Power Analysis for CSI online typing with patients with aphasia - Interaction effect: Session x Ordinal position (wh estimates)

Kirsten Stark

07 Dezember, 2021

load packages

```
library(tidyr)
library(dplyr)
library(devtools)
library(MASS)
library(lme4)
library(lmerTest)
library(simr)
library(pbkrtest)
library(testthat)
library(ggplot2)

rm(list = ls())

today <- Sys.Date()
today <- format(today, format="%d%m%y")
```

Load data from Lorenz, Doering, van Scherpenberg, Pino, Abdel Rahman, & Obrig (2021)

In this lab-based study, people with Aphasia did a CSI task with 3 repetitions in two subsequent weeks. Both testing sessions had different items. Additionally, participants also did a CSI task with compounds (not relevant here).
The data loaded here are already cleaned for errors and participants.

```
# Set number of iterations
n_it = 1000
# load data
df <- read.csv2(here::here("data", "power-analysis",
                "Doeringetal_PWA_naming_simple_nouns_data_for_modelling.csv"))

# subset the relevant columns
df <- df %>%
  filter(error==1) %>%
```

```r
    # repet_trig = repetition (151,152,153 is 1,2,3),
    # wh = testing session 1 and 2,
    # cat_nr = 18 categories (à 5 members each)
  dplyr::select(c(subject, Ordinal_position, RT,
                  repet_trig, wh, item_id, cat_nr)) %>%
  droplevels() %>%
  dplyr::rename(repet = repet_trig,
                OrdPos = Ordinal_position) %>%
  # factorize colums
  mutate(subject = as.factor(subject),
         repet = as.factor(repet),
         wh = as.factor(wh),
         item_id = as.factor(item_id),
         cat_nr = as.factor(cat_nr))
```

The data structure is somewhat different from the planned experiment. Our experiment will have no repetition, but three testing sessions with the same items. Does the variance differ between the repetitions within a session and the two testing sessions?

```r
df %>% group_by(wh, repet, OrdPos) %>%
  summarize(sd = sd(RT, na.rm = T))
```

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

```
## # A tibble: 30 x 4
## # Groups:   wh, repet [6]
##    wh    repet OrdPos    sd
##    <fct> <fct> <chr>  <dbl>
##  1 1     151   OP1     694.
##  2 1     151   OP2     896.
##  3 1     151   OP3     929.
##  4 1     151   OP4     706.
##  5 1     151   OP5    1048.
##  6 1     152   OP1     830.
##  7 1     152   OP2     907.
##  8 1     152   OP3     726.
##  9 1     152   OP4     766.
## 10 1     152   OP5     926.
## # ... with 20 more rows
```

```r
df %>% group_by(wh)%>% summarize(sd = sd(RT, na.rm = T))
```

```
## # A tibble: 2 x 2
##   wh       sd
##   <fct> <dbl>
## 1 1      859.
## 2 2      958.
```

```r
df %>% group_by(repet)%>% summarize(sd = sd(RT, na.rm = T))
```

```
## # A tibble: 3 x 2
```

```
##   repet    sd
##   <fct> <dbl>
## 1 151   1003.
## 2 152    865.
## 3 153    857.
```

```
df %>% group_by(wh, repet)%>%summarize(sd = sd(RT, na.rm = T))
```

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

```
## # A tibble: 6 x 3
## # Groups:   wh [2]
##   wh    repet    sd
##   <fct> <fct> <dbl>
## 1 1     151    870.
## 2 1     152    833.
## 3 1     153    872.
## 4 2     151   1112.
## 5 2     152    895.
## 6 2     153    840.
```

```
df %>% group_by(OrdPos)%>%summarize(sd = sd(RT,na.rm=T))
```

```
## # A tibble: 5 x 2
##   OrdPos    sd
##   <chr>  <dbl>
## 1 OP1     776.
## 2 OP2     880.
## 3 OP3     944.
## 4 OP4     968.
## 5 OP5     971.
```

Overall, the variances are not too different (but still). The variance in the second session seems to be somewhat higher and the variances seem to increase with ordinal position.

Comparison 1: Power analysis for the interaction of Ordinal position and Session

Subset the data to the first repetition of the first and second session.

```
df_Ord <- df %>% filter(repet == 151) %>% droplevels()
```

