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

Kirsten Stark

17 Oktober, 2022

Load packages

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

```
## Lade nötiges Paket: Matrix
```

```
##
## Attache Paket: 'Matrix'
```

```
## Die folgenden Objekte sind maskiert von 'package:tidyr':
##
##     expand, pack, unpack
```

```
library(lmerTest)
```

```
##
## Attache Paket: 'lmerTest'
```

```
## Das folgende Objekt ist maskiert 'package:lme4':
##
##     lmer
```

```
## Das folgende Objekt ist maskiert 'package:stats':
##
##     step
```

```
library(Rmisc)
```

```
## Lade nötiges Paket: lattice
```

```
## Lade nötiges Paket: plyr
```

```
library(Cairo)
#library(strengejacke)
library(ggplot2)
library(sjPlot)
```

```
## #refugeeswelcome
```

```
library(dplyr)
```

```
##
## Attache Paket: 'dplyr'
```

```
## Die folgenden Objekte sind maskiert von 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
## Die folgenden Objekte sind maskiert von 'package:stats':
##
##     filter, lag
```

```
## Die folgenden Objekte sind maskiert von 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
options(scipen=999)

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

Load and preprocess data

```
# 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

```
# 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)
```

```
##
##         1   2   3
##   101 160 160 160
##   102 160 160 160
##   103 160 160 160
##   104 160 160 160
##   105 160 160 160
##   106 160 160 160
##   107 160 160 160
##   108 160 160 160
##   109 160 160 160
##   110 160 160 160
##   111 160 160 160
##   112 160 160 160
##   113 160 160 160
##   114 160 160 160
##   115 160 160 160
##   116 160 160 160
##   117 160 160 160
##   118 160 160 160
##   119 160 160 160
##   120 160 160 160
##   201 160 160 160
##   202 160 160 160
##   203 160 160 160
##   204 160 160 160
##   205 160 160 160
##   206 160 160 160
##   207 160 160 160
##   208 160 160 160
##   209 160 160 160
##   210 160 160 160
##   211 160 160 160
##   212 160 160 160
##   213 160 160 160
##   214 160 160 160
##   215 160 160 160
##   216 160 160 160
##   217 160 160 160
##   218 160 160 160
##   219 160 160 160
##   220 160 160 160
```

```
# 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 control     202       1             1
##  2 control     203       2             1
##  3 control     204       2             1
##  4 control     204       3             2
##  5 control     205       1             3
##  6 control     205       2             1
##  7 control     205       3             3
##  8 control     207       1             2
##  9 control     208       1             1
## 10 control     208       2             1
## # ... with 76 more rows
```

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

Drop filler trials

```
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))
```

Classified 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

[*] Bei 1.3 wollten wir schauen wie oft die Artikel mitgenannt wurden, aber das können wir ja erstmal vernachlässigen. Für die exakten Ergebnisse wollten wir dann ja nochmal besprechen, weil die Bestimmung der VOT nicht 100% möglich ist mit Artikel.

```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)
```

```
##  [1] NA          "4"         "99"        "5"         "2"
##  [6] "7"         "1"         "9"         "3"         "6"
## [11] "8"         "4;;;;;;"   "1;;;;;;"   "9;;;;;;"   "6;;;;;;"
## [16] "1;;;;;;ok?"
```

```r
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)
```

```
##  [1] NA   "4"  "99" "5"  "2"  "7"  "1"  "9"  "3"  "6"  "8"
```

```r
unique(df$correct)
```

```
## [1] "1"       "1.2"     "1.1"     "0"       "1.3"     "1.4"     "1;;;;;;"
```

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

```
## [1] "1"   "1.2" "1.1" "0"   "1.3" "1.4"
```

```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] ### Was bedeutet das? VOT ist ja nicht immer NA
```

```
##   [1] 2533 1867 1632 1541 1801 2820 2194 2564 2914 1750 1297 1483 1761 1231 2457
##  [16] 2321 1627 1523 1176 1810 1555 1455 1846 1283 1198 1478 1701 1932 1564 1376
##  [31] 1450 1713 1916 3122 2026 1141 1188 1219 1443 2664 1822 1000 1238 3071 1191
##  [46] 1942 1228  923 1648 1251 2061 2254 3208 2245 1423 1376 2057 2214 1718 2773
##  [61] 2022 2088 2766 1470 2558 2045 1509 2116 2880 1662  655 1205 2301 1740 2241
##  [76] 1286 3196 2809 2969 1478 2852 2546 1259 2237 1317 1162 1705 1086 1771 1286
##  [91] 1645 2695 2723 1110 2793 2355 1861 1976 1247 1474 2112  998 1998 1501 1895
## [106] 1352 2570 2405 1522 1662 1814 2668 1509 2907 1521 1129 2965 2088 1959 1630
## [121] 2884 1749 1591 1184 1994 1034 1491 2266 1422 1633 1181 1019 2203 2615
```

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

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

```
sum(df$correct == 0 & !is.na(df$error)) # einmal fehlt die Fehlerklassifizierung!!
```

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

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

```
## # A tibble: 1 x 32
## # 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
## # ... with 22 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>, PosOr <fct>, group <fct>, PosOr_cont <dbl>
```

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 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
```

Errors

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

```
##
##   1   2   3   4   5   6   7   8   9  99
##  56 188  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 incorrrct trials that were not technical errors per ordinal position?
table(df$PosOr[df$category != "Filler" & df$correct == 0 &
                 df$error != 99])
```

```
##
##   1   2   3   4   5
## 302 333 347 331 394
```

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
## # ... with 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 experimen
                         error == "6" ~
                           "wrong: replacement without connection to the word (word not in the exper
                         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
## # ... with 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)


df %>% filter(is.na(error) & group== "PWA" & session == 2 & correct == "0")
```

```
## # A tibble: 1 x 32
## # 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
## # ... with 22 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>, PosOr <fct>, group <fct>, PosOr_cont <dbl>
```

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 | correct ==1.1 | correct == 1.2 |correct == 1.3 ~ 1,
  correct == 1.4 | correct == 0 ~ 0)) -> df
# Fillers included
df %>% group_by(group, session) %>% dplyr::count(correct_class)
```

```
## # A tibble: 12 x 4
## # Groups:   group, session [6]
##    group   session correct_class     n
##    <fct>   <fct>           <dbl> <int>
##  1 control 1                   0   124
##  2 control 1                   1  2276
##  3 control 2                   0    88
##  4 control 2                   1  2312
##  5 control 3                   0    51
##  6 control 3                   1  2349
##  7 PWA     1                   0   677
##  8 PWA     1                   1  1723
##  9 PWA     2                   0   536
## 10 PWA     2                   1  1864
## 11 PWA     3                   0   469
## 12 PWA     3                   1  1931
```

```
table(df$correct_class)
```

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

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

```
## # A tibble: 12 x 4
## # Groups:   group, session [6]
```

```
##    group   session correct_class     n
##    <fct>   <fct>           <dbl> <int>
##  1 control 1                   0   124
##  2 control 1                   1  2276
##  3 control 2                   0    88
##  4 control 2                   1  2312
##  5 control 3                   0    51
##  6 control 3                   1  2349
##  7 PWA     1                   0   677
##  8 PWA     1                   1  1723
##  9 PWA     2                   0   536
## 10 PWA     2                   1  1864
## 11 PWA     3                   0   469
## 12 PWA     3                   1  1931
```

```
table(df$correct_class[df$category != "Filler"])
```

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

```
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)
```

```
## # 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
## # ... with 14 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)
```

```
## # A tibble: 12 x 4
## # Groups:   group, session [6]
##    group   session error_class     n
##    <fct>   <fct>         <dbl> <int>
##  1 control 1                 0  2323
##  2 control 1                 1    77
##  3 control 2                 0  2343
##  4 control 2                 1    57
##  5 control 3                 0  2367
##  6 control 3                 1    33
##  7 PWA     1                 0  1791
##  8 PWA     1                 1   609
##  9 PWA     2                 0  1895
## 10 PWA     2                 1   505
## 11 PWA     3                 0  1974
## 12 PWA     3                 1   426
```

```
table(df$error_class)
```

```
##
##     0     1
## 12693  1707
```

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

```
## # A tibble: 12 x 4
## # Groups:   group, session [6]
##    group   session error_class     n
##    <fct>   <fct>         <dbl> <int>
##  1 control 1                 0  2323
##  2 control 1                 1    77
##  3 control 2                 0  2343
```

```
##  4 control 2                1    57
##  5 control 3                0  2367
##  6 control 3                1    33
##  7 PWA     1                0  1791
##  8 PWA     1                1   609
##  9 PWA     2                0  1895
## 10 PWA     2                1   505
## 11 PWA     3                0  1974
## 12 PWA     3                1   426
```

```
table(df$error_class[df$category != "Filler"])
```

```
##
##     0     1
## 12693  1707
```

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

REACTION TIMES

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

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

Descriptives

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

```
means_final$increase <- NA
for(k in 1:length(unique(means_final$group))){
  for(i in 1:length(unique(means_final$session))){
    for(j in 2:length(unique(means_final$PosOr))) {
      means_final$increase[means_final$session==unique(means_final$session)[i] &
                             means_final$PosOr==unique(means_final$PosOr)[j] &
                             means_final$group == unique(means_final$group)[k]] <-
       means_final$VOT[means_final$session==unique(means_final$session)[i] &
                             means_final$PosOr==unique(means_final$PosOr)[j] &
                             means_final$group == unique(means_final$group)[k]] -
        means_final$VOT[means_final$session==unique(means_final$session)[i] &
                             means_final$PosOr==unique(means_final$PosOr)[j-1] &
                             means_final$group == unique(means_final$group)[k]]
    }
}}
# means_final

## Calculate overall mean increase per session (weighted)
```

```r
## PWA
mean(means_final$increase[means_final$session==1 & means_final$group == "PWA"], na.rm=T)
```

```
## [1] 27.30054
```

```r
## control
mean(means_final$increase[means_final$session==1 & means_final$group == "control"], na.rm=T)
```

```
## [1] 12.6036
```

```r
means_final$PosOr_effect <- NA
means_final$PosOr_effect[means_final$PosOr==1] <- 1
for(k in 1:length(unique(means_final$group))){
for(i in 1:length(unique(means_final$session))){
  for(j in 2:length(unique(means_final$PosOr))) {
    means_final$PosOr_effect[
      means_final$session==unique(means_final$session)[i] &
        means_final$PosOr==unique(means_final$PosOr)[1] &
        means_final$group == unique(means_final$group)[k]] <-
      means_final$PosOr_effect[
      means_final$session==unique(means_final$session)[i] &
        means_final$PosOr==unique(means_final$PosOr)[1] &
        means_final$group == unique(means_final$group)[k]] +
      means_final$increase[means_final$session==unique(means_final$session)[i] &
                             means_final$PosOr==unique(means_final$PosOr)[j]&
                             means_final$group == unique(means_final$group)[k]]*
     (means_final$N[means_final$session==unique(means_final$session)[i] &
                           means_final$PosOr==unique(means_final$PosOr)[j]&
                             means_final$group == unique(means_final$group)[k]]+
       means_final$N[means_final$session==unique(means_final$session)[i] &
                           means_final$PosOr==unique(means_final$PosOr)[j-1]&
                             means_final$group == unique(means_final$group)[k]])
  }
  means_final$PosOr_effect[means_final$session==unique(means_final$session)[i] &
                             means_final$PosOr==unique(means_final$PosOr)[1]&
                             means_final$group == unique(means_final$group)[k]] <-
    means_final$PosOr_effect[means_final$session==unique(means_final$session)[i] &
                               means_final$PosOr==unique(means_final$PosOr)[1]&
                             means_final$group == unique(means_final$group)[k]]/
    (sum(means_final$N[means_final$session==unique(means_final$session)[i]&
                             means_final$group == unique(means_final$group)[k]])+
       sum(means_final$N[means_final$session==unique(means_final$session)[i] &
                           (means_final$PosOr=="2" |
                              means_final$PosOr=="3" |
                              means_final$PosOr=="4")&
                             means_final$group == unique(means_final$group)[k]]))
}}
means_final
```

```
##      group session PosOr   N      VOT       sd       se       ci   increase
## 1  control       1     1 461 1297.252 386.6087 18.00616 35.38453         NA
## 2  control       1     2 456 1278.075 368.6706 17.26458 33.92821 -19.176859
```

```
## 3  control       1     3 446 1352.942 419.3224 19.85549 39.02218  74.867353
## 4  control       1     4 457 1329.444 375.5740 17.56861 34.52548 -23.498158
## 5  control       1     5 456 1347.666 395.4900 18.52052 36.39636  18.222054
## 6  control       2     1 466 1204.941 327.2233 15.15832 29.78730         NA
## 7  control       2     2 462 1244.987 364.0399 16.93667 33.28264  40.046543
## 8  control       2     3 464 1213.458 289.9285 13.45959 26.44945 -31.529745
## 9  control       2     4 466 1244.242 296.8703 13.75225 27.02425  30.784342
## 10 control       2     5 454 1266.531 329.4120 15.46007 30.38236  22.289241
## 11 control       3     1 469 1199.645 323.1567 14.92199 29.32239         NA
## 12 control       3     2 471 1173.122 282.9187 13.03621 25.61646 -26.522941
## 13 control       3     3 467 1208.308 305.2131 14.12358 27.75378  35.185898
## 14 control       3     4 470 1232.751 306.1228 14.12039 27.74706  24.443562
## 15 control       3     5 472 1263.283 349.0940 16.06835 31.57453  30.531484
## 16     PWA       1     1 362 1280.805 463.4011 24.35582 47.89711         NA
## 17     PWA       1     2 345 1352.300 518.0623 27.89154 54.85942  71.495131
## 18     PWA       1     3 337 1374.660 515.0242 28.05516 55.18588  22.360233
## 19     PWA       1     4 357 1334.771 468.7956 24.81129 48.79513 -39.889411
## 20     PWA       1     5 322 1390.007 538.3805 30.00277 59.02691  55.236193
## 21     PWA       2     1 375 1168.998 438.9850 22.66909 44.57485         NA
## 22     PWA       2     2 386 1188.463 443.3652 22.56670 44.36939  19.465021
## 23     PWA       2     3 370 1212.971 435.7079 22.65138 44.54198  24.508033
## 24     PWA       2     4 378 1245.738 457.2034 23.51600 46.23896  32.766688
## 25     PWA       2     5 355 1308.230 496.4997 26.35147 51.82511  62.491968
## 26     PWA       3     1 395 1144.157 373.7086 18.80332 36.96739         NA
## 27     PWA       3     2 386 1183.630 457.0922 23.26538 45.74310  39.473388
## 28     PWA       3     3 396 1224.247 442.1547 22.21911 43.68250  40.616714
## 29     PWA       3     4 384 1223.097 437.4259 22.32230 43.88959  -1.149424
## 30     PWA       3     5 370 1273.044 469.5947 24.41307 48.00619  49.946521
##    PosOr_effect
## 1      12.47980
## 2             NA
## 3             NA
## 4             NA
## 5             NA
## 6      15.41663
## 7             NA
## 8             NA
## 9             NA
## 10            NA
## 11     15.90047
## 12            NA
## 13            NA
## 14            NA
## 15            NA
## 16     27.37866
## 17            NA
## 18            NA
## 19            NA
## 20            NA
## 21     34.57573
## 22            NA
## 23            NA
## 24            NA
## 25            NA
```

```
## 26     32.08108
## 27           NA
## 28           NA
## 29           NA
## 30           NA
```

Types of correctness classification

```
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
## # ... with 26 more rows
```

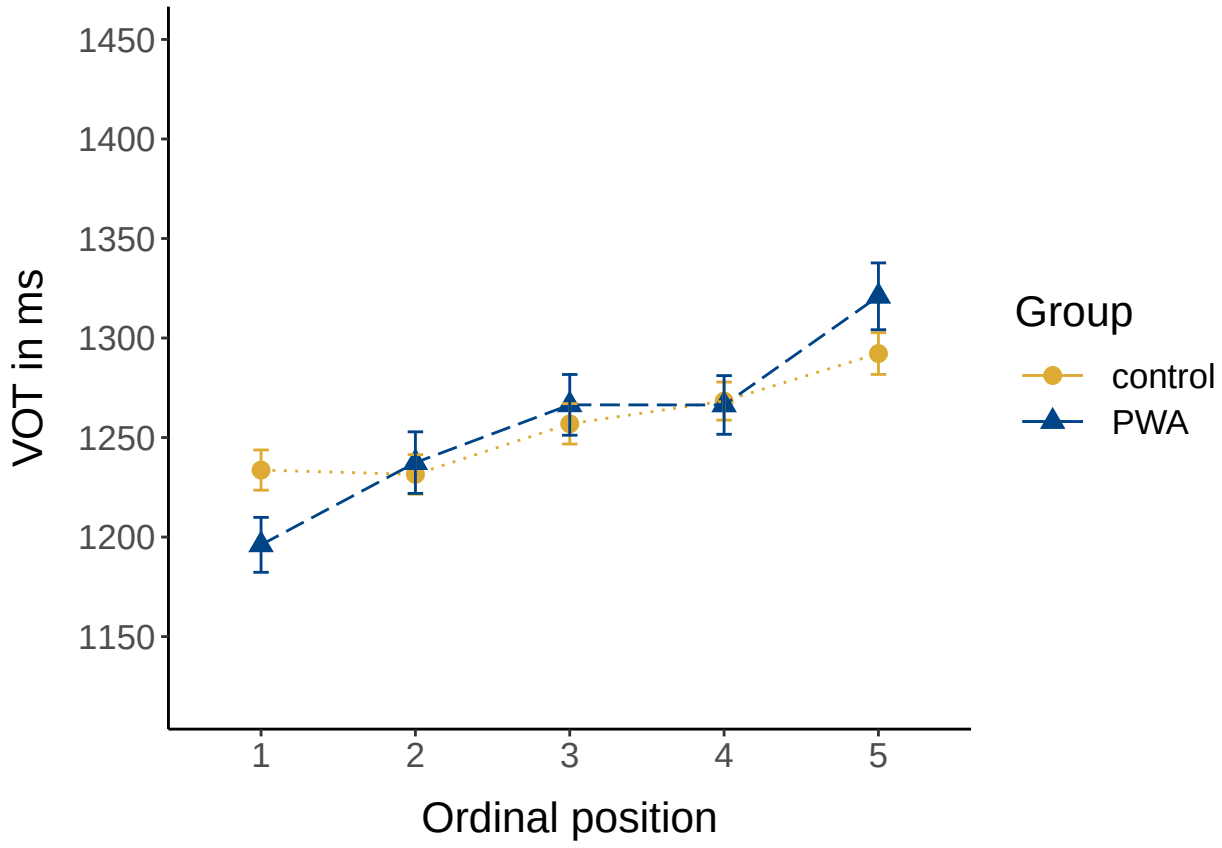
Plotting

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 <- "#DDAA33"
PWA_color <- "#004488"
```

