

Validation of Superpower

Aaron Caldwell & Daniël Lakens

2019-09-12

Contents

1	Introduction	5
2	One-way ANOVA Part 1	7
2.1	Validation of Power in One-Way ANOVA	7
2.2	Two conditions	8
3	One-way ANOVA Part 2	17
3.1	Validation of Power in One-Way ANOVA with Brysbaert example	17
3.2	Power for simple effects	19
3.3	One-way ANOVA Part 3	26
3.4	Three conditions replication	27
3.5	Variation 1	28
3.6	Three conditions replication	30
3.7	Variation 2	31
3.8	Three conditions replication	33
4	Repeated Measures-ANOVA Part 1	35
4.1	Two conditions, medium effect size	35
5	Repeated Measures-ANOVA Part 2	45
5.1	The relation between Cohen's f and Cohen's d	45
5.2	Three within conditions, medium effect size	47

6 Repeated Measures-ANOVA Part 3	53
6.1 Reproducing Brysbaert	54
6.2 Reproducing Brysbaert Variation 1 Changing Correlation	58

This is a compilation of validation documents for **Superpower** written in **Mark-down** and compiled by **Bookdown**.

```
install.packages("bookdown")  
# or the development version  
# devtools::install_github("rstudio/bookdown")
```

Chapter 1

Introduction

Below we have included various examples of the performance of **Superpower** against other R packages and statistical programs (such as GPower).

Chapter 2

One-way ANOVA Part 1

```
knitr::opts_chunk$set(echo = TRUE)
nsims <- 100 #set number of simulations
require(mvtnorm, quietly = TRUE)
require(MASS, quietly = TRUE)
require(afex, quietly = TRUE)
require(emmeans, quietly = TRUE)
require(ggplot2, quietly = TRUE)
require(gridExtra, quietly = TRUE)
require(reshape2, quietly = TRUE)
require(pwr, quietly = TRUE)
# Install functions from GitHub by running the code below:
library(Superpower)
```

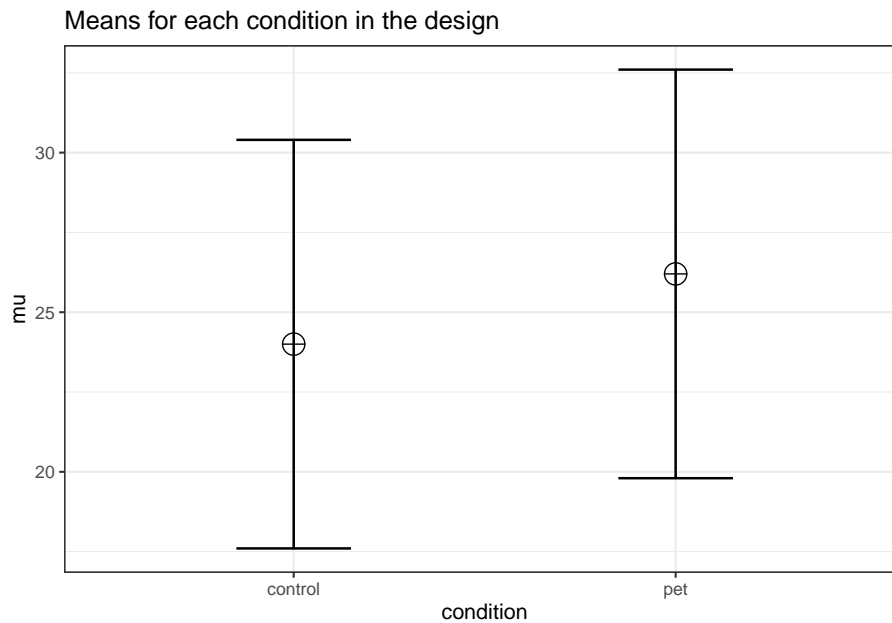
2.1 Validation of Power in One-Way ANOVA

Using the formula also used in Albers & Lakens (2018), we can determine the means that should yield a specified effect sizes (expressed in Cohen's f). Eta-squared (identical to partial eta-squared for One-Way ANOVA's) has benchmarks of .0099, .0588, and .1379 for small, medium, and large effect sizes (Cohen, 1988). Although these benchmarks are quite random, and researchers should only use such benchmarks for power analyses as a last resort, we will demonstrate a-priori power analysis for these values.

2.2 Two conditions

Imagine we aim to design a study to test the hypothesis that giving people a pet to take care of will increase their life satisfaction. We have a control condition, and a condition where people get a pet, and randomly assign participants to either condition. We can simulate a One-Way ANOVA with a specified alpha, sample size, and effect size, on see the statistical power we would have for the ANOVA and the follow-up comparisons. We expect pets to increase life-satisfaction compared to the control condition. Based on work by Pavot and Diener (1993) we believe that we can expect responses on the life-satisfaction scale to have a mean of approximately 24 in our population, with a standard deviation of 6.4. We expect having a pet increases life satisfaction with approximately 2.2 scale points for participants who get a pet. 200 participants in total, with 100 participants in each condition. But before we proceed with the data collection, we examine the statistical power our design would have to detect the differences we predict.

```
string <- "2b"
n <- 100
# We are thinking of running 50 people in each condition
mu <- c(24, 26.2)
# Enter means in the order that matches the labels below.
# In this case, control, cat, dog.
sd <- 6.4
labelnames <- c("condition", "control", "pet") #
# the label names should be in the order of the means specified above.
design_result <- ANOVA_design(design = string,
                             n = n,
                             mu = mu,
                             sd = sd,
                             labelnames = labelnames)
```

```
alpha_level <- 0.05
# You should think carefully about how to justify your alpha level.
# We will give some examples later, but for now, use 0.05.
ANOVA_power(design_result, alpha_level = alpha_level, nsims = nsims)
```

```
## Power and Effect sizes for ANOVA tests
##           power effect_size
## anova_condition    69    0.03316
##
## Power and Effect sizes for contrasts
##                               power effect_size
## p_condition_control_condition_pet    69    0.341
```

The result shows that we have exactly the same power for the ANOVA, as we have for the t -test. This is because when there are only two groups, these tests are mathematically identical. In a study with 100 participants, we would have quite low power (around 67.7%). An ANOVA with 2 groups is identical to a t -test. For our example, Cohen's d (the standardized mean difference) is $2.2/6.4$, or $d = 0.34375$ for the difference between the control condition and pets, which we can use to easily compute the expected power for these simple comparisons using the `pwr` package.

```
pwr.t.test(d = 2.2/6.4,
           n = 100,
           sig.level = 0.05,
           type="two.sample",
           alternative="two.sided")$power
```

```
## [1] 0.6768572
```

We can also directly compute Cohen's f from Cohen's d for two groups, as Cohen (1988) describes, because $f = 1/2d$. So $f = 0.5 \cdot 0.34375 = 0.171875$. And indeed, power analysis using the `pwr` package yields the same result using the `pwr.anova.test` as the `pwr.t.test`.

```
K <- 2
n <- 100
f <- 0.171875
pwr.anova.test(n = n,
               k = K,
               f = f,
               sig.level = alpha_level)$power
```

```
## [1] 0.6768572
```

This analysis tells us that running the study with 100 participants in each condition is too likely to *not* yield a significant test result, even if our expected pattern of differences is true. This is not optimal.

Let's mathematically explore which pattern of means we would need to expect to have 90% power for the ANOVA with 50 participants in each group. We can use the `pwr` package in R to compute a sensitivity analysis that tells us the effect size, in Cohen's f , that we are able to detect with 3 groups and 50 participants in each group, in order to achieve 90% power with an alpha level of 5%.

```
K <- 2
n <- 100
sd <- 6.4
r <- 0
#Calculate f when running simulation
f <- pwr.anova.test(n = n,
                   k = K,
                   power = 0.9,
                   sig.level = alpha_level)$f
f
```

```
## [1] 0.2303587
```

This sensitivity analysis shows we have 90% power in our planned design to detect effects of Cohen's f of 0.2303587. Benchmarks by Cohen (1988) for small, medium, and large Cohen's f values are 0.1, 0.25, and 0.4, which correspond to eta-squared values of small (.0099), medium (.0588), and large (.1379), in line with $d = .2$, $.5$, or $.8$. So, at least based on these benchmarks, we have 90% power to detect effects that are slightly below a medium effect benchmark.

```
f2 <- f^2
ES <- f2/(f2+1)
ES
```

```
## [1] 0.0503911
```

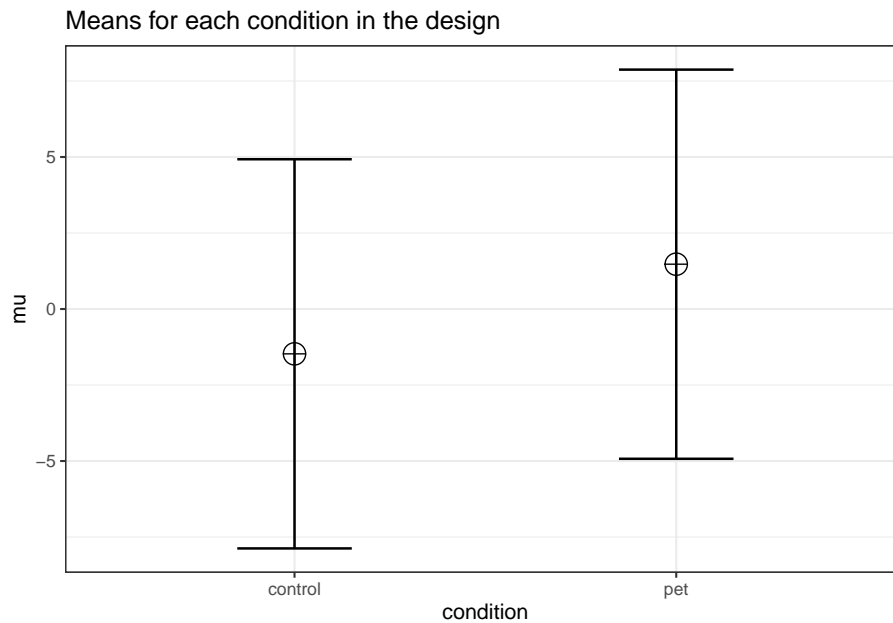
Expressed in eta-squared, we can detect values of eta-squared = 0.05 or larger.

```
mu <- mu_from_ES(K = K, ES = ES)
mu <- mu * sd
mu
```

```
## [1] -1.474295  1.474295
```

We can compute a pattern of means, given a standard deviation of 6.4, that would give us an effect size of $f = 0.23$, or eta-squared of 0.05. We should be able to accomplish this if the means are -1.474295 and 1.474295. We can use these values to confirm the ANOVA has 90% power.