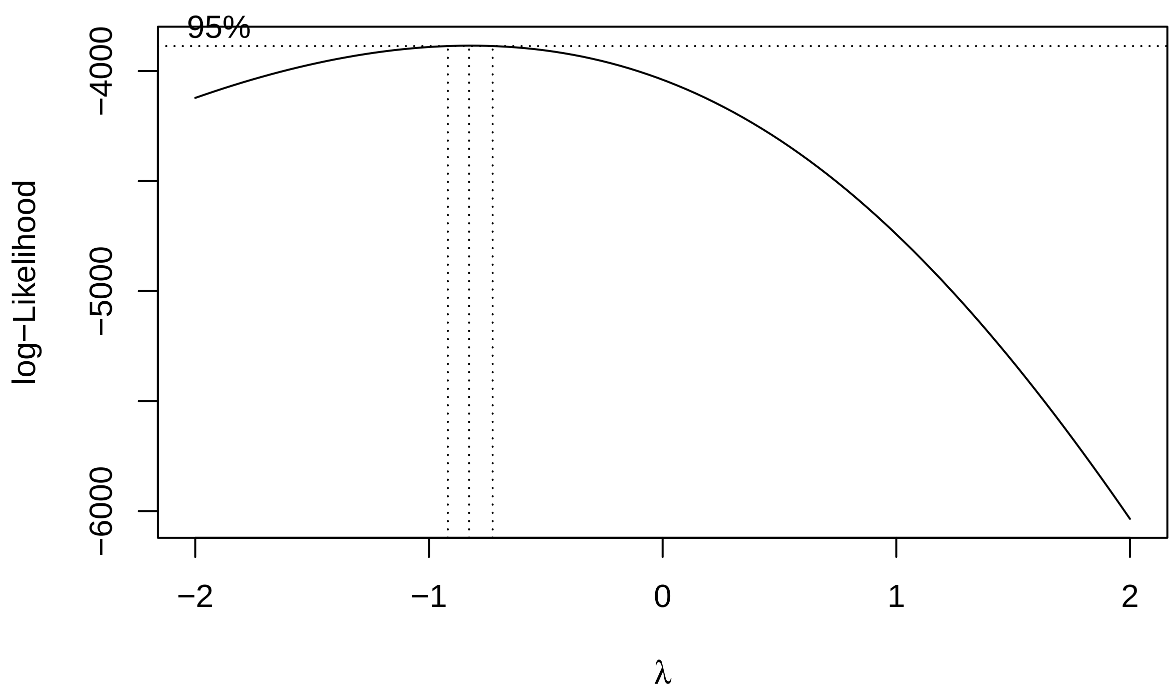
1) Set up models based on the structure of the online CSI experiment

Unfortunately, simr does not work properly with GLMMs. Therefore, for the power analysis, we will set up an LMM with transformed RTs
a) center OrdPos (continuous predictor)

```r
# center continuous predictor
df_Ord %>% mutate(OrdPos_num = case_when(OrdPos == "OP1" ~1,
                                         OrdPos == "OP2" ~2,
                                         OrdPos == "OP3" ~3,
                                         OrdPos == "OP4" ~4,
                                         OrdPos == "OP5" ~5)) %>%
                  mutate(OrdPos.c=
                    scale(as.numeric(as.character(OrdPos_num)),
                     center = TRUE, scale = FALSE)) %>%
        mutate(wh=as.factor(wh))-> df_Ord
```

b) Check distribution of RTs

```r
# Boxcox plot suggests inverse transformation:
boxcox(df_Ord$RT ~ df_Ord$OrdPos*df_Ord$wh)
```




```r
# # check distribution of RTs (by eyeballing)
# # 1) density plot of RTs
# qplot(data=df_Ord, RT, geom="density", na.rm=TRUE)+ theme_bw()
# # 2) plot data against real normal distribution -> is it way off?
# qqnorm(df_Ord$RT); qqline(df_Ord$RT)
#
# # check distribution of logRTs (by eyeballing)
# df_Ord$lRT <- log(df_Ord$RT)
```

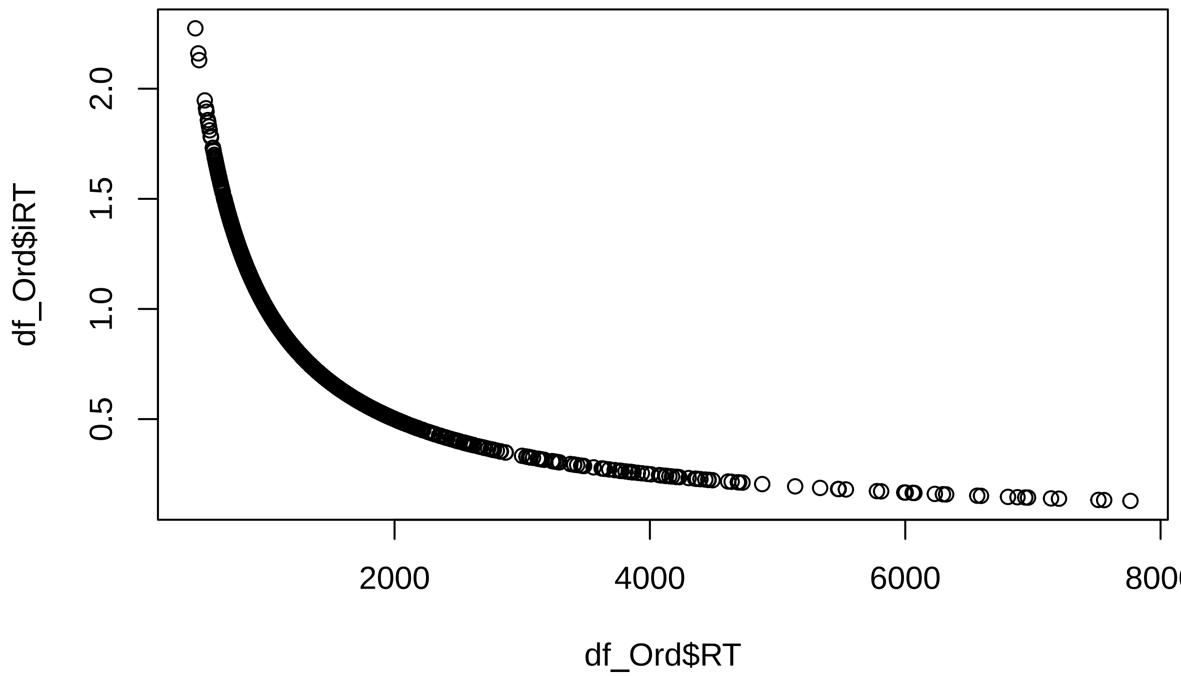
```r
# # 1) density plot of logRTs
# qplot(data=df_Ord, lRT, geom="density", na.rm=TRUE)+ theme_bw()
# # 2) plot data against real normal distribution -> is it way off?
# qqnorm(df_Ord$lRT); qqline(df_Ord$lRT)
# ### data kind of normally distributed. Log RTs fit better
#
# # check distribution of 1/RT (by eyeballing)
# df_Ord$iRT <- 1/df_Ord$RT
# # 1) density plot of logRTs
# qplot(data=df_Ord, iRT, geom="density", na.rm=TRUE)+ theme_bw()
# # 2) plot data against real normal distribution -> is it way off?
# qqnorm(df_Ord$iRT); qqline(df_Ord$iRT)
# ### data kind of normally distributed. Inverse RTs fit well
```

