Power Analysis for CSI online typing with patients with aphasia - Summary

Kirsten Stark

08 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")

# Set number of iterations
n_iter = 1000
set.seed(99)
```

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.

```
# 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 %>%
```

```r
  filter(error==1) %>%
    # 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. Still, the variances between repetitions and sessions is somewhat comparable. Therefore, we will run our power analyses based on the variance estimates from the three repetitions. An additional analysis with the factor session (two levels) has also been run.

Comparison 1: Power analysis for the ordinal position effect in the first repetition only

Subset the data to the first session and first repetition

```r
df_Ord <- df %>% filter(wh == 1, repet == 151) %>% droplevels()
```

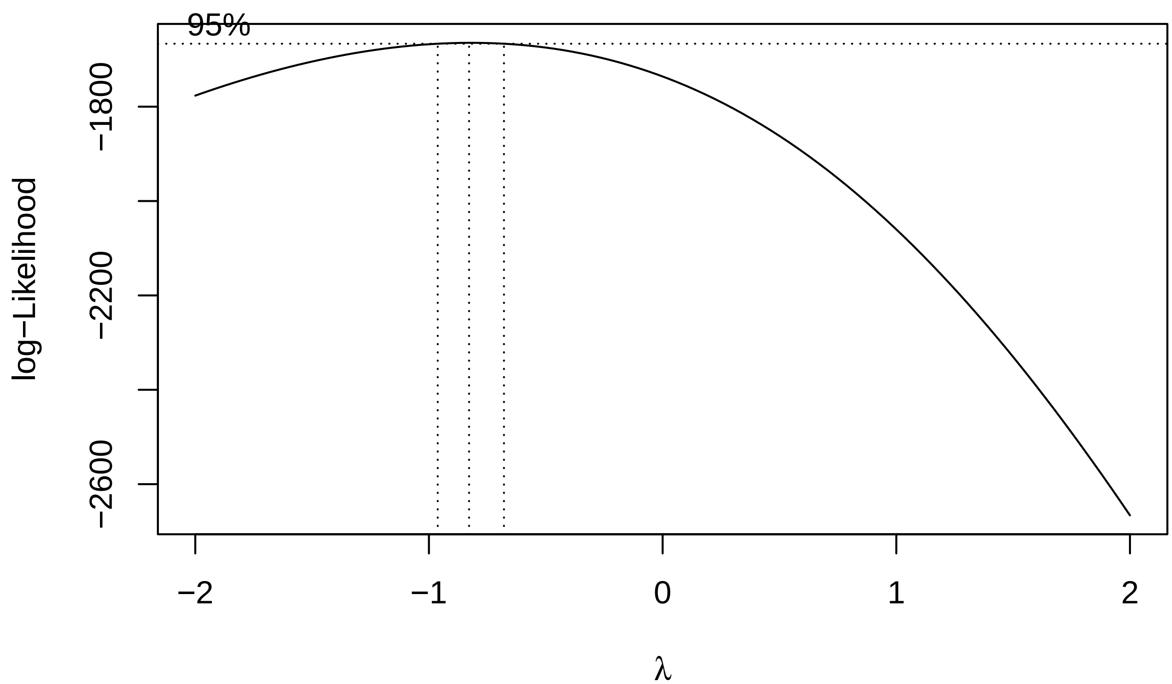
1) Set up models based on the structure of the online CSI experiment

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)