RTs across session, by ordinal position and group Line graph (only correct trials, without fillers). 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")))
```



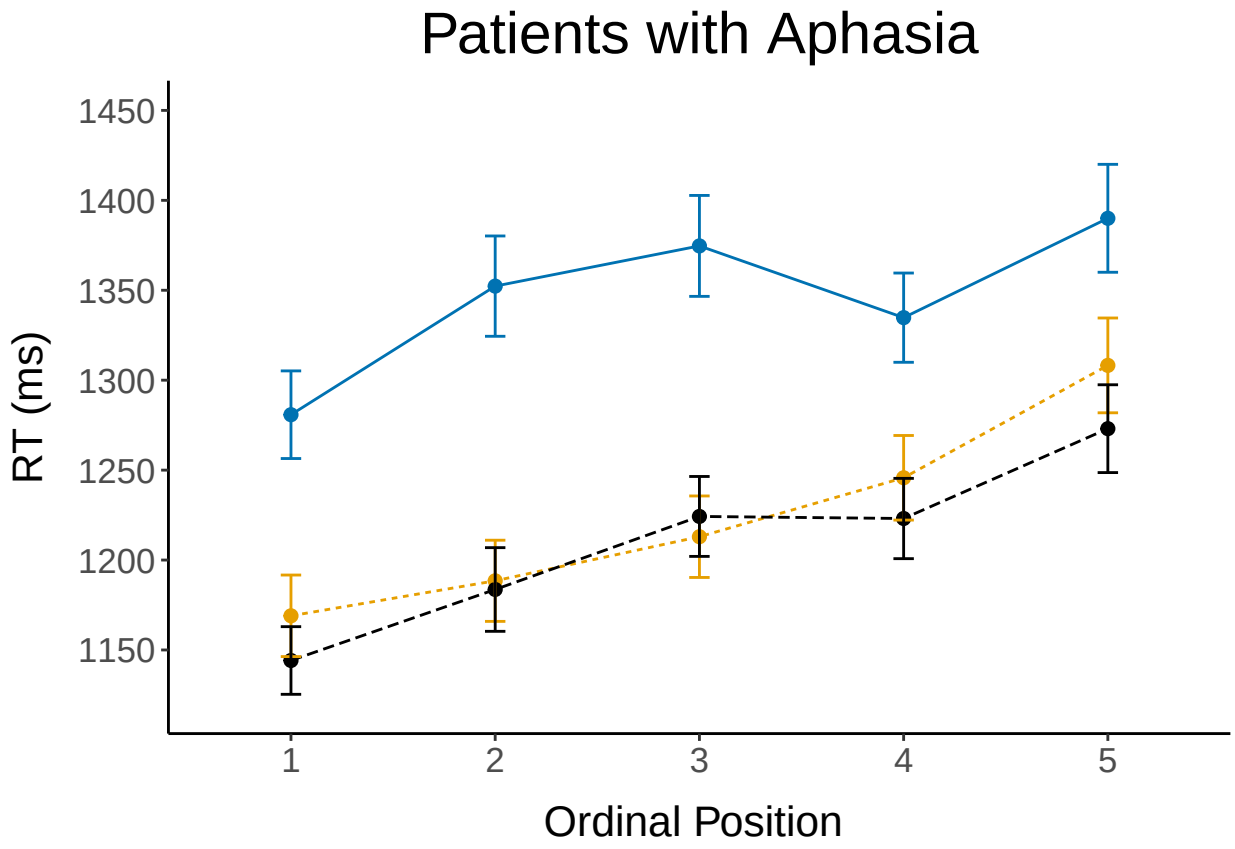

```
filename <- "CSI_online_aphasia_spoken_plot_rt_across_sessions.pdf"
ggsave(plot_vot, filename =
         here::here("results", "figures", filename),
       width = 18, height = 13, units = "cm",
       dpi = 300, device = cairo_pdf)
```

```
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") + #+
  # annotate(geom="text", x=1.5, y=1330, label="n = 30",
  #          color="black", size = 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="none")+
   guides(color=guide_legend(
     override.aes=list(linetype=override.linetype)))+
    scale_linetype(guide="none"))
```

RTs by Group, session, and ordinal position

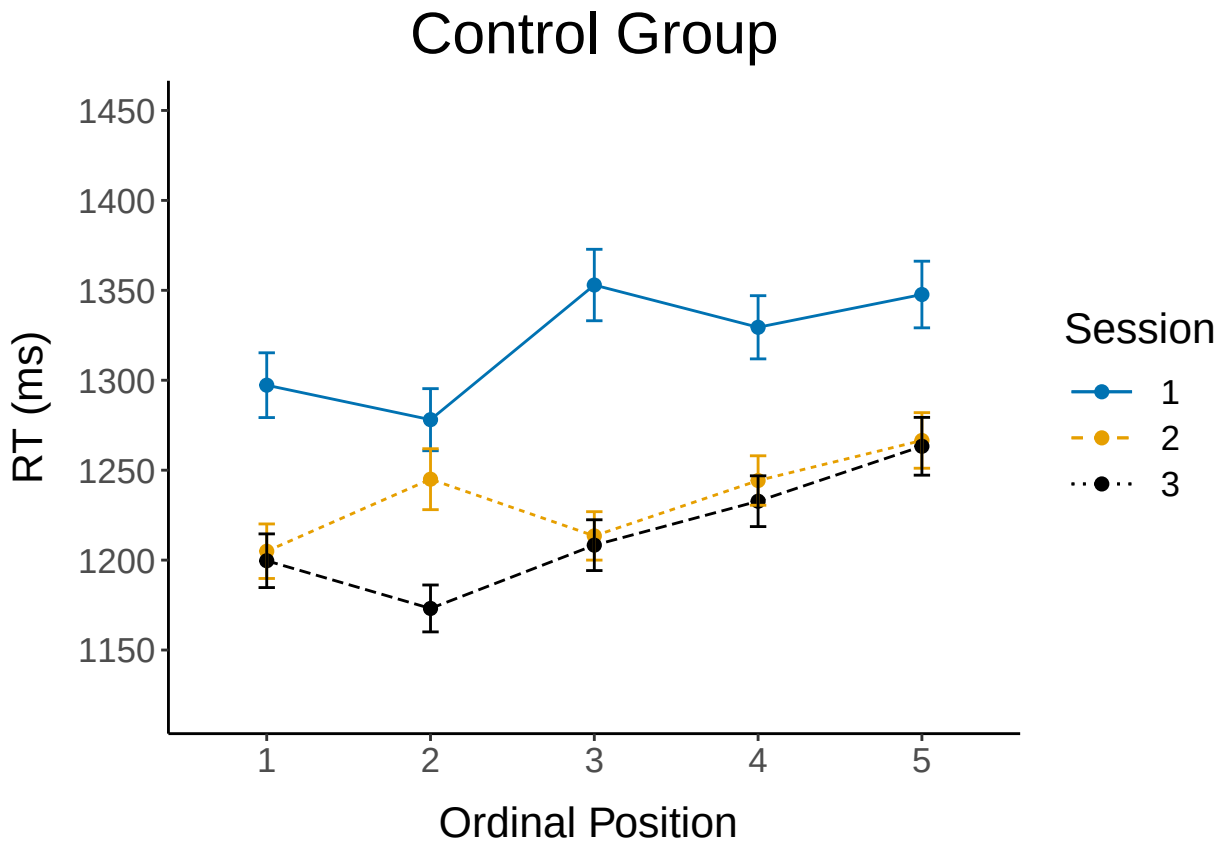
```
## Scale for 'linetype' is already present. Adding another scale for 'linetype',
## which will replace the existing scale.
```




```
(plot_rt_repetition_control <- means_final %>% filter(group=="control") %>%
    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)) +
    #breaks = c(1100, 1150, 1200, 1250, 1300, 1350)) +
    labs(x="Ordinal Position ",y ="RT (ms)", colour="Session", linetype="Session",
         title = "Control Group") + #+
  # annotate(geom="text", x=1.5, y=1330, label="n = 30",
  #          color="black", size = 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"))+
   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")
```

Normalized boxplot

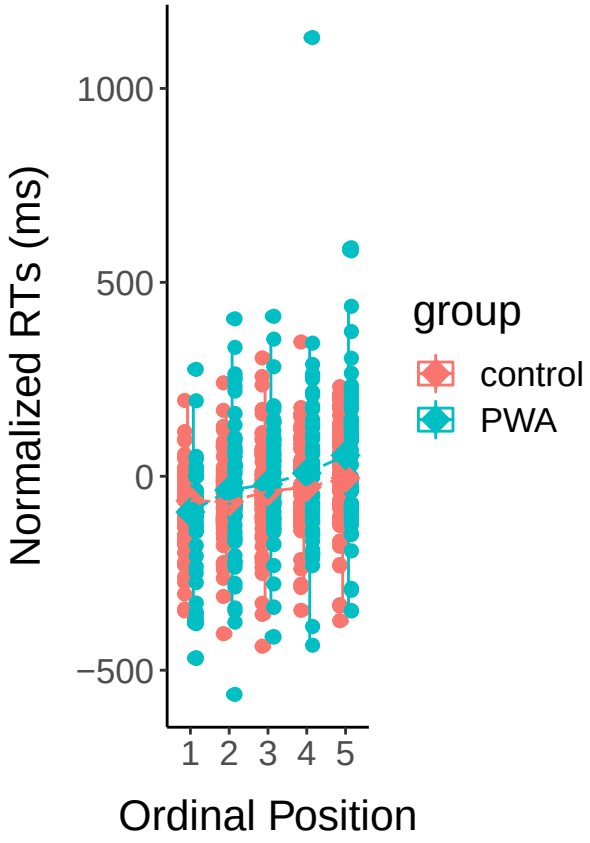
```
## Warning in qt(conf.interval/2 + 0.5, datac$N - 1): NaNs wurden erzeugt
```

```
(means_subject <- means_subject %>%
  group_by(subject) %>%
  dplyr::mutate(VOT_norm = VOT - first(VOT)))
```

```
## # A tibble: 600 x 10
## # Groups:   subject [40]
##    group   subject session PosOr     N   VOT    sd    se    ci VOT_norm
##    <fct>   <fct>   <fct>   <fct> <dbl> <dbl> <dbl> <dbl> <dbl>    <dbl>
##  1 control 201     1       1        24 1435.  390.  79.6 165.       0
##  2 control 201     1       2        23 1339.  278.  57.9 120.     -96.1
##  3 control 201     1       3        24 1529.  477.  97.4 202.      93.8
##  4 control 201     1       4        24 1318.  211.  43.0  89.0  -117.
##  5 control 201     1       5        24 1440.  343.  70.0 145.      4.83
##  6 control 201     2       1        24 1093.  149.  30.5  63.1  -342.
##  7 control 201     2       2        24 1302.  452.  92.3 191.   -134.
##  8 control 201     2       3        24 1283.  341.  69.7 144.   -152.
##  9 control 201     2       4        24 1295.  221.  45.2  93.4  -141.
## 10 control 201     2       5        23 1204.  258.  53.7 111.   -231.
## # ... with 590 more rows
```

```
(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))
```

Export plot grid

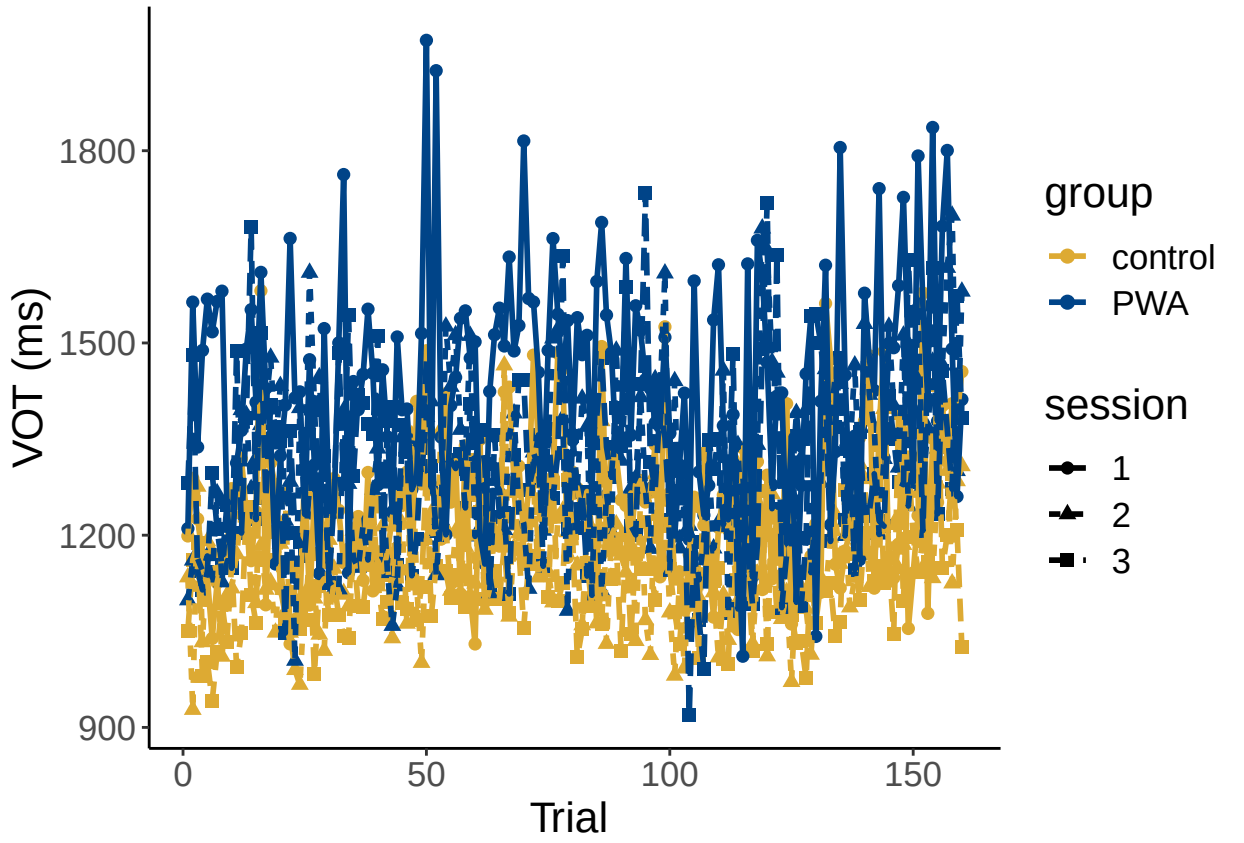
```
# cowplot::plot_grid(plot_rt, boxplot,
#           nrow = 1, labels = c("A", "B"), label_fontfamily = "Arial") %>%
#   ggsave(filename = here::here("results", "figures",
#                                "CSI_online_aphasia_typing_RTs_and_normalized_RTs"),
#          width = 18, height = 13, units = "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)))#+
```




```
  # annotate(geom="text", x=20, y=1800, label="n = 19",
  #          color="black", size = 8))

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: GLMM (Gamma distribution) with ordinal position as a continuous predictor

Contrast coding *Center predictor variable* Across both groups

```
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
```

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

```
## [1] -0.00000000000000002197888
```

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

For PWA only

```
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
```

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

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

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

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
```

```
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"
```

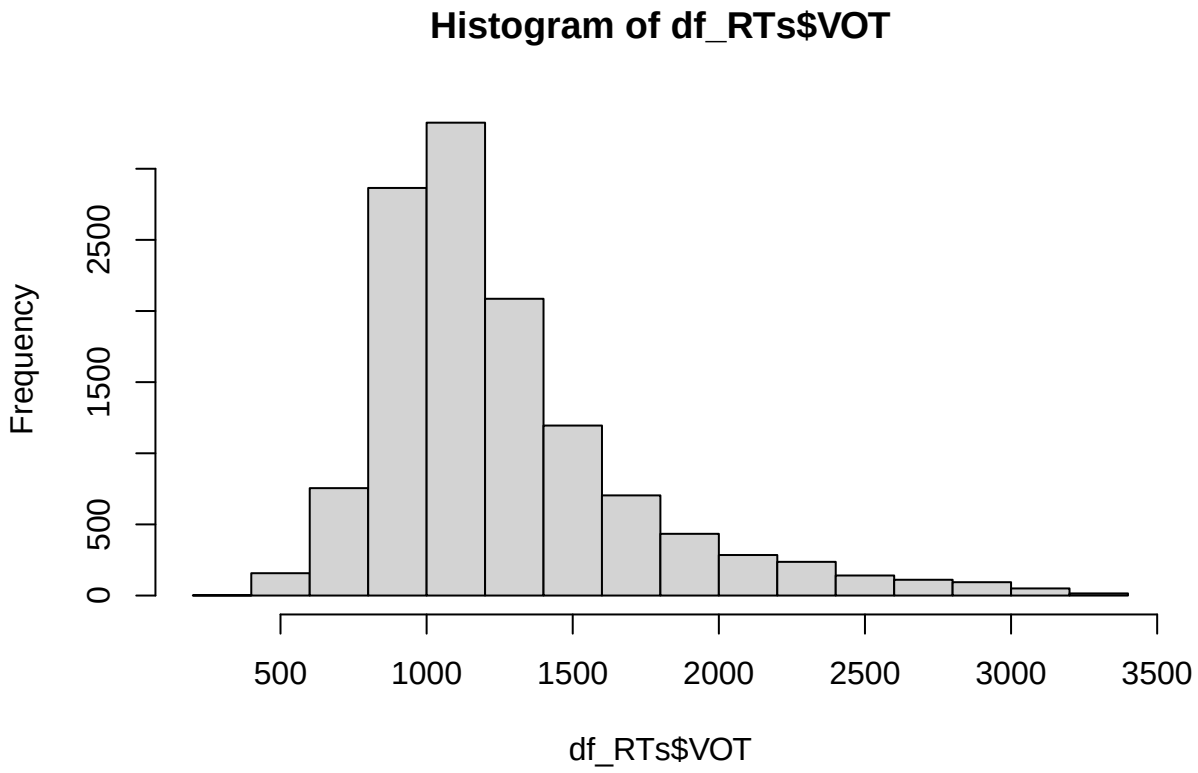
```
levels(df_RTs_PWA$group)
```

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

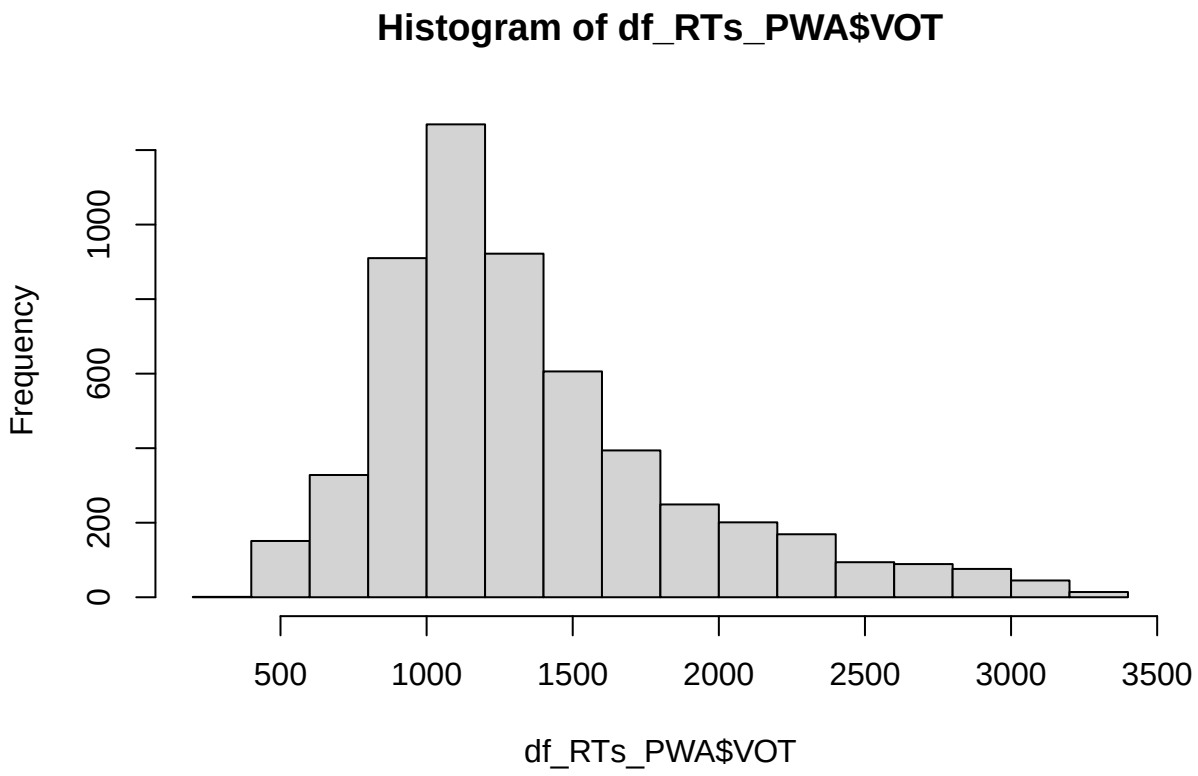
Check distribution of data Are the data normally distributed or does a gamma distribution fit the data better?

Histogram of the reaction time data

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