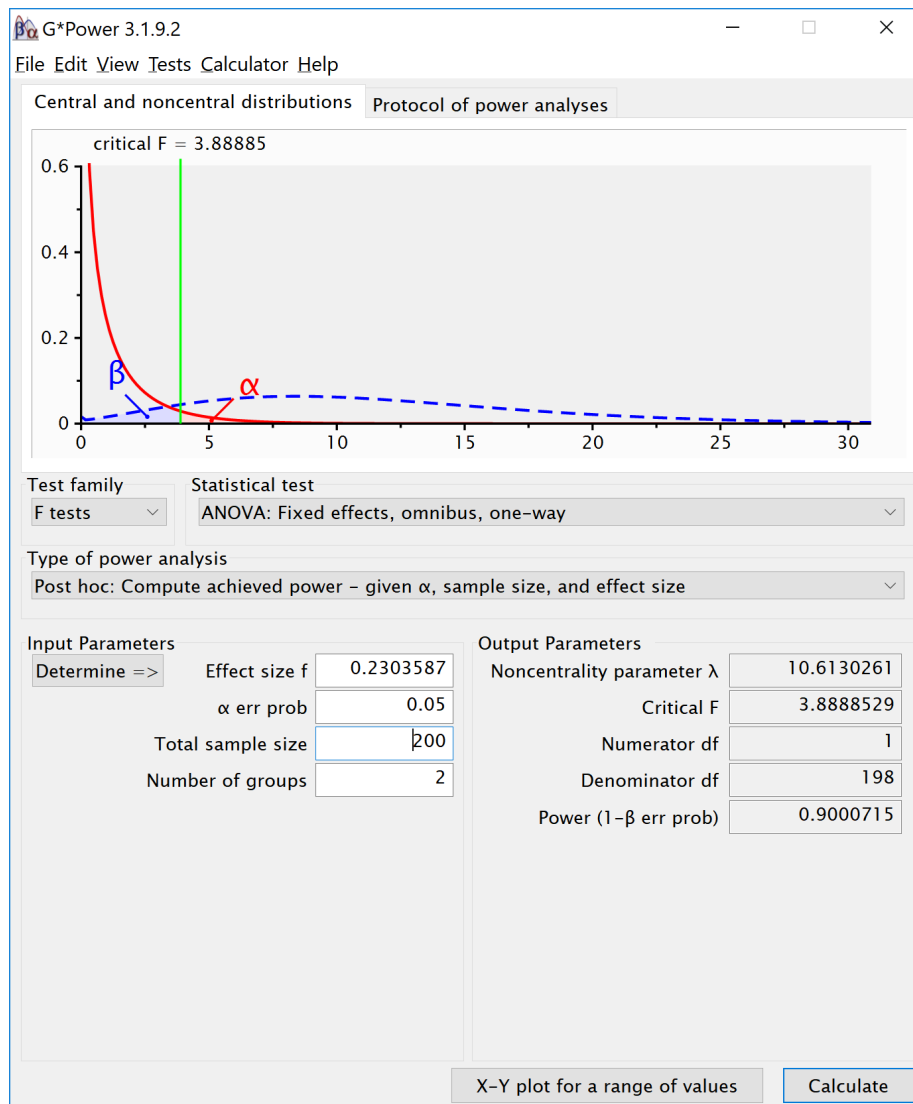
```
design_result <- ANOVA_design(design = string,
                             n = n,
                             mu = mu,
                             sd = sd,
                             labelnames = labelnames)
```



```
ANOVA_power(design_result, alpha_level = alpha_level, nsims = nsims)
```

```
## Power and Effect sizes for ANOVA tests
##           power effect_size
## anova_condition      88    0.04836
##
## Power and Effect sizes for contrasts
##           power effect_size
## p_condition_control_condition_pet      88    0.4328
```

The simulation confirms that for the F -test for the ANOVA we have 90% power. This is also what g^* power tells us what would happen based on a post-hoc power analysis with an f of 0.2303587, 2 groups, 200 participants in total (100 in each between subject condition), and an alpha of 5%.

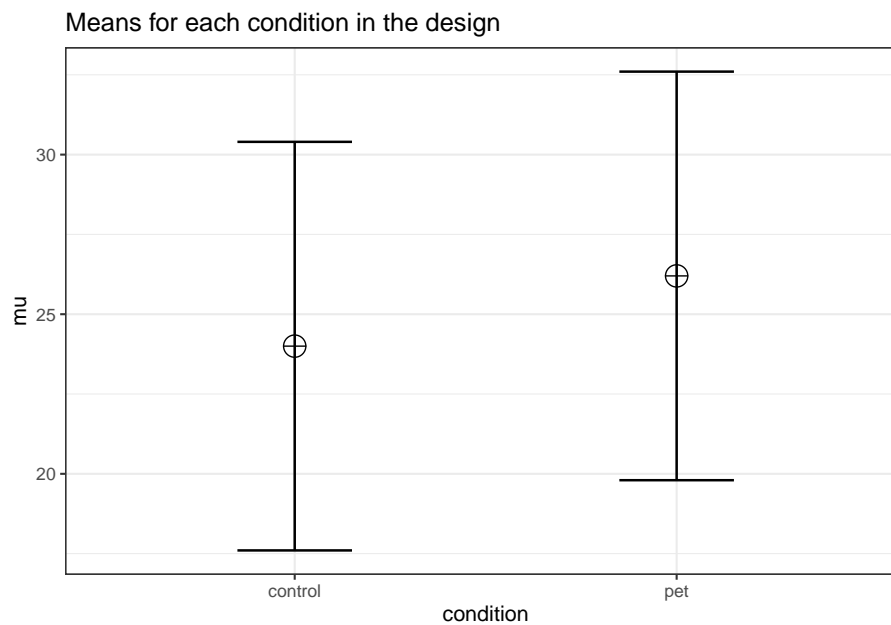


If we return to our expected means, how many participants do we need for sufficient power? Given the expected difference and standard deviation, $d = 0.34375$, and $f = 0.171875$. We can perform an a-priori power analysis for this simple case, which tells us we need 179 participants in each group (we can't split people in parts, and thus always round a power analysis upward), or 358 in total.

```
K <- 2
power <- 0.9
f <- 0.171875
pwr.anova.test(power = power,
```

```
##
##      Balanced one-way analysis of variance power calculation
##
##          k = 2
##          n = 178.8104
##          f = 0.171875
##      sig.level = 0.05
##          power = 0.9
##
## NOTE: n is number in each group
```

```
string <- "2b"
n <- 179
mu <- c(24, 26.2)
# Enter means in the order that matches the labels below.
# In this case, control, pet.
sd <- 6.4
labelnames <- c("condition", "control", "pet") #
# the label names should be in the order of the means specified above.
design_result <- ANOVA_design(design = string,
                             n = n,
                             mu = mu,
                             sd = sd,
                             labelnames = labelnames)
```

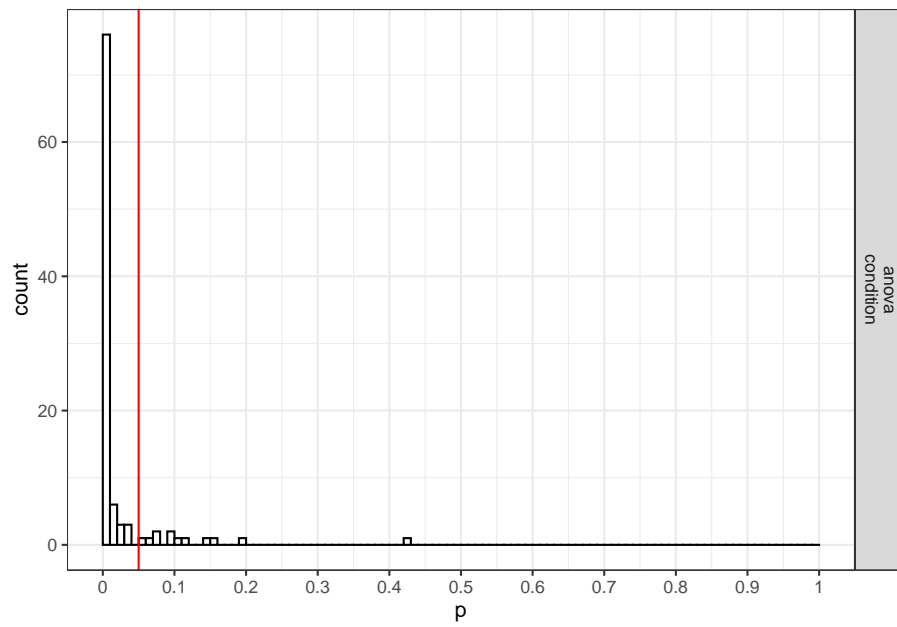


```
alpha_level <- 0.05
power_result <- ANOVA_power(design_result, alpha_level = alpha_level, nsims = nsims)
```

```
## Power and Effect sizes for ANOVA tests
##           power effect_size
## anova_condition    88    0.03289
##
## Power and Effect sizes for contrasts
##           power effect_size
## p_condition_control_condition_pet    88    0.3533
```

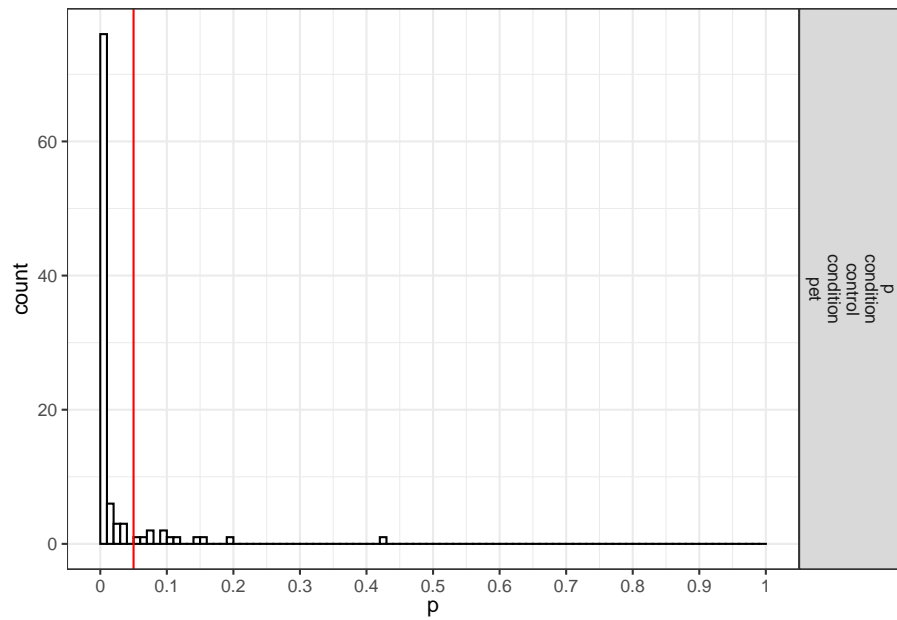
We stored the result from the power analysis in an object. This allows us to request plots (which are not printed automatically) showing the p -value distribution. If we request `power_result$plot1` we get the p -value distribution for the ANOVA:

```
power_result$plot1
```



If we request `power_result$plot2` we get the p-value distribution for the paired comparisons (in this case only one):

```
power_result$plot2
```



Chapter 3

One-way ANOVA Part 2

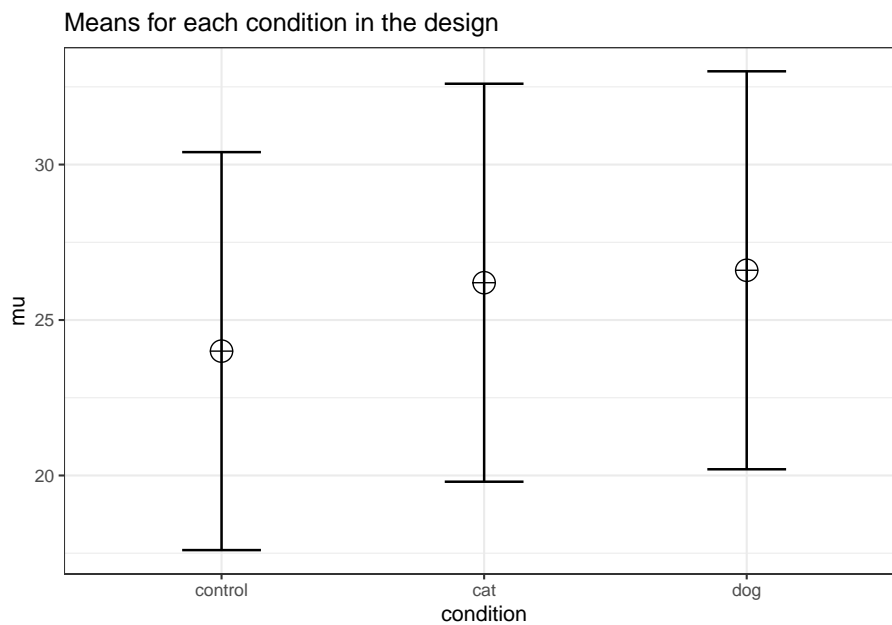
3.1 Validation of Power in One-Way ANOVA with Brysbaert example