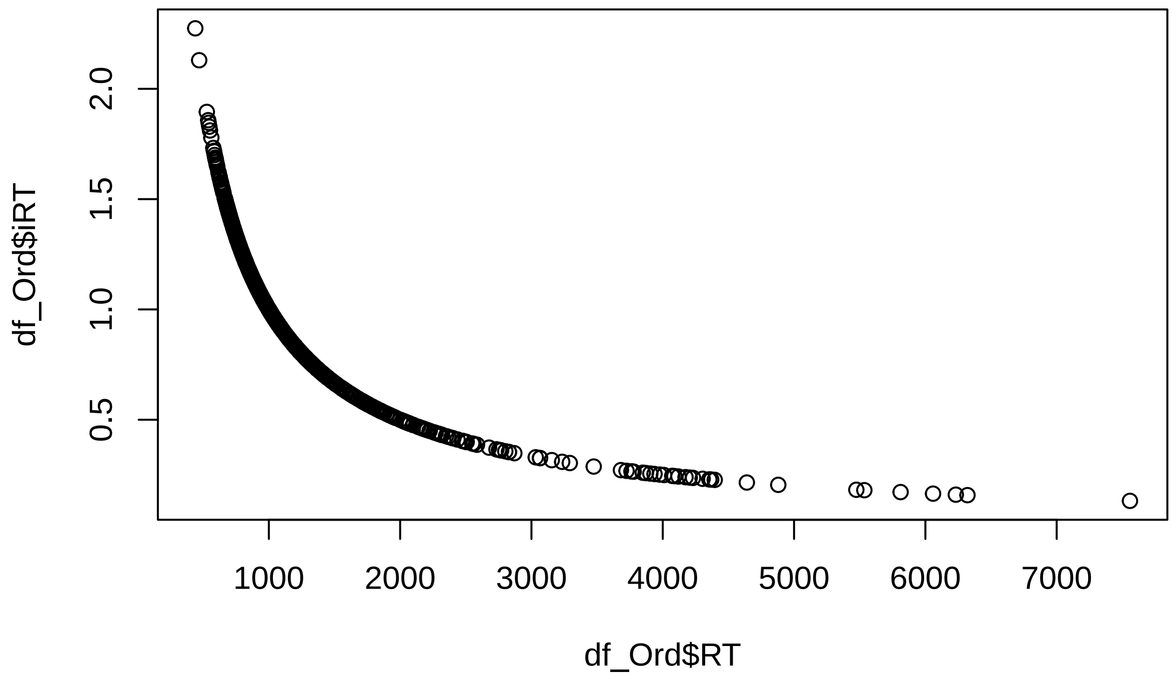
```r
# center continuous predictor
df_Ord %>% mutate(OrdPos_num = case_when(OrdPos == "OP1" ~1,
                                         OrdPos == "OP2" ~2,
                                         OrdPos == "OP3" ~3,
                                         OrdPos == "OP4" ~4,
                     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

b) Check distribution of RTs: An inverse transformation (1000/x) is needed and appropriate

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



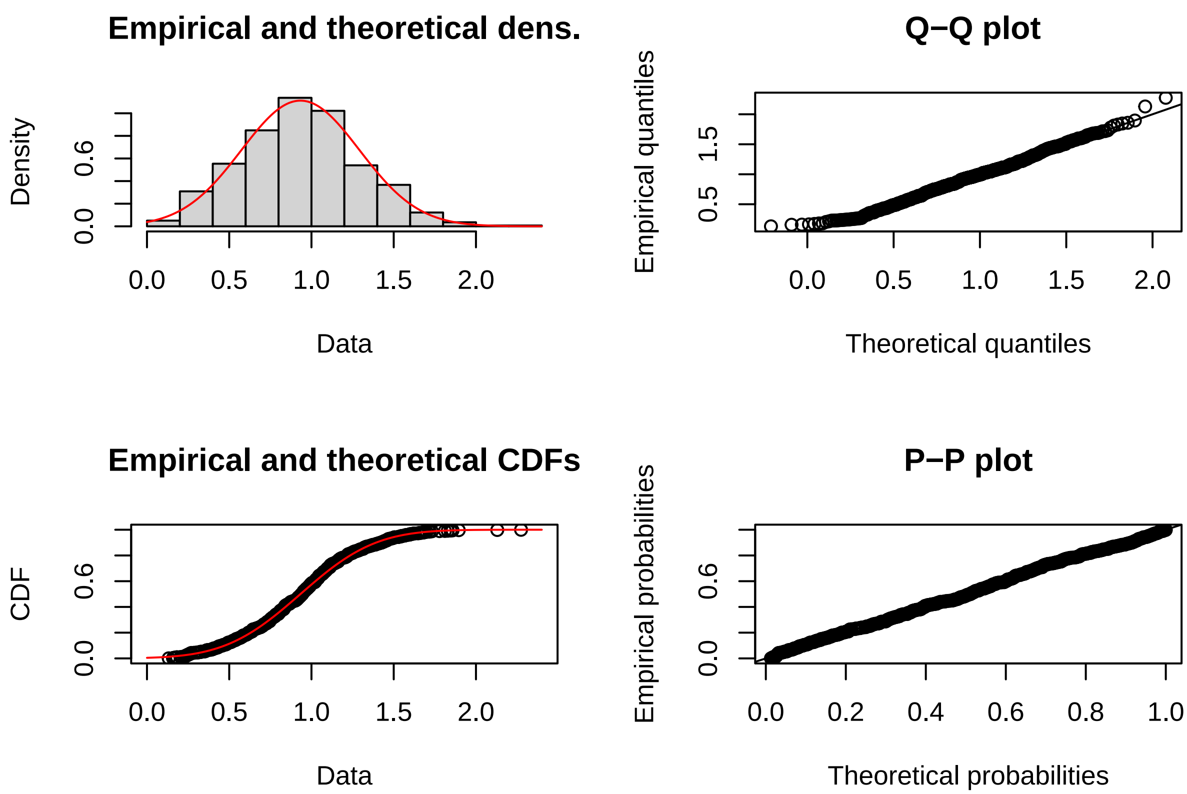

```
# apply transformation
df_Ord$iRT <-1000/df_Ord$RT
plot(df_Ord$RT, df_Ord$iRT)
```




```r
# check fit
fit.normal_inv<- fitdistrplus::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 0.9329607 0.013612620
## sd   0.3588675 0.009625239
## Loglikelihood:  -273.9248   AIC:  551.8497   BIC:  560.9375
## Correlation matrix:
##      mean sd
## mean    1  0
## sd      0  1
```

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



c) Set up the model *Model 1: Linear model with continuous predictor "Ordinal position" as the only predictor and inversely transformed RT data*