```
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)
```

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)
```

```
## Lade nötiges Paket: MASS
```

```
##
## Attache Paket: 'MASS'
```

```
## Das folgende Objekt ist maskiert 'package:dplyr':
##
##     select
```

```
## Lade nötiges Paket: survival
```

```
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)
```

```
##
## Attache Paket: 'actuar'
```

```
## Die folgenden Objekte sind maskiert von 'package:stats':
##
##     sd, var
```

```
## Das folgende Objekt ist maskiert 'package:grDevices':
##
##     cm
```

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

```
## 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), metho
```

```
## 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 wurden erzeugt
```

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

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

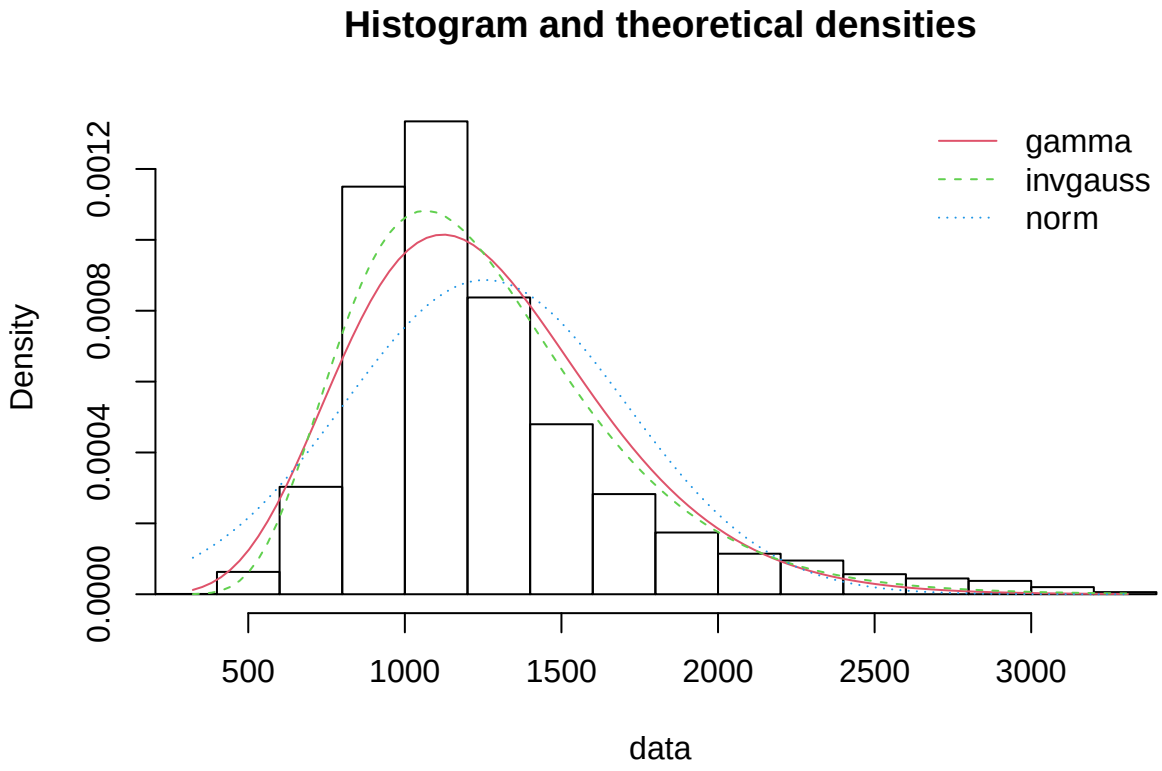
```
summary(fit.invgauss_PWA)
```

```
## 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
```

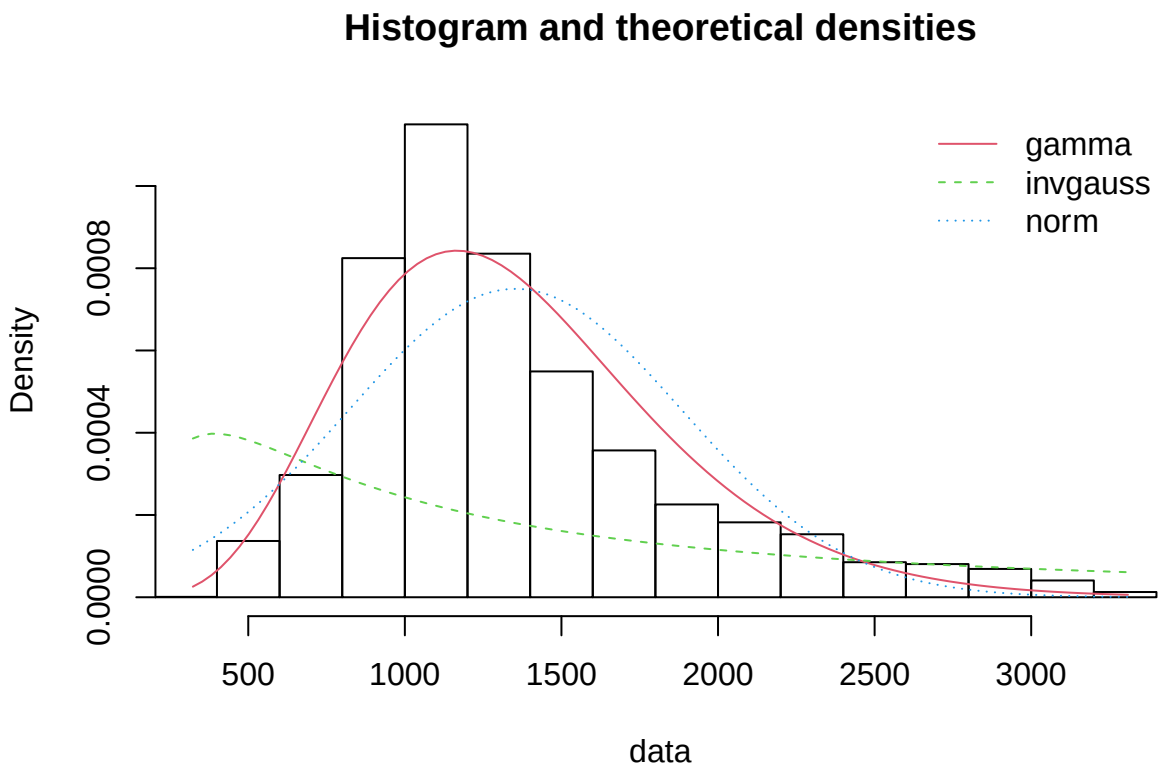
```
#plot(fit.invgauss_PWA)
```

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

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

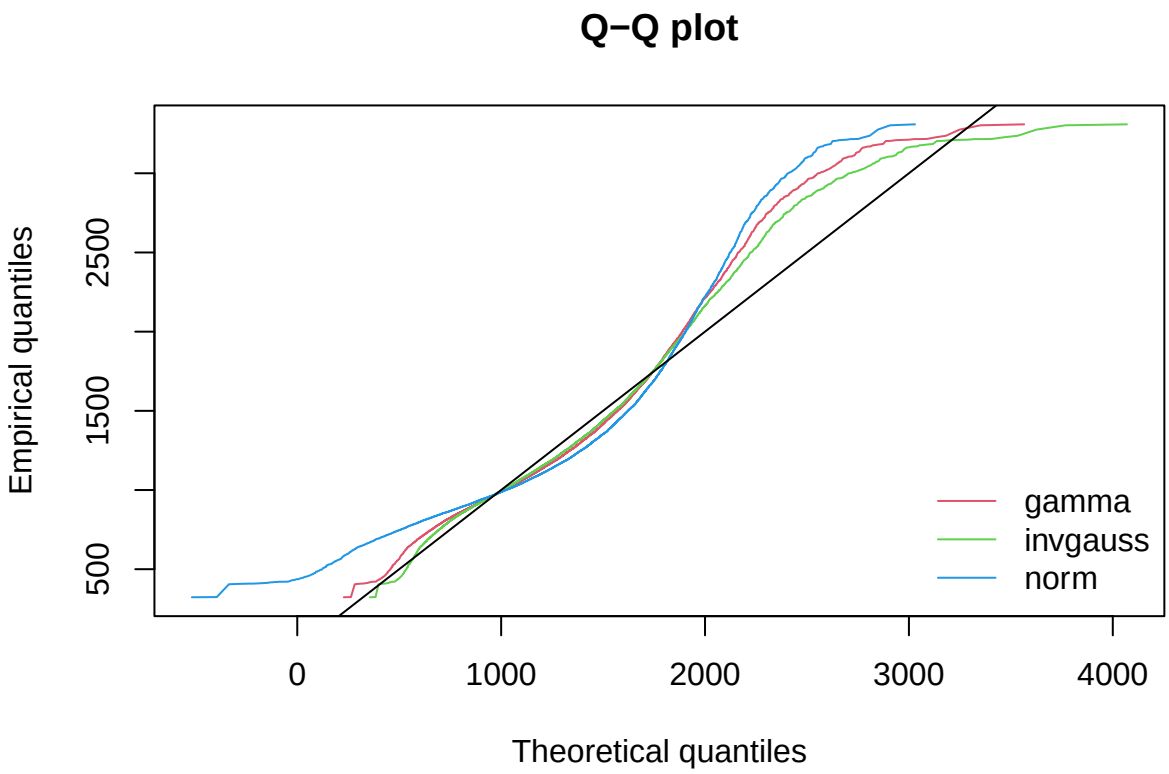



```
denscomp(list(fit.gamma_PWA, fit.invgauss_PWA, fit.normal_PWA))
```

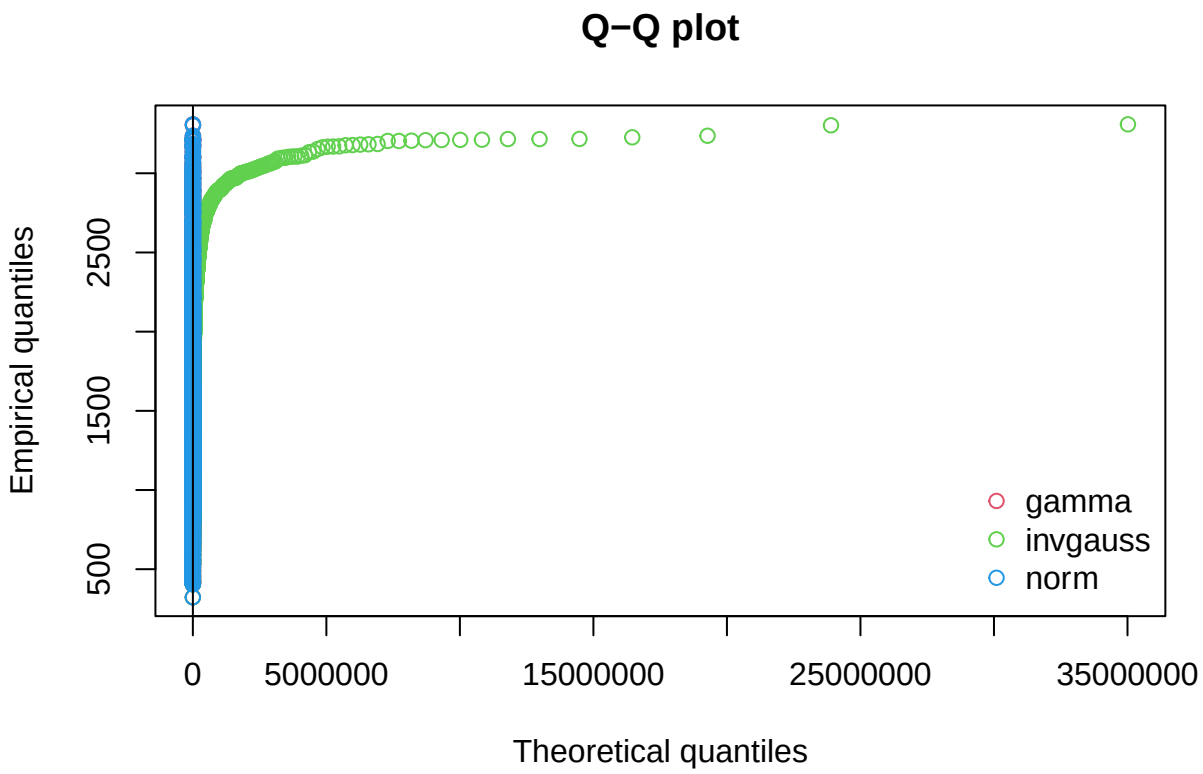


Visually compare fit of both distributions in Q-Q-plots

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




```
qqcomp(list(fit.gamma_PWA, fit.invgauss_PWA, fit.normal_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. Therefor we will use the Gamma distribution.

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)
```

```
# 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"))
```

Main 1: 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)))
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## unable to evaluate scaled gradient
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge: degenerate Hessian with 2 negative eigenvalues
```

```
didLmerConverge(m1)
```

```
##  The relative maximum gradient of 0.000321 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)
```

```
## Warning in vcov.merMod(object, use.hessian = use.hessian): variance-covariance matrix computed from
## not positive definite or contains NA values: falling back to var-cov estimated from RX
```

```
## Warning in vcov.merMod(object, correlation = correlation, sigm = sig): variance-covariance matrix co
## not positive definite or contains NA values: falling back to var-cov estimated from RX
```

```
## 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  1388.0698  37.2568
##  category.1 re2.PosOr.cont_by_session2  1746.8806  41.7957
##  category.2 re2.session3                2080.6078  45.6137
##  category.3 re2.session2                1856.3845  43.0858
##  category.4 re2.PosOr.cont               234.6952  15.3198
##  category.5 (Intercept)                 5374.9183  73.3138
##  subject    re1.PosOr.cont_by_session3   600.7206  24.5096
##  subject.1  re1.PosOr.cont_by_session2   965.2898  31.0691
##  subject.2  re1.session3                3742.7339  61.1779
##  subject.3  re1.session2                2955.9548  54.3687
##  subject.4  re1.PosOr.cont               361.3714  19.0098
```

```
##  subject.5  (Intercept)               20917.5451 144.6290
##  Residual                                 0.1001   0.3164
## Number of obs: 5518, groups:  category, 24; subject, 20
##
## Fixed effects:
##                     Estimate Std. Error t value             Pr(>|z|)
## (Intercept)         1545.138     36.384  42.467 < 0.0000000000000002 ***
## PosOr.cont            31.556      6.806   4.637      0.0000035423972 ***
## session2            -112.883     21.078  -5.355      0.0000000853240 ***
## session3            -144.044     22.181  -6.494      0.0000000000836 ***
## PosOr.cont:session2   14.552     15.059   0.966                0.334
## PosOr.cont:session3    9.690     13.695   0.708                0.479
## ---
## 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   0.009
## session2    -0.010  0.003
## session3    -0.016 -0.001  0.240
## PsOr.cnt:s2  0.000 -0.037  0.026  0.008
## PsOr.cnt:s3 -0.001 -0.052  0.009  0.023  0.259
## optimizer (bobyqa) convergence code: 0 (OK)
## unable to evaluate scaled gradient
## Model failed to converge: degenerate  Hessian with 2 negative eigenvalues
```

```
anova(m1)
```

```
## Analysis of Variance Table
##                     npar Sum Sq Mean Sq F value
## PosOr.cont             1 2.2285 2.22855 22.2598
## session                2 5.7971 2.89854 28.9520
## PosOr.cont:session     2 0.1159 0.05794  0.5788
```

```
# 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 only
          pred.labels = c("(Intercept)", "Ordinal Position",
                          "Session 2 vs 1",
                          "Session 3 vs 1", "Ord.Pos. x Session2-1",
                          "Ord.Pos. x Session3-1"),
          dv.labels = "Vocal Onset Time",
          #string.pred = "",
          string.stat = "t-Value",
          file = here::here("results", "tables", "CSI_online_aphasia_PWA_glmm_cont.html"))
```

```
## Warning in vcov.merMod(object, use.hessian = use.hessian): variance-covariance matrix computed from
## not positive definite or contains NA values: falling back to var-cov estimated from RX
```

```
## Warning in vcov.merMod(object, use.hessian = use.hessian): variance-covariance matrix computed from
## not positive definite or contains NA values: falling back to var-cov estimated from RX
```

```
## Warning in vcov.merMod(object, use.hessian = use.hessian): variance-covariance matrix computed from
## not positive definite or contains NA values: falling back to var-cov estimated from RX

## Warning in vcov.merMod(object, correlation = correlation, sigm = sig): variance-covariance matrix co
## not positive definite or contains NA values: falling back to var-cov estimated from RX

## Warning in vcov.merMod(object, use.hessian = use.hessian): variance-covariance matrix computed from
## not positive definite or contains NA values: falling back to var-cov estimated from RX

## Warning in vcov.merMod(object, correlation = correlation, sigm = sig): variance-covariance matrix co
## not positive definite or contains NA values: falling back to var-cov estimated from RX

## Warning in vcov.merMod(object, use.hessian = use.hessian): variance-covariance matrix computed from
## not positive definite or contains NA values: falling back to var-cov estimated from RX

## Warning in vcov.merMod(object, correlation = correlation, sigm = sig): variance-covariance matrix co
## not positive definite or contains NA values: falling back to var-cov estimated from RX
```

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

Vocal Onset Time

Predictors

Estimates

CI

t-Value

p

(Intercept)

1545.14

1473.81 – 1616.47

42.47

<0.001

Ordinal Position

31.56

18.21 – 44.90

4.64

<0.001

Session 2 vs 1

-112.88

-154.20 – -71.56

-5.36

<0.001

Session 3 vs 1

-144.04

-187.53 – -100.56

-6.49

<0.001

Ord.Pos. x Session2-1

14.55

-14.97 – 44.07

0.97

0.334

Ord.Pos. x Session3-1

9.69

-17.16 – 36.54

0.71

0.479

N subject

20

N category

24

Observations

5518

Exloratoy follow-up with polynomial contrasts Ordinal position effect is only a trend in session 1. Does another than a linear trend describe the data better?

