Simulation and Power analysis

Frank Ma

21/04/2021

```
library(afex)
```

```
## Warning: package 'afex' was built under R version 4.0.5
```

```
## Loading required package: lme4
```

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

```
## Registered S3 methods overwritten by 'car':
##   method                          from
##   influence.merMod                lme4
##   cooks.distance.influence.merMod lme4
##   dfbeta.influence.merMod         lme4
##   dfbetas.influence.merMod        lme4
```

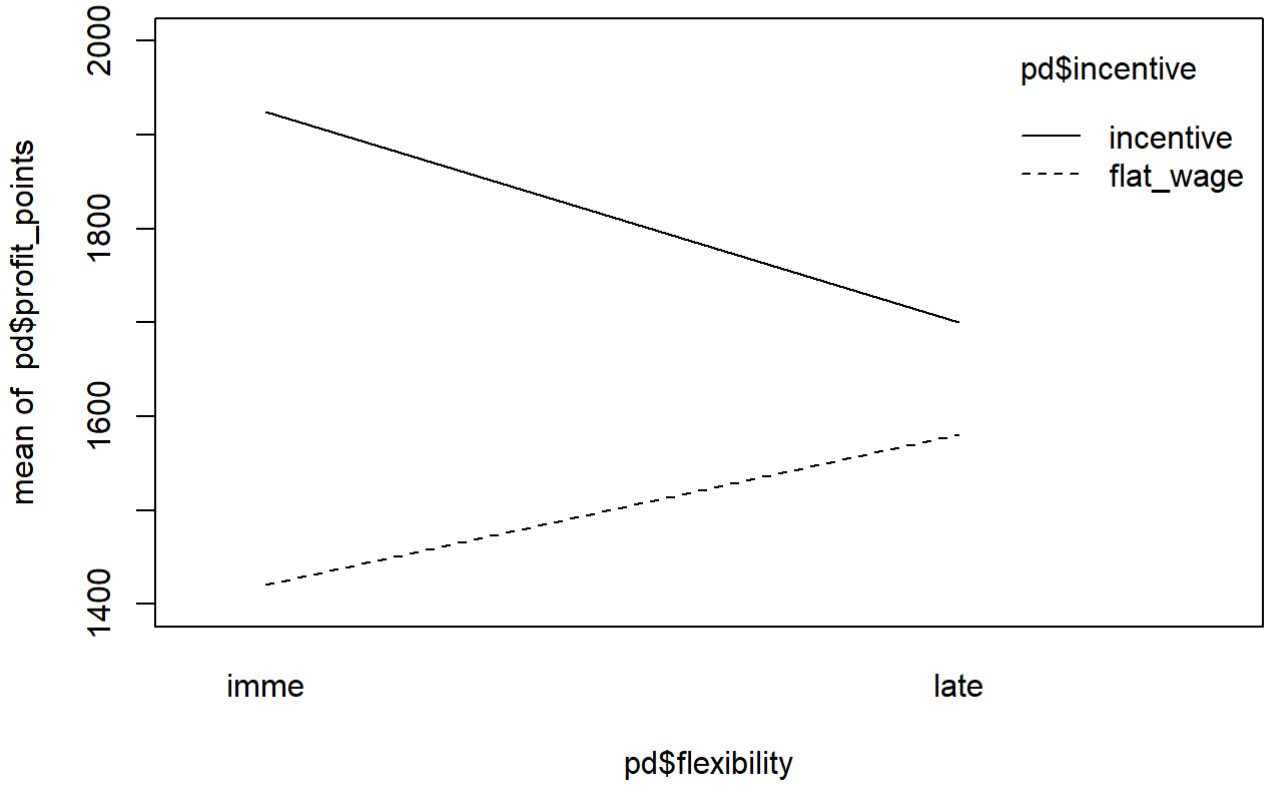
```
## ************
## Welcome to afex. For support visit: http://afex.singmann.science/
```