```
lmm1 <- lmer(iRT ~ OrdPos.c +
               (OrdPos.c|subject) +(OrdPos.c|cat_nr) ,
            data = df_Ord, REML = FALSE,
            control=lmerControl(optimizer = "bobyqa"))
isSingular(lmm1)
```

```
## [1] FALSE
```

```
summary(lmm1)
```

```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
## Formula: iRT ~ OrdPos.c + (OrdPos.c | subject) + (OrdPos.c | cat_nr)
##    Data: df_Ord
## Control: lmerControl(optimizer = "bobyqa")
##
##      AIC      BIC   logLik deviance df.resid
##    354.3    395.2   -168.1    336.3      686
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -2.94617 -0.64056  0.07423  0.61505  3.04551
##
## Random effects:
##  Groups   Name        Variance  Std.Dev. Corr
##  subject  (Intercept) 0.0334551 0.18291
##           OrdPos.c    0.0003164 0.01779  -0.58
##  cat_nr   (Intercept) 0.0106732 0.10331
##           OrdPos.c    0.0003476 0.01864  -0.24
##  Residual             0.0840919 0.28999
## Number of obs: 695, groups:  subject, 18; cat_nr, 17
##
## Fixed effects:
##             Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)  0.92215    0.05157 25.89346  17.881 4.33e-16 ***
## OrdPos.c    -0.03048    0.01013 11.39100  -3.008   0.0115 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr)
## OrdPos.c -0.259
```

2) Extend dataset

We already know that we want 24 categories (we use the same stimuli as in Stark, van Scherpenberg et al., 2021). So we extend along categories.
Additionally, we also reduce along participants: Our power curve suggested that 16 participants would be sufficient to achieve a power of .80 based on the effect size from Lorenz et al., 2016.

```
lmm1_2 <- extend(lmm1, along="cat_nr", n=24)
# m2data <- getData(lmm1_2)
# ## ok, data were indeed extended to n categories ;-)
# str(m2data)
# str(df_Ord)
```

3) Specify effect size and run power analysis

We use the experimental effect size from Lorenz et al

Linear model with continuous predictor "Ordinal position" and inversely transformed RT data

```
PowerLMM_OrdPos_Lorenz <- powerCurve(lmm1_2, along = "subject", breaks=16, test=fixed("OrdPos.c","t"), r
lastResult()$warnings
lastResult()$errors
```

```
PowerLMM_OrdPos_Lorenz
```

```
## Power for predictor 'OrdPos.c', (95% confidence interval),
## by number of levels in subject:
##      16: 90.90% (88.94, 92.61) - 943 rows
##
## Time elapsed: 0 h 3 m 59 s
```

5) Use effect from Stark, van Scherpenberg et al., 2021

Our effect size from a neurotypical sample in an online experiment was smaller. We we take this effect on 16 participants + 25 % (=20 participants), the power is substantially reduced.

Extend to 20 participants

```
lmm24_20_Stark <- extend(lmm1_2, along="subject", n=20)
#str(lmm24_20_Stark)
```

Specify effect size
What effect size is 31 ms in 1/(RT/1000) depends on the position: We take the grand mean

```
# calculate the effect size in the transformed scale
gm <- mean(df_Ord$RT)
eff1 <- 1000/(gm+(31/2)) - 1000/(gm-(31/2))

# use this as the fixed effect of interest
fixef(lmm24_20_Stark)["OrdPos.c"] <- eff1
```

Power analysis: Linear model with continuous predictor "Ordinal position" (effect size from Stark et al., 2021) and inversely transformed RT data

```
PowerLMM_OrdPos_Stark <-
  powerSim(lmm24_20_Stark, test=fixed("OrdPos.c","t"), nsim = n_iter) # increase to nsim > 1000 for rea
lastResult()$warnings
lastResult()$errors
```

```
PowerLMM_OrdPos_Stark
```

```
## Power for predictor 'OrdPos.c', (95% confidence interval):
##       50.80% (47.65, 53.94)
##
## Test: t-test with Satterthwaite degrees of freedom (package lmerTest)
##       Effect size for OrdPos.c is -0.017
##
## Based on 1000 simulations, (5 warnings, 0 errors)
## alpha = 0.05, nrow = 1193
##
## Time elapsed: 0 h 4 m 46 s
```

Unfortunately, if the effect was that small and the variance estimates realistic, the power would be too low to reliably detect the effect. A power curve suggests that the power would still be below 80 % with 60 participants. Thus, this seems to be a risk that we need to take. Moreover, our main interest is the analyis across the three sessions.

Comparison 2: Power analysis for the ordinal position effect across all three sessions (Ordinal position and interaction effect)

As a proxy for our sessions, we will use the variances from the three repetitions of the first session in Lorenz et al (2021)

Subset the data to the first session