```
x <- df_RTs_PWA %>% filter(session == "1")
x$PosOr <- as.factor(x$PosOr)
levels(x$PosOr)
```

```
## [1] "1" "2" "3" "4" "5"
```

```
contrasts(x$PosOr) <- contr.poly(5)
contrasts(x$PosOr)
```

```
##           .L         .Q                         .C         ^4
## 1 -0.6324555  0.5345225 -0.31622776601683810770990  0.1195229
## 2 -0.3162278 -0.2672612  0.63245553203367621541990 -0.4780914
## 3  0.0000000 -0.5345225 -0.00000000000000040959720  0.7171372
## 4  0.3162278 -0.2672612 -0.63245553203367543826370 -0.4780914
## 5  0.6324555  0.5345225  0.31622776601683783015420  0.1195229
```

```
# m1_poly <- glmer(VOT ~ PosOr +
#                  (PosOr|subject) +
#                (PosOr|category),
#               data = x,
#              family =Gamma(link ="identity"),
#              control=glmerControl(optimizer = "bobyqa",
#                                   optCtrl = list(maxfun = 2e5)))
# m1_poly <- afex::lmer_alt(VOT ~ PosOr +
#                  (PosOr||subject) +
#                (PosOr||category),
#               data = x,
#              family =Gamma(link ="identity"),
#              control=glmerControl(optimizer = "bobyqa",
#                                   optCtrl = list(maxfun = 2e5)))
# didLmerConverge(m1_poly)
# rePCA(m1_poly)
# summary(m1_poly)
# m1_poly <- afex::lmer_alt(VOT ~ PosOr +
#                  (1|subject) +
#                (PosOr||category),
#               data = x,
#              family =Gamma(link ="identity"),
#              control=glmerControl(optimizer = "bobyqa",
#                                   optCtrl = list(maxfun = 2e5)))
m1_poly <- glmer(VOT ~ PosOr +
                  (1|subject) +
                (1|category),
              data = x,
             family =Gamma(link ="identity"),
             control=glmerControl(optimizer = "bobyqa",
                                  optCtrl = list(maxfun = 2e5)))
summary(m1_poly)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: Gamma  ( identity )
## Formula: VOT ~ PosOr + (1 | subject) + (1 | category)
##    Data: x
## Control: glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 200000))
##
##      AIC      BIC   logLik deviance df.resid
##  25711.3  25754.9 -12847.7  25695.3     1715
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.7030 -0.6496 -0.2616  0.3853  6.2919
##
## Random effects:
##  Groups   Name        Variance   Std.Dev.
##  category (Intercept) 10337.6692 101.6743
##  subject  (Intercept) 34984.6830 187.0419
##  Residual                 0.1173   0.3425
## Number of obs: 1723, groups:  category, 24; subject, 20
```

```
##
## Fixed effects:
##             Estimate Std. Error t value             Pr(>|z|)
## (Intercept) 1658.824     15.504 106.992 < 0.0000000000000002 ***
## PosOr.L       44.069     12.603   3.497             0.000471 ***
## PosOr.Q       -4.651     11.108  -0.419             0.675433
## PosOr.C       48.397     10.437   4.637           0.00000354 ***
## PosOr^4       41.526     10.991   3.778             0.000158 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##         (Intr) PsOr.L PsOr.Q PsOr.C
## PosOr.L  0.171
## PosOr.Q  0.035  0.018
## PosOr.C  0.098 -0.032 -0.015
## PosOr^4  0.022 -0.173 -0.021 -0.057
```

```
saveRDS(m1_poly, file = here::here("results", "tables",
                           "CSI_online_aphasia_PWA_session1_glmm_poly_contrast.RDS"))
tab_model(m1_poly,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 (polynomial contrasts)in S
          dv.labels = "Vocal Onset Time",
          #string.pred = "",
          string.stat = "t-Value",
          file = here::here("results", "tables", "CSI_online_aphasia_PWA_session1_glmm_poly_contrast.ht
```

GLMM (Gamma distribution) of VOTs Predicted by Ordinal Position (polynomial contrasts)in Session 1, PWA only

Vocal Onset Time

Predictors

Estimates

CI

t-Value

p

(Intercept)

1658.82

1628.42 – 1689.23

106.99

<0.001

PosOr L

44.07

19.35 – 68.79

3.50

<0.001

PosOr Q

-4.65

-26.44 – 17.14

-0.42

0.675

PosOr C

48.40

27.93 – 68.87

4.64

<0.001

PosOr^4

41.53

19.97 – 63.08

3.78

<0.001

N subject

20

N category

24

Observations

1723

Secondary analysis with factor 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 0.0000976 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)                              11534.61991 107.3993
##  subject.1   re1.PosOr.cont                             230.18246  15.1718
##  subject.2   re1.session2                              3987.79084  63.1490
##  subject.3   re1.session3                              3759.76860  61.3170
##  subject.4   re1.PosOr.cont_by_session2                 799.74318  28.2797
##  subject.5   re1.PosOr.cont_by_session3                 337.51050  18.3715
##  category    (Intercept)                              3218.45026  56.7314
##  category.1  re2.PosOr.cont                             107.70336  10.3780
##  category.2  re2.session2                               675.13294  25.9833
##  category.3  re2.session3                               671.60568  25.9154
##  category.4  re2.group2.1                              1350.47307  36.7488
##  category.5  re2.PosOr.cont_by_session2                 201.20951  14.1848
##  category.6  re2.PosOr.cont_by_session3                 294.13083  17.1502
##  category.7  re2.PosOr.cont_by_group2.1                 180.27911  13.4268
##  category.8  re2.session2_by_group2.1                  3093.03280  55.6150
##  category.9  re2.session3_by_group2.1                  2663.60056  51.6101
##  category.10 re2.PosOr.cont_by_session2_by_group2.1  2197.50571  46.8776
##  category.11 re2.PosOr.cont_by_session3_by_group2.1  1097.44406  33.1277
##  Residual                                               0.08145   0.2854
## Number of obs: 12455, groups:  subject, 40; category, 24
##
## Fixed effects:
##                              Estimate Std. Error t value             Pr(>|z|)
## (Intercept)                  1390.761      3.066 453.562 < 0.0000000000000002
## PosOr.cont                     22.805      2.589   8.808 < 0.0000000000000002
## session2                      -98.052      3.044 -32.207 < 0.0000000000000002
## session3                     -125.370      5.071 -24.724 < 0.0000000000000002
## group2-1                      338.061      3.110 108.715 < 0.0000000000000002
## PosOr.cont:session2             6.166      2.775   2.222              0.02625
## PosOr.cont:session3             5.635      2.726   2.067              0.03870
## PosOr.cont:group2-1            15.688      2.184   7.181 0.00000000000689930
## session2:group2-1             -30.703      2.588 -11.865 < 0.0000000000000002
## session3:group2-1             -42.602      2.243 -18.991 < 0.0000000000000002
## PosOr.cont:session2:group2-1   17.959      2.232   8.045 0.00000000000000866
## PosOr.cont:session3:group2-1    6.851      2.392   2.865              0.00418
##
## (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  -0.113
## session2    -0.223 -0.159
## session3     0.492 -0.079 -0.469
## group2-1    -0.129  0.080  0.075 -0.147
## PsOr.cnt:s2  0.107 -0.055 -0.066  0.166 -0.156
## PsOr.cnt:s3  0.241 -0.010 -0.232  0.342  0.123 -0.085
## PsOr.cn:2-1  0.045  0.094 -0.037 -0.087  0.076 -0.050  0.002
## sssn2:gr2-1  0.137 -0.156 -0.062  0.078 -0.102 -0.003  0.068 -0.113
## sssn3:gr2-1  0.057  0.023 -0.190  0.300  0.132 -0.047  0.125 -0.053 -0.024
## PsOr.:2:2-1 -0.126  0.072  0.178 -0.181 -0.004  0.095 -0.198  0.050 -0.067
## PsOr.:3:2-1  0.035 -0.069 -0.002  0.143 -0.172  0.062 -0.162 -0.179  0.049
##             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 -0.051
## PsOr.:3:2-1  0.099 -0.005
```

```
anova(m2)
```

```
## Analysis of Variance Table
##                           npar Sum Sq Mean Sq F value
## PosOr.cont                   1 2.4994  2.4994 30.6842
## session                      2 8.7006  4.3503 53.4079
## group                        1 7.2714  7.2714 89.2695
## PosOr.cont:session           2 0.0633  0.0316  0.3886
## PosOr.cont:group             1 0.3947  0.3947  4.8463
## session:group                2 0.2507  0.1254  1.5391
## PosOr.cont:session:group     2 0.0888  0.0444  0.5451
```

```r
# 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",
          pred.labels = c("(Intercept)", "Ordinal Position",
                          "Session 2 vs 1",
                          "Session 3 vs 1",
                          "Group (PWA-Control)",
                          "Ord.Pos. x Session2-1",
                          "Ord.Pos. x Session3-1",
                          "Ord.Pos. x Group",
                          "Session 2-1 x Group",
                          "Session 3-1 x Group",
                          "Ord.Pos. x Session 2-1 x Group",
                          "Ord.Pos. x Session 3-1 x 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)

1390.76

1384.75 – 1396.77

453.56

<0.001

Ordinal Position

22.80

17.73 – 27.88

8.81

<0.001

Session 2 vs 1

-98.05

-104.02 – -92.08

-32.21

<0.001

Session 3 vs 1

-125.37

-135.31 – -115.43

-24.72

<0.001

Group (PWA-Control)

338.06

331.97 – 344.16

108.72

<0.001

Ord.Pos. x Session2-1

6.17

0.73 – 11.60

2.22

0.026

Ord.Pos. x Session3-1

5.63

0.29 – 10.98

2.07

0.039

Ord.Pos. x Group

15.69

11.41 – 19.97

7.18

<0.001

Session 2-1 x Group

-30.70

-35.77 – -25.63

-11.87

<0.001

Session 3-1 x Group

-42.60

-47.00 – -38.21

-18.99

<0.001

Ord.Pos. x Session 2-1 x Group

17.96

13.58 – 22.33

8.04

<0.001

Ord.Pos. x Session 3-1 x Group

6.85

2.16 – 11.54

2.86

0.004

N subject

40

N category

24

Observations

12455

```
m2_nested1 <- afex::lmer_alt(VOT ~ group/(PosOr.cont*session) +
                (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_nested1)
```

Follow-up: Nested model

```
##  The relative maximum gradient of 0.000166 is less than our 0.001 criterion.
##  You can safely ignore any warnings about a claimed convergence failure.
```

```
summary(m2_nested1)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: Gamma  ( identity )
## Formula:
## VOT ~ group/(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.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)                              11534.85697 107.4005
##  subject.1   re1.PosOr.cont                             230.18078  15.1717
##  subject.2   re1.session2                              3987.88094  63.1497
##  subject.3   re1.session3                              3759.79911  61.3172
##  subject.4   re1.PosOr.cont_by_session2                 799.74202  28.2797
##  subject.5   re1.PosOr.cont_by_session3                 337.49125  18.3709
##  category    (Intercept)                              3218.39802  56.7309
##  category.1  re2.PosOr.cont                            107.70371  10.3780
##  category.2  re2.session2                              675.09636  25.9826
##  category.3  re2.session3                              671.59755  25.9152
##  category.4  re2.group2.1                             1350.41903  36.7480
##  category.5  re2.PosOr.cont_by_session2                201.21027  14.1849
##  category.6  re2.PosOr.cont_by_session3                294.13884  17.1505
##  category.7  re2.PosOr.cont_by_group2.1                180.27836  13.4268
##  category.8  re2.session2_by_group2.1                 3093.25531  55.6170
##  category.9  re2.session3_by_group2.1                 2663.74031  51.6114
##  category.10 re2.PosOr.cont_by_session2_by_group2.1   2197.54909  46.8780
##  category.11 re2.PosOr.cont_by_session3_by_group2.1   1097.52308  33.1289
##  Residual                                               0.08145   0.2854
## Number of obs: 12455, groups:  subject, 40; category, 24
##
## Fixed effects:
##                                 Estimate Std. Error t value
## (Intercept)                     1390.793     20.911  66.510
## group2-1                         338.029     35.614   9.492
## groupcontrol:PosOr.cont           14.961      5.091   2.939
## groupPWA:PosOr.cont               30.648      5.696   5.380
## groupcontrol:session2            -82.696     18.989  -4.355
## groupPWA:session2               -113.407     21.181  -5.354
## groupcontrol:session3           -104.064     18.495  -5.627
## groupPWA:session3               -146.680     20.631  -7.110
## groupcontrol:PosOr.cont:session2  -2.812     11.002  -0.256
## groupPWA:PosOr.cont:session2      15.146     12.526   1.209
## groupcontrol:PosOr.cont:session3   2.208      9.420   0.234
## groupPWA:PosOr.cont:session3       9.061     10.933   0.829
##                                        Pr(>|z|)
## (Intercept)                     < 0.0000000000000002 ***
## group2-1                        < 0.0000000000000002 ***
## groupcontrol:PosOr.cont                    0.0033 **
```

```
## groupPWA:PosOr.cont                  0.00000007427945 ***
## groupcontrol:session2                0.00001330259837 ***
## groupPWA:session2                    0.00000008589352 ***
## groupcontrol:session3                0.00000001836973 ***
## groupPWA:session3                    0.00000000000116 ***
## groupcontrol:PosOr.cont:session2              0.7982
## groupPWA:PosOr.cont:session2                  0.2266
## groupcontrol:PosOr.cont:session3              0.8147
## groupPWA:PosOr.cont:session3                  0.4073
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) grp2-1 gr:PO. grPWA:PO. grpc:2 gPWA:2 grpc:3 gPWA:3 g:PO.:2
## group2-1     0.017
## grpcntr:PO.  0.002 -0.002
## grpPWA:PsO.  0.008  0.009  0.091
## grpcntrl:s2 -0.003  0.004 -0.001  0.000
## grpPWA:sss2 -0.007 -0.008  0.000  0.002    -0.010
## grpcntrl:s3 -0.004  0.005  0.000  0.000     0.153  0.000
## grpPWA:sss3 -0.013 -0.015  0.000 -0.001     0.000  0.217  0.001
## grpcn:PO.:2  0.000  0.000 -0.021  0.000     0.009  0.000  0.006  0.000
## grPWA:PO.:2  0.000  0.000  0.000 -0.043     0.000  0.031  0.000  0.011 -0.107
## grpcn:PO.:3  0.000  0.000 -0.033  0.000     0.006  0.000  0.012  0.000  0.253
## grPWA:PO.:3 -0.001 -0.001  0.000 -0.062     0.000  0.013  0.000  0.030  0.000
##             gPWA:PO.:2 g:PO.:3
## group2-1
## grpcntr:PO.
## grpPWA:PsO.
## grpcntrl:s2
## grpPWA:sss2
## grpcntrl:s3
## grpPWA:sss3
## grpcn:PO.:2
## grPWA:PO.:2
## grpcn:PO.:3  0.000
## grPWA:PO.:3  0.315      0.008
## optimizer (bobyqa) convergence code: 0 (OK)
## unable to evaluate scaled gradient
## Model failed to converge: degenerate  Hessian with 1 negative eigenvalues
```

```
saveRDS(m2_nested1, file = here::here("results", "tables",
                        "CSI_online_aphasia_Group_nest_PosOrxSession_glmm_cont.RDS"))
tab_model(m2_nested1,transform = NULL,
          show.re.var = F, show.stat = T,show.r2 = F,show.icc = F,
          title = "GLMM (Gamma distribution) of VOTs with Ordinal Position and Session nested into Grou
          # pred.labels = c("(Intercept)", "Ordinal Position",
          #                 "Session 2 vs 1",
          #                 "Session 3 vs 1", "Ord.Pos. x Session2-1",
          #                 "Ord.Pos. x Session3-1"),
          dv.labels = "Vocal Onset Time",
          #string.pred = "",
          string.stat = "t-Value",
          file = here::here("results", "tables","CSI_online_aphasia_Group_nest_PosOrxSession_glmm_cont.
```

GLMM (Gamma distribution) of VOTs with Ordinal Position and Session nested into Group

Vocal Onset Time

Predictors

Estimates

CI

t-Value

p

(Intercept)

1390.79

1349.80 – 1431.78

66.51

<0.001

group2-1

338.03

268.22 – 407.84

9.49

<0.001

group [control] * PosOrcont

14.96

4.98 – 24.94

2.94

0.003

group [PWA] * PosOr cont

30.65

19.48 – 41.81

5.38

<0.001

group [control] *session2

-82.70

-119.92 – -45.48

-4.36

<0.001

group [PWA] * session2

-113.41

-154.92 – -71.89

-5.35

<0.001

group [control] *session3

-104.06

-140.32 – -67.81

-5.63

<0.001

group [PWA] * session3

-146.68

-187.12 – -106.24

-7.11

<0.001

group [control] * PosOrcont * session2

-2.81

-24.38 – 18.75

-0.26

0.798

group [PWA] * PosOr cont* session2

15.15

-9.41 – 39.70

1.21

0.227

group [control] * PosOrcont * session3

2.21

-16.26 – 20.67

0.23

0.815

group [PWA] * PosOr cont* session3

9.06

-12.37 – 30.49

0.83

0.407

N subject

40

N category

24

Observations

12455

```
m2_nested1.2 <- afex::lmer_alt(VOT ~ session/(PosOr.cont*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_nested1.2)
```

```
##  The relative maximum gradient of 0.000146 is less than our 0.001 criterion.
##  You can safely ignore any warnings about a claimed convergence failure.
```

```
summary(m2_nested1.2)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: Gamma  ( identity )
## Formula:
## VOT ~ session/(PosOr.cont * 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)                           11534.94021 107.4008
##  subject.1   re1.PosOr.cont                          230.18084  15.1717
##  subject.2   re1.session2                           3987.82103  63.1492
##  subject.3   re1.session3                           3759.79038  61.3171
##  subject.4   re1.PosOr.cont_by_session2              799.76244  28.2801
##  subject.5   re1.PosOr.cont_by_session3              337.51203  18.3715
##  category    (Intercept)                           3218.46193  56.7315
##  category.1  re2.PosOr.cont                          107.70349  10.3780
##  category.2  re2.session2                           675.10427  25.9828
##  category.3  re2.session3                           671.60612  25.9154
##  category.4  re2.group2.1                          1350.47048  36.7487
##  category.5  re2.PosOr.cont_by_session2              201.20413  14.1846
```

```
##  category.6  re2.PosOr.cont_by_session3               294.13331  17.1503
##  category.7  re2.PosOr.cont_by_group2.1               180.27936  13.4268
##  category.8  re2.session2_by_group2.1                3093.44618  55.6188
##  category.9  re2.session3_by_group2.1                2663.72072  51.6112
##  category.10 re2.PosOr.cont_by_session2_by_group2.1  2197.62435  46.8788
##  category.11 re2.PosOr.cont_by_session3_by_group2.1  1097.58646  33.1298
##  Residual                                              0.08145   0.2854
## Number of obs: 12455, groups:  subject, 40; category, 24
##
## Fixed effects:
##                              Estimate Std. Error t value             Pr(>|z|)
## (Intercept)                  1390.790     20.911  66.510 < 0.0000000000000002
## session2                      -98.051     14.150  -6.930 0.000000000004223156
## session3                     -125.370     13.858  -9.047 < 0.0000000000000002
## session1:PosOr.cont            18.871      5.839   3.232              0.00123
## session2:PosOr.cont            25.038      6.417   3.902 0.000095370749907444
## session3:PosOr.cont            24.506      6.112   4.010 0.000060786008773415
## session1:group2-1             362.445     38.637   9.381 < 0.0000000000000002
## session2:group2-1             331.731     40.601   8.171 0.000000000000000307
## session3:group2-1             319.841     40.264   7.944 0.000000000000001965
## session1:PosOr.cont:group2-1    7.417     11.489   0.646              0.51856
## session2:PosOr.cont:group2-1   25.377     13.411   1.892              0.05845
## session3:PosOr.cont:group2-1   14.269     11.992   1.190              0.23412
##
## (Intercept)                  ***
## session2                     ***
## session3                     ***
## session1:PosOr.cont          **
## session2:PosOr.cont          ***
## session3:PosOr.cont          ***
## session1:group2-1            ***
## session2:group2-1            ***
## session3:group2-1            ***
## session1:PosOr.cont:group2-1
## session2:PosOr.cont:group2-1 .
## session3:PosOr.cont:group2-1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) sessn2 sessn3 ss1:PO. ss2:PO. ss3:PO. s1:2-1 s2:2-1 s3:2-1
## session2    -0.008
## session3    -0.012  0.189
## sssn1:PsOr.  0.005 -0.014 -0.014
## sssn2:PsOr.  0.004  0.016 -0.001  0.176
## sssn3:PsOr.  0.004 -0.001  0.013  0.266  -0.019
## sssn1:gr2-1  0.018 -0.043 -0.046  0.007   0.001   0.001
## sssn2:gr2-1  0.015  0.036 -0.007  0.001   0.009   0.000   0.741
## sssn3:gr2-1  0.013 -0.005  0.031  0.002   0.000   0.006   0.754  0.606
## ss1:PO.:2-1  0.003 -0.007 -0.007  0.153   0.007   0.016   0.012  0.001  0.002
## ss2:PO.:2-1  0.003  0.010 -0.001  0.006   0.101  -0.002   0.001  0.011  0.000
## ss3:PO.:2-1  0.003 -0.002  0.007  0.016  -0.002   0.102   0.001  0.000  0.010
##             s1:PO.: s2:PO.:
## session2
```