3.1.1 Three conditions

Imagine we aim to design a study to test the hypothesis that giving people a pet to take care of will increase their life satisfaction. We have a control condition, a ‘cat’ pet condition, and a ‘dog’ pet condition. We can simulate a One-Way ANOVA with a specified alpha, sample size, and effect size, on see the statistical power we would have for the ANOVA and the follow-up comparisons. We expect all pets to increase life-satisfaction compared to the control condition. Obviously, we also expect the people who are in the ‘dog’ pet condition to have even greater life-satisfaction than people in the ‘cat’ pet condition. Based on work by Pavot and Diener (1993) we believe that we can expect responses on the life-satisfaction scale to have a mean of approximately 24 in our population, with a standard deviation of 6.4. We expect having a pet increases life satisfaction with approximately 2.2 scale points for participants who get a cat, and 2.6 scale points for participants who get a dog. We initially consider collecting data from 150 participants in total, with 50 participants in each condition. But before we proceed with the data collection, we examine the statistical power our design would have to detect the differences we predict.

```
string <- "3b"
n <- 50
# We are thinking of running 50 people in each condition
mu <- c(24, 26.2, 26.6)
# Enter means in the order that matches the labels below.
# In this case, control, cat, dog.
```

```
sd <- 6.4
labelnames <- c("condition", "control", "cat", "dog") #
# the label names should be in the order of the means specified above.
design_result <- ANOVA_design(design = string,
                             n = n,
                             mu = mu,
                             sd = sd,
                             labelnames = labelnames)
```



```
alpha_level <- 0.05
# You should think carefully about how to justify your alpha level.
# We will give some examples later, but for now, use 0.05.
ANOVA_power(design_result, alpha_level = alpha_level, nsims = nsims)
```

```
## Power and Effect sizes for ANOVA tests
##           power effect_size
## anova_condition    46    0.04169
##
## Power and Effect sizes for contrasts
##           power effect_size
## p_condition_control_condition_cat    41    0.33459
## p_condition_control_condition_dog    46    0.37274
## p_condition_cat_condition_dog        7    0.03855
```

```
#should yield
#0.3983064
#0.5205162
#0.06104044
```

The result shows that you would have quite low power with 50 participants, both for the overall ANOVA (just around 50% power), as for the follow up comparisons (approximately 40% power for the control vs cat condition, around 50% for the control vs dogs condition, and a really low power (around 6%, just above the Type 1 error rate of 5%) for the expected difference between cats and dogs.

3.2 Power for simple effects

We are typically not just interested in the ANOVA, but also in follow up comparisons. In this case, we would perform a *t*-test comparing the control condition against the cat and dog condition, and we would compare the cat and dog conditions against each other, in independent *t*-tests.

For our example, Cohen's *d* (the standardized mean difference) is $2.2/6.4$, or $d = 0.34375$ for the difference between the control condition and cats, $2.6/6.4$ or $d = 0.40625$ for the difference between the control condition and dogs, and $0.4/6.4$ or $d = 0.0625$ for the difference between cats and dogs as pets.

We can easily compute the expected power for these simple comparisons using the *pwr* package.

```
pwr.t.test(d = 2.2/6.4,
           n = 50,
           sig.level = 0.05,
           type="two.sample",
           alternative="two.sided")$power
```

```
## [1] 0.3983064
```

```
pwr.t.test(d = 2.6/6.4,
           n = 50,
           sig.level = 0.05,
           type="two.sample",
           alternative="two.sided")$power
```

```
## [1] 0.5205162
```

```
pwr.t.test(d = 0.4/6.4,
           n = 50,
           sig.level = 0.05,
           type="two.sample",
           alternative="two.sided")$power
```

```
## [1] 0.06104044
```

This analysis tells us that running the study with 50 participants in each condition is more likely to *not* yield a significant test result, even if our expected pattern of differences is true, than that we will observe a p -value smaller than our alpha level. This is not optimal.

Let's mathematically explore which pattern of means we would need to expect to have 90% power for the ANOVA with 50 participants in each group. We can use the `pwr` package in R to compute a sensitivity analysis that tells us the effect size, in Cohen's f , that we are able to detect with 3 groups and 50 participants in each group, in order to achieve 90% power with an alpha level of 5%.

```
K <- 3
n <- 50
sd <- 6.4
r <- 0
#Calculate f when running simulation
f <- pwr.anova.test(n = n,
                   k = K,
                   power = 0.9,
                   sig.level = alpha_level)$f
f
```

```
## [1] 0.2934417
```

This sensitivity analysis shows we have 90% power in our planned design to detect effects of Cohen's f of 0.2934417. Benchmarks by Cohen (1988) for small, medium, and large Cohen's f values are 0.1, 0.25, and 0.4, which correspond to eta-squared values of small (.0099), medium (.0588), and large (.1379), in line with $d = .2$, $.5$, or $.8$. So, at least based on these benchmarks, we have 90% power to detect effects that are somewhat sizeable.

```
f2 <- f^2
ES <- f2/(f2+1)
ES
```

```
## [1] 0.07928127
```

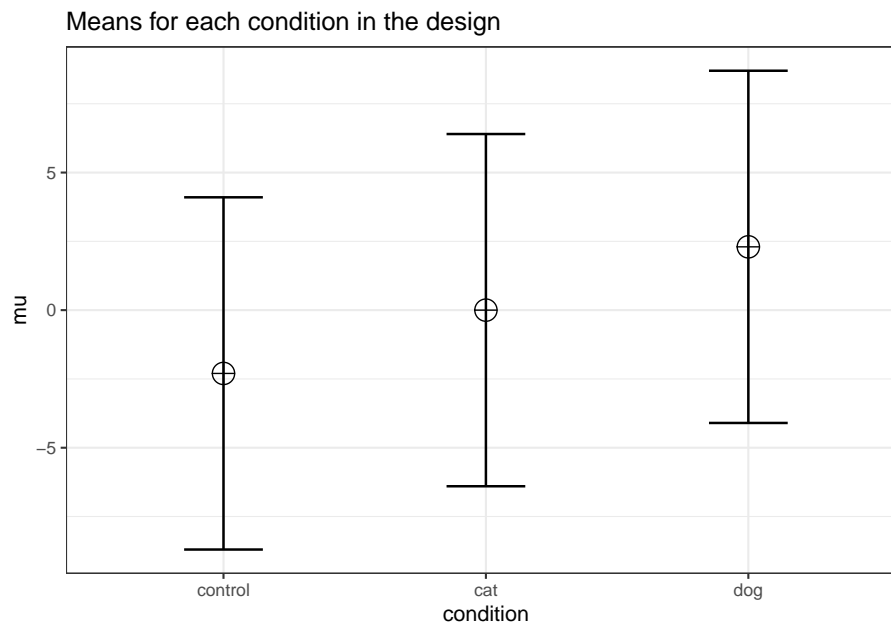
Expressed in eta-squared, we can detect values of eta-squared = 0.0793 or larger.

```
mu <- mu_from_ES(K = K, ES = ES)
mu <- mu * sd
mu
```

```
## [1] -2.300104  0.000000  2.300104
```

We can compute a pattern of means, given a standard deviation of 6.4, that would give us an effect size of $f = 0.2934$, or eta-squared of 0.0793. We should be able to accomplish this if the means are -2.300104, 0.000000, and 2.300104. We can use these values to confirm the ANOVA has 90% power.

```
design_result <- ANOVA_design(design = string,
                             n = n,
                             mu = mu,
                             sd = sd,
                             labelnames = labelnames)
```

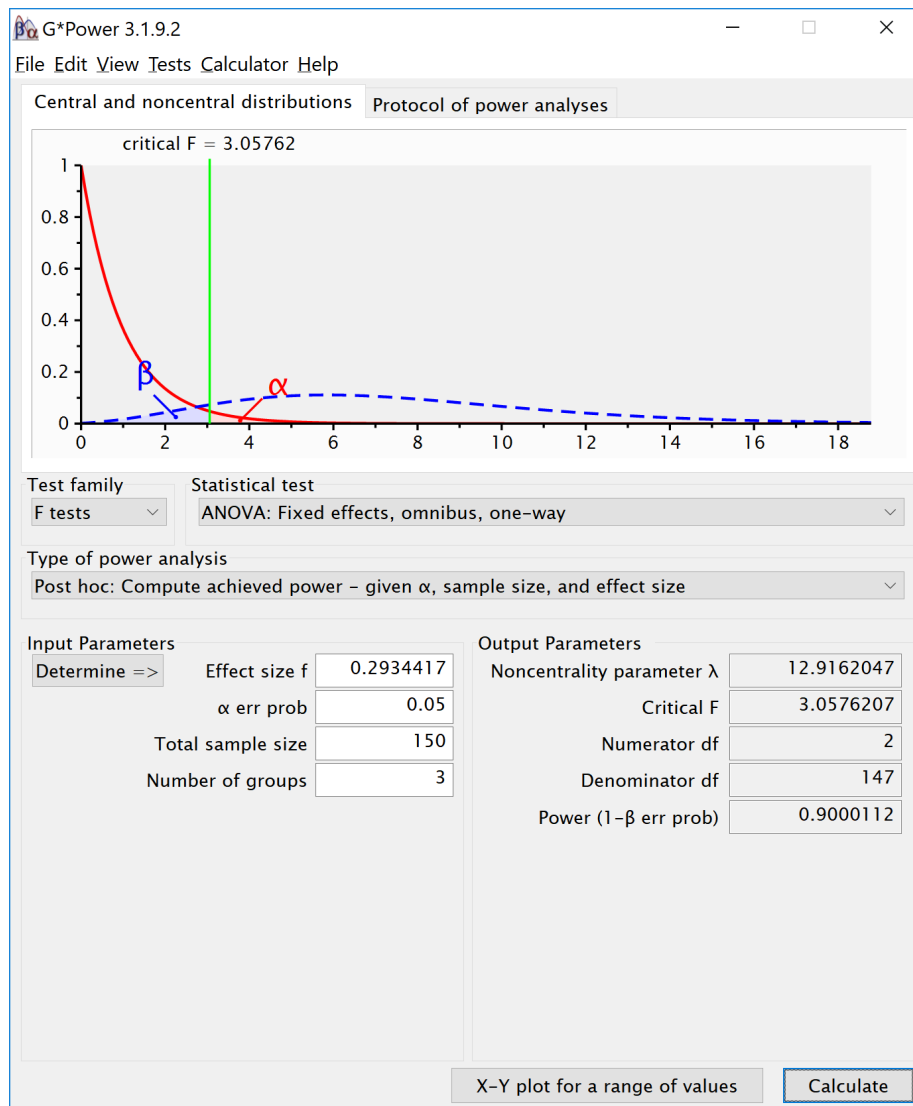


```
ANOVA_power(design_result, alpha_level = alpha_level, nsims = nsims)
```

```
## Power and Effect sizes for ANOVA tests
##           power effect_size
```

```
## anova_condition      88      0.08837
##
## Power and Effect sizes for contrasts
##
##           power effect_size
## p_condition_control_condition_cat      48      0.3741
## p_condition_control_condition_dog      91      0.7049
## p_condition_cat_condition_dog        39      0.3247
```

The simulation confirms that for the F -test for the ANOVA we have 90% power. This is also what g*power tells us what would happen based on a post-hoc power analysis with an f of 0.2934417, 3 groups, 150 participants in total (50 in each between subject condition), and an alpha of 5%.



We can also compute the power for the ANOVA and simple effects in R with the `pwr` package. The calculated effect sizes and power match those from the simulation.

```
K <- 3
n <- 50
sd <- 6.4
f <- 0.2934417
pwr.anova.test(n = n,
               k = K,
```

```
f = f,
sig.level = alpha_level)$power
```

```
## [1] 0.9000112
```

```
d <- 2.300104/6.4
d
```

```
## [1] 0.3593912
```

```
pwr.t.test(d = 2.300104/6.4,
  n = 50,
  sig.level = 0.05,
  type="two.sample",
  alternative="two.sided")$power
```

```
## [1] 0.4284243
```

```
d <- 2*2.300104/6.4
d
```

```
## [1] 0.7187825
```

```
pwr.t.test(d = d,
  n = 50,
  sig.level = 0.05,
  type="two.sample",
  alternative="two.sided")$power
```

```
## [1] 0.9450353
```

We can also compare the results against the analytic solution by Aberson (2019).