```
df_full <- df %>% filter(wh == 1) %>% droplevels()
```

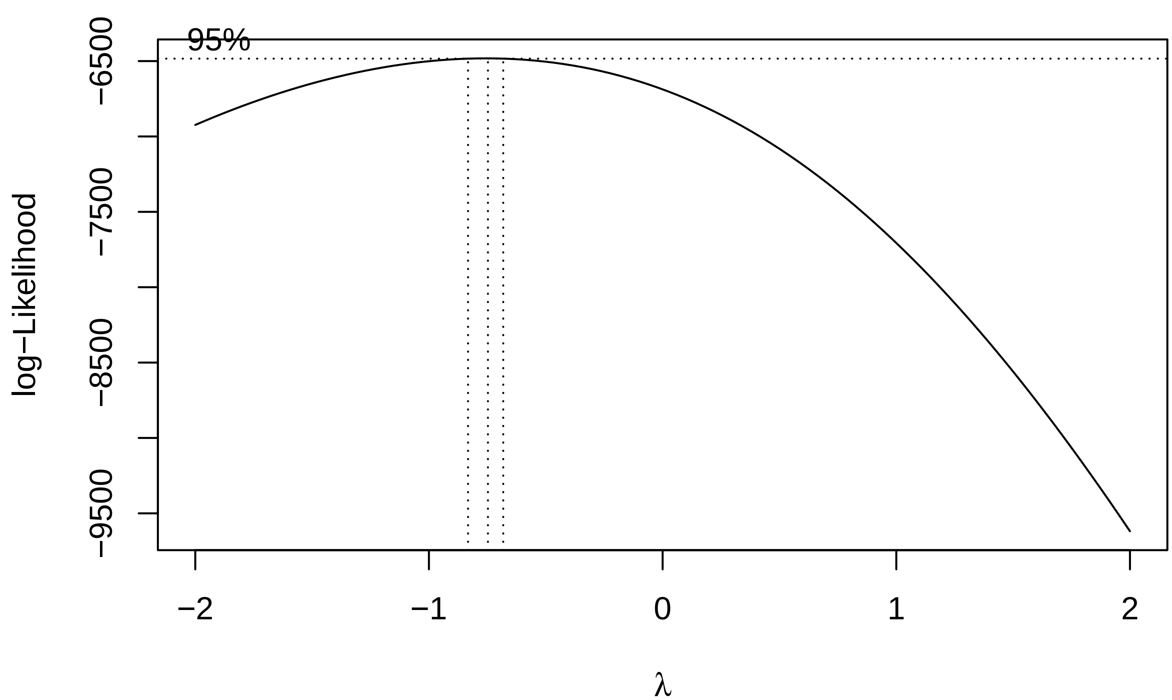
1) Set up models based on the structure of the new experiment

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)

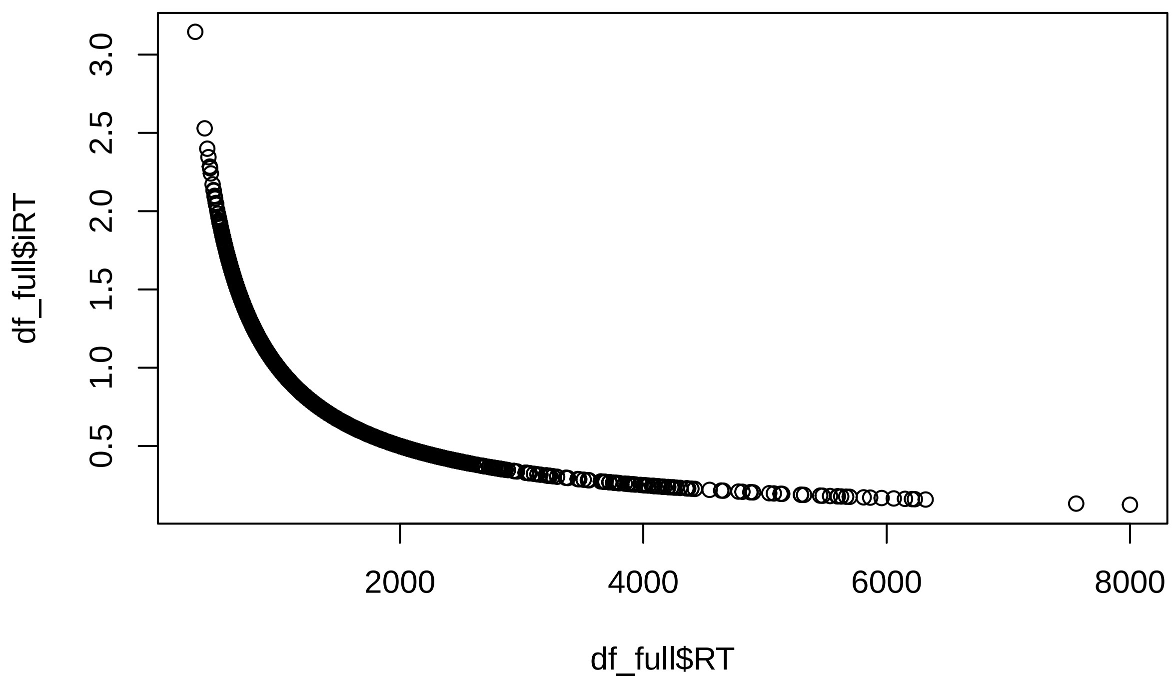
```
# center continuous predictor
df_full %>% mutate(OrdPos_num = case_when(OrdPos == "OP1" ~1,
                                         OrdPos == "OP2" ~2,
                                         OrdPos == "OP3" ~3,
                                         OrdPos == "OP4" ~4,
                     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

b) Check distribution of RTs: An inverse transformation (1000/x) is needed and appropriate

```
# Boxcox plot suggests inverse transformation:
boxcox(df_full$RT ~ df_full$OrdPos)
```