```
## session3
## sssn1:PsOr.
## sssn2:PsOr.
## sssn3:PsOr.
## sssn1:gr2-1
## sssn2:gr2-1
## sssn3:gr2-1
## ss1:PO.:2-1
## ss2:PO.:2-1  0.014
## ss3:PO.:2-1  0.251  -0.161
## optimizer (bobyqa) convergence code: 0 (OK)
## unable to evaluate scaled gradient
## Model failed to converge: degenerate  Hessian with 3 negative eigenvalues
```

```
saveRDS(m2_nested1.2, file = here::here("results", "tables",
                          "CSI_online_aphasia_Session_nest_PosOrxGroup_glmm_cont.RDS"))
tab_model(m2_nested1.2,transform = NULL,
          show.re.var = F, show.stat = T,show.r2 = F,show.icc = F,
          title = "GLMM (Gamma distribution) of VOTs with Ordinal Position and Group nested into Sessio
          # pred.labels = c("(Intercept)", "Ordinal Position",
          #                 "Session 2 vs 1",
          #                 "Session 3 vs 1", "Ord.Pos. x Session2-1",
          #                 "Ord.Pos. x Session3-1"),
          dv.labels = "Vocal Onset Time",
          #string.pred = "",
          string.stat = "t-Value",
          file = here::here("results", "tables","CSI_online_aphasia_Session_nest_PosOrxGroup_glmm_cont.
```

GLMM (Gamma distribution) of VOTs with Ordinal Position and Group nested into Session

Vocal Onset Time

Predictors

Estimates

CI

t-Value

p

(Intercept)

1390.79

1349.80 – 1431.78

66.51

<0.001

session [2]

-98.05

-125.79 – -70.32

-6.93

<0.001
session [3]
-125.37
-152.53 – -98.21
-9.05
<0.001
session [1] * PosOr cont
18.87
7.43 – 30.32
3.23
0.001
session [2] * PosOr cont
25.04
12.46 – 37.62
3.90
<0.001
session [3] * PosOr cont
24.51
12.53 – 36.49
4.01
<0.001
session1:group2-1
362.44
286.71 – 438.18
9.38
<0.001
session2:group2-1
331.73
252.15 – 411.32
8.17
<0.001
session3:group2-1
319.84
240.92 – 398.77
7.94
<0.001

session1:PosOr.cont:group2-1

7.42

-15.10 – 29.94

0.65

0.519

session2:PosOr.cont:group2-1

25.38

-0.91 – 51.66

1.89

0.058

session3:PosOr.cont:group2-1

14.27

-9.24 – 37.78

1.19

0.234

N subject

40

N category

24

Observations

12455

```
# m2_nested2 <- afex::lmer_alt(VOT ~ group/session/PosOr.cont +
#                 (PosOr.cont*session||subject) +
#                (PosOr.cont*session*group||category),
#               data = df_RTs,
#              family =Gamma(link ="identity"),
#              control=glmerControl(optimizer = "bobyqa",
#                                   optCtrl = list(maxfun = 2e5)))
m2_nested2 <- afex::lmer_alt(VOT ~ group/session/PosOr.cont +
               (PosOr.cont*session||subject) +
              (PosOr.cont*session*group-PosOr.cont||category),
             data = df_RTs,
            family =Gamma(link ="identity"),
            control=glmerControl(optimizer = "bobyqa",
                                 optCtrl = list(maxfun = 2e5)))
didLmerConverge(m2_nested2)
```

```
##  The relative maximum gradient of 0.000378 is less than our 0.001 criterion.
##  You can safely ignore any warnings about a claimed convergence failure.
```

```
summary(m2_nested2)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: Gamma  ( identity )
## Formula: VOT ~ group/session/PosOr.cont + (1 + re1.PosOr.cont + re1.session2 +
##     re1.session3 + re1.PosOr.cont_by_session2 + re1.PosOr.cont_by_session3 ||
##     subject) + (1 + re2.session2 + re2.session3 + re2.group2.1 +
##     re2.PosOr.cont_by_session1 + 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
## 178418.1 178648.4 -89178.0 178356.1    12424
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.5274 -0.6239 -0.2391  0.3345  7.9190
##
## Random effects:
##  Groups      Name                                    Variance    Std.Dev.
##  subject     (Intercept)                             11536.17980 107.4066
##  subject.1   re1.PosOr.cont                            229.26791  15.1416
##  subject.2   re1.session2                             4004.14658  63.2783
##  subject.3   re1.session3                             3775.60699  61.4460
##  subject.4   re1.PosOr.cont_by_session2                798.97698  28.2662
##  subject.5   re1.PosOr.cont_by_session3                340.20850  18.4447
##  category    (Intercept)                             3215.36617  56.7042
##  category.1  re2.session2                             674.46362  25.9704
##  category.2  re2.session3                             671.72882  25.9177
##  category.3  re2.group2.1                            1348.76715  36.7256
##  category.4  re2.PosOr.cont_by_session1               296.08110  17.2070
##  category.5  re2.PosOr.cont_by_session2               249.94246  15.8096
##  category.6  re2.PosOr.cont_by_session3               174.27048  13.2012
##  category.7  re2.PosOr.cont_by_group2.1               178.29181  13.3526
##  category.8  re2.session2_by_group2.1                3079.60866  55.4942
##  category.9  re2.session3_by_group2.1                2660.32755  51.5784
##  category.10 re2.PosOr.cont_by_session2_by_group2.1  2398.75557  48.9771
##  category.11 re2.PosOr.cont_by_session3_by_group2.1  1178.68325  34.3320
##  Residual                                              0.08139   0.2853
## Number of obs: 12455, groups:  subject, 40; category, 24
##
## Fixed effects:
##                                 Estimate Std. Error t value
## (Intercept)                     1390.945     20.908  66.526
## group2-1                         338.069     35.614   9.493
## groupcontrol:session2            -82.922     19.004  -4.363
## groupPWA:session2               -113.298     21.195  -5.346
## groupcontrol:session3           -104.463     18.513  -5.643
## groupPWA:session3               -147.174     20.647  -7.128
## groupcontrol:session1:PosOr.cont   15.420      7.923   1.946
## groupPWA:session1:PosOr.cont       22.983      9.125   2.519
## groupcontrol:session2:PosOr.cont   12.098      8.902   1.359
```

```
## groupPWA:session2:PosOr.cont       37.390      9.830   3.804
## groupcontrol:session3:PosOr.cont   17.170      7.923   2.167
## groupPWA:session3:PosOr.cont       31.421      8.813   3.565
##                                            Pr(>|z|)
## (Intercept)                       < 0.0000000000000002 ***
## group2-1                          < 0.0000000000000002 ***
## groupcontrol:session2                0.00001280272193 ***
## groupPWA:session2                    0.00000009012739 ***
## groupcontrol:session3                0.00000001675474 ***
## groupPWA:session3                    0.00000000000102 ***
## groupcontrol:session1:PosOr.cont             0.051633 .
## groupPWA:session1:PosOr.cont                 0.011782 *
## groupcontrol:session2:PosOr.cont             0.174153
## groupPWA:session2:PosOr.cont                 0.000142 ***
## groupcontrol:session3:PosOr.cont             0.030235 *
## groupPWA:session3:PosOr.cont                 0.000364 ***
## ---
## 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     0.017
## grpcntrl:s2 -0.003  0.004
## grpPWA:sss2 -0.007 -0.008 -0.010
## grpcntrl:s3 -0.004  0.005  0.152  0.000
## grpPWA:sss3 -0.013 -0.015  0.000  0.217   0.001
## grpcn:1:PO.  0.001 -0.001 -0.007  0.000  -0.008  0.000
## grPWA:1:PO.  0.005  0.006  0.000 -0.018   0.000 -0.018   0.089
## grpcn:2:PO.  0.001 -0.001  0.005  0.000   0.000  0.000   0.031  0.029
## grPWA:2:PO.  0.005  0.006  0.000  0.023   0.000 -0.002   0.030  0.035 -0.045
## grpcn:3:PO.  0.001 -0.001  0.000  0.000   0.007  0.000   0.226 -0.027 -0.109
## grPWA:3:PO.  0.004  0.005  0.000 -0.003   0.000  0.018  -0.028  0.205  0.083
##             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.  0.084
## grPWA:3:PO. -0.092  -0.041
## optimizer (bobyqa) convergence code: 0 (OK)
## unable to evaluate scaled gradient
## Model failed to converge: degenerate  Hessian with 1 negative eigenvalues
```

```
saveRDS(m2_nested2, file = here::here("results", "tables",
                          "CSI_online_aphasia_Group_nest_Session_nest_PosOr_glmm_cont.RDS"))
tab_model(m2_nested2,transform = NULL,
          show.re.var = F, show.stat = T,show.r2 = F,show.icc = F,
          title = "GLMM (Gamma distribution) of VOTs with Ordinal Position nested into Session nested i
          # pred.labels = c("(Intercept)", "Ordinal Position",
```

```
          #                   "Session 2 vs 1",
          #                   "Session 3 vs 1", "Ord.Pos. x Session2-1",
          #                   "Ord.Pos. x Session3-1"),
          dv.labels = "Vocal Onset Time",
          #string.pred = "",
          string.stat = "t-Value",
          file = here::here("results", "tables","CSI_online_aphasia_Group_nest_Session_nest_PosOr_glmm_
```

GLMM (Gamma distribution) of VOTs with Ordinal Position nested into Session nested into Group

Vocal Onset Time

Predictors

Estimates

CI

t-Value

p

(Intercept)

1390.95

1349.96 – 1431.93

66.53

<0.001

group2-1

338.07

268.26 – 407.88

9.49

<0.001

group [control] *session2

-82.92

-120.17 – -45.67

-4.36

<0.001

group [PWA] * session2

-113.30

-154.84 – -71.75

-5.35

<0.001

group [control] *session3

-104.46

-140.75 – -68.17

-5.64

<0.001

group [PWA] * session3

-147.17

-187.65 – -106.70

-7.13

<0.001

group [control] *session1* PosOr cont

15.42

-0.11 – 30.95

1.95

0.052

group [PWA] * session1 *PosOr cont

22.98

5.10 – 40.87

2.52

0.012

group [control] *session2* PosOr cont

12.10

-5.35 – 29.55

1.36

0.174

group [PWA] * session2 *PosOr cont

37.39

18.12 – 56.66

3.80

<0.001

group [control] *session3* PosOr cont

17.17

1.64 – 32.70

2.17

0.030

group [PWA] * session3 *PosOr cont

31.42

14.15 – 48.70

3.57

<0.001

N subject

40

N category

24

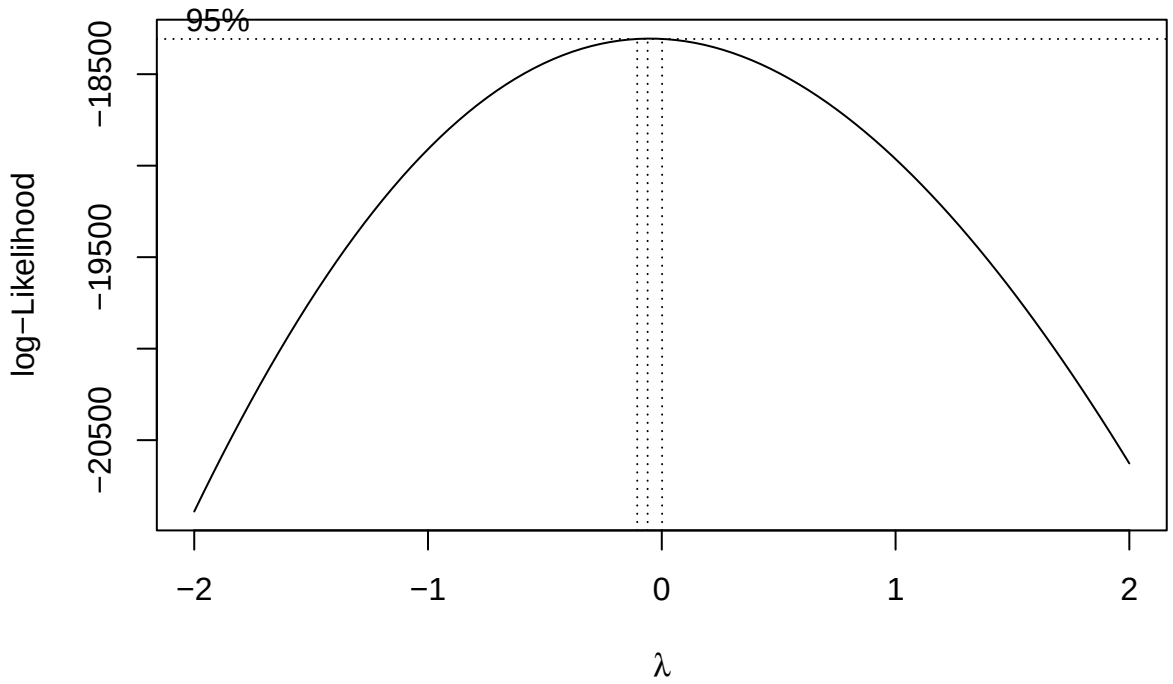
Observations

12455

Control: Comparison with transformed RTs

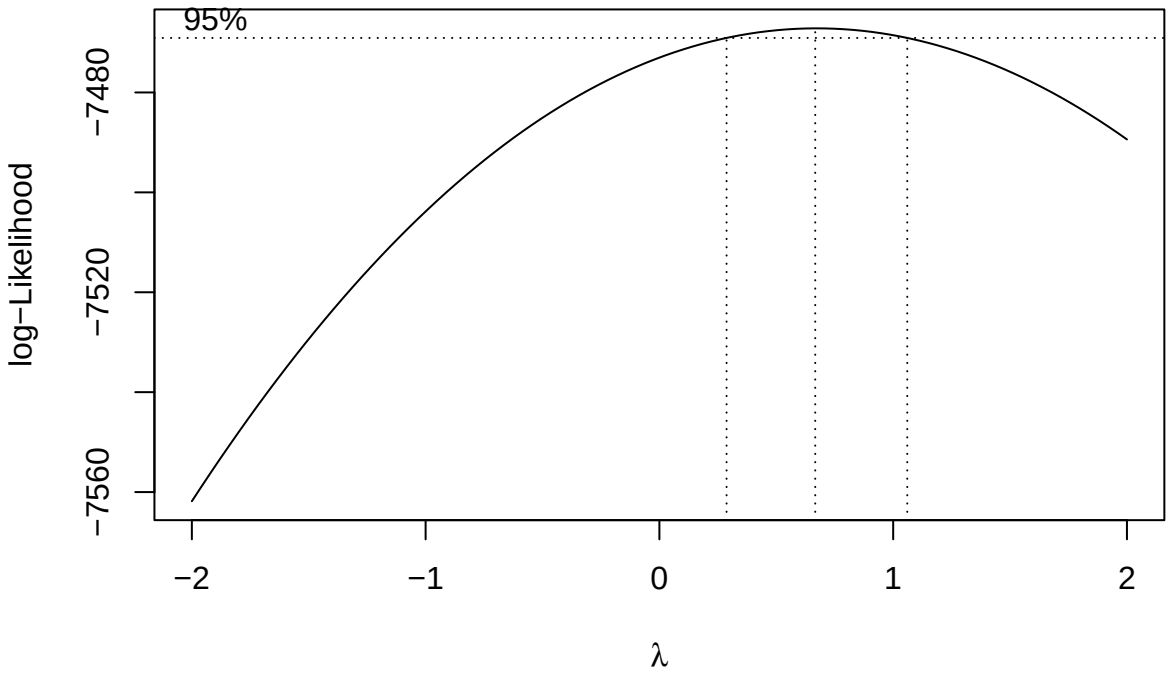
Box-cox test
(common transformations: -2 -> 1/(Yˆ2), -1 -> 1/y, -0.5 -> 1/(sqrt(y))), 0 -> log(y), 0.5 -> sqrt(y), *1* -> *y*, 2 -> yˆ2, 3 -> yˆ3)

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



PWA only

```
## Box-Cox suggests log transformation --> compute with log-transformed RTs as s control analysis
## for the main analyses we will use a GLMM
boxcox(log(df_RTs_PWA$VOT)~  df_RTs_PWA$PosOr*df_RTs_PWA$session)
```




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

```
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
```

```
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",
          pred.labels = c("(Intercept)", "Ordinal Position", "Session 2 vs 1",
                          "Session 3 vs 1", "Ord.Pos. x Session2-1",
                          "Ord.Pos. x Session3-1"),
          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