Check the goodness of fit of an inverse transformation (with RT divided by 1000: 1/(x/1000) = 1000/x):

```r
library(fitdistrplus)
```

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

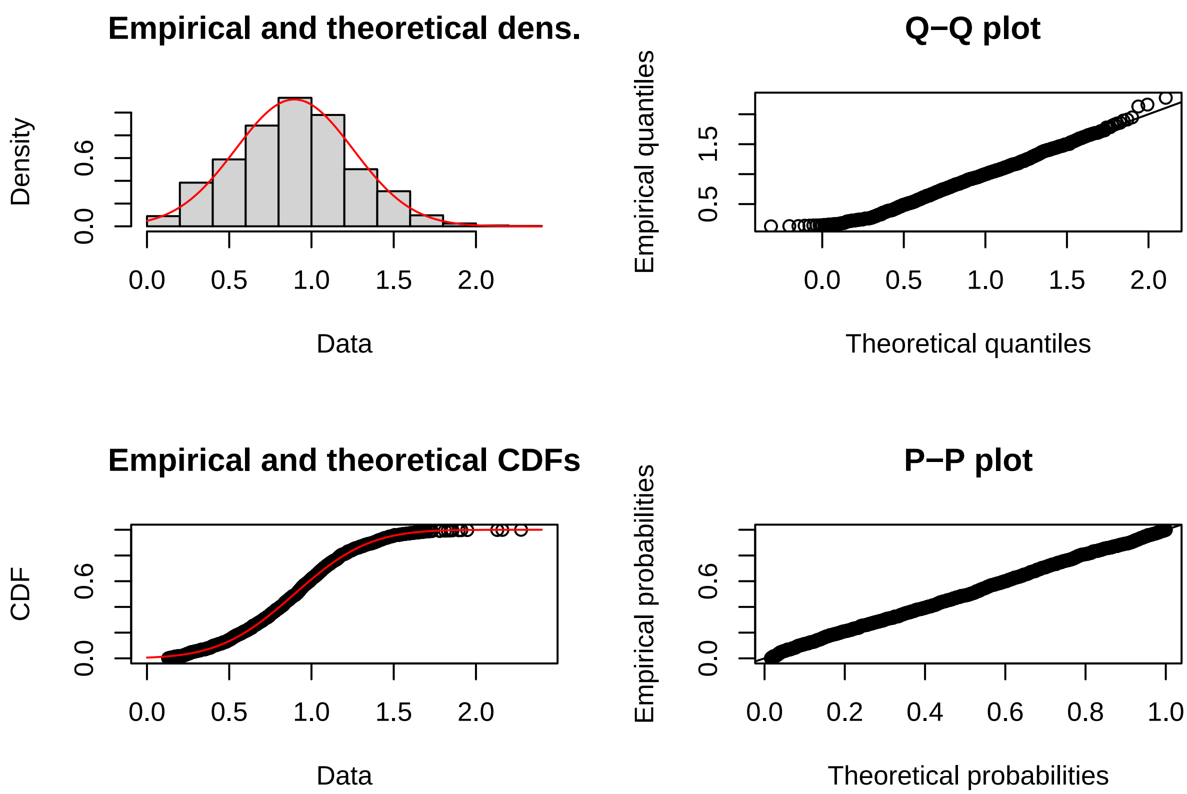
```r
# fit.normal<- fitdist(df_Ord$RT, distr = "norm", method = "mle")
# summary(fit.normal)
# plot(fit.normal)
df_Ord$iRT <-1000/df_Ord$RT
# plot this transformation distribution
plot(df_Ord$RT, df_Ord$iRT)
```




```r
# check fit
fit.normal_inv<- fitdist(df_Ord$iRT, distr = "norm", method = "mle")
summary(fit.normal_inv)
```

```
## Fitting of the distribution ' norm ' by maximum likelihood
## Parameters :
##       estimate  Std. Error
## mean 0.8957700 0.009580682
## sd   0.3577073 0.006774327
## Loglikelihood:  -544.9123   AIC:  1093.825   BIC:  1104.304
## Correlation matrix:
##      mean sd
## mean    1  0
## sd      0  1
```

```r
plot(fit.normal_inv)
```