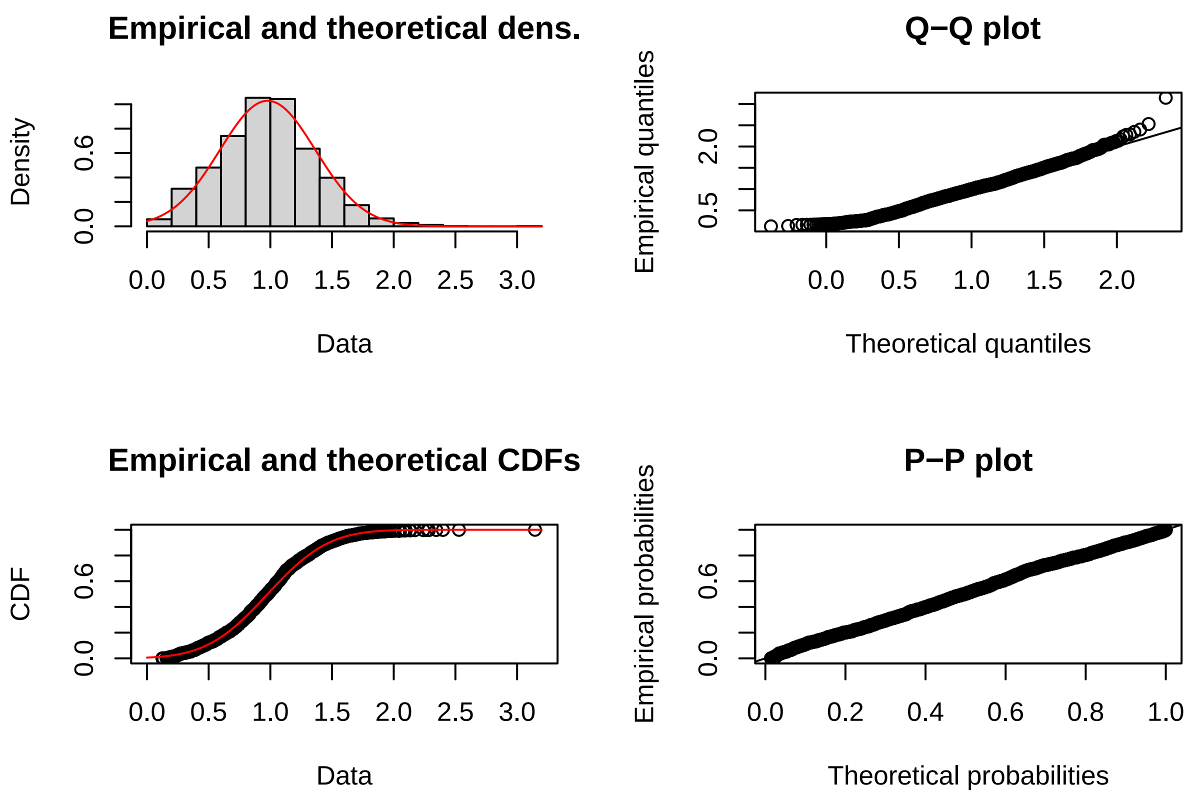
```
# apply transformation
df_full$iRT <-1000/df_full$RT
plot(df_full$RT, df_full$iRT)
```




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

```
## Fitting of the distribution ' norm ' by maximum likelihood
## Parameters :
##       estimate  Std. Error
## mean 0.9781205 0.008345432
## sd   0.3879504 0.005900935
## Loglikelihood:  -1020.123   AIC:  2044.246   BIC:  2055.603
## Correlation matrix:
##      mean sd
## mean    1  0
## sd      0  1
```

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



c) Set up the model *Linear model with continuous predictor "Ordinal position" and fatcor "repet" (sliding difference contrast) as the predictor variables and inversely transformed RT data*

```
# compute sliding difference contrast: Intercept is grand mean, second level is compared to first level
contrasts(df_full$repet) <- contr.sdif(3)
# maximal model has singular fit - stepwise reduction
# lmm1 <- lmer(iRT ~ OrdPos.c*repet +
#                (OrdPos.c*repet|subject) +(OrdPos.c*repet|cat_nr) ,
#             data = df_full, REML = FALSE,
#             control=lmerControl(optimizer = "bobyqa"))

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