Ordinal Position

0.02

0.01 – 0.03

5.57

<0.001

Session 2 vs 1

-0.09

-0.11 – -0.06

-7.56

<0.001

Session 3 vs 1

-0.10

-0.13 – -0.06

-6.08

<0.001

Ord.Pos. x Session2-1

0.01

-0.00 – 0.02

1.75

0.081

Ord.Pos. x Session3-1

0.01

-0.01 – 0.02

0.82

0.413

N subject

20

N category

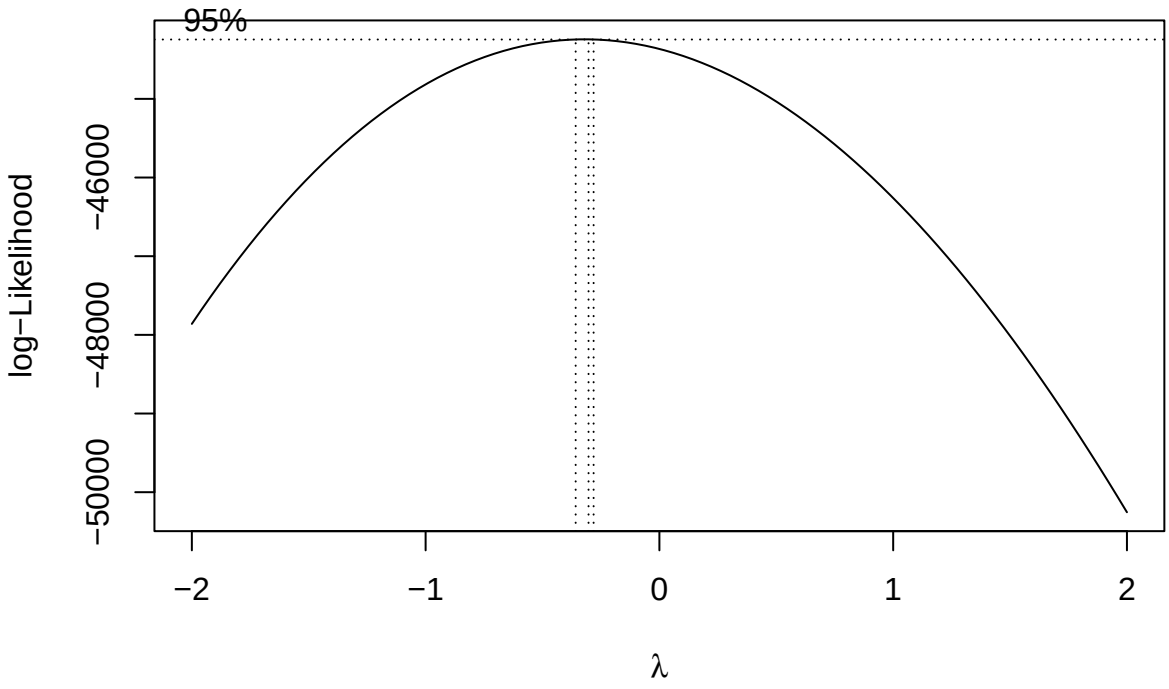
24

Observations

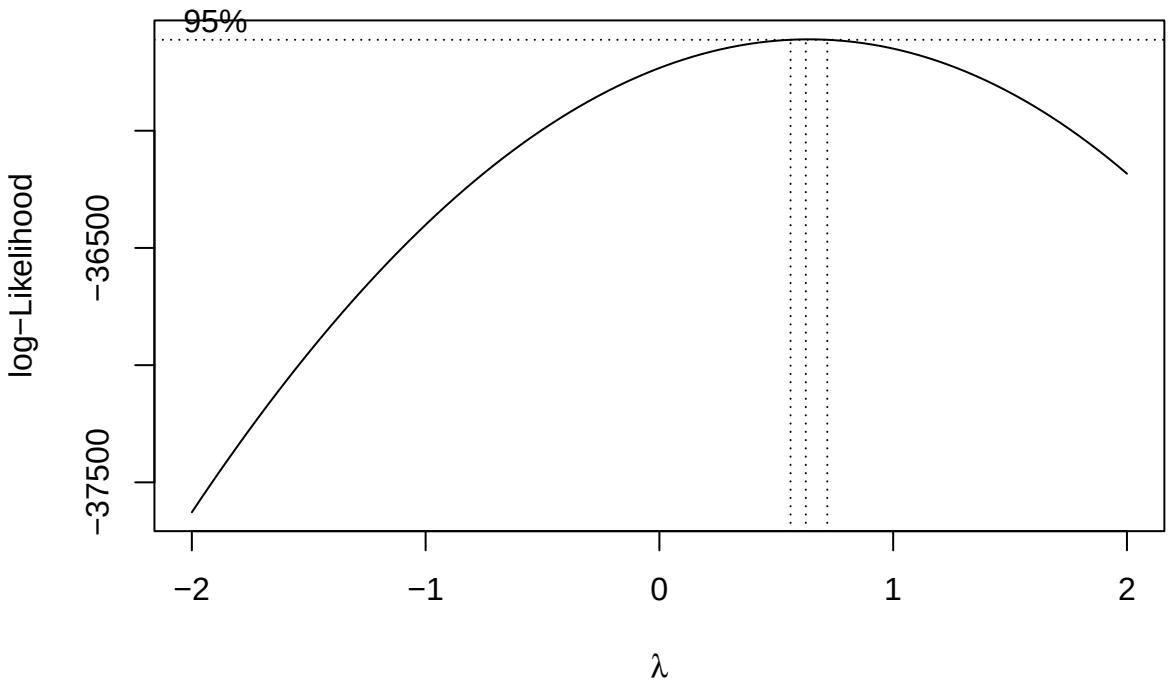
5518

Secondary analysis with factor group *Box-cox test*
(common transformations: -2 -> 1/(Yˆ2), -1 -> 1/y, -0.5 -> 1/(sqrt(y))), 0 -> log(y), 0.5 -> sqrt(y), *1* -> *y*, 2 -> yˆ2, 3 -> yˆ3)

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




```
## Box-Cox suggests 1/sqrt transformation --> compute with transformed RTs as s control analysis
## for the main analyses we will use a GLMM
# boxcox(log(df_RTs$VOT) ~  df_RTs$PosOr*df_RTs$session*df_RTs$group)
# boxcox(1000/df_RTs$VOT~  df_RTs$PosOr*df_RTs$session*df_RTs$group)
boxcox(1/sqrt(df_RTs$VOT)~  df_RTs$PosOr*df_RTs$session*df_RTs$group)
```




```
df_RTs$VOTsqrt <- 1/sqrt(df_RTs$VOT)
```

```
library(lmerTest)
# m2_lmm <- lmer(VOTsqrt ~ 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(VOTsqrt ~ PosOr.cont*session*group +
#                  (PosOr.cont*session|subject) +
#                (PosOr.cont*session*group|category),
#               data = df_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(VOTsqrt ~ 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(VOTsqrt ~ PosOr.cont*session*group +
#                 (PosOr.cont*session||subject) +
#                (PosOr.cont*session*group-session-PosOr.cont:group-session:group||category),
#              data = df_RTs,
#             control=lmerControl(optimizer = "bobyqa",
#                                 optCtrl = list(maxfun = 2e5)))

  # m2_lmm <- afex::lmer_alt(VOTsqrt ~ PosOr.cont*session*group +
  #                 (PosOr.cont*session||subject) +
  #                (PosOr.cont*session*group-session-PosOr.cont:group-session:group||category),
  #              data = df_RTs,
  #             control=lmerControl(optimizer = "bobyqa",
  #                                   optCtrl = list(maxfun = 2e5)))
# m2_lmm <- afex::lmer_alt(VOTsqrt ~ 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(VOTsqrt ~ 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 0.0000047 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: VOTsqrt ~ 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: -105142.7
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -5.7850 -0.5434  0.1263  0.6635  7.1916
##
## Random effects:
##  Groups   Name        Variance       Std.Dev.   Corr
##  subject  (Intercept) 0.000007924962 0.00281513
##           PosOr.cont  0.000000020840 0.00014436  0.13
##           session2    0.000000663998 0.00081486 -0.04  0.26
##           session3    0.000000944836 0.00097203 -0.50  0.24  0.57
##  category (Intercept) 0.000001815120 0.00134726
##           PosOr.cont  0.000000005342 0.00007309 -0.07
##           group2-1    0.000000171640 0.00041430 -0.05  0.15
##  Residual             0.000011981558 0.00346144
## Number of obs: 12455, groups:  subject, 40; category, 24
##
## Fixed effects:
##                                  Estimate     Std. Error             df
## (Intercept)                     0.02888938     0.00052429    57.94410593
## PosOr.cont                     -0.00023864     0.00003539    31.07110095
## session2                        0.00102977     0.00015167    38.93606557
## session3                        0.00118586     0.00017291    37.37342619
## group2-1                       -0.00199987     0.00089675    38.53838639
## PosOr.cont:session2            -0.00007206     0.00005475 12261.36986564
## PosOr.cont:session3            -0.00004999     0.00005435 12262.24941430
## PosOr.cont:group2-1            -0.00007212     0.00006418    37.60062787
## session2:group2-1               0.00031995     0.00030333    38.93568415
## session3:group2-1               0.00017637     0.00034582    37.37348419
## PosOr.cont:session2:group2-1   -0.00020843     0.00010951 12259.75846929
## PosOr.cont:session3:group2-1   -0.00005121     0.00010870 12261.82843912
##                               t value             Pr(>|t|)
## (Intercept)                    55.102 < 0.0000000000000002 ***
## PosOr.cont                     -6.743         0.0000001493 ***
## session2                        6.790         0.0000000421 ***
## session3                        6.858         0.0000000416 ***
## group2-1                       -2.230               0.0316 *
## PosOr.cont:session2            -1.316               0.1881
## PosOr.cont:session3            -0.920               0.3577
## PosOr.cont:group2-1            -1.124               0.2682
## session2:group2-1               1.055               0.2980
## session3:group2-1               0.510               0.6131
## PosOr.cont:session2:group2-1   -1.903               0.0570 .
## PosOr.cont:session3:group2-1   -0.471               0.6375
## ---
## 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   0.055
## session2    -0.029  0.144
## session3    -0.382  0.138  0.557
## group2-1    -0.001  0.007 -0.001 -0.001
## 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  0.068  0.001  0.001  0.090 -0.011 -0.016
## sssn2:gr2-1  0.000  0.001  0.051  0.026 -0.034  0.007  0.005  0.159
```

```
## sssn3:gr2-1 -0.001  0.001  0.026  0.036 -0.446  0.004  0.007  0.152  0.557
## PsOr.:2:2-1  0.000 -0.010  0.007  0.004  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
## PosOr.cont                0.00054479 0.00054479     1    31.1 45.4687
## session                   0.00071671 0.00035835     2    38.1 29.9088
## group                     0.00005959 0.00005959     1    38.5  4.9735
## PosOr.cont:session        0.00002171 0.00001086     2 12261.3  0.9061
## PosOr.cont:group          0.00001513 0.00001513     1    37.6  1.2630
## session:group             0.00001344 0.00000672     2    38.1  0.5607
## PosOr.cont:session:group  0.00004762 0.00002381     2 12260.2  1.9872
##                                 Pr(>F)
## PosOr.cont                0.00000014933 ***
## session                   0.00000001555 ***
## group                           0.03164 *
## PosOr.cont:session              0.40411
## PosOr.cont:group                0.26821
## session:group                   0.57546
## PosOr.cont:session:group        0.13712
## ---
## 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, show.stat = T,show.r2 = F,show.icc = F,
          title = "LMM of VOTs Predicted by Ordinal Position and Session",
          # pred.labels = c("(Intercept)", "Ordinal Position", "Session 2 vs 1",
          #                 "Session 3 vs 1", "Ord.Pos. x Session2-1",
          #                 "Ord.Pos. x Session3-1"),
          dv.labels = "Vocal Onset Time (1/sqrt-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 (1/sqrt-transformed)

Predictors

Estimates

CI

t-Value

p

(Intercept)

0.03

0.03 – 0.03

55.10

<0.001

PosOr cont

-0.00

-0.00 – -0.00

-6.74

<0.001

session [2]

0.00

0.00 – 0.00

6.79

<0.001

session [3]

0.00

0.00 – 0.00

6.86

<0.001

group2-1

-0.00

-0.00 – -0.00

-2.23

0.032

PosOr cont * session [2]

-0.00

-0.00 – 0.00

-1.32

0.188

PosOr cont * session [3]

-0.00

-0.00 – 0.00

-0.92

0.358

PosOr.cont:group2-1

-0.00

-0.00 – 0.00

-1.12

0.268

session2:group2-1

0.00

-0.00 – 0.00

1.05

0.298

session3:group2-1

0.00

-0.00 – 0.00

0.51

0.613

PosOr.cont:session2:group2-1

-0.00

-0.00 – 0.00

-1.90

0.057

PosOr.cont:session3:group2-1

-0.00

-0.00 – 0.00

-0.47

0.638

N subject

40

N category

24

Observations

12455

Exploratory control: Add Array to model

Center array

```
df$array <- as.factor(df$array)
contrasts(df$array) <- contr.sdif(30)
```

Add array to fixed structure of intercept only model (otherwise it takes veeery long to converge)

```
# m2_array <- afex::lmer_alt(VOT ~ PosOr.cont*session*group*array +
#                 (1|subject) +
#               (1|category),
#              data = df_RTs,
#             family =Gamma(link ="identity"),
#             control=glmerControl(optimizer = "bobyqa",
#                                  optCtrl = list(maxfun = 2e5)))
# didLmerConverge(m2_array)
# ## The warnings can be safely ignored
#
# # inspect model
# summary(m2_array)
# anova(m2_array)
#
# # save model output
# saveRDS(m2_array, file = here::here("results", "tables",
#                            "CSI_online_aphasia_SessionxGroupxArray.RDS"))
```

Model still fails to converge, but there seems to be some influence of array.
What if we add array to the random structure only?

```
# m2_array_rand <- afex::lmer_alt(VOT ~ PosOr.cont*session*group +
#                 (PosOr.cont*session*array||subject) +
#               (PosOr.cont*session*group*array||category),
#              data = df_RTs,
#             family =Gamma(link ="identity"),
#             control=glmerControl(optimizer = "bobyqa",
#                                  optCtrl = list(maxfun = 2e5)))
# summary(m2_array_rand)
# # save model output
# saveRDS(m2_array_rand, file = here::here("results", "tables",
#                            "CSI_online_aphasia_SessionxGroupxArray_random_structure.RDS"))
```

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  7033     0.977
## 2 control           1   167     0.0232
## 3 PWA               0  5660     0.786
## 4 PWA               1  1540     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  2323     0.968
##  2 control 1                 1    77     0.0321
##  3 control 2                 0  2343     0.976
##  4 control 2                 1    57     0.0238
##  5 control 3                 0  2367     0.986
##  6 control 3                 1    33     0.0138
##  7 PWA     1                 0  1791     0.746
##  8 PWA     1                 1   609     0.254
##  9 PWA     2                 0  1895     0.790
## 10 PWA     2                 1   505     0.210
## 11 PWA     3                 0  1974     0.822
## 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: 22 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 99       81   0.0112
## # ... with 12 more rows
```

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

```
## # A tibble: 60 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       99       39   0.0162
##  8 control 1       <NA>   2284   0.952
##  9 control 2       1         1   0.000417
## 10 control 2       2        17   0.00708
## # ... with 50 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  99
##   0   0   0   0   0   0   0   0   0   0 103
##   1  56 188  94 851  33  46 137  32 270   0
```

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

```
##
##     0
## 12590
```

```
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` argumen
```

```
## # 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
## # ... with 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):

```
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
```

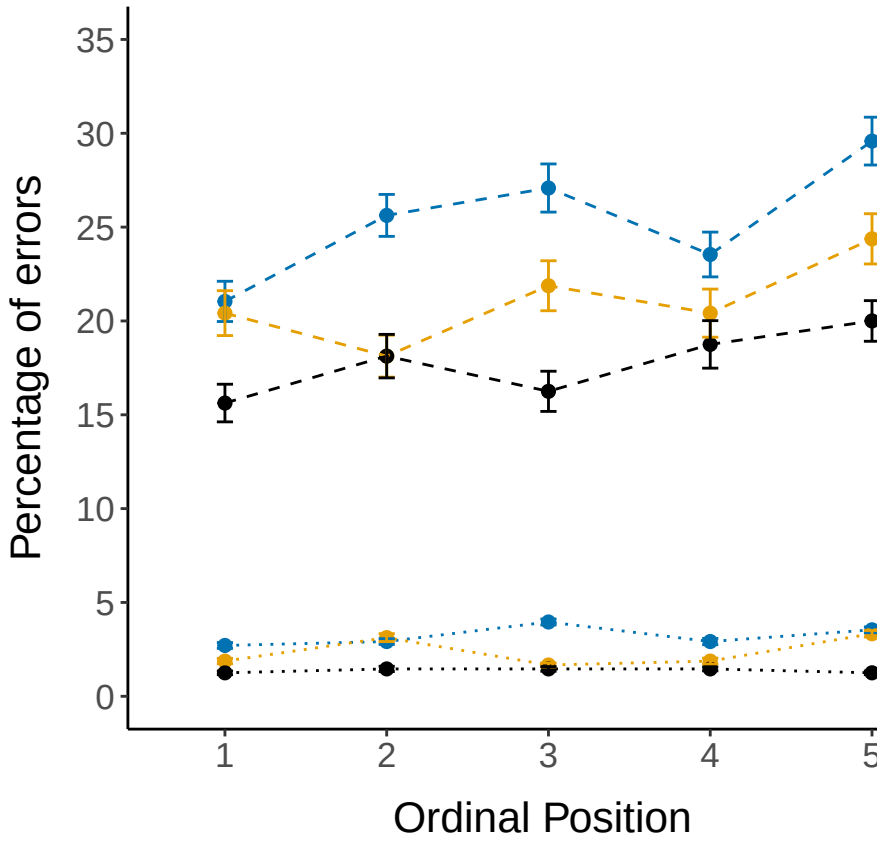
```
means_final_errors$PosOr_effect <- NA
means_final_errors$PosOr_effect[means_final_errors$PosOr==1] <- 1
for(k in 1:length(unique(means_final_errors$group))){
for(i in 1:length(unique(means_final_errors$session))){
  means_final_errors$PosOr_effect[means_final_errors$session==unique(means_final_errors$session)[i] &
        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)[
        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)[
        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)[
        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
```

```
## # ... with 20 more rows, and 3 more variables: increase_count <dbl>,
## #   increase_mean <dbl>, PosOr_effect <dbl>
```

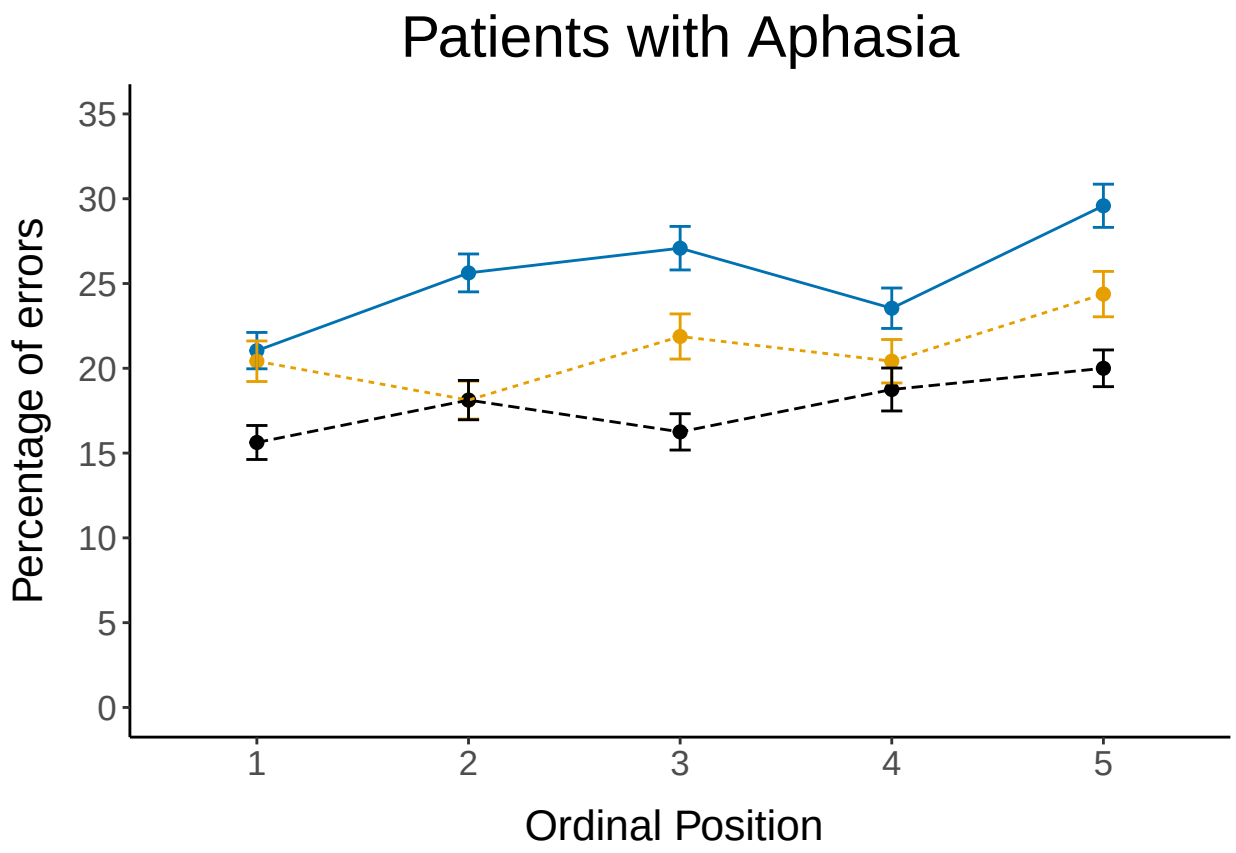
Plotting