```
## - Functions for ANOVAs: aov_car(), aov_ez(), and aov_4()
## - Methods for calculating p-values with mixed(): 'KR', 'S', 'LRT', and 'PB'
## - 'afex_aov' and 'mixed' objects can be passed to emmeans() for follow-up tests
## - NEWS: library('emmeans') now needs to be called explicitly!
## - Get and set global package options with: afex_options()
## - Set orthogonal sum-to-zero contrasts globally: set_sum_contrasts()
## - For example analyses see: browseVignettes("afex")
## ************
```

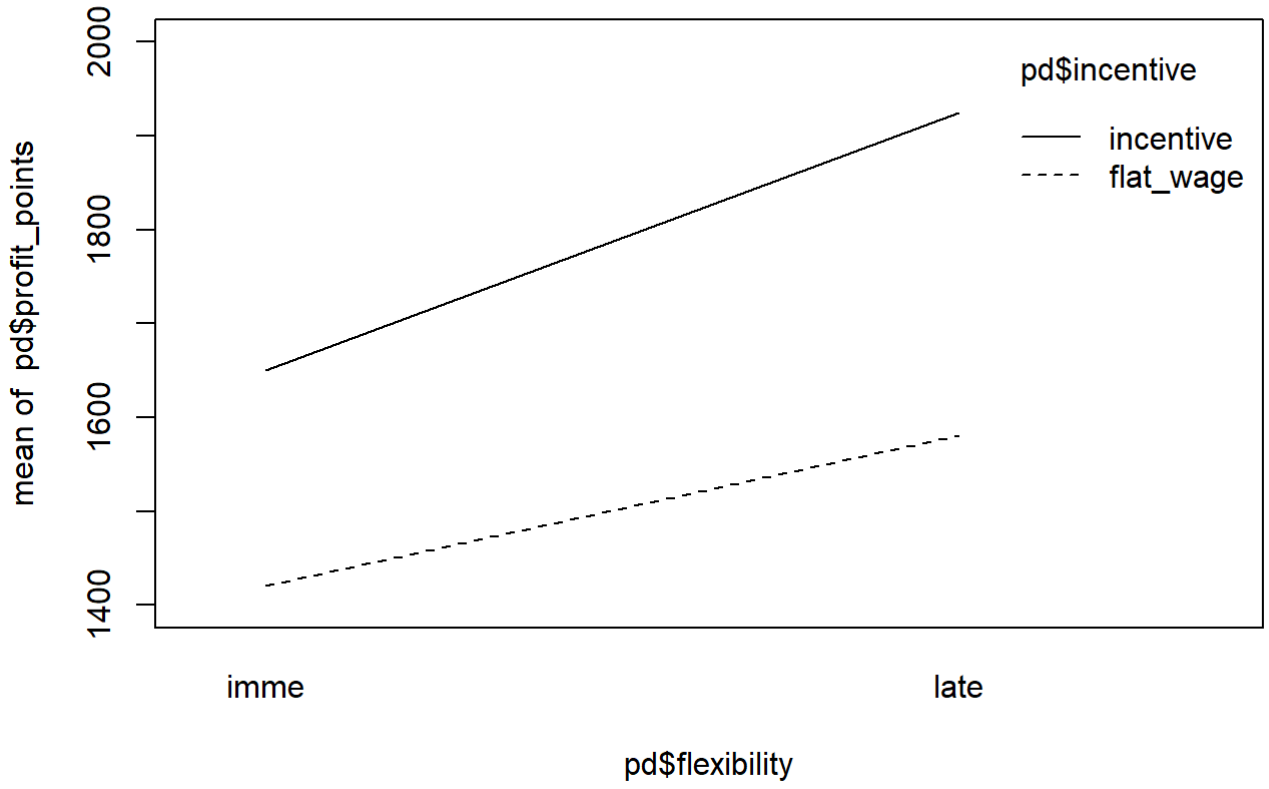
```
##
## Attaching package: 'afex'
```

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

```
## there are four groups:
## group_I_IF will receive immediate flexibility with incentive
## group_I_LF will receive late flexibility with incentive
## group_NI_IF will receive immediate flexibility with flat wage
## group_NI_LF will receive late flexibility with flat wage

## I expect:
## 1 a main effect of flexibility: performance is higher in late flexibility than early flexi
bility
## 2 a main effect of incentive: performance is higher with incentive than flat wage
## 3 interaction of flexibility and incentive: Employees with late flexibility outperform emp
loyees with early flexibility more with flat-wage than with  incentive


##motivation argument
incentive <- rep(c("incentive", "flat_wage"), each = 2)
flexibility <- rep(c("imme", "late"), time = 2)
mean_I_LF  <- 1700
mean_I_IF  <- 1925
mean_NI_LF <- 1581
mean_NI_IF <- 1421

pd <- data.frame(profit_points = c(mean_I_IF, mean_I_LF, mean_NI_IF, mean_NI_LF), flexibility
= flexibility, incentive = incentive)

interaction.plot(pd$flexibility, pd$incentive, pd$profit_points, ylim = c(1400,2000))
```



```
##cognitive load argument
incentive <- rep(c("incentive", "flat_wage"), each = 2)
flexibility <- rep(c("imme", "late"), time = 2)
mean_I_LF <-  1925