First, load the function for a 3-way ANOVA.

```
anova1f_3<-function(m1=NULL,m2=NULL,m3=NULL,s1=NULL,s2=NULL,s3=NULL,n1=NULL,n2=NULL,n3=NULL){
  x<-stats::rnorm(n1,m1,s1)
  X<-x
  MEAN<-m1
  SD<-s1
  Z <- (((X - mean(X, na.rm = TRUE))/stats::sd(X, na.rm = TRUE))) * SD
  y<-MEAN + Z
```



```

group<-rep("A1",n1)
l1<-data.frame(y, group)
x<-stats::rnorm(n2,m2,s2)
X<-x
MEAN<-m2
SD<-s2
Z <- (((X - mean(X, na.rm = TRUE))/stats::sd(X, na.rm = TRUE))) * SD
y<-MEAN + Z
group<-rep("A2",n2)
l2<-data.frame(y, group)
x<-stats::rnorm(n3,m3,s3)
X<-x
MEAN<-m3
SD<-s3
Z <- (((X - mean(X, na.rm = TRUE))/stats::sd(X, na.rm = TRUE))) * SD
y<-MEAN + Z
group<-rep("A3",n3)
l3<-data.frame(y, group)
simdat<-rbind(l1,l2,l3)
anova<-stats::aov(y~group, data=simdat)
anova<-car::Anova(anova, type="III")
SSA<-anova[2,1] #column, row
SSwin<-anova[3,1]
dfwin<-anova[3,2]
dfbg<-anova[2,2]
eta2<-SSA/(SSA+SSwin)
f2<-eta2/(1-eta2)
lambda<-f2*dfwin
minusalpha<-1-alpha
Ft<-stats::qf(minusalpha, dfbg, dfwin)
power<-1-stats::pf(Ft, dfbg,dfwin,lambda)
list(Power = power)}

```

Then we use the function to calculate power.

```

#Initial example, low power
anova1f_3(m1=24, m2=26.2, m3=26.6, s1=6.4, s2=6.4, s3=6.4, n1=50, n2=50, n3=50, alpha=.05)

```

```

## $Power
## [1] 0.4769468

```

```

#From: Aberson, Christopher L. Applied Power Analysis for the Behavioral Sciences, 2nd Edition.
# $Power [1] 0.4769468
#Later example, based on larger mean difference
anova1f_3(m1=-2.300104, m2=0, m3=2.300104, s1=6.4, s2=6.4, s3=6.4, n1=50, n2=50, n3=50, alpha=.05)

```

```
## $Power
## [1] 0.9000112
```

```
# $Power [1] 0.9000112
```

3.3 One-way ANOVA Part 3

We first repeat the simulation by Brysbaert:

```
# Simulations to estimate the power of an ANOVA with three unrelated groups
# the effect between the two extreme groups is set to d = .4, the effect for the third
# we use the built-in aov-test command
# give sample sizes (all samples sizes are equal)
N = 90
# give effect size d
d1 = .4 #difference between the extremes
d2 = .4 #third condition goes with the highest extreme
# give number of simulations
nSim = nsims
# give alpha levels
alpha1 = .05 #alpha level for the omnibus ANOVA
alpha2 = .05 #alpha level for three post hoc one-tailed t-tests Bonferroni correction
# create progress bar in case it takes a while
#pb <- winProgressBar(title = "progress bar", min = 0, max = nSim, width = 300)
# create vectors to store p-values
p1 <-numeric(nSim) #p-value omnibus ANOVA
p2 <-numeric(nSim) #p-value first post hoc test
p3 <-numeric(nSim) #p-value second post hoc test
p4 <-numeric(nSim) #p-value third post hoc test
pes1 <-numeric(nSim) #partial eta-squared
pes2 <-numeric(nSim) #partial eta-squared two extreme conditions
library(lsr)
for(i in 1:nSim){ #for each simulated experiment
  # setWinProgressBar(pb, i, title=paste(round(i/nSim*100, 1), "% done"))
  x<-rnorm(n = N, mean = 0, sd = 1)
  y<-rnorm(n = N, mean = d1, sd = 1)
  z<-rnorm(n = N, mean = d2, sd = 1)
  data = c(x,y,z)
  groups= factor(rep(letters[24:26], each = N))
  test <- aov(data~groups)
  pes1[i] <- etaSquared(test)[1,2]
  p1[i] <- summary(test)[[1]][["Pr(>F)"]][[1]]
  p2[i] <- t.test(x,y)$p.value
  p3[i] <- t.test(x,z)$p.value
```

```

p4[i] <- t.test(y,z)$p.value
data = c(x,y)
groups= factor(rep(letters[24:25], each = N))
test <- aov(data~groups)
pes2[i] <- etaSquared(test)[1,2]
}
#close(pb)#close progress bar
# results are as predicted when omnibus ANOVA is significant, t-tests are significant between x and y
#printing all unique tests (adjusted code by DL)
sum(p1<alpha1)/nSim

```

```
## [1] 0.71
```

```
sum(p2<alpha2)/nSim
```

```
## [1] 0.63
```

```
sum(p3<alpha2)/nSim
```

```
## [1] 0.73
```

```
sum(p4<alpha2)/nSim
```

```
## [1] 0.07
```

```
mean(pes1)
```

```
## [1] 0.03808708
```

```
mean(pes2)
```

```
## [1] 0.03816216
```

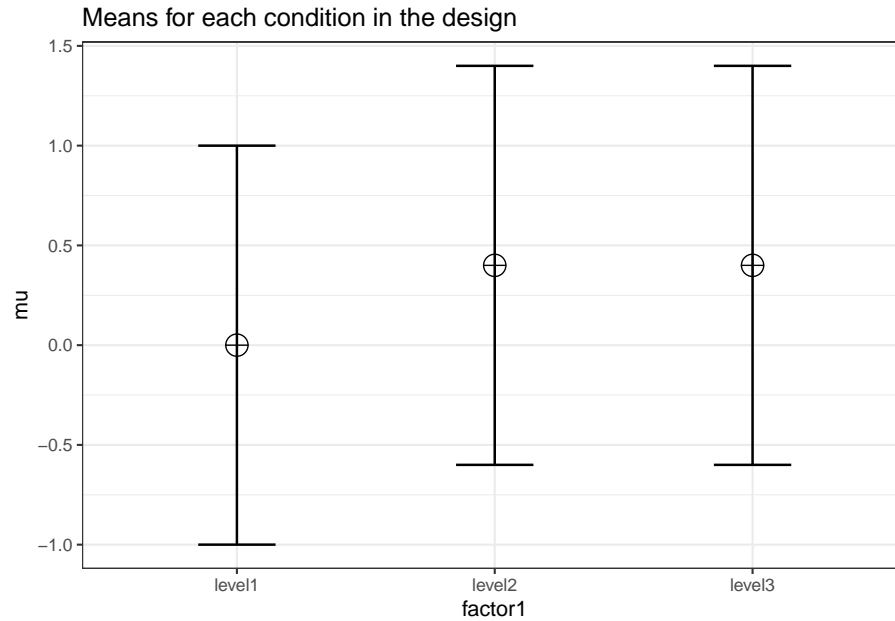
3.4 Three conditions replication

```

K <- 3
mu <- c(0, 0.4, 0.4)
n <- 90
sd <- 1
r <- 0
design =paste(K,"b",sep="")

```

```
design_result <- ANOVA_design(design =string,
                             n = n,
                             mu = mu,
                             sd = sd,
                             labelnames = c("factor1", "level1", "level2", "level3"))
```



```
ANOVA_power(design_result, nsims = nsims)
```

```
## Power and Effect sizes for ANOVA tests
##           power effect_size
## anova_factor1      81      0.04089
##
## Power and Effect sizes for contrasts
##           power effect_size
## p_factor1_level1_factor1_level2      82      0.40691
## p_factor1_level1_factor1_level3      75      0.39780
## p_factor1_level2_factor1_level3       3     -0.01002
```

3.5 Variation 1

```

# give sample sizes (all samples sizes are equal)
N = 145
# give effect size d
d1 = .4 #difference between the extremes
d2 = .0 #third condition goes with the highest extreme
# give number of simulations
nSim = nsims
# give alpha levels
alpha1 = .05 #alpha level for the omnibus ANOVA
alpha2 = .05 #alpha level for three post hoc one-tailed t-tests Bonferroni correction
# create progress bar in case it takes a while
#pb <- winProgressBar(title = "progress bar", min = 0, max = nSim, width = 300)
# create vectors to store p-values
p1 <-numeric(nSim) #p-value omnibus ANOVA
p2 <-numeric(nSim) #p-value first post hoc test
p3 <-numeric(nSim) #p-value second post hoc test
p4 <-numeric(nSim) #p-value third post hoc test
pes1 <-numeric(nSim) #partial eta-squared
pes2 <-numeric(nSim) #partial eta-squared two extreme conditions
library(lsr)
for(i in 1:nSim){ #for each simulated experiment
  # setWinProgressBar(pb, i, title=paste(round(i/nSim*100, 1), "% done"))
  x<-rnorm(n = N, mean = 0, sd = 1)
  y<-rnorm(n = N, mean = d1, sd = 1)
  z<-rnorm(n = N, mean = d2, sd = 1)
  data = c(x,y,z)
  groups= factor(rep(letters[24:26], each = N))
  test <- aov(data~groups)
  pes1[i] <- etaSquared(test)[1,2]
  p1[i] <- summary(test)[[1]][["Pr(>F)"]][[1]]
  p2[i] <- t.test(x,y)$p.value
  p3[i] <- t.test(x,z)$p.value
  p4[i] <- t.test(y,z)$p.value
  data = c(x,y)
  groups= factor(rep(letters[24:25], each = N))
  test <- aov(data~groups)
  pes2[i] <- etaSquared(test)[1,2]
}
#close(pb)#close progress bar
# results are as predicted when omnibus ANOVA is significant, t-tests are significant between x and y
#printing all unique tests (adjusted code by DL)
sum(p1<alpha1)/nSim

```

```
## [1] 0.9
```

```
sum(p2<alpha2)/nSim
```

```
## [1] 0.91
```

```
sum(p3<alpha2)/nSim
```

```
## [1] 0.03
```

```
sum(p4<alpha2)/nSim
```

```
## [1] 0.91
```

```
mean(pes1)
```

```
## [1] 0.03644733
```

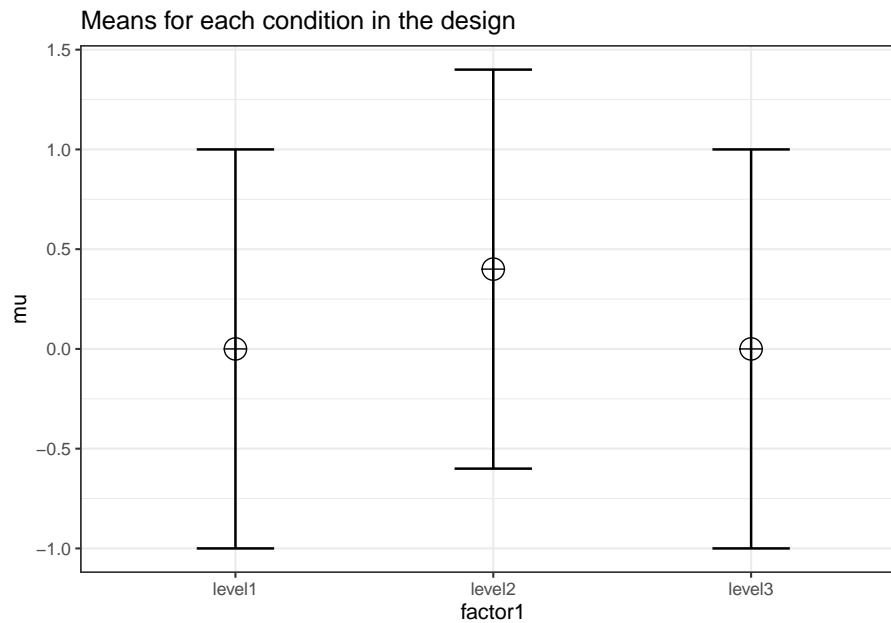
```
mean(pes2)
```

```
## [1] 0.04017308
```