```
## [1] FALSE
```

```
summary(lmm1)
```

```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
```

```
## Formula: iRT ~ OrdPos.c * repet + (1 + re1.repet2.1 + re1.repet3.2 ||
##     subject) + (1 + re2.OrdPos.c || cat_nr)
##    Data: data
## Control: lmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 2e+05))
##
##      AIC      BIC   logLik deviance df.resid
##   1139.4   1207.5   -557.7   1115.4     2149
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.2704 -0.6453  0.0760  0.6398  5.1565
##
## Random effects:
##  Groups    Name          Variance  Std.Dev.
##  subject   (Intercept)   0.0457486 0.21389
##  subject.1 re1.repet2.1 0.0011736 0.03426
##  subject.2 re1.repet3.2 0.0006358 0.02522
##  cat_nr    (Intercept)   0.0138004 0.11748
##  cat_nr.1  re2.OrdPos.c 0.0001398 0.01182
##  Residual                0.0922992 0.30381
## Number of obs: 2161, groups:  subject, 18; cat_nr, 17
##
## Fixed effects:
##                      Estimate Std. Error         df t value Pr(>|t|)
## (Intercept)        9.488e-01  5.848e-02  2.721e+01  16.223 1.62e-15 ***
## OrdPos.c          -1.976e-02  5.580e-03  1.681e+01  -3.541  0.00255 **
## repet2-1           5.028e-02  1.806e-02  2.069e+01   2.785  0.01120 *
## repet3-2           4.273e-02  1.696e-02  2.002e+01   2.520  0.02034 *
## OrdPos.c:repet2-1  1.442e-02  1.136e-02  2.094e+03   1.269  0.20449
## OrdPos.c:repet3-2  3.068e-03  1.117e-02  2.092e+03   0.275  0.78359
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) OrdPs. rpt2-1 rpt3-2 OP.:2-
## OrdPos.c     0.000
## repet2-1    -0.002 -0.004
## repet3-2    -0.001 -0.004 -0.415
## OrdPs.c:2-1 -0.001 -0.014  0.007  0.000
## OrdPs.c:3-2 -0.001 -0.009  0.000 -0.007 -0.499
```

2) Extend dataset

We already know that we want 24 categories (we use the same stimuli as in Stark, van Scherpenberg et al., 2021). So we extend along categories.
Additionally, we also reduce along participants: Our power curve suggested that 16 participants would be sufficient to achieve a power of .80 based on the effect size from Lorenz et al., 2016.

```
lmm1_2 <- extend(lmm1, along="cat_nr", n=24)
# m2data <- getData(lmm1_2)
# ## ok, data were indeed extended to n categories ;-)
# str(m2data)
# str(df_full)
```

3) Specify effect size and run power analysis

We use the experimental effect size from Lorenz et al to see the power if all effects remained the same

Linear model with continuous predictor "Ordinal position", sliding difference contrasted predictor repetition and inversely transformed RT data

```
PowerLMM_OrdPos_Lorenz <- powerCurve(lmm1_2, along = "subject", breaks=16, test=fixed("OrdPos.c","t"), n
lastResult()$warnings
lastResult()$errors
```

```
PowerLMM_OrdPos_Lorenz
```

```
## Power for predictor 'OrdPos.c', (95% confidence interval),
## by number of levels in subject:
##      16: 98.20% (97.17, 98.93) - 2906 rows
##
## Time elapsed: 0 h 14 m 41 s
```

5) Use effect from Stark, van Scherpenberg et al., 2021

Our effect size from a neurotypical sample in an online experiment was smaller. We take this effect on 16 participants + 25 % (=20 participants), the power is substantially reduced.

Extend to 20 participants

```
lmm24_20_Stark <- extend(lmm1_2, along="subject", n=20)
#str(lmm24_20_Stark)
```

Specifiy effect size
What effect size is 31 ms in 1/(RT/1000) depends on the position: We take the grand mean

```
# calculate the effect size in the transformed scale
gm <- mean(df_Ord$RT)
eff1 <- 1000/(gm+(31/2)) - 1000/(gm-(31/2))

# use this as the fixed effect of interest
fixef(lmm24_20_Stark)["OrdPos.c"] <- eff1
```

Power analysis: Linear model with continuous predictor "Ordinal position" (effect size from Stark et al., 2021) and inversely transformed RT data

```
PowerLMM_OrdPos_Stark <-
  powerSim(lmm24_20_Stark, test=fixed("OrdPos.c","t"), nsim = n_iter) # increase to nsim > 1000 for rea
lastResult()$warnings
lastResult()$errors
```

```
PowerLMM_OrdPos_Stark
```

```
## Power for predictor 'OrdPos.c', (95% confidence interval):
##       97.50% (96.33, 98.38)
##
```

```
## Test: t-test with Satterthwaite degrees of freedom (package lmerTest)
##       Effect size for OrdPos.c is -0.017
##
## Based on 1000 simulations, (6 warnings, 0 errors)
## alpha = 0.05, nrow = 3685
##
## Time elapsed: 0 h 17 m 44 s
```

Now, the power is high even with the more conservative effect size estimate.

6) Estimate the effect size of the interaction effect of the ordinal position x Session we can reliably detect with our 20 VP - with oridnal position effect from Lorenz et al (2021)

Extend data set