mean_I_IF  <- 1650
mean_NI_LF <- 1581
mean_NI_IF <- 1421

pd <- data.frame(profit_points = c(mean_I_IF, mean_I_LF, mean_NI_IF, mean_NI_LF), flexibility
= flexibility, incentive = incentive)

interaction.plot(pd$flexibility, pd$incentive, pd$profit_points, ylim = c(1400,2000))
```



```
n <- 50

    group_I_IF <-  rnorm(n, 1925, 599)
    group_I_LF <-  rnorm(n, 1700, 479)
    group_NI_LF <-  rnorm(n, 1581, 378)
    group_NI_IF <-  rnorm(n, 1421, 433)



participant <- c(1:(n*4))
incentive <- rep(c("incentive", "flat_wage"), each = n*2)
flexibility <- rep(c("late", "imme"), each = n, time = 2)


aov_dat <- data.frame(participant = participant, flexibility = flexibility, incentive = incen
tive, score = c(group_I_IF, group_I_LF, group_NI_IF, group_NI_LF))


aov_car(score ~ flexibility*incentive+ Error(participant), data = aov_dat, type = 3)
```

```
## Converting to factor: flexibility, incentive
```

```
## Contrasts set to contr.sum for the following variables: flexibility, incentive
```

```
## Anova Table (Type 3 tests)
##
## Response: score
##                   Effect     df       MSE         F   ges p.value
## 1           flexibility 1, 196 235773.07      0.17 <.001    .682
## 2             incentive 1, 196 235773.07 19.28 ***  .090   <.001
## 3 flexibility:incentive 1, 196 235773.07      1.96  .010    .163
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '+' 0.1 ' ' 1
```

```
aov_dat$flexibility_sum <- factor(aov_dat$flexibility)
contrasts(aov_dat$flexibility_sum) <- contr.sum(2)
contrasts(aov_dat$flexibility_sum)
```

```
##      [,1]
## imme    1
## late   -1
```

```
aov_dat$incentive_sum <- factor(aov_dat$incentive)
contrasts(aov_dat$incentive_sum) <- contr.sum(2)
contrasts(aov_dat$incentive_sum)
```

```
##           [,1]
## flat_wage    1
## incentive   -1
```

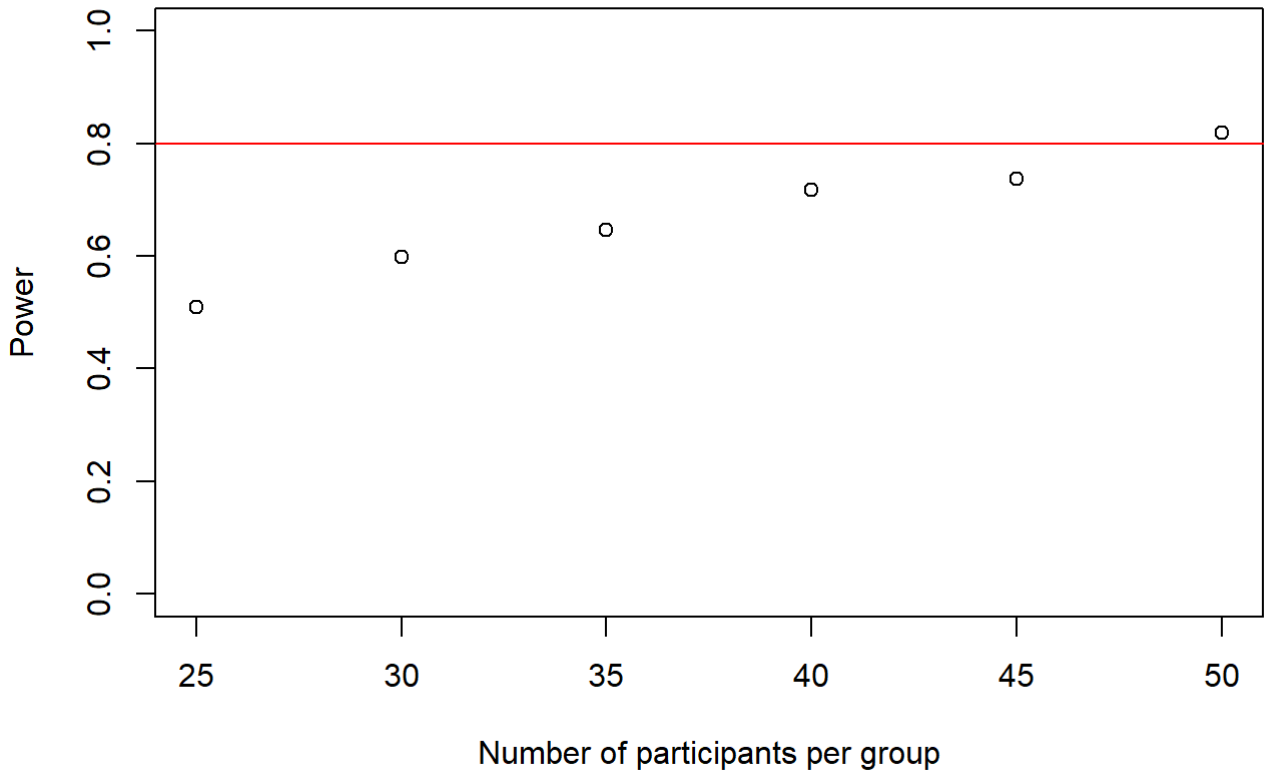