```
means_final_errors$session_group <- paste0(means_final_errors$group,
                                           means_final_errors$session)
override.linetype<-c("dotted", "dashed")
(plot_error <- means_final_errors %>%
    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"))+
  # guides(color=guide_legend(
  #    override.aes=list(linetype=override.linetype)))+
  labs(x="Ordinal Position ",y ="Percentage of errors"))
```

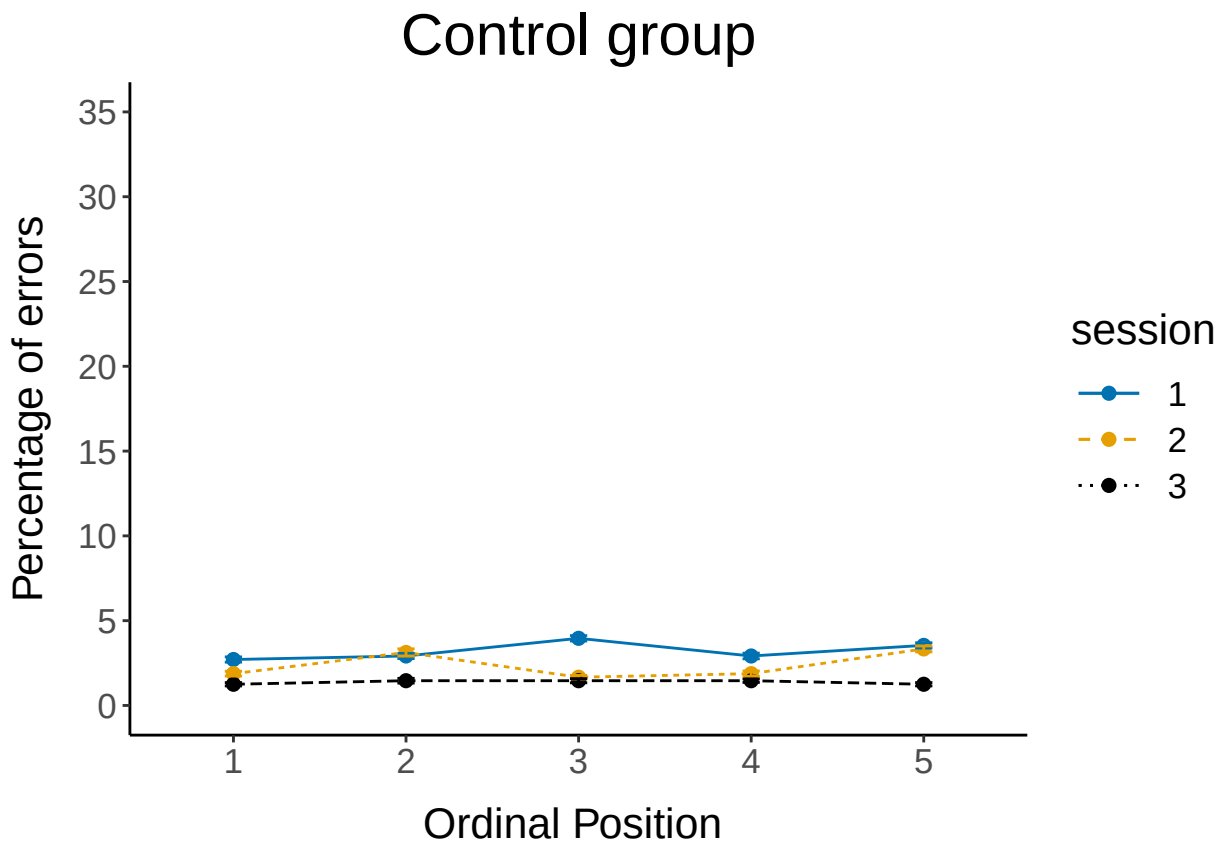


Errors by ordinal position and repetition

```
override.linetype<-c("solid", "dashed", "dotted")
(plot_error_PWA <- means_final_errors %>% filter(group=="PWA") %>%
    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_linetype_manual(values=c("solid", "dashed", "dotted"))+
    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"),
    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"))
```

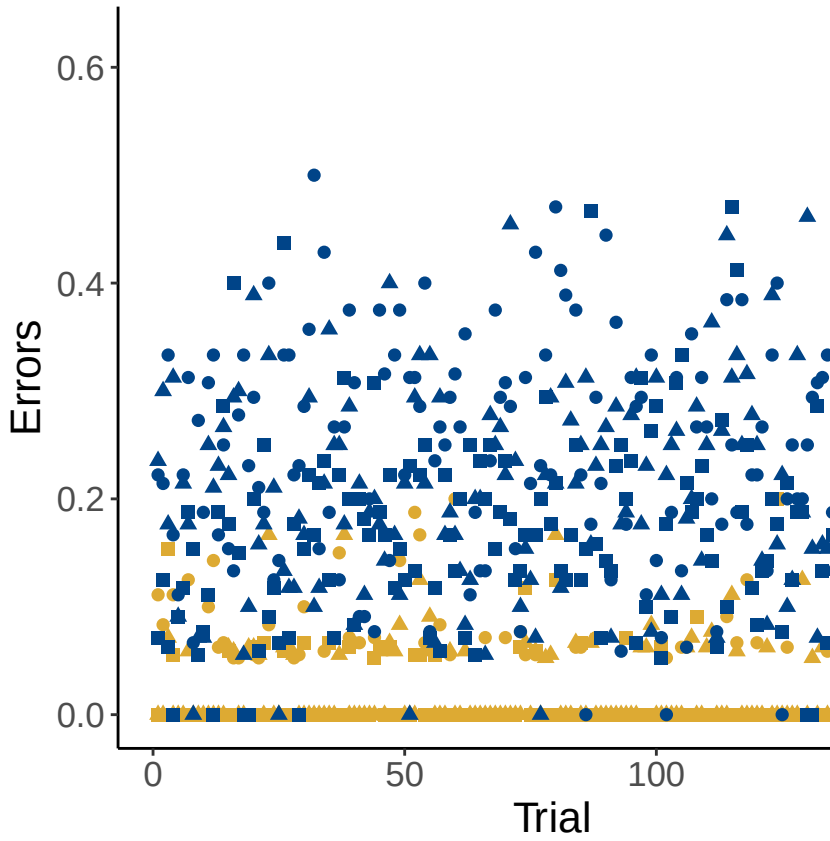



```
(plot_error_control <- means_final_errors %>% filter(group=="control") %>%
    ggplot(., aes(x=PosOr, y=mean_p, group=session, color = session)) +
  geom_point( size = 2)+
      stat_summary(aes(linetype=session),fun=mean,  geom="line", size = 0.5) +
    #scale_linetype_manual(values=c("solid", "dashed", "dotted"))+
    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"))+
  guides(color=guide_legend(
     override.aes=list(linetype=override.linetype)))+
  labs(x="Ordinal Position ",y ="Percentage of errors",
       title="Control group"))
```




```
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))
```

```
(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)+
  #stat_summary(aes(color=group, linetype=session),fun=mean,  geom="line", size = 1) +
  apatheme+
  labs(x="Trial ",y ="Errors")+
  scale_color_manual(values=c(control_color, PWA_color)))
```



Control: Plot Errors accross the experiment

```
 # annotate(geom="text", x=20, y=200, label="n = 19",
  #          color="black", size = 8))


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"
```

PWA only *GLMM*

```
# 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] 0.5903587
##
## Rotation (n x k) = (1 x 1):
##      [,1]
## [1,]    1
##
## $subject
## Standard deviations (1, .., p=2):
## [1] 1.4819260 0.1130569
##
## Rotation (n x k) = (2 x 2):
##             [,1]        [,2]
## [1,] -0.99975217 -0.02226226
## [2,] -0.02226226  0.99975217
##
## attr(,"class")
## [1] "prcomplist"
```

```
didLmerConverge(m1_error)
```

```
##  The relative maximum gradient of 0.00000345 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
##   5371.6   5440.4  -2675.8   5351.6     7190
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.4699 -0.3963 -0.2435 -0.1231 11.3501
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  category (Intercept) 0.34852  0.5904
##  subject  (Intercept) 2.19502  1.4816
##           PosOr.cont  0.01386  0.1177   0.28
## Number of obs: 7200, groups:  category, 24; subject, 20
##
## Fixed effects:
##                     Estimate Std. Error z value            Pr(>|z|)
## (Intercept)         -1.87170    0.35550  -5.265 0.00000014021408657 ***
## PosOr.cont           0.08597    0.03990   2.155              0.0312 *
## session2            -0.36621    0.08400  -4.360 0.00001301942954910 ***
## session3            -0.68018    0.08716  -7.803 0.00000000000000602 ***
## PosOr.cont:session2 -0.03151    0.05936  -0.531              0.5956
## PosOr.cont:session3 -0.03200    0.06154  -0.520              0.6032
## ---
## 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   0.168
## session2     0.009  0.001
## session3     0.017 -0.003  0.460
## PsOr.cnt:s2  0.000  0.052 -0.034 -0.016
## PsOr.cnt:s3  0.000  0.098 -0.016 -0.035  0.459
```

```
# 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",
          pred.labels = c("(Intercept)", "Ordinal Position", "Session 2 vs 1",
                          "Session 3 vs 1", "Ord.Pos. x Session2-1",
                          "Ord.Pos. x Session3-1"),
          dv.labels = "Error Rate",
```

```
          #string.pred = "",
          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)

-1.87

-2.57 – -1.17

-5.26

<0.001

Ordinal Position

0.09

0.01 – 0.16

2.15

0.031

Session 2 vs 1

-0.37

-0.53 – -0.20

-4.36

<0.001

Session 3 vs 1

-0.68

-0.85 – -0.51

-7.80

<0.001

Ord.Pos. x Session2-1

-0.03

-0.15 – 0.08

-0.53

0.596

Ord.Pos. x Session3-1

-0.03

-0.15 – 0.09

-0.52

0.603

N subject

20

N category

24

Observations

7200

Secondary analysis: Session x Group *GLMM*

```
# 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 -> reduce 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)))
# rePCA(m2_error)
didLmerConverge(m2_error)
```

```
##  The relative maximum gradient of 0.0000134 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 + (1 + re1.PosOr.cont ||
##     subject) + (1 + re2.group2.1 || category)
##    Data: data
## Control: glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 200000))
##
##      AIC      BIC   logLik deviance df.resid
##   6803.7   6924.9  -3385.9   6771.7    14384
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.3138 -0.2733 -0.1470 -0.0762 18.2499
##
## Random effects:
##  Groups     Name           Variance Std.Dev.
##  subject    (Intercept)    1.329936 1.15323
##  subject.1  re1.PosOr.cont 0.008825 0.09394
##  category   (Intercept)    0.587998 0.76681
##  category.1 re2.group2.1   0.544928 0.73819
## Number of obs: 14400, groups:  subject, 40; category, 24
##
## Fixed effects:
##                               Estimate Std. Error z value             Pr(>|z|)
## (Intercept)                  -3.186771   0.249517 -12.772 < 0.0000000000000002
## PosOr.cont                    0.071264   0.035463   2.010             0.044479
## session2                     -0.347771   0.097312  -3.574             0.000352
## session3                     -0.791853   0.111849  -7.080     0.00000000000145
## group2-1                      2.627177   0.415965   6.316     0.00000000026866
## PosOr.cont:session2          -0.005382   0.068631  -0.078             0.937498
## PosOr.cont:session3          -0.042242   0.078844  -0.536             0.592122
## PosOr.cont:group2-1           0.053342   0.070945   0.752             0.452131
## session2:group2-1            -0.037144   0.194597  -0.191             0.848620
## session3:group2-1             0.222045   0.223657   0.993             0.320809
```

```
## PosOr.cont:session2:group2-1 -0.047417   0.137257  -0.345             0.729750
## PosOr.cont:session3:group2-1  0.029547   0.157687   0.187             0.851362
##
## (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  -0.009
## session2     0.020 -0.013
## session3     0.057  0.012  0.382
## group2-1    -0.066  0.005 -0.018 -0.058
## PsOr.cnt:s2 -0.002  0.082 -0.074 -0.024  0.003
## PsOr.cnt:s3  0.003  0.252 -0.024 -0.024 -0.004  0.382
## PsOr.cn:2-1  0.005 -0.535  0.012 -0.014 -0.011 -0.064 -0.220
## sssn2:gr2-1 -0.014  0.012 -0.627 -0.227  0.023  0.058  0.017 -0.013
## sssn3:gr2-1 -0.047 -0.014 -0.227 -0.696  0.067  0.017  0.011  0.012  0.382
## PsOr.:2:2-1  0.003 -0.064  0.058  0.017 -0.003 -0.626 -0.228  0.082 -0.074
## PsOr.:3:2-1 -0.003 -0.221  0.017  0.011  0.004 -0.228 -0.697  0.252 -0.024
##             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 -0.024
## PsOr.:3:2-1 -0.024  0.382
```

```
# 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",
          # pred.labels = c("(Intercept)", "Ordinal Position", "Session 2 vs 1",
          #                 "Session 3 vs 1", "Ord.Pos. x Session2-1",
          #                 "Ord.Pos. x Session3-1"),
          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)
-3.19
-3.68 – -2.70
-12.77
<0.001
PosOr cont
0.07
0.00 – 0.14
2.01
0.044
session [2]
-0.35
-0.54 – -0.16
-3.57
<0.001
session [3]
-0.79
-1.01 – -0.57
-7.08
<0.001
group2-1
2.63
1.81 – 3.44
6.32
<0.001

PosOr cont * session [2]

-0.01

-0.14 – 0.13

-0.08

0.937

PosOr cont * session [3]

-0.04

-0.20 – 0.11

-0.54

0.592

PosOr.cont:group2-1

0.05

-0.09 – 0.19

0.75

0.452

session2:group2-1

-0.04

-0.42 – 0.34

-0.19

0.849

session3:group2-1

0.22

-0.22 – 0.66

0.99

0.321

PosOr.cont:session2:group2-1

-0.05

-0.32 – 0.22

-0.35

0.730

PosOr.cont:session3:group2-1

0.03

-0.28 – 0.34

0.19

0.851

N subject

40

N category

24

Observations

14400

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

```
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

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

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

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

Descriptives

```
(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
```

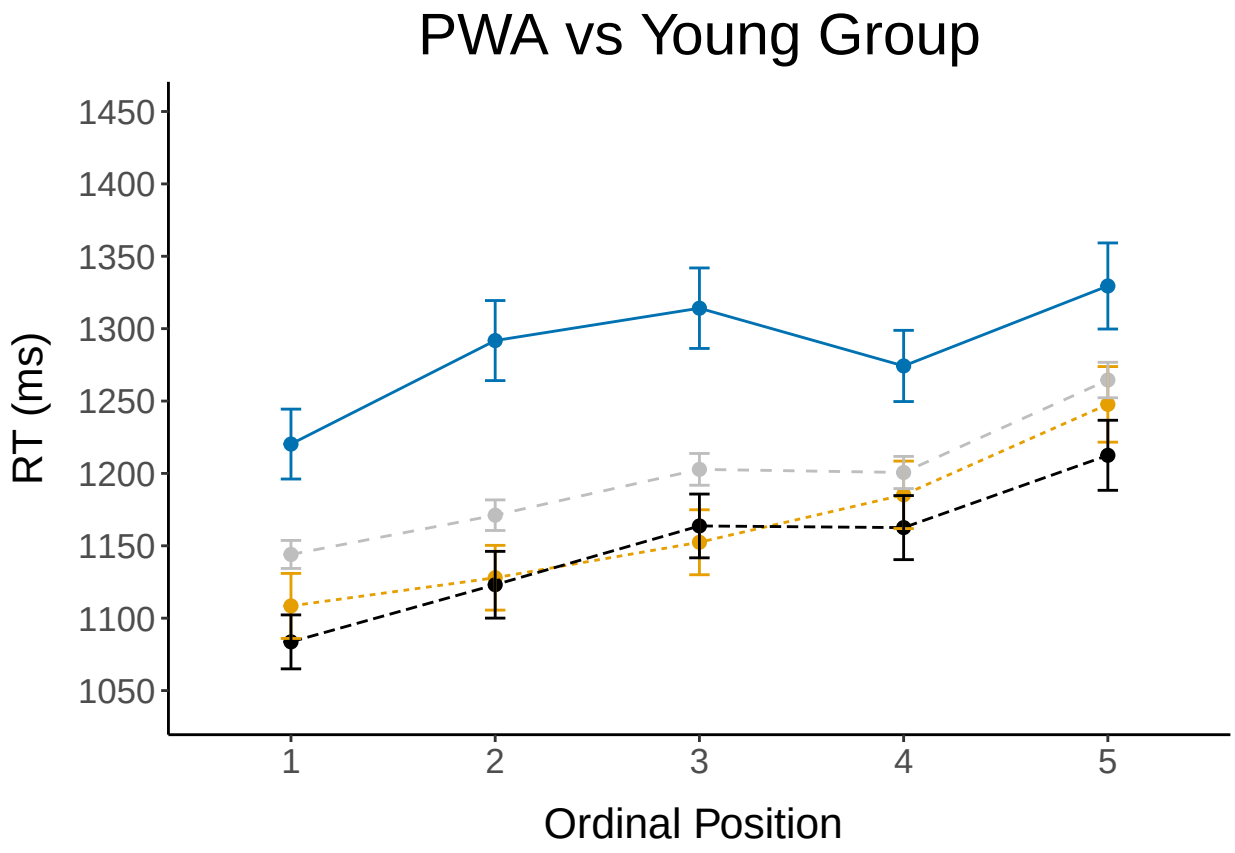
Plotting

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 +
```

```
    # stat_summary(df_combi[df_combi$group=="young",],
    #              aes(x=PosOr, y=VOT), fun=mean,  geom="line", size = 1) +
    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") + #+
  # annotate(geom="text", x=1.5, y=1330, label="n = 30",
  #          color="black", size = 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="none")+
   guides(color=guide_legend(
     override.aes=list(linetype=override.linetype)))+
    scale_linetype(guide="none"))
```

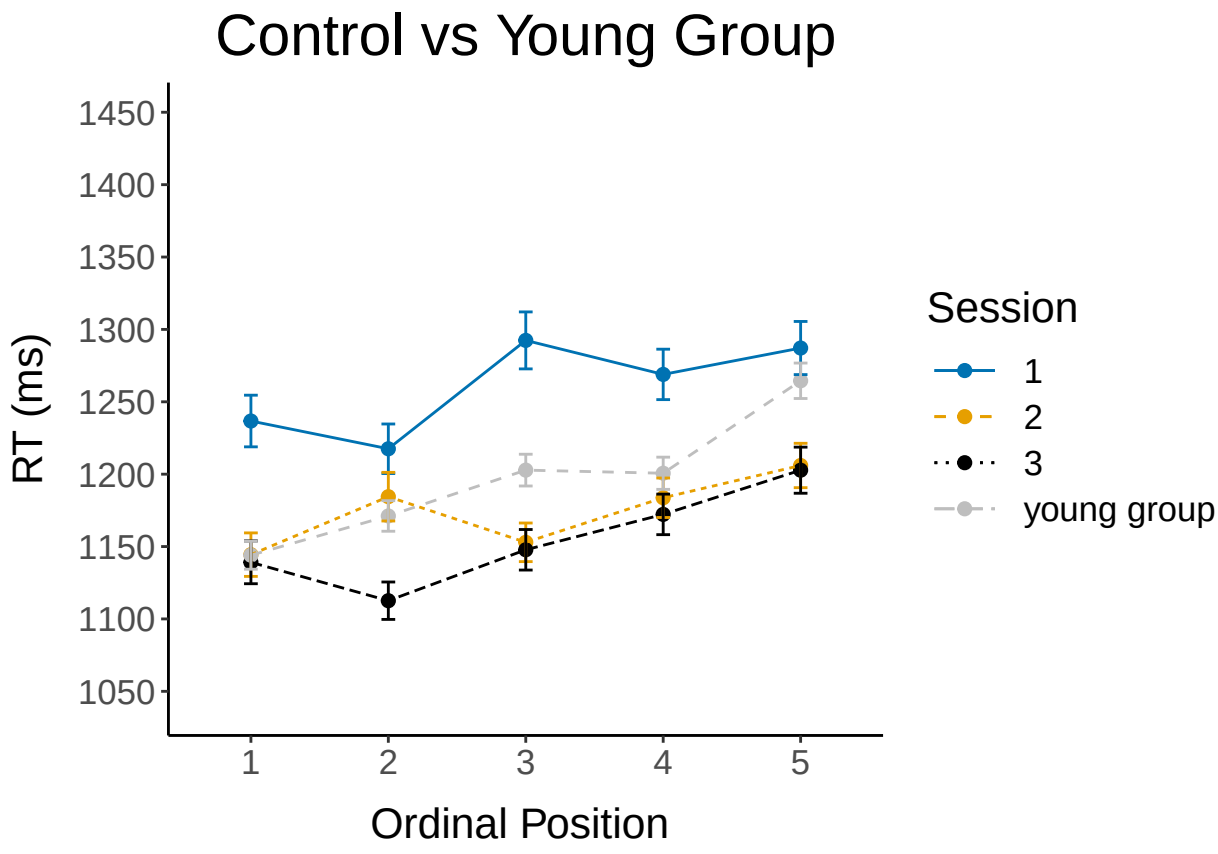
```
## Scale for 'linetype' is already present. Adding another scale for 'linetype',
## which will replace the existing scale.
```