An inverse transformation fits the data well. We will thus use this kind of transformation

c) Set up the models

Model 1: Linear model with continuous predictor "Ordinal position", treatment contrasted predictor "repetition" and inversely transformed RT data

```
# compute sliding difference contrast: Intercept is grand mean, second level is compared to first level
contrasts(df_Ord$wh) <- contr.sdif(2)
# maximal model has singular fit - stepwise reduction
#lmm1 <- lmer(iRT ~ OrdPos.c*wh +
#                (OrdPos.c*wh|subject) +(OrdPos.c*wh|cat_nr) ,
#             data = df_Ord, REML = FALSE,
#             control=lmerControl(optimizer = "bobyqa"))
lmm1 <- lmer(iRT ~ OrdPos.c*wh +
               (wh|subject) +(wh|cat_nr) ,
             data = df_Ord, REML = FALSE,
             control=lmerControl(optimizer = "bobyqa",
                                 optCtrl=list(maxfun=2e5)))


# lmm1 <- afex::lmer_alt(iRT ~ OrdPos.c*wh +
#                (wh||subject) +(OrdPos.c*wh||cat_nr) ,
#              data = df_Ord, REML = FALSE,
#              control=lmerControl(optimizer = "bobyqa",
#                                  optCtrl=list(maxfun=2e5)))
```

```r
# isSingular(lmm1)
# summary(lmm1)
```

A) Power analysis for the Ordinal position effect in this more complex model

2a) Extend dataset

Other than Lorenz et al, we have 24 categories (we use the same stimuli as in Stark, van Scherpenberg et al., 2021). So we extend along categories.
Additionally, we also extend along participants: 25 participants (20 + 25%)

```r
lmm24 <- extend(lmm1, along="cat_nr", n=24)
m2data <- getData(lmm24)
## ok, data were indeed extended to n categories ;-)
str(m2data)
```

```
## 'data.frame':    1871 obs. of  10 variables:
##  $ subject   : Factor w/ 18 levels "1","2","3","4",..: 1 1 1 1 2 2 2 3 3 3 ...
##  $ OrdPos    : chr  "OP5" "OP1" "OP2" "OP3" ...
##  $ RT        : num  7558 735 678 860 1208 ...
##  $ repet     : Factor w/ 1 level "151": 1 1 1 1 1 1 1 1 1 1 ...
##  $ wh        : Factor w/ 2 levels "1","2": 1 1 1 1 1 1 1 1 1 1 ...
##   ..- attr(*, "contrasts")= num [1:2, 1] -0.5 0.5
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:2] "1" "2"
##   .. .. ..$ : chr "2-1"
##  $ item_id   : Factor w/ 90 levels "1","2","3","4",..: 20 16 17 18 18 20 16 18 19 20 ...
##  $ cat_nr    : Factor w/ 24 levels "a","b","c","d",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ OrdPos_num: num  5 1 2 3 2 4 5 1 2 3 ...
##  $ OrdPos.c  : num [1:1871, 1] 2.0136 -1.9864 -0.9864 0.0136 -0.9864 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : NULL
##  $ iRT       : num  0.132 1.36 1.476 1.163 0.828 ...
```

```r
str(df_Ord)
```

```
## 'data.frame':    1394 obs. of  10 variables:
##  $ subject   : Factor w/ 18 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ OrdPos    : chr  "OP5" "OP1" "OP1" "OP1" ...
##  $ RT        : num  7558 652 735 728 552 ...
##  $ repet     : Factor w/ 1 level "151": 1 1 1 1 1 1 1 1 1 1 ...
##  $ wh        : Factor w/ 2 levels "1","2": 1 1 1 1 1 1 1 1 1 1 ...
##   ..- attr(*, "contrasts")= num [1:2, 1] -0.5 0.5
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:2] "1" "2"
##   .. .. ..$ : chr "2-1"
##  $ item_id   : Factor w/ 90 levels "1","2","3","4",..: 20 1 16 56 31 61 2 32 62 57 ...
##  $ cat_nr    : Factor w/ 18 levels "1","2","3","4",..: 4 1 4 12 7 13 1 7 13 12 ...
##  $ OrdPos_num: num  5 1 1 1 1 1 2 2 2 2 ...
##  $ OrdPos.c  : num [1:1394, 1] 2.01 -1.99 -1.99 -1.99 -1.99 ...
```

```
##   ..- attr(*, "scaled:center")= num 2.99
##  $ iRT       : num  0.132 1.533 1.36 1.373 1.811 ...
```

```r
lmm24_20 <- extend(lmm24, along="subject", n=20)
 m2data <- getData(lmm24_20)

lmm24_25 <- extend(lmm24, along="subject", n=25)
 m2data <- getData(lmm24_25)
# ## ok, data were indeed extended to n subjects ;-)
str(m2data)
```

```
## 'data.frame':    2567 obs. of  10 variables:
##  $ subject   : Factor w/ 25 levels "a","b","c","d",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ OrdPos    : chr  "OP5" "OP1" "OP2" "OP3" ...
##  $ RT        : num  7558 735 678 860 652 ...
##  $ repet     : Factor w/ 1 level "151": 1 1 1 1 1 1 1 1 1 1 ...
##  $ wh        : Factor w/ 2 levels "1","2": 1 1 1 1 1 1 1 1 1 1 ...
##   ..- attr(*, "contrasts")= num [1:2, 1] -0.5 0.5
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:2] "1" "2"
##   .. .. ..$ : chr "2-1"
##  $ item_id   : Factor w/ 90 levels "1","2","3","4",..: 20 16 17 18 1 2 3 4 5 56 ...
##  $ cat_nr    : Factor w/ 24 levels "a","b","c","d",..: 1 1 1 1 2 2 2 2 2 3 ...
##  $ OrdPos_num: num  5 1 2 3 1 2 3 4 5 1 ...
##  $ OrdPos.c  : num [1:2567, 1] 2.0136 -1.9864 -0.9864 0.0136 -1.9864 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : NULL
##  $ iRT       : num  0.132 1.36 1.476 1.163 1.533 ...
```

```r
# str(df_Ord)
```