3.6 Three conditions replication

```
K <- 3
mu <- c(0, 0.4, 0.0)
n <- 145
sd <- 1
r <- 0
design =paste(K,"b",sep="")
```

```
design_result <- ANOVA_design(design =string,
                             n = n,
                             mu = mu,
                             sd = sd,
                             labelnames = c("factor1", "level1", "level2", "level3"))
```



```
ANOVA_power(design_result, nsims = nsims)
```

```
## Power and Effect sizes for ANOVA tests
##           power effect_size
## anova_factor1    92    0.03871
##
## Power and Effect sizes for contrasts
##           power effect_size
## p_factor1_level1_factor1_level2    91    0.397884
## p_factor1_level1_factor1_level3     5   -0.005665
## p_factor1_level2_factor1_level3    91   -0.403446
```

3.7 Variation 2

```
# give sample sizes (all samples sizes are equal)
N = 82
# give effect size d
d1 = .4 #difference between the extremes
d2 = .2 #third condition goes with the highest extreme
# give number of simulations
nSim = nsims
```

```

# give alpha levels
alpha1 = .05 #alpha level for the omnibus ANOVA
alpha2 = .05 #alpha level for three post hoc one-tailed t-tests Bonferroni correction
# create progress bar in case it takes a while
#pb <- winProgressBar(title = "progress bar", min = 0, max = nSim, width = 300)
# create vectors to store p-values
p1 <-numeric(nSim) #p-value omnibus ANOVA
p2 <-numeric(nSim) #p-value first post hoc test
p3 <-numeric(nSim) #p-value second post hoc test
p4 <-numeric(nSim) #p-value third post hoc test
pes1 <-numeric(nSim) #partial eta-squared
pes2 <-numeric(nSim) #partial eta-squared two extreme conditions
library(lsr)
for(i in 1:nSim){ #for each simulated experiment
  # setWinProgressBar(pb, i, title=paste(round(i/nSim*100, 1), "% done"))
  x<-rnorm(n = N, mean = 0, sd = 1)
  y<-rnorm(n = N, mean = d1, sd = 1)
  z<-rnorm(n = N, mean = d2, sd = 1)
  data = c(x,y,z)
  groups= factor(rep(letters[24:26], each = N))
  test <- aov(data~groups)
  pes1[i] <- etaSquared(test)[1,2]
  p1[i] <- summary(test)[[1]][["Pr(>F)"]][[1]]
  p2[i] <- t.test(x,y)$p.value
  p3[i] <- t.test(x,z)$p.value
  p4[i] <- t.test(y,z)$p.value
  data = c(x,y)
  groups= factor(rep(letters[24:25], each = N))
  test <- aov(data~groups)
  pes2[i] <- etaSquared(test)[1,2]
}
#close(pb)#close progress bar
# results are as predicted when omnibus ANOVA is significant, t-tests are significant
#printing all unique tests (adjusted code by DL)
sum(p1<alpha1)/nSim

```

```
## [1] 0.58
```

```
sum(p2<alpha2)/nSim
```

```
## [1] 0.62
```



```
sum(p3<alpha2)/nSim
```

```
## [1] 0.26
```

```
sum(p4<alpha2)/nSim
```

```
## [1] 0.2
```

```
mean(pes1)
```

```
## [1] 0.03198582
```

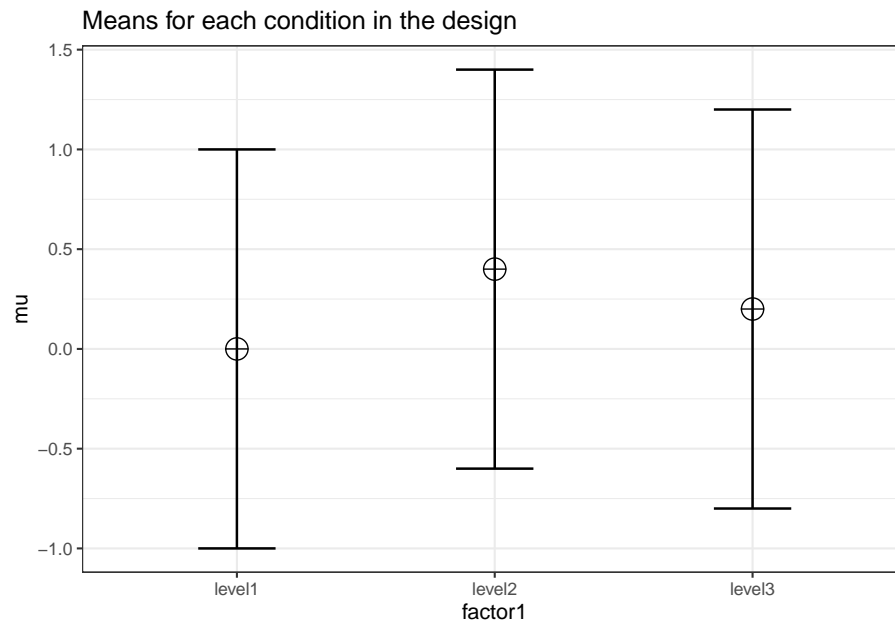
```
mean(pes2)
```

```
## [1] 0.04184699
```

3.8 Three conditions replication

```
K <- 3  
mu <- c(0, 0.4, 0.2)  
n <- 82  
sd <- 1  
design =paste(K,"b",sep="")
```

```
design_result <- ANOVA_design(design =string,  
                             n = n,  
                             mu = mu,  
                             sd = sd,  
                             labelnames = c("factor1", "level1", "level2", "level3"))
```



```
ANOVA_power(design_result, nsims = nsims)
```

```
## Power and Effect sizes for ANOVA tests
##           power effect_size
## anova_factor1    58    0.03266
##
## Power and Effect sizes for contrasts
##                               power effect_size
## p_factor1_level1_factor1_level2    71    0.3891
## p_factor1_level1_factor1_level3    24    0.1995
## p_factor1_level2_factor1_level3    20   -0.1892
```

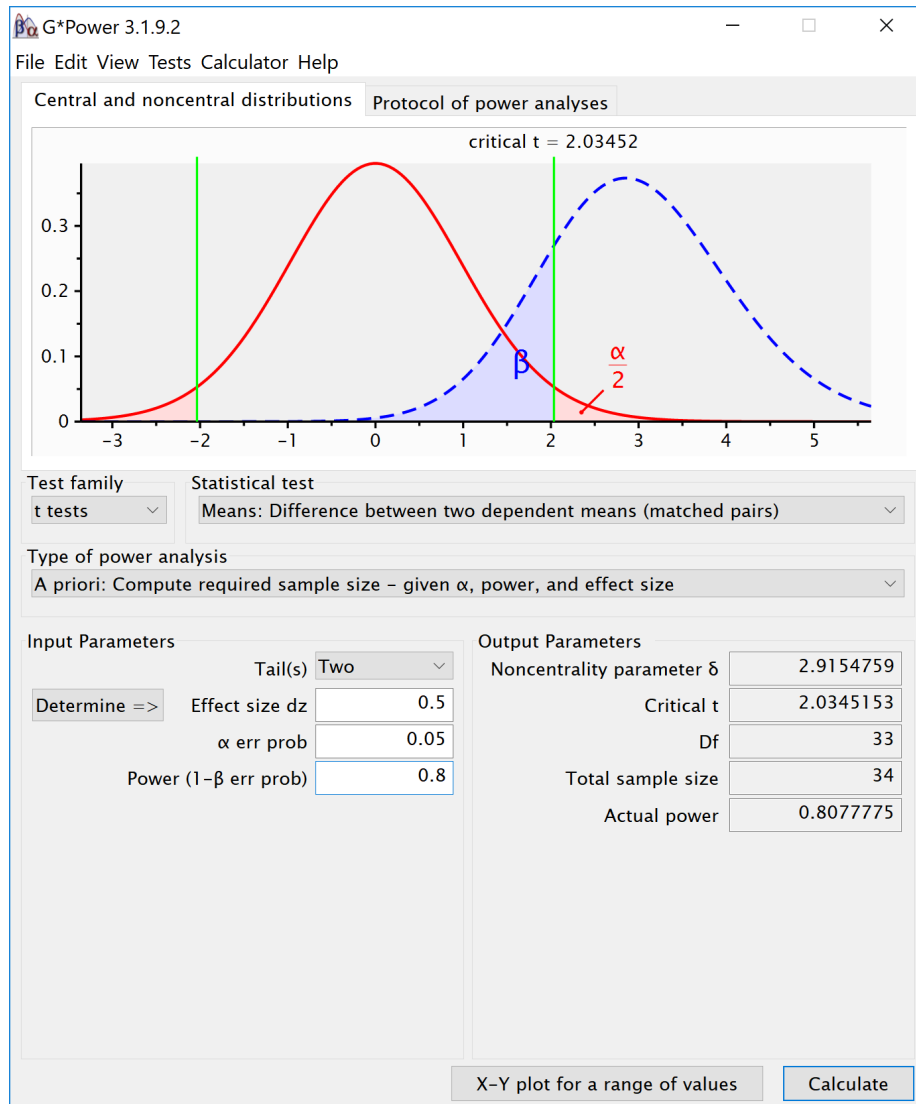
Chapter 4

Repeated Measures-ANOVA Part 1

In a repeated measures design multiple observations are collected from the same participants. In the simplest case, where there are two repeated observations, a repeated measures ANOVA equals a dependent or paired t -test. The difference compared to a between subject design is that repeated measures can be correlated, and in psychology, they often are. Let's first explore the impact of this correlation on the power of a repeated measures ANOVA.

4.1 Two conditions, medium effect size

To illustrate the effect of correlated observations, we start by simulating data for a medium effect size for a dependent (or paired, or within-subject) t -test. Let's first look at G*power. If we want to perform an a-priori power analysis, we are asked to fill in the effect size d_z . As Cohen (1988) writes, "The Z subscript is used to emphasize the fact that our raw score unit is no longer X or Y, but Z", where Z are the difference scores of X-Y.



Within designs can have greater power to detect differences than between designs because the values are correlated, and a within design requires less participants because each participant provides multiple observations. One difference between an independent t -test and a dependent t -test is that an independent t -test has $2(n-1)$ degrees of freedom, while a dependent t -test has $(n-1)$ degrees of freedom. The sample size needed in a two-group within-design (NW) relative to the sample needed in two-group between-designs (NB), assuming normal distributions, and ignoring the difference in degrees of freedom between the two types of tests, is (from Maxwell & Delaney, 2004, p. 561, formula 45):

$$N_W = \frac{N_B(1-\rho)}{2}$$

The division by 2 in the equation is due to the fact that in a two-condition within design every participant provides two data-points. The extent to which this reduces the sample size compared to a between-subject design depends on the correlation (r) between the two dependent variables, as indicated by the $1-r$ part of the equation. If the correlation is 0, a within-subject design needs half as many participants as a between-subject design (e.g., 64 instead 128 participants), simply because every participants provides 2 datapoints. The higher the correlation, the larger the relative benefit of within designs, and whenever the correlation is negative (up to -1) the relative benefit disappears.

Whereas in an independent t -test the two observations are uncorrelated, in a within design the observations are correlated. This has an effect on the standard deviation of the difference scores. In turn, because the standardized effect size is the mean difference divided by the standard deviation of the difference scores, the correlation has an effect on the standardized mean difference in a within design, Cohen's d_z . The relation, as Cohen (1988, formula 2.3.7) explains, is:

$$\sigma_z = \sigma \sqrt{2(1 - \rho)}$$

Therefore, the relation between d_z and d is $\sqrt{2(1 - \rho)}$. As Cohen (1988) writes: "In other words, a given difference between population means for matched (dependent) samples is standardized by a value which is $\sqrt{2(1 - \rho)}$ as large as would be the case were they independent. If we enter a correlation of 0.5 in the formula, we get $\sqrt{2(0.5)} = 1$. In other words, when the correlation is 0.5, $d = d_z$. When there is a strong correlation between dependent variables, for example $r = 0.9$, we get $d = d_z \sqrt{2(1 - 0.9)}$, and a d_z of 1 would be a $d = 0.45$. Reversely, $d_z = \frac{d}{\sqrt{2(1-r)}}$, so with a $r = 0.9$, a d of 1 would be a $d_z = 2.24$.