```
(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)) +
    #breaks = c(1100, 1150, 1200, 1250, 1300, 1350)) +
    labs(x="Ordinal Position ",y ="RT (ms)", colour="Session", linetype="Session",
         title = "Control vs Young Group") + #+
  # annotate(geom="text", x=1.5, y=1330, label="n = 30",
  #         color="black", size = 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"))+
   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))
```

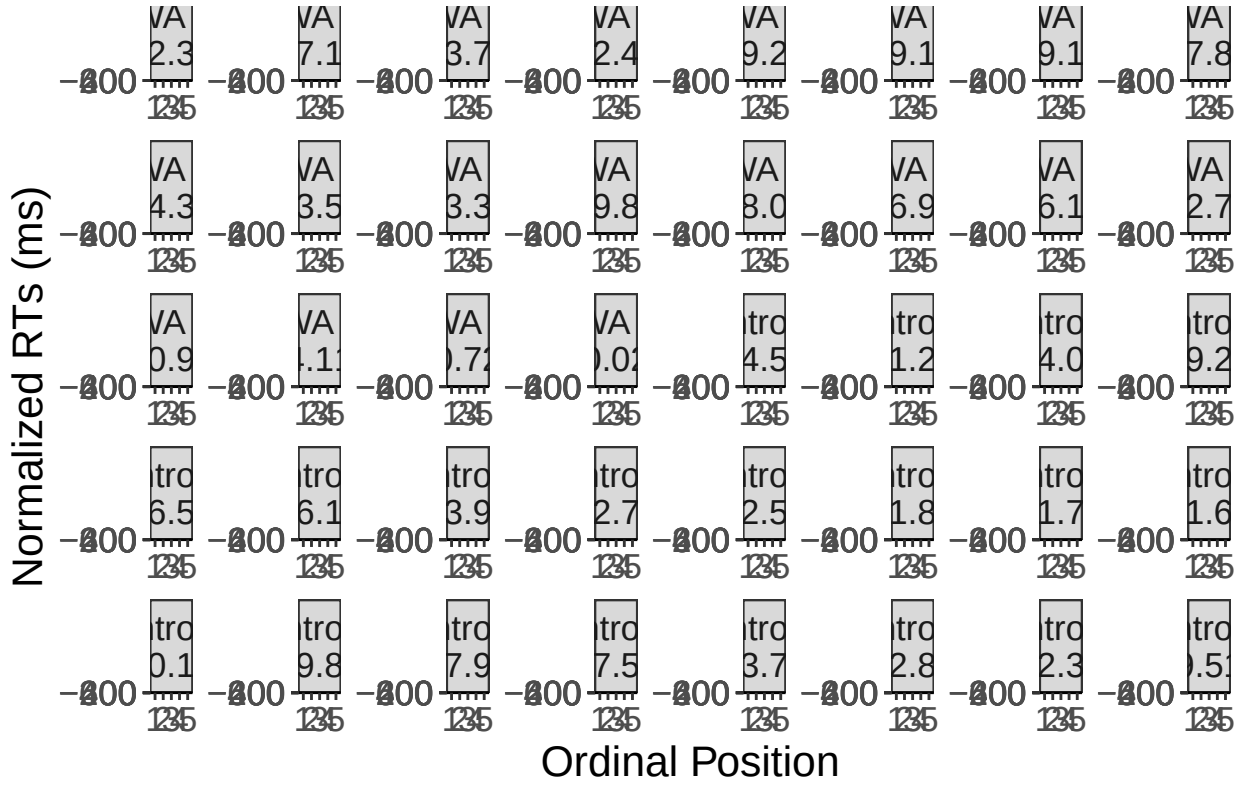
Additional plots

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"),#, "session"),
                          #betweenvars="group",
                          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$subject[i] & means_final
                                  means_final_subject$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$effect, levels=rev(levels(means_final_subject$


# 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(0.72)","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"))
```




```
#plot_rt <- lemon::reposition_legend(plot_rt, "bott1.1.om right",panel='panel-5-5')

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))
```

RTs by category

Line graph for each category:

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

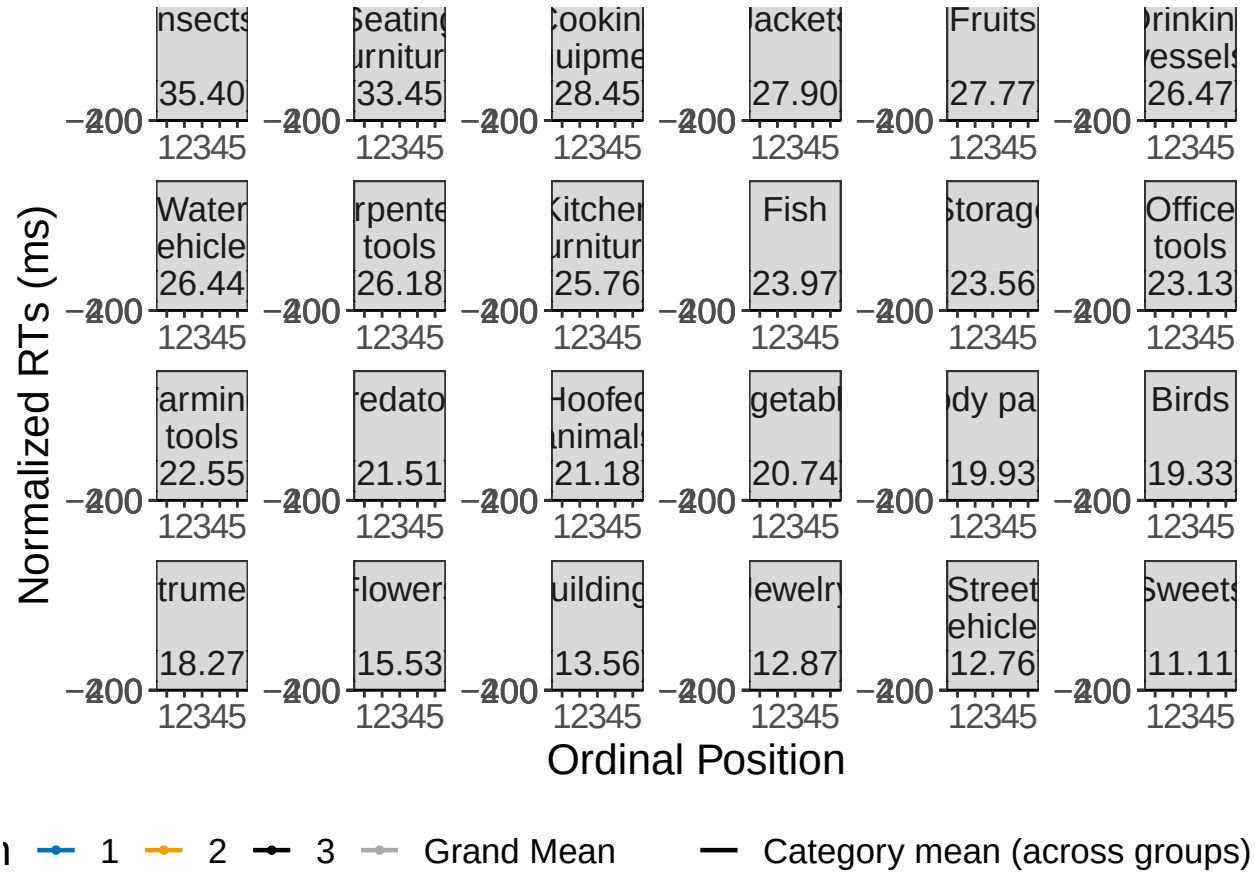
```
## Automatically converting the following non-factors to factors: category
```

```
                    #betweenvars="group")
means_final<- df_RTs %>%
   Rmisc::summarySEwithin(.,"VOT",idvar = "category",
                          withinvars = c("PosOr"),#, "session"),
                          #betweenvars="group",
```

```
                          na.rm = T)

for(i in 1:nrow(means_final_category)) {
  means_final_category$grandmean[i] <- means_final$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$category[i] & means_f
                                  means_final_category$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$effect, levels=rev(levels(means_final_categor
means_final_category$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(
      "Instruments\n\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(20.74)" ,
     "Body parts\n\n(19.93)","Birds\n\n(19.33)",
     "Instruments\n\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=6)+
    scale_y_continuous(limits = c(-500, 500),
                       breaks = c(-400,-200,0,200,400)) +
    scale_x_discrete(breaks=c(1,2,3,4,5))+
    theme(legend.position = "bottom"))
```




```
#plot_rt <- lemon::reposition_legend(plot_rt, "bott1.1.om right",panel='panel-5-5')

filename <- "CSI_online_aphasia_effect_by_category.pdf"
ggsave(plot_rt_category, filename =
```

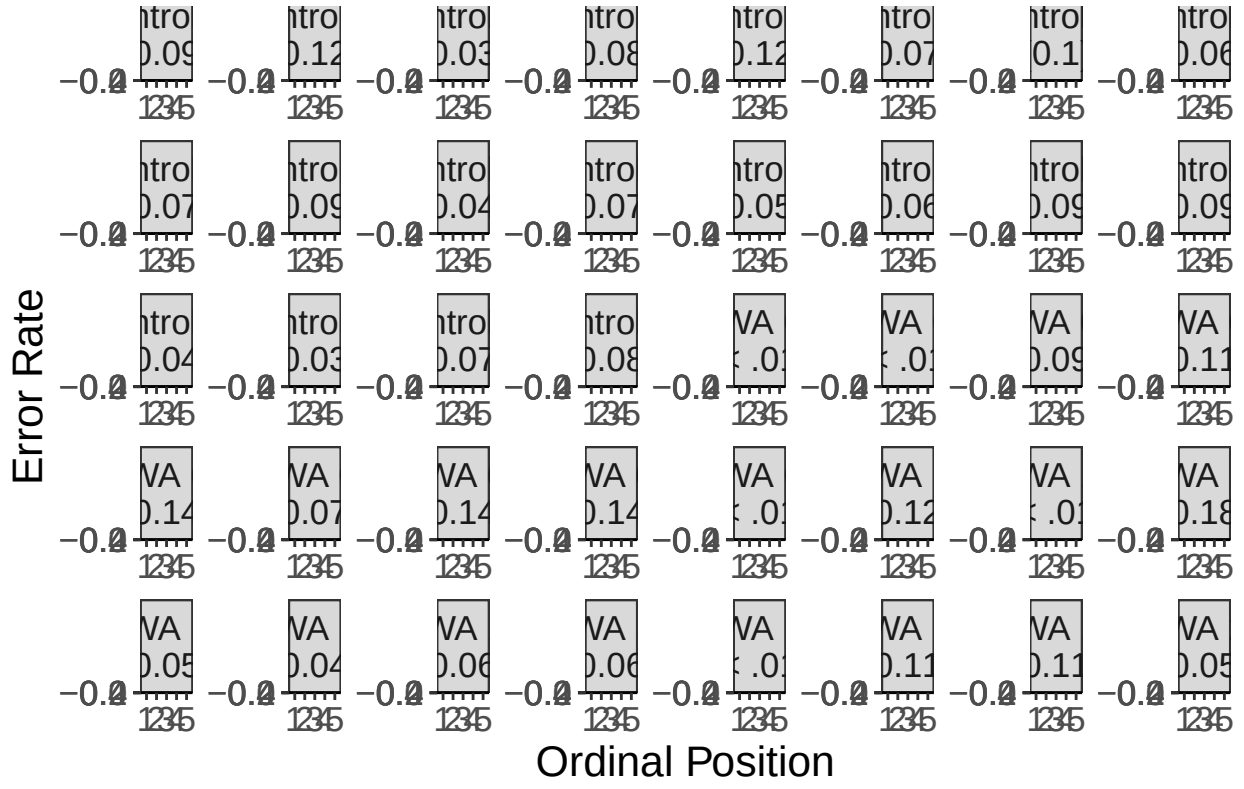
```
        here::here("results", "figures", filename),
       width = 26, height = 20,units = "cm",
       dpi = 300, device = cairo_pdf)
#embedFonts(file = here::here("results", "figures", filename))
```

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"),#, "session"),
                          #betweenvars="group",
                          na.rm = T)

for(i in 1:nrow(means_final_subject)) {
  means_final_subject$grandmean[i] <- means_final$error_class[means_final$PosOr == means_final_subject$
    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$subject[i] & mean
                                    means_final_subject$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$effect, levels=rev(levels(means_final_subject$


# add participant number
means_final_subject <- means_final_subject %>%
  mutate(subject_en = case_when(
     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,levels=c(
      "Control 01\n(0.09)", "Control 02\n(0.12)", "Control 03\n(0.03)",
      "Control 04\n(0.08)", "Control 05\n(0.12)", "Control 06\n(0.07)",
      "Control 07\n(0.1)","Control 08\n(0.06)", "Control 09\n(0.07)",
      "Control 10\n(0.09)", "Control 11\n(0.04)", "Control 12\n(0.07)",
      "Control 13\n(0.05)", "Control 14\n(0.06)","Control 15\n(0.09)",
      "Control 16\n(0.09)", "Control 17\n(0.04)", "Control 18\n(0.03)",
      "Control 19\n(0.07)", "Control 20\n(0.08)", "PWA 01\n(< .01)",
      "PWA 02\n(< .01)","PWA 03\n(0.09)","PWA 04\n(0.11)","PWA 05\n(0.14)",
      "PWA 06\n(0.07)","PWA 07\n(0.14)","PWA 08\n(0.14)",
      "PWA 09\n(< .01)","PWA 10\n(0.12)","PWA 11\n(< .01)","PWA 12\n(0.18)",
      "PWA 13\n(0.05)","PWA 14\n(0.04)","PWA 15\n(0.06)","PWA 16\n(0.06)",
      "PWA 17\n(< .01)","PWA 18\n(0.11)","PWA 19\n(0.11)","PWA 20\n(0.05)")))

# Plotting
(plot_error_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 ="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"))
```




```
#plot_rt <- lemon::reposition_legend(plot_rt, "bott1.1.om right",panel='panel-5-5')

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))
```

Errors by categry

Line graph for each participant:

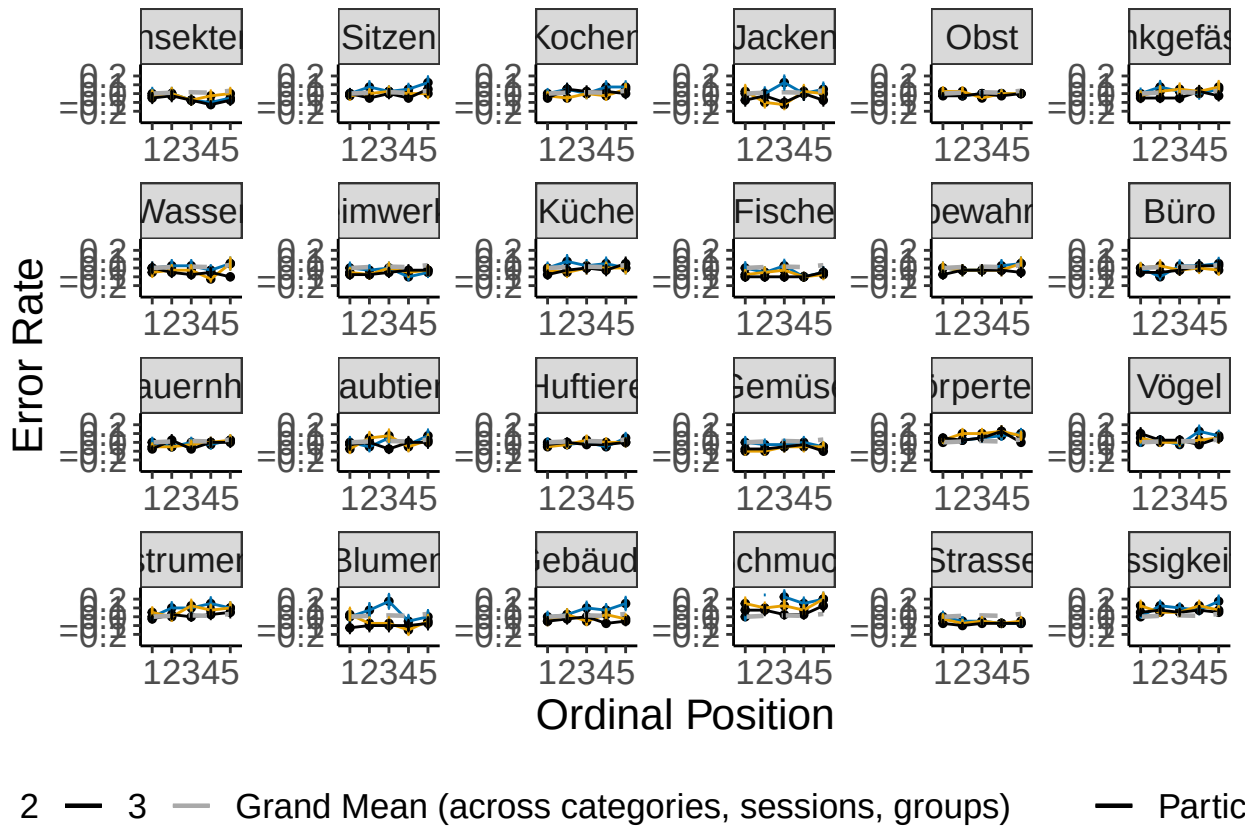
```
m2_error <- readRDS(here::here("results", "tables", "CSI_online_aphasia_SessionxGroup_glmm_errors.RDS"))
modeloutput <- coef(m2_error)$category
means_final_category <- df_errors %>%
   summarySEwithin(.,"error_class",withinvars = c("category","PosOr", "session"))
```

```
## 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$PosOr == means_final_category
    means_final$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[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_catego

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 = category)
    #
    #        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"))

# Plotting
(plot_error_category <- means_final_category %>%
    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("Participant mean",
                                   "Grand Mean"))+
    apatheme+
    labs(x="Ordinal Position",y ="Error Rate") +
     facet_wrap(means_final_category$category_en, scales='free', ncol=6)+
    scale_y_continuous(limits = c(-0.3, 0.3),
                       breaks = c(-0.2,-0.1,0,0.1,0.2)) +
    scale_x_discrete(breaks=c(1,2,3,4,5))+
    theme(legend.position = "bottom"))
```




```
#plot_rt <- lemon::reposition_legend(plot_rt, "bott1.1.om right",panel='panel-5-5')

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))
```