3a) Specify effect sizes: Use the ordinal position effect from Lorenz et al 2021

Model 1: Linear model with continuous predictor "Ordinal position" and inversely transformed RT data Ordinal position effect: ~77 ms

```r
# use this as the fixed effect of interest
fixef(lmm24_20)
```

```
##    (Intercept)       OrdPos.c          wh2-1 OrdPos.c:wh2-1
##    0.882846217   -0.027034979   -0.062846528    0.007402801
```

```r
fixef(lmm24_25)
```

```
##    (Intercept)       OrdPos.c          wh2-1 OrdPos.c:wh2-1
##    0.882846217   -0.027034979   -0.062846528    0.007402801
```

4a) Run the power analysis for ordinal position based on effect size from Lorenz et al (2021)

```
set.seed(99)
```

Model 1: Linear model with continuous predictor "Ordinal position" and the factor "repetition" (sliding diff. contrast) and inversely transformed RT data (sample size: 16)

```
PowerLMM_OrdPos_16 <- powerCurve(lmm24_20,along = "subject",
                                 test=fixed("OrdPos.c","t"),
                          breaks = c(16), nsim = n_it)
lastResult()$warnings
lastResult()$errors
```

```
PowerLMM_OrdPos_16
```

```
## Power for predictor 'OrdPos.c', (95% confidence interval),
## by number of levels in subject:
##      16: 100.0% (99.63, 100.0) - 1649 rows
##
## Time elapsed: 0 h 5 m 33 s
```

Model 2: Linear model with continuous predictor "Ordinal position" and the factor "repetition" (sliding diff. contrast) and inversely transformed RT data (sample size: 16)

```
PowerLMM_OrdPos_20 <- powerSim(lmm24_20, test=fixed("OrdPos.c","t"), nsim = n_it) # increase to nsim >
lastResult()$warnings
lastResult()$errors
```

```
PowerLMM_OrdPos_20
```

```
## Power for predictor 'OrdPos.c', (95% confidence interval):
##       100.0% (99.63, 100.0)
##
## Test: t-test with Satterthwaite degrees of freedom (package lmerTest)
##       Effect size for OrdPos.c is -0.027
##
## Based on 1000 simulations, (1 warning, 0 errors)
## alpha = 0.05, nrow = 2081
##
## Time elapsed: 0 h 6 m 38 s
##
## nb: result might be an observed power calculation
```

Model 3: Linear model with continuous predictor "Ordinal position" and the factor "repetition" (sliding diff. contrast) and inversely transformed RT data (sample size: 25)

```
PowerLMM_OrdPos_25 <- powerSim(lmm24_25, test=fixed("OrdPos.c","t"), nsim = n_it) # increase to nsim >
lastResult()$warnings
lastResult()$errors
```

```
PowerLMM_OrdPos_25
```

```
## Power for predictor 'OrdPos.c', (95% confidence interval):
##       100.0% (99.63, 100.0)
##
## Test: t-test with Satterthwaite degrees of freedom (package lmerTest)
##       Effect size for OrdPos.c is -0.027
##
## Based on 1000 simulations, (4 warnings, 0 errors)
## alpha = 0.05, nrow = 2567
##
## Time elapsed: 0 h 7 m 37 s
##
## nb: result might be an observed power calculation
```

Do the same with the effect size from Stark et al (2021)