```
aov_dat$flexibility_sum_num <- ifelse(aov_dat$flexibility == "late", 1, -1)
aov_dat$incentive_sum_num <- ifelse(aov_dat$incentive == "incentive", 1, -1)

lm_int_sum <- lm(score ~ 1 + incentive_sum_num + flexibility_sum_num + incentive_sum_num:flex
ibility_sum_num, data = aov_dat)
lm_int_sum
```

```
##
## Call:
## lm(formula = score ~ 1 + incentive_sum_num + flexibility_sum_num +
##     incentive_sum_num:flexibility_sum_num, data = aov_dat)
##
## Coefficients:
##                             (Intercept)                      incentive_sum_num
##                                 1687.84                                 150.77
##                     flexibility_sum_num  incentive_sum_num:flexibility_sum_num
##                                   14.09                                  48.02
```

```
##the motivation hypothesis on performance
set.seed(1)
n_sims <- 1000 # we want 1000 simulations
p_vals <- c()
power_at_n <- c(0) # this vector will contain the power for each sample-size (it needs the in
itial 0 for the while-loop to work)
n <- 25 # sample-size
n_increase <- 5 # by which stepsize should n be increased
i <- 2

power_crit <- .80
alpha <- .05

while(power_at_n[i-1] < power_crit){
  for(sim in 1:n_sims){
    group_I_IF <-  rnorm(n, 1925, 599)
    group_I_LF <-  rnorm(n, 1700, 479)
    group_NI_LF <-  rnorm(n, 1581, 378)
    group_NI_IF <-  rnorm(n, 1421, 433)

participant <- c(1:(n*4))
incentive <- rep(c("incentive", "flat_wage"), each = n*2)
flexibility <- rep(c("late", "imme"), each = n, time = 2)

    aov_dat <- data.frame(participant = participant, incentive = incentive, flexibility = fle
xibility, score = c(group_I_IF, group_I_LF, group_NI_IF, group_NI_LF))
    aov_dat$incentive_sum_num <- ifelse(aov_dat$incentive == "incentive", 1, -1) # apply sum-
to-zero coding
    aov_dat$flexibility_sum_num <- ifelse(aov_dat$flexibility == "late", 1, -1)