```
lmm1_2 <- extend(lmm1, along="cat_nr", n=24)
lmm24_20 <- extend(lmm1, along="subject", n=20)
```

Specify effect sizes and run different power analyses

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

```
gm <- mean(df_Ord$RT)
eff_40ms <- (1000/(gm+(40/2))) - (1000/(gm-(40/2)))
fixef(lmm24_20)["OrdPos.c:repet2-1"]<- eff_40ms
fixef(lmm24_20)["OrdPos.c:repet3-2"]<- eff_40ms
fixef(lmm24_20_Stark)["OrdPos.c:repet2-1"]<- eff_40ms
fixef(lmm24_20_Stark)["OrdPos.c:repet3-2"]<- eff_40ms
```

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

```
PowerLMM1_Interaction40_20 <- powerSim(lmm24_20, test=fixed("OrdPos.c:repet2-1","t"), nsim = n_iter) #
lastResult()$warnings
lastResult()$errors
```

```
PowerLMM1_Interaction40_20
```

```
## Power for predictor 'OrdPos.c:repet2-1', (95% confidence interval):
##       53.10% (49.95, 56.23)
##
## Test: t-test with Satterthwaite degrees of freedom (package lmerTest)
##       Effect size for OrdPos.c:repet2-1 is -0.022
##
```

```
## Based on 1000 simulations, (7 warnings, 0 errors)
## alpha = 0.05, nrow = 2391
##
## Time elapsed: 0 h 11 m 44 s
```

```
PowerLMM1_Interaction40_20 <- powerSim(lmm24_20, test=fixed("OrdPos.c:repet3-2","t"), nsim = n_iter) #
lastResult()$warnings
lastResult()$errors
```

```
PowerLMM1_Interaction40_20
```

```
## Power for predictor 'OrdPos.c:repet3-2', (95% confidence interval):
##       56.50% (53.36, 59.60)
##
## Test: t-test with Satterthwaite degrees of freedom (package lmerTest)
##       Effect size for OrdPos.c:repet3-2 is -0.022
##
## Based on 1000 simulations, (7 warnings, 0 errors)
## alpha = 0.05, nrow = 2391
##
## Time elapsed: 0 h 11 m 46 s
```

Model B2: Linear model with interaction effect "Ordinal position x Session" of 40 ms and ordinal position effect from Stark et al., 2021

```
PowerLMM1_Interaction24_20_Stark <- powerSim(lmm24_20_Stark, test=fixed("OrdPos.c:repet2-1","t"), nsim =
lastResult()$warnings
lastResult()$errors
```

```
PowerLMM1_Interaction24_20_Stark
```

```
## Power for predictor 'OrdPos.c:repet2-1', (95% confidence interval):
##       72.80% (69.93, 75.54)
##
## Test: t-test with Satterthwaite degrees of freedom (package lmerTest)
##       Effect size for OrdPos.c:repet2-1 is -0.022
##
## Based on 1000 simulations, (12 warnings, 0 errors)
## alpha = 0.05, nrow = 3685
##
## Time elapsed: 0 h 17 m 56 s
```

```
PowerLMM1_Interaction24_20_Stark <- powerSim(lmm24_20_Stark, test=fixed("OrdPos.c:repet3-2","t"), nsim =
lastResult()$warnings
lastResult()$errors
```

```
PowerLMM1_Interaction24_20_Stark
```

```
## Power for predictor 'OrdPos.c:repet3-2', (95% confidence interval):
##       74.70% (71.89, 77.37)
##
```

```
## Test: t-test with Satterthwaite degrees of freedom (package lmerTest)
##       Effect size for OrdPos.c:repet3-2 is -0.022
##
## Based on 1000 simulations, (9 warnings, 0 errors)
## alpha = 0.05, nrow = 3685
##
## Time elapsed: 0 h 17 m 49 s
```

```
gm <- mean(df_Ord$RT)
eff_45ms <- (1000/(gm+(45/2))) - (1000/(gm-(45/2)))
fixef(lmm24_20)["OrdPos.c:repet2-1"]<- eff_45ms
fixef(lmm24_20)["OrdPos.c:repet3-2"]<- eff_45ms
fixef(lmm24_20_Stark)["OrdPos.c:repet2-1"]<- eff_45ms
fixef(lmm24_20_Stark)["OrdPos.c:repet3-2"]<- eff_45ms
```

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

```
PowerLMM1_Interaction45_20 <- powerSim(lmm24_20, test=fixed("OrdPos.c:repet2-1","t"), nsim = n_iter) #
lastResult()$warnings
lastResult()$errors
```

```
PowerLMM1_Interaction45_20
```

```
## Power for predictor 'OrdPos.c:repet2-1', (95% confidence interval):
##       63.80% (60.73, 66.78)
##
## Test: t-test with Satterthwaite degrees of freedom (package lmerTest)
##       Effect size for OrdPos.c:repet2-1 is -0.025
##
## Based on 1000 simulations, (6 warnings, 0 errors)
## alpha = 0.05, nrow = 2391
##
## Time elapsed: 0 h 11 m 50 s
```

