.3         coda_0.19-4.1           parallel_4.3.1
## [118] ellipsis_0.3.2          assertthat_0.2.1        bayestestR_0.13.1
## [121] opdisDownsampling_1.0.1 bitops_1.0-8            viridisLite_0.4.2
## [124] mvtnorm_1.2-2           scales_1.2.1            insight_0.19.3
## [127] purrr_1.0.1             crayon_1.5.2            rlang_1.1.1
## [130] cowplot_1.1.1
```