Some consider this increase in d_z compared to d when observations are strongly correlated an 'inflation' when estimating effect sizes, but since the reduction in the standard deviation of the difference scores due to the correlation makes it easier to distinguish signal from noise in a hypothesis test, it leads to a clear power benefit.

```
# Check sample size formula Maxwell
# Power is pretty similar with n/2, same d (assuming r = 0.5).
# Small differences due to df = 2(n-1) vs df = n-1
pwr.t.test(d = 0.05,
            n = c(2000, 4000, 8000),
            sig.level = 0.05,
            type = "two.sample",
            alternative = "two.sided")
```

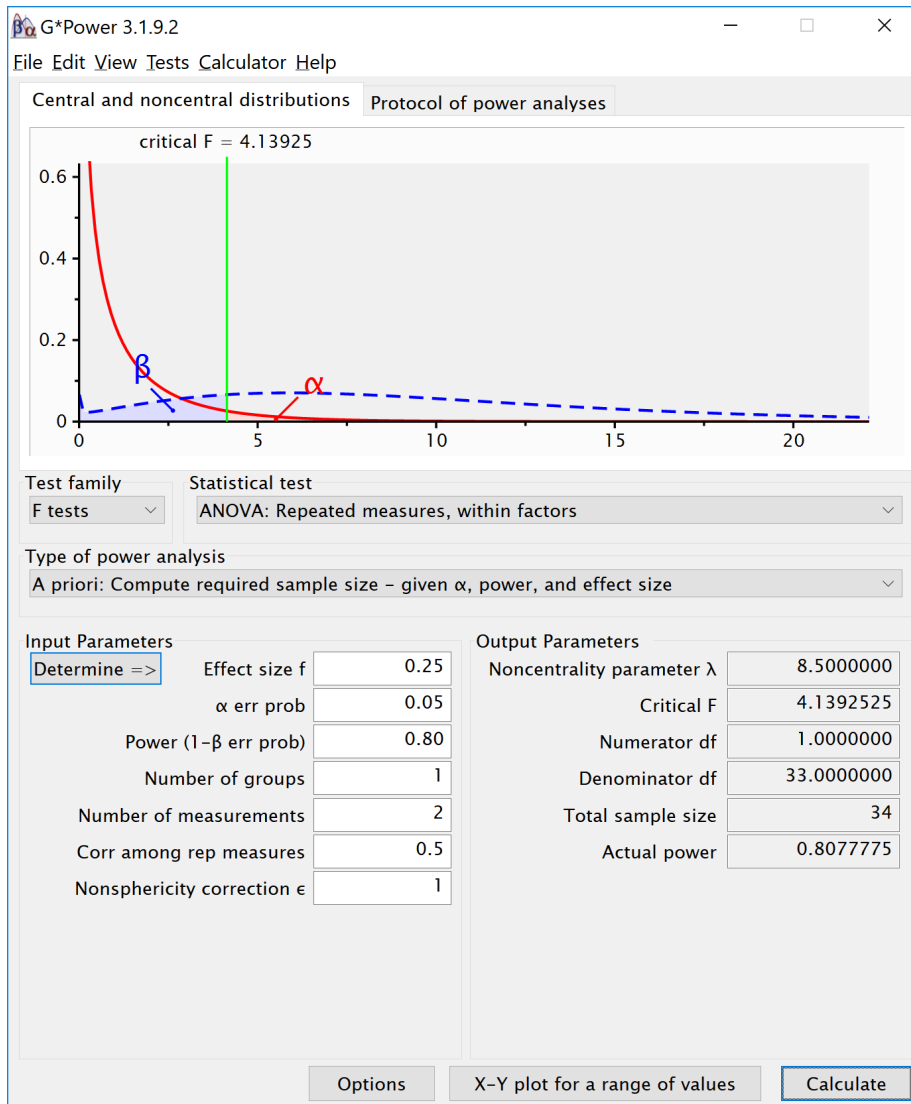
```
##
##      Two-sample t test power calculation
##
##              n = 2000, 4000, 8000
```

```
##          d = 0.05
##      sig.level = 0.05
##      power = 0.3524674, 0.6086764, 0.8853424
##      alternative = two.sided
##
## NOTE: n is number in each group
```

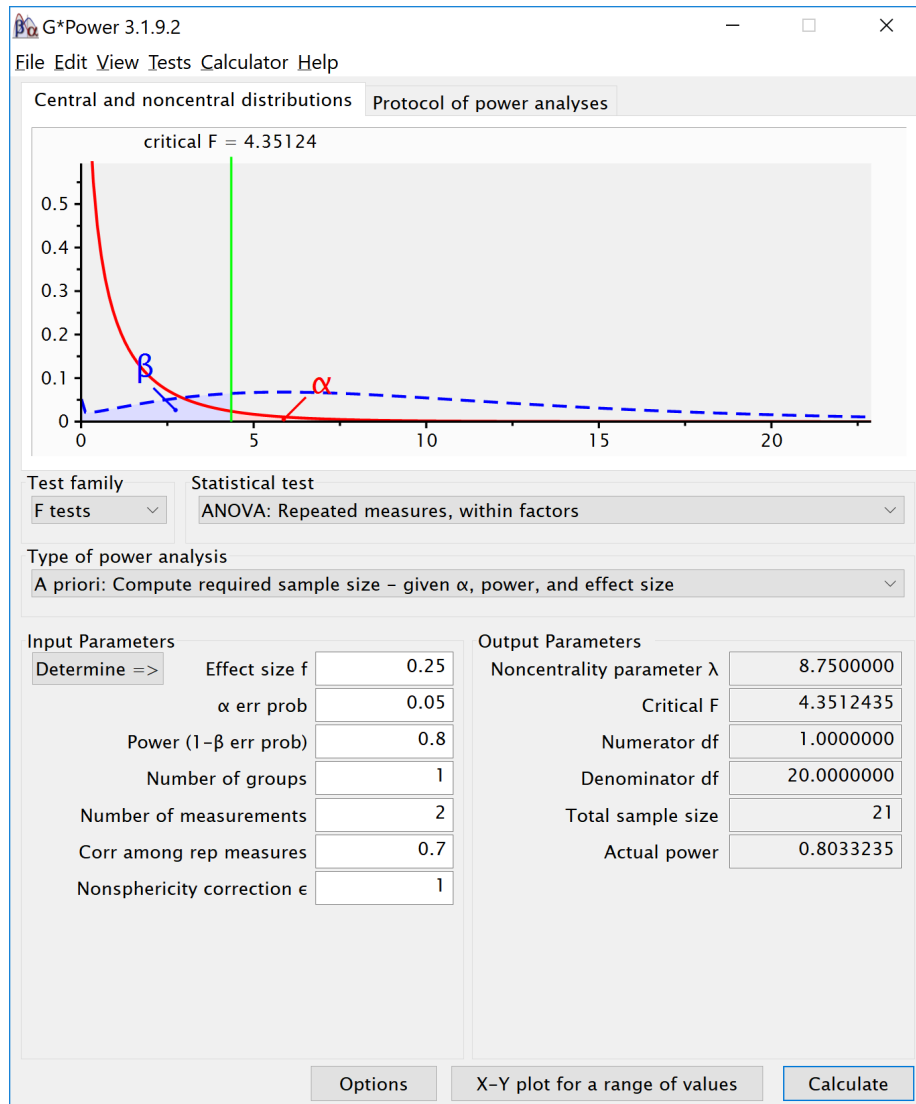
```
pwr.t.test(d = 0.05,
           n = c(1000, 2000, 4000),
           sig.level = 0.05,
           type = "paired",
           alternative = "two.sided")
```

```
##
##      Paired t test power calculation
##
##          n = 1000, 2000, 4000
##          d = 0.05
##      sig.level = 0.05
##      power = 0.3520450, 0.6083669, 0.8852320
##      alternative = two.sided
##
## NOTE: n is number of pairs
```

There is no equivalent f_z for Cohen's f for a within subject ANOVA. For two groups, we can directly compute Cohen's f from Cohen's d for two groups, as Cohen (1988) describes, because $f = 1/2d$. For a $d = 0.5$, $f = 0.25$. In Gpower we can run a 2 group within-subject power analysis for ANOVA. We plan for 80% power, and reproduce the analysis above for the dependent t -test. This works because the correlation is set to 0.5, when $d = dz$, and thus the transformation of $f=1/2d$ works.

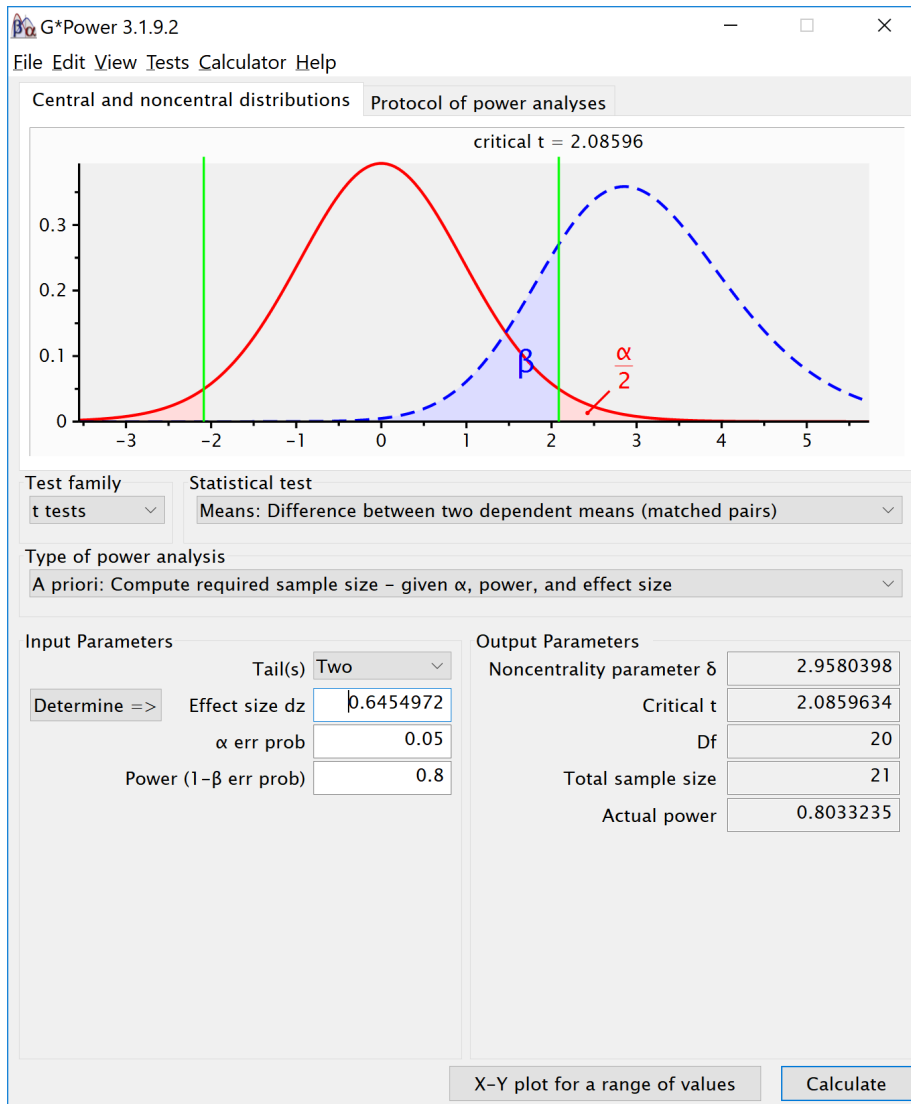


If we change the correlation to 0.7 and keep all other settings the same, the repeated measure a-priori power analysis yields a sample of 21. The correlation increases the power for the test.