3a) Specify effect sizes: Use the ordinal position effect from Stark et al 2021 *Model 1: Linear model with continuous predictor "Ordinal position" and inversely transformed RT data* Ordinal position effect: ~31

```
# take the grand mean
gm <- mean(df_Ord$RT)
fixeff_Stark <- (1000/(gm+(31/2))) - (1000/(gm-(31/2)))
# use this as the fixed effect of interest
lmm20_Stark <- lmm24_20
fixef(lmm20_Stark)["OrdPos.c"] <- fixeff_Stark
lmm25_Stark <- lmm24_25
fixef(lmm25_Stark)["OrdPos.c"] <- fixeff_Stark
```

```
set.seed(99)
```

4a) Run the power analysis for ordinal position based on effect size from Stark et al (2021) *Model 1: Linear model with continuous predictor "Ordinal position" and the factor "repetition" (sliding diff. contrast) and inversely transformed RT data (sample size: 16)*

```
PowerLMM_OrdPos_16_Stark <- powerCurve(lmm24_20,along = "subject",
                                       test=fixed("OrdPos.c","t"),
                          breaks = c(16), nsim = n_it)
lastResult()$warnings
lastResult()$errors
```

```
PowerLMM_OrdPos_16_Stark
```

```
## Power for predictor 'OrdPos.c', (95% confidence interval),
## by number of levels in subject:
##      16: 100.0% (99.63, 100.0) - 1649 rows
##
## Time elapsed: 0 h 5 m 36 s
```

Model 2: Linear model with continuous predictor "Ordinal position" and the factor "repetition" (sliding diff. contrast) and inversely transformed RT data (sample size: 20)

```
PowerLMM_OrdPos_20_Stark <- powerSim(lmm20_Stark, test=fixed("OrdPos.c","t"), nsim = n_it) # increase t
lastResult()$warnings
lastResult()$errors
```

```
PowerLMM_OrdPos_20_Stark
```

```
## Power for predictor 'OrdPos.c', (95% confidence interval):
##       92.90% (91.13, 94.41)
##
## Test: t-test with Satterthwaite degrees of freedom (package lmerTest)
##       Effect size for OrdPos.c is -0.015
##
## Based on 1000 simulations, (3 warnings, 0 errors)
## alpha = 0.05, nrow = 2081
##
## Time elapsed: 0 h 6 m 40 s
```

Model 3: Linear model with continuous predictor "Ordinal position" and the factor "repetition" (sliding diff. contrast) and inversely transformed RT data (sample size: 25)

```
PowerLMM_OrdPos_25_Stark <- powerSim(lmm25_Stark, test=fixed("OrdPos.c","t"), nsim = n_it) # increase t
lastResult()$warnings
lastResult()$errors
```

```
PowerLMM_OrdPos_25_Stark
```

```
## Power for predictor 'OrdPos.c', (95% confidence interval):
##       96.90% (95.63, 97.88)
##
## Test: t-test with Satterthwaite degrees of freedom (package lmerTest)
##       Effect size for OrdPos.c is -0.015
##
## Based on 1000 simulations, (5 warnings, 0 errors)
## alpha = 0.05, nrow = 2567
##
## Time elapsed: 0 h 7 m 37 s
```

B) Power analysis for the Interaction effect Ordinal position x Session

2b) Extend data set

We will use the sample size as for the effect of main interest (ordinal position)

3b)+ 4b) Specify effect sizes and run different power analyses

To specify the effect sizes, we always take the grand mean and take +- 1/2 the inversly transformed difference we want to implement. We keep the two main effects unchanged.

```r
gm <- mean(df_Ord$RT)
eff_35ms <- (1000/(gm+(35/2))) - (1000/(gm-(35/2)))
fixef(lmm24_20)["OrdPos.c:wh2-1"]<- eff_35ms
```

35 ms interaction effect, 20 VP *Model B1: Linear model with interaction effect “Ordinal position x Session” of 35 ms*

```r
PowerLMM1_Interaction35_20 <- powerSim(lmm24_20, test=fixed("OrdPos.c:wh2-1","t"), nsim = n_it) # incre
lastResult()$warnings
lastResult()$errors
```

```r
PowerLMM1_Interaction35_20
```

```
## Power for predictor 'OrdPos.c:wh2-1', (95% confidence interval):
##       48.30% (45.16, 51.45)
##
## Test: t-test with Satterthwaite degrees of freedom (package lmerTest)
##       Effect size for OrdPos.c:wh2-1 is -0.017
##
## Based on 1000 simulations, (2 warnings, 0 errors)
## alpha = 0.05, nrow = 2081
##
## Time elapsed: 0 h 6 m 39 s
```

```r
gm <- mean(df_Ord$RT)
eff_35ms <- (1000/(gm+(35/2))) - (1000/(gm-(35/2)))
fixef(lmm24_25)["OrdPos.c:wh2-1"]<- eff_35ms
```