    lm_int <- lm(score ~ 1 + flexibility_sum_num + incentive_sum_num + flexibility_sum_num:in
centive_sum_num, data = aov_dat) # fit the model with the interaction
    lm_null <- lm(score ~ 1 + flexibility_sum_num + incentive_sum_num, data = aov_dat) # fit
 the model without the interaction
    p_vals[sim] <- anova(lm_int, lm_null)$`Pr(>F)`[2] # put the p-values in a list
  }
    print(n)
    power_at_n[i] <- mean(p_vals < alpha) # check power (i.e. proportion of p-values that are
smaller than alpha-level of .10)
    names(power_at_n)[i] <- n
    n <- n+n_increase # increase sample-size by 100 for low-resolution testing first
    i <- i+1 # increase index of the while-loop by 1 to save power and cohens d to vector
}
```

```
## [1] 25
## [1] 30
## [1] 35
## [1] 40
## [1] 45
## [1] 50
```

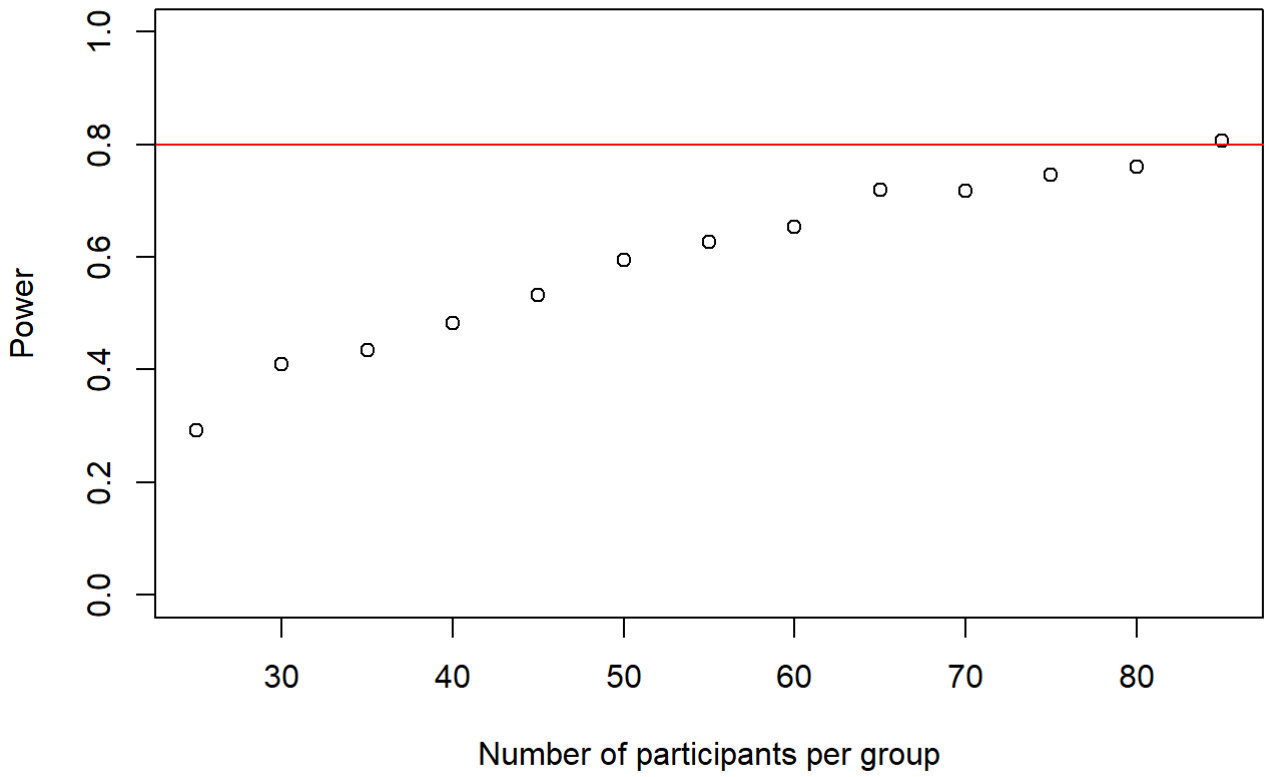
```
power_at_n <- power_at_n[-1] # delete first 0 from the vector

plot(as.numeric(names(power_at_n)), power_at_n, xlab = "Number of participants per group", yl
ab = "Power", ylim = c(0,1), axes = TRUE)
abline(h = .80, col = "red")
```


Power
0.0
0.2
0.4
0.6
0.8
1.0
25
30
35
40
45
50
Number of participants per group

```
## cognitive load hypothesis

n_sims <- 1000 # we want 1000 simulations
p_vals <- c()
power_at_n <- c(0) # this vector will contain the power for each sample-size (it needs the in
itial 0 for the while-loop to work)
n <- 25 # sample-size
n_increase <- 5 # by which stepsize should n be increased
i <- 2

power_crit <- .80
alpha <- .05

while(power_at_n[i-1] < power_crit){
  for(sim in 1:n_sims){
    group_I_IF <-  rnorm(n, 1700, 450)
    group_I_LF <-  rnorm(n, 2150, 599)
    group_NI_LF <-  rnorm(n, 1581, 378)
    group_NI_IF <-  rnorm(n, 1421, 433)

participant <- c(1:(n*4))
incentive <- rep(c("incentive", "flat_wage"), each = n*2)