To reproduce this analysis in Gpower with a dependent t -test we need to change d_z following the formula above, $d_z = \frac{0.5}{\sqrt{2(1-0.7)}}$, which yields $d_z = 0.6454972$.

If we enter this value in Gpower for an a-priori power analysis, we get the exact same results (as we should, since an repeated measures ANOVA with 2 groups equals a dependent t -test). This example illustrates that the correlation between dependent variables always factors into a power analysis, both for a dependent t -test, and for a repeated measures ANOVA. Because a dependent t -test uses d_z the correlation might be less visible, but given the relation between d and d_z , the correlation is always taken into account and can greatly improve power for within designs compared to between designs.

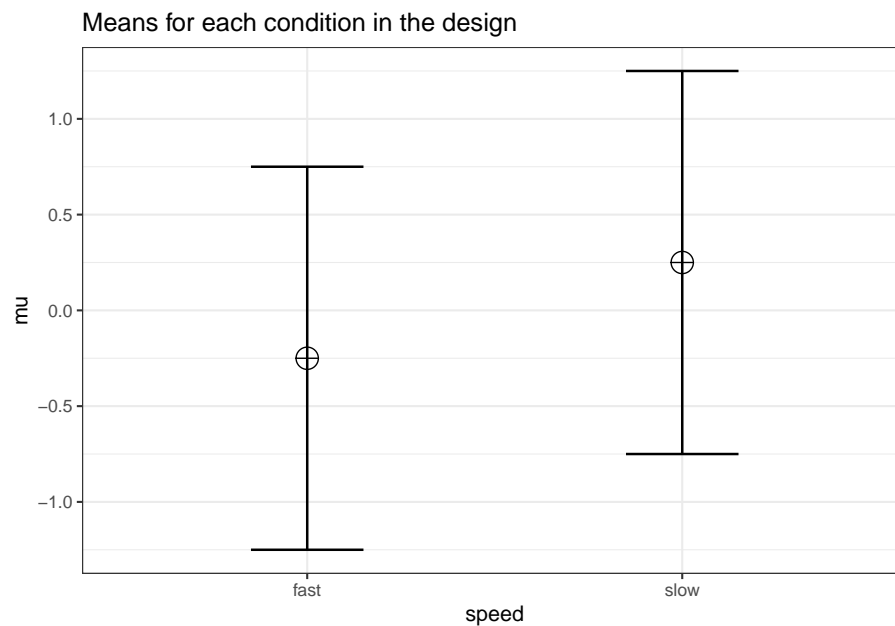


We can perform both these power analyses using simulations as well. We set groups to 2 for the simulation, $n = 34$ (which should give 80.777 power, according to the g*power program), a correlation among repeated measures of 0.5, and an alpha of 0.05. In this case, we simulate data with means -0.25 and 0.25, and set the sd to 1. This means we have a mean difference of 0.5, and a Cohen's d of $0.5/1 = 0.5$. In the first example, we set the correlation to 0.5, and the result should be 80.77% power, and an effect size estimate of 0.5 for the simple effect. We also calculate partial eta-squared for the ANOVA, which equals $\frac{f^2}{f^2+1}$, or 0.05882353.

```
K <- 2
n <- 34
sd <- 1
r <- 0.5
alpha = 0.05
f <- 0.25
f2 <- f^2
ES <- f2/(f2+1)
ES
```

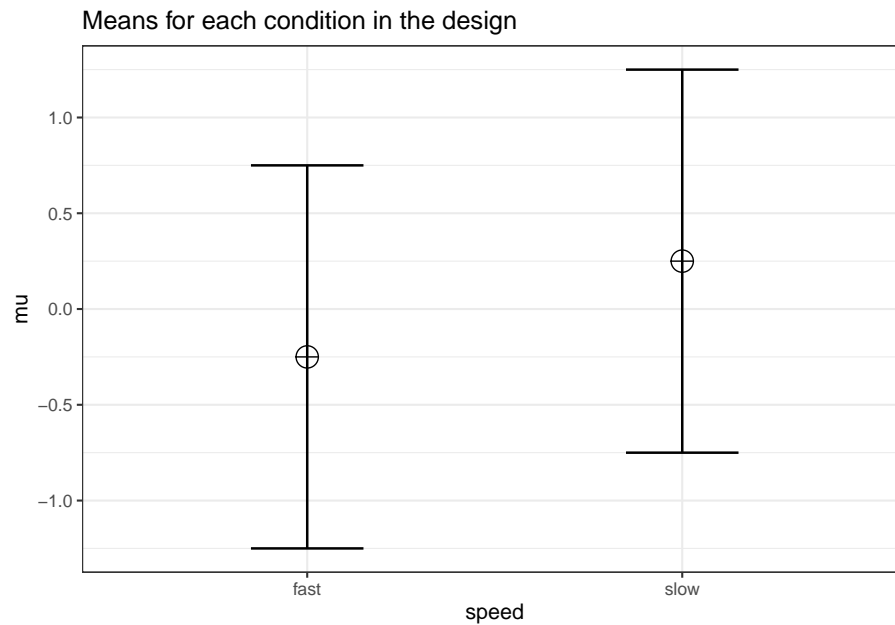
```
## [1] 0.05882353
```

```
mu <- mu_from_ES(K = K, ES = ES)
design = paste(K, "w", sep = "")
labelnames <- c("speed", "fast", "slow")
design_result <- ANOVA_design(design = design,
                             n = n,
                             mu = mu,
                             sd = sd,
                             r = r,
                             labelnames = labelnames)
```



```
alpha_level <- 0.05
ANOVA_power(design_result, nsims = nsims)

## Power and Effect sizes for ANOVA tests
##           power effect_size
## anova_speed      83      0.2252
##
## Power and Effect sizes for contrasts
##           power effect_size
## p_speed_fast_speed_slow      83      0.5273
##
## Within-Subject Factors Included: Check MANOVA Results
```



```
alpha_level <- 0.05
design_result$sigmatrix
```

```
##      fast slow
## fast  1.0  0.7
## slow  0.7  1.0
```

```
ANOVA_power(design_result, nsims = nsims)
```

```
## Power and Effect sizes for ANOVA tests
##           power effect_size
## anova_speed    81      0.3359
##
## Power and Effect sizes for contrasts
##           power effect_size
## p_speed_fast_speed_slow    81      0.7132
##
## Within-Subject Factors Included: Check MANOVA Results
```

```
#relation dz and f for within designs
```

```
f <- 0.5*0.6454972
```

```
# Entering this f in G*power, with a correlation of 0.5, yields the same as entering f
```

Chapter 5

Repeated Measures-ANOVA Part 2

In a repeated measures design multiple observations are collected from the same participants. Here, we will examine a repeated measures experiment with 3 within-subject conditions, to illustrate how a repeated measures ANOVA extends a dependent *t*-test with 3 groups.

In the example for a two-group within design we provided a specific formula for the sample size benefit for two groups. The sample size needed in within-designs (NW) with more than 2 conditions, relative to the sample needed in between-designs (NB), assuming normal distributions and compound symmetry, and ignoring the difference in degrees of freedom between the two types of tests, is (from Maxwell & Delaney, 2004, p. 562, formula 47):

$$N_W = \frac{N_B(1-\rho)}{a}$$

Where *a* is the number of within-subject levels.

5.1 The relation between Cohen's *f* and Cohen's *d*

Whereas in the case of a repeated measures ANOVA with 2 groups we could explain the principles of a power analysis by comparing our test against a *t*-test and Cohen's *d*, this becomes more difficult when we have more than 2 groups. It is more useful to explain how to directly calculate Cohen's *f*, the effect size used in power analyses for ANOVA. Cohen's *f* is calculated following Cohen, 1988, formula 8.2.1 and 8.2.2:

$$f = \sqrt{\frac{\sum (\mu - \bar{\mu})^2}{\frac{N}{\sigma}}}$$

Imagine we have a within-subject experiment with 3 conditions. We ask people what their mood is when their alarm clock wakes them up, when they wake up naturally on a week day, and when they wake up naturally on a weekend day. Based on pilot data, we expect the means (on a 7 point validated mood scale) are 3.8, 4.2, and 4.3. The standard deviation is 0.9, and the correlation between the dependent measurements is 0.7. We can calculate Cohen's f for the ANOVA, and Cohen's d_z for the contrasts:

```
mu <- c(3.8, 4.2, 4.3)
sd <- 0.9
f <- sqrt(sum((mu-mean(mu))^2)/length(mu))/sd #Cohen, 1988, formula 8.2.1 and 8.2.2
f
```

```
## [1] 0.2400274
```

```
r <- 0.7
(4.2-3.8)/0.9/sqrt(2*(1-r))
```

```
## [1] 0.5737753
```

```
(4.3-3.8)/0.9/sqrt(2*(1-r))
```

```
## [1] 0.7172191
```

```
(4.3-4.2)/0.9/sqrt(2*(1-r))
```

```
## [1] 0.1434438
```

The relation between Cohen's d or d_z and Cohen's f becomes more difficult when there are multiple groups, because the relationship depends on the pattern of the means. Cohen (1988) presents calculations for three patterns, minimal variability (for example, for 5 means: -0.25, 0, 0, 0, 0.25), medium variability (for example, for 5 means: -0.25, -0.25, 0.25, 0.25, 0.25 or -0.25, -0.25, -0.25, 0.25, 0.25). For these three patterns, formula's are available that compute Cohen's f from Cohen's d , where d is the effect size calculated for the difference between the largest and smallest mean (if the largest mean is 0.25 and the smallest mean is -0.25, $0.25 - -0.25 = 0.5$, so d is 0.5 divided by the standard deviation of 0.9). In our example, d would be $(4.3-3.8)/0.9 = 0.5555556$. If we divide this value by $\sqrt{2*(1-r)}$ we have $d_z = 0.5555556/0.7745967 = 0.7172191$.

I have created a custom function that will calculate f from d , based on a specification of one of the three patterns of means. Our pattern is most similar (but not identical) to a maximum variability pattern (two means are high, one is lower). So we could attempt to calculate f from d (0.5555556), by calculating d from the largest and smallest mean:

```
source("https://raw.githubusercontent.com/Lakens/ANOVA_power_simulation/master/calc_f_d_eta.R")
res <- calc_f_d_eta(mu = mu, sd = sd, variability = "maximum")
res$f
```

```
## [1] 0.2618914
```

```
res$d
```

```
## [1] 0.5555556
```

We see the Cohen's f value is 0.2618914 and $d = 0.5555556$. The Cohen's f is not perfectly accurate - it is assuming the pattern of means is 3.8, 4.3, 4.3, and not 3.8, 4.2, 4.3. If the means and sd is known, it is best to calculate Cohen's f directly from these values.

5.2 Three within conditions, medium effect size

We can perform power analyses for within designs using simulations. We set groups to 3 for the simulation, $n = 20$, and the correlation between dependent variables to 0.8. If the true effect size is $f = 0.25$, and the alpha level is 0.05, the power is 96.6%.

In this case, we simulate data with means -0.3061862, 0.0000000, and 0.3061862, and set the sd to 1.