35 ms interaction effect, 25 VP *Model B1: Linear model with interaction effect “Ordinal position x Session” of 35 ms*

```r
PowerLMM1_Interaction35 <- powerSim(lmm24_25, test=fixed("OrdPos.c:wh2-1","t"), nsim = n_it) # increase
lastResult()$warnings
lastResult()$errors
```

```r
PowerLMM1_Interaction35
```

```
## Power for predictor 'OrdPos.c:wh2-1', (95% confidence interval):
##       55.90% (52.76, 59.01)
##
## Test: t-test with Satterthwaite degrees of freedom (package lmerTest)
##       Effect size for OrdPos.c:wh2-1 is -0.017
##
## Based on 1000 simulations, (3 warnings, 0 errors)
## alpha = 0.05, nrow = 2567
##
## Time elapsed: 0 h 7 m 35 s
```

```r
gm <- mean(df_Ord$RT)
eff_40ms <- (1000/(gm+(40/2))) - (1000/(gm-(40/2)))
fixef(lmm24_20)["OrdPos.c:wh2-1"]<- eff_40ms
```

40 ms interaction effect, 20 VP *Model B2: Linear model with interaction effect "Ordinal position x Session" of 40 ms*

```r
PowerLMM1_Interaction40_20 <- powerSim(lmm24_20, test=fixed("OrdPos.c:wh2-1","t"), nsim = n_it) # incre
lastResult()$warnings
lastResult()$errors
```

```r
PowerLMM1_Interaction40_20
```

```
## Power for predictor 'OrdPos.c:wh2-1', (95% confidence interval):
##       60.20% (57.09, 63.25)
##
## Test: t-test with Satterthwaite degrees of freedom (package lmerTest)
##       Effect size for OrdPos.c:wh2-1 is -0.020
##
## Based on 1000 simulations, (2 warnings, 0 errors)
## alpha = 0.05, nrow = 2081
##
## Time elapsed: 0 h 6 m 40 s
```

```r
gm <- mean(df_Ord$RT)
eff_40ms <- (1000/(gm+(40/2))) - (1000/(gm-(40/2)))
fixef(lmm24_25)["OrdPos.c:wh2-1"]<- eff_40ms
```

40 ms interaction effect, 25 VPs *Model B2: Linear model with interaction effect "Ordinal position x Session" of 40 ms*

```r
PowerLMM1_Interaction40 <- powerSim(lmm24_25, test=fixed("OrdPos.c:wh2-1","t"), nsim = n_it) # increase
lastResult()$warnings
lastResult()$errors
```

```r
PowerLMM1_Interaction40
```

```
## Power for predictor 'OrdPos.c:wh2-1', (95% confidence interval):
##       65.30% (62.26, 68.25)
##
## Test: t-test with Satterthwaite degrees of freedom (package lmerTest)
##       Effect size for OrdPos.c:wh2-1 is -0.020
##
## Based on 1000 simulations, (0 warnings, 0 errors)
## alpha = 0.05, nrow = 2567
##
## Time elapsed: 0 h 7 m 36 s
```

```r
gm <- mean(df_Ord$RT)
eff_45ms <- (1000/(gm+(45/2))) - (1000/(gm-(45/2)))
fixef(lmm24_20)["OrdPos.c:wh2-1"]<- eff_45ms
```

45 ms interaction effect, 20 VP *Model B3: Linear model with interaction effect "Ordinal position x Session" of 45 ms*

```r
PowerLMM1_Interaction45_20 <- powerSim(lmm24_20, test=fixed("OrdPos.c:wh2-1","t"), nsim = n_it) # incre
lastResult()$warnings
lastResult()$errors
```

```r
PowerLMM1_Interaction45_20
```

```
## Power for predictor 'OrdPos.c:wh2-1', (95% confidence interval):
##       67.60% (64.60, 70.50)
##
## Test: t-test with Satterthwaite degrees of freedom (package lmerTest)
##       Effect size for OrdPos.c:wh2-1 is -0.022
##
## Based on 1000 simulations, (1 warning, 0 errors)
## alpha = 0.05, nrow = 2081
##
## Time elapsed: 0 h 6 m 39 s
```

```r
gm <- mean(df_Ord$RT)
eff_45ms <- (1000/(gm+(45/2))) - (1000/(gm-(45/2)))
fixef(lmm24_25)["OrdPos.c:wh2-1"]<- eff_45ms
```

45 ms interaction effect, 25 VP *Model B3: Linear model with interaction effect "Ordinal position x Session" of 45 ms*

```r
PowerLMM1_Interaction45 <- powerSim(lmm24_25, test=fixed("OrdPos.c:wh2-1","t"), nsim = n_it) # increase
lastResult()$warnings
lastResult()$errors
```

```r
PowerLMM1_Interaction45
```

```
## Power for predictor 'OrdPos.c:wh2-1', (95% confidence interval):
##       77.90% (75.20, 80.44)
##
## Test: t-test with Satterthwaite degrees of freedom (package lmerTest)
##       Effect size for OrdPos.c:wh2-1 is -0.022
##
## Based on 1000 simulations, (6 warnings, 0 errors)
## alpha = 0.05, nrow = 2567
##
## Time elapsed: 0 h 7 m 35 s
```

```r
gm <- mean(df_Ord$RT)
eff_50ms <- (1000/(gm+(50/2))) - (1000/(gm-(50/2)))
fixef(lmm24_20)["OrdPos.c:wh2-1"]<- eff_50ms
```