flexibility <- rep(c("late", "imme"), each = n, time = 2)

    aov_dat <- data.frame(participant = participant, incentive = incentive, flexibility = fle
xibility, score = c(group_I_IF, group_I_LF, group_NI_IF, group_NI_LF))
    aov_dat$incentive_sum_num <- ifelse(aov_dat$incentive == "incentive", 1, -1) # apply sum-
to-zero coding
    aov_dat$flexibility_sum_num <- ifelse(aov_dat$flexibility == "late", 1, -1)
    lm_int <- lm(score ~ 1 + flexibility_sum_num + incentive_sum_num + flexibility_sum_num:in
centive_sum_num, data = aov_dat) # fit the model with the interaction
    lm_null <- lm(score ~ 1 + flexibility_sum_num + incentive_sum_num, data = aov_dat) # fit
 the model without the interaction
    p_vals[sim] <- anova(lm_int, lm_null)$`Pr(>F)`[2] # put the p-values in a list
  }
    print(n)
    power_at_n[i] <- mean(p_vals < alpha) # check power (i.e. proportion of p-values that are
smaller than alpha-level of .10)
    names(power_at_n)[i] <- n
    n <- n+n_increase # increase sample-size by 100 for low-resolution testing first
    i <- i+1 # increase index of the while-loop by 1 to save power and cohens d to vector
}
```

```
## [1] 25
## [1] 30
## [1] 35
## [1] 40
## [1] 45
## [1] 50
## [1] 55
## [1] 60
## [1] 65
## [1] 70
## [1] 75
## [1] 80
## [1] 85
```

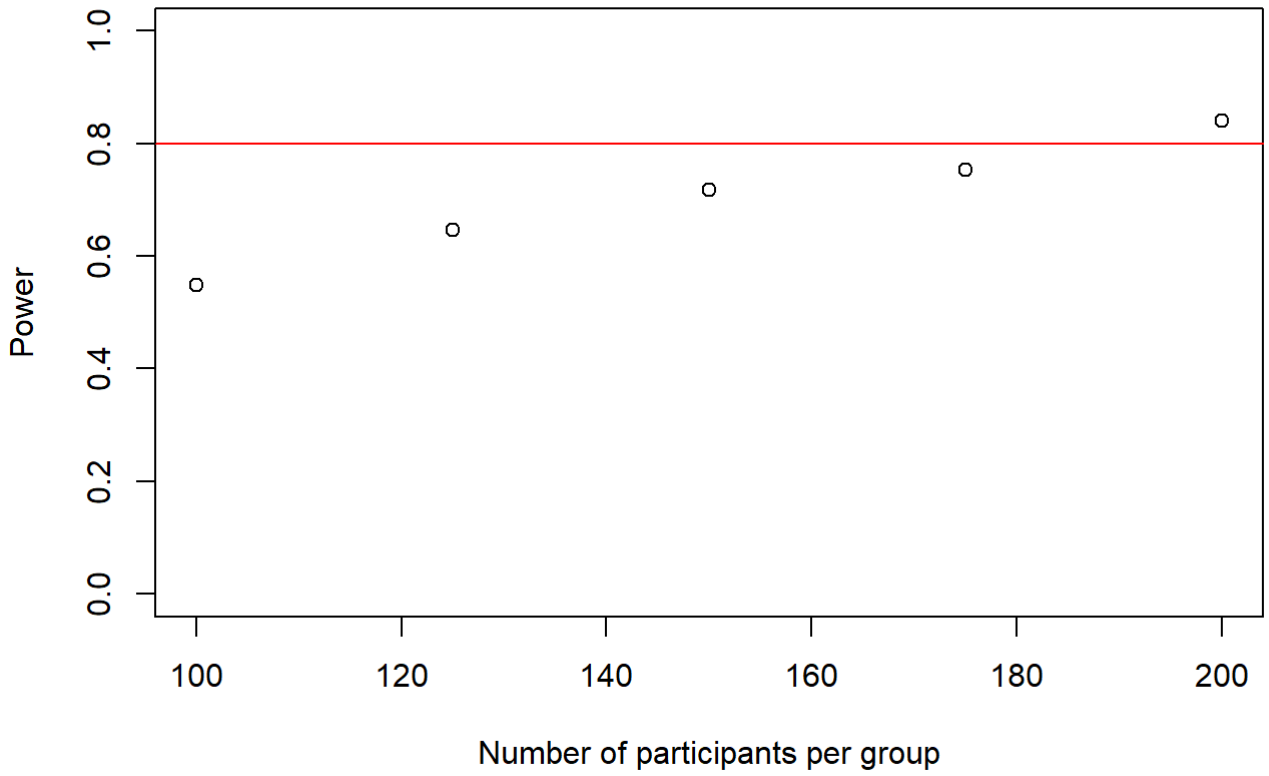
```
power_at_n <- power_at_n[-1] # delete first 0 from the vector

plot(as.numeric(names(power_at_n)), power_at_n, xlab = "Number of participants per group", yl
ab = "Power", ylim = c(0,1), axes = TRUE)
abline(h = .80, col = "red")
```



```
## hypothesis on learning


n_sims <- 1000 # we want 1000 simulations
p_vals <- c()
power_at_n <- c(0) # this vector will contain the power for each sample-size (it needs the in
itial 0 for the while-loop to work)
n <- 100 # sample-size
n_increase <- 25 # by which stepsize should n be increased
i <- 2