```
PowerLMM1_Interaction45_20 <- powerSim(lmm24_20, test=fixed("OrdPos.c:repet3-2","t"), nsim = n_iter) #
lastResult()$warnings
lastResult()$errors
```

```
PowerLMM1_Interaction45_20
```

```
## Power for predictor 'OrdPos.c:repet3-2', (95% confidence interval):
##       65.10% (62.05, 68.06)
##
## Test: t-test with Satterthwaite degrees of freedom (package lmerTest)
##       Effect size for OrdPos.c:repet3-2 is -0.025
##
## Based on 1000 simulations, (5 warnings, 0 errors)
## alpha = 0.05, nrow = 2391
##
## Time elapsed: 0 h 11 m 48 s
```

Model B4: Linear model with interaction effect "Ordinal position x Session" of 45 ms and ordinal position effect from Stark et al., 2021

```
PowerLMM1_Interaction45_20_Stark <- powerSim(lmm24_20_Stark, test=fixed("OrdPos.c:repet2-1","t"), nsim =
lastResult()$warnings
lastResult()$errors
```

```
PowerLMM1_Interaction45_20_Stark
```

```
## Power for predictor 'OrdPos.c:repet2-1', (95% confidence interval):
##       81.80% (79.27, 84.15)
##
## Test: t-test with Satterthwaite degrees of freedom (package lmerTest)
##       Effect size for OrdPos.c:repet2-1 is -0.025
##
## Based on 1000 simulations, (9 warnings, 0 errors)
## alpha = 0.05, nrow = 3685
##
## Time elapsed: 0 h 17 m 50 s
```

```
PowerLMM1_Interaction45_20_Stark <- powerSim(lmm24_20_Stark, test=fixed("OrdPos.c:repet3-2","t"), nsim =
lastResult()$warnings
lastResult()$errors
```

```
PowerLMM1_Interaction45_20_Stark
```

```
## Power for predictor 'OrdPos.c:repet3-2', (95% confidence interval):
##       83.90% (81.47, 86.13)
##
## Test: t-test with Satterthwaite degrees of freedom (package lmerTest)
##       Effect size for OrdPos.c:repet3-2 is -0.025
##
## Based on 1000 simulations, (11 warnings, 0 errors)
## alpha = 0.05, nrow = 3685
##
## Time elapsed: 0 h 17 m 46 s
```

What effect size does this mean?

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 <-  0.0457486+0.0011736+0.0006358+0.0138004+0.0001398+0.0922992

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

```
## [1] -0.05700982
```

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

```
## [1] -0.06413986
```

7) Power for the Ordinal position effect if the effect of the interaction effect was 45 ms and 20 VP

```r
gm <- mean(df_Ord$RT)
eff_45ms <- (1000/(gm+(45/2))) - (1000/(gm-(45/2)))
fixef(lmm24_20)["OrdPos.c:repet2-1"]<- eff_45ms
fixef(lmm24_20)["OrdPos.c:repet3-2"]<- eff_45ms

eff1 <- 1000/(gm+(31/2)) - 1000/(gm-(31/2))
fixef(lmm24_20_Stark)["OrdPos.c"] <- eff1
fixef(lmm24_20_Stark)["OrdPos.c:repet2-1"]<- eff_45ms
fixef(lmm24_20_Stark)["OrdPos.c:repet3-2"]<- eff_45ms
```

Linear model with continuous predictor "Ordinal position" and factor "Session" and inversely transformed RT data - Ordinal position effect from Lorenz et al

```r
PowerLMM_OrdPos_Lorenz <- powerSim(lmm24_20, test=fixed("OrdPos.c","t"), nsim = n_iter) # increase to n
lastResult()$warnings
lastResult()$errors
```

```r
PowerLMM_OrdPos_Lorenz
```

```
## Power for predictor 'OrdPos.c', (95% confidence interval):
##       92.60% (90.80, 94.15)
##
## Test: t-test with Satterthwaite degrees of freedom (package lmerTest)
##       Effect size for OrdPos.c is -0.020
##
## Based on 1000 simulations, (6 warnings, 0 errors)
## alpha = 0.05, nrow = 2391
##
## Time elapsed: 0 h 11 m 46 s
```

Linear model with continuous predictor "Ordinal position" and factor "Session" and inversely transformed RT data - Ordinal position effect from Stark et al

```r
PowerLMM_OrdPos_Stark <-
  powerSim(lmm24_20_Stark, test=fixed("OrdPos.c","t"), nsim = n_iter) # increase to nsim > 1000 for rea
lastResult()$warnings
lastResult()$errors
```

```r
PowerLMM_OrdPos_Stark
```

```
## Power for predictor 'OrdPos.c', (95% confidence interval):
##       97.10% (95.86, 98.05)
```

```
##
## Test: t-test with Satterthwaite degrees of freedom (package lmerTest)
##       Effect size for OrdPos.c is -0.017
##
## Based on 1000 simulations, (9 warnings, 0 errors)
## alpha = 0.05, nrow = 3685
##
## Time elapsed: 0 h 17 m 48 s
```