50 ms interaction effect, 20 VP *Model B4: Linear model with interaction effect "Ordinal position x Session" of 50 ms*

```r
PowerLMM1_Interaction50_20 <- powerSim(lmm24_20, test=fixed("OrdPos.c:wh2-1","t"), nsim = n_it) # incre
lastResult()$warnings
lastResult()$errors
```

```r
PowerLMM1_Interaction50_20
```

```
## Power for predictor 'OrdPos.c:wh2-1', (95% confidence interval):
##       78.50% (75.82, 81.01)
##
## Test: t-test with Satterthwaite degrees of freedom (package lmerTest)
##       Effect size for OrdPos.c:wh2-1 is -0.024
##
## Based on 1000 simulations, (2 warnings, 0 errors)
## alpha = 0.05, nrow = 2081
##
## Time elapsed: 0 h 6 m 39 s
```

```r
gm <- mean(df_Ord$RT)
eff_50ms <- (1000/(gm+(50/2))) - (1000/(gm-(50/2)))
fixef(lmm24_25)["OrdPos.c:wh2-1"]<- eff_50ms
```

50 ms interaction effect, 25 VP *Model B4: Linear model with interaction effect "Ordinal position x Session" of 50 ms*

```r
PowerLMM1_Interaction50 <- powerSim(lmm24_25, test=fixed("OrdPos.c:wh2-1","t"), nsim = n_it) # increase
lastResult()$warnings
lastResult()$errors
```

```r
PowerLMM1_Interaction50
```

```
## Power for predictor 'OrdPos.c:wh2-1', (95% confidence interval):
##       85.20% (82.85, 87.34)
##
## Test: t-test with Satterthwaite degrees of freedom (package lmerTest)
##       Effect size for OrdPos.c:wh2-1 is -0.024
##
## Based on 1000 simulations, (2 warnings, 0 errors)
## alpha = 0.05, nrow = 2567
##
## Time elapsed: 0 h 7 m 36 s
```

```r
gm <- mean(df_Ord$RT)
eff_55ms <- (1000/(gm+(55/2))) - (1000/(gm-(55/2)))
fixef(lmm24_20)["OrdPos.c:wh2-1"]<- eff_55ms
```

55 ms interaction effect, 20 VP *Model B5: Linear model with interaction effect "Ordinal position x Session" of 55 ms*

```r
PowerLMM1_Interaction55_20 <- powerSim(lmm24_20, test=fixed("OrdPos.c:wh2-1","t"), nsim = n_it) # incre
lastResult()$warnings
lastResult()$errors
```

```r
PowerLMM1_Interaction55_20
```

```
## Power for predictor 'OrdPos.c:wh2-1', (95% confidence interval):
##       82.20% (79.69, 84.52)
##
## Test: t-test with Satterthwaite degrees of freedom (package lmerTest)
##       Effect size for OrdPos.c:wh2-1 is -0.027
##
## Based on 1000 simulations, (2 warnings, 0 errors)
## alpha = 0.05, nrow = 2081
##
## Time elapsed: 0 h 6 m 37 s
```

```r
gm <- mean(df_Ord$RT)
eff_55ms <- (1000/(gm+(55/2))) - (1000/(gm-(55/2)))
fixef(lmm24_25)["OrdPos.c:wh2-1"]<- eff_55ms
```

55 ms interaction effect, 25 VP *Model B5: Linear model with interaction effect "Ordinal position x Session" of 55 ms*

```r
PowerLMM1_Interaction55 <- powerSim(lmm24_25, test=fixed("OrdPos.c:wh2-1","t"), nsim = n_it) # increase
lastResult()$warnings
lastResult()$errors
```

```r
PowerLMM1_Interaction55
```

```
## Power for predictor 'OrdPos.c:wh2-1', (95% confidence interval):
##       89.60% (87.54, 91.42)
##
## Test: t-test with Satterthwaite degrees of freedom (package lmerTest)
##       Effect size for OrdPos.c:wh2-1 is -0.027
##
## Based on 1000 simulations, (1 warning, 0 errors)
## alpha = 0.05, nrow = 2567
##
## Time elapsed: 0 h 7 m 36 s
```

C) Calculate effect size for interaction effect

Use the method by Westfall, Judd, and Kenny (2014), as reviewed in Brysbaert & Stevens (2018)
Westfall d = (difference between the means)/sqrt(variances of all random effects)
We use the experimental variances and the simulated effect sizes

```
x <- summary(lmm1)
randomvars <- 0.031756+0.001310+0.011178+0.002588+0.082964
(d_35 <- eff_35ms /sqrt(randomvars))
```

```
## [1] -0.04750136
```

```
(d_40 <- eff_40ms /sqrt(randomvars))
```

```
## [1] -0.05428975
```

```
(d_45 <- eff_45ms /sqrt(randomvars))
```

```
## [1] -0.06107914
```

```
(d_50 <- eff_50ms /sqrt(randomvars))
```

```
## [1] -0.06786966
```

```
(d_55 <- eff_55ms /sqrt(randomvars))
```

```
## [1] -0.07466141
```