power_crit <- .80
alpha <- .05

while(power_at_n[i-1] < power_crit){
  for(sim in 1:n_sims){
    group_I_IF <-  rnorm(n, 100, 66)
    group_I_LF <-  rnorm(n, 90, 66)
    group_NI_LF <-  rnorm(n, 77, 182)
    group_NI_IF <-  rnorm(n, 30, 187)

participant <- c(1:(n*4))
incentive <- rep(c("incentive", "flat_wage"), each = n*2)
flexibility <- rep(c("late", "imme"), each = n, time = 2)

    aov_dat <- data.frame(participant = participant, incentive = incentive, flexibility = fle
xibility, score = c(group_I_IF, group_I_LF, group_NI_IF, group_NI_LF))
    aov_dat$incentive_sum_num <- ifelse(aov_dat$incentive == "incentive", 1, -1) # apply sum-
to-zero coding
    aov_dat$flexibility_sum_num <- ifelse(aov_dat$flexibility == "late", 1, -1)
    lm_int <- lm(score ~ 1 + flexibility_sum_num + incentive_sum_num + flexibility_sum_num:in
centive_sum_num, data = aov_dat) # fit the model with the interaction
    lm_null <- lm(score ~ 1 + flexibility_sum_num + incentive_sum_num, data = aov_dat) # fit
 the model without the interaction
    p_vals[sim] <- anova(lm_int, lm_null)$`Pr(>F)`[2] # put the p-values in a list
  }
    print(n)
    power_at_n[i] <- mean(p_vals < alpha) ## check power (i.e. proportion of p-values that ar
e smaller than alpha-level of .10)
    names(power_at_n)[i] <- n
    n <- n+n_increase # increase sample-size by 100 for low-resolution testing first
    i <- i+1 # increase index of the while-loop by 1 to save power and cohens d to vector
}
```

```
## [1] 100
## [1] 125
## [1] 150
## [1] 175
## [1] 200
```

```
power_at_n <- power_at_n[-1] # delete first 0 from the vector

plot(as.numeric(names(power_at_n)), power_at_n, xlab = "Number of participants per group", yl
ab = "Power", ylim = c(0,1), axes = TRUE)
abline(h = .80, col = "red")
```


Power
0.0
0.2
0.4
0.6
0.8
1.0
100
120
140
160
180
200
Number of participants per group