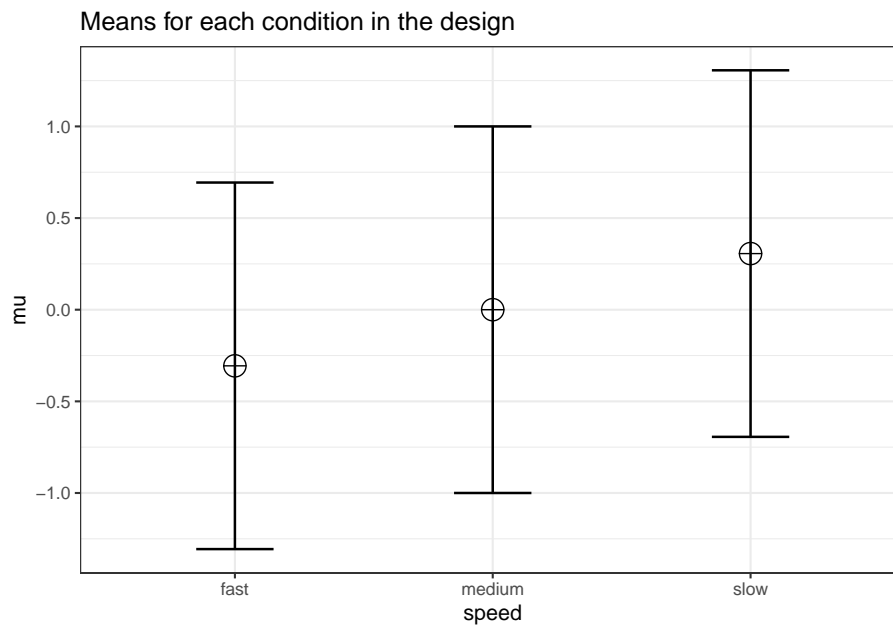
```
K <- 3
n <- 20
sd <- 1
r <- 0.8
alpha = 0.05
f <- 0.25
f2 <- f^2
ES <- f2/(f2+1)
ES
```

```
## [1] 0.05882353
```

```
mu <- mu_from_ES(K = K, ES = ES)
sqrt(sum((mu-mean(mu))^2)/length(mu))/sd #Cohen, 1988, formula 8.2.1 and 8.2.2
```

```
## [1] 0.25
```

```
design =paste(K,"w",sep="")
labelnames <- c("speed", "fast", "medium", "slow")
design_result <- ANOVA_design(design = design,
                             n = n,
                             mu = mu,
                             sd = sd,
                             r = r,
                             labelnames = labelnames)
```



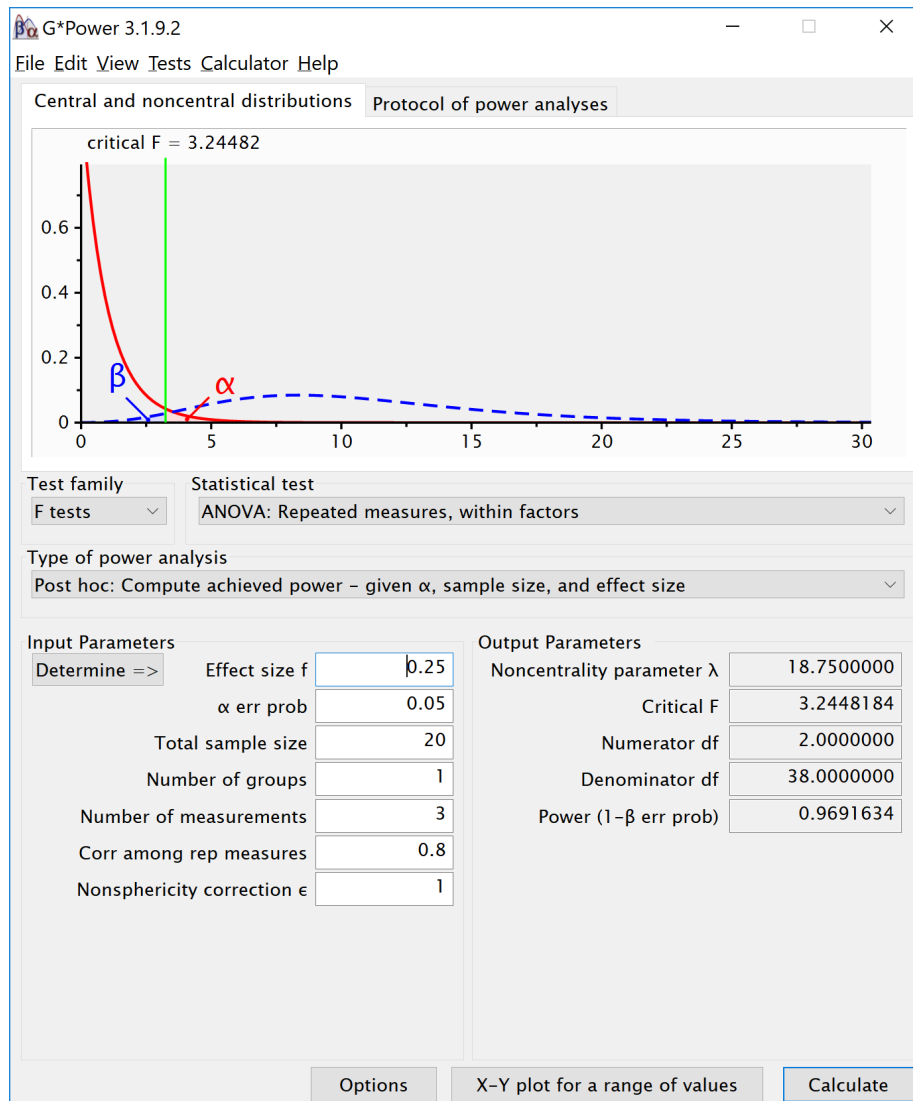
```
alpha_level <- 0.05
ANOVA_power(design_result, nsims = nsims)
```

```
## Power and Effect sizes for ANOVA tests
##           power effect_size
## anova_speed    98      0.351
##
## Power and Effect sizes for contrasts
##           power effect_size
```



```
## p_speed_fast_speed_medium    52    0.5183
## p_speed_fast_speed_slow      100    1.0454
## p_speed_medium_speed_slow    58    0.5029
##
## Within-Subject Factors Included: Check MANOVA Results
```

The results of the simulation are indeed very close to 96.9%. We can see this is in line with the power estimate from Gpower:

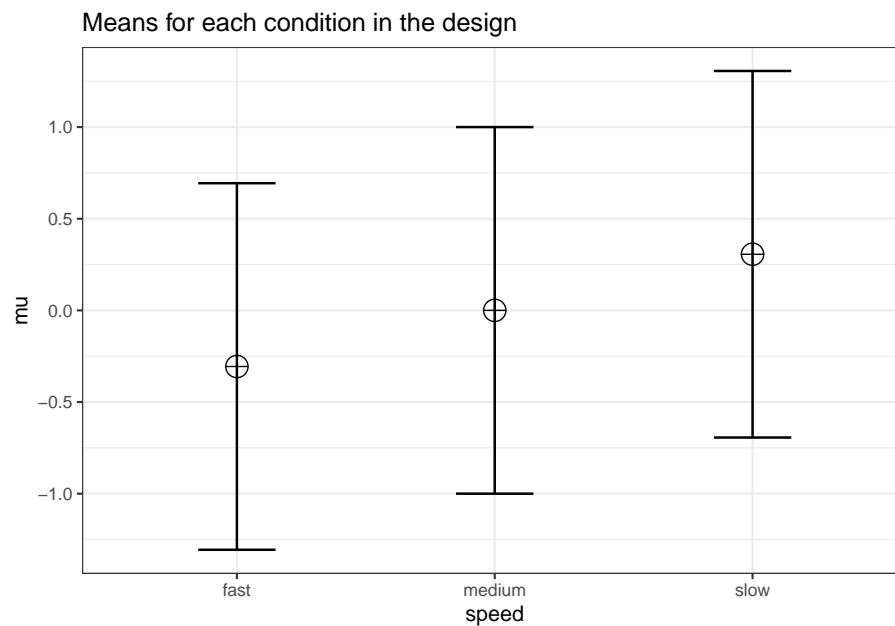


We can also validate this by creating the code to do a power analysis in R from scratch:

```
K <- 3 #three groups
n <- 20
sd <- 1
r <- 0.8
alpha = 0.05
f <- 0.25
f2 <- f^2
ES <- f2/(f2+1)
ES
```

```
## [1] 0.05882353
```

```
mu <- mu_from_ES(K = K, ES = ES)
design = paste(K, "w", sep = "")
labelnames <- c("speed", "fast", "medium", "slow")
design_result <- ANOVA_design(design = design,
                             n = n,
                             mu = mu,
                             sd = sd,
                             r = r,
                             labelnames = labelnames)
```



```
power_oneway_within(design_result)$power
```

```
## [1] 0.9691634
```

```
power_oneway_within(design_result)$eta_p_2
```

```
## [1] 0.05882353
```

```
power_oneway_within(design_result)$eta_p_2_SPSS
```

```
## [1] 0.3303965
```

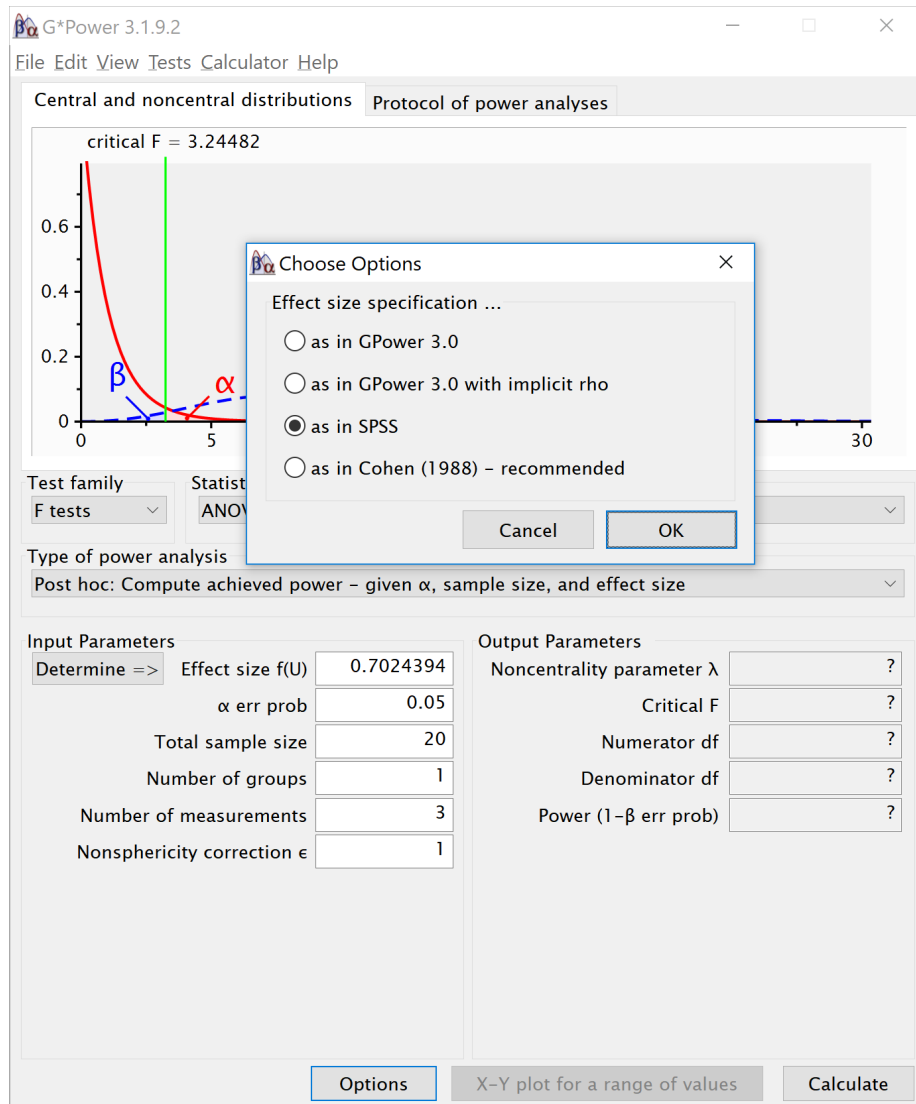
```
power_oneway_within(design_result)$Cohen_f
```

```
## [1] 0.25
```

```
power_oneway_within(design_result)$Cohen_f_SPSS
```

```
## [1] 0.7024394
```

We can even check the calculation of Cohen's f SPSS style in GPower. We take the GPower settings as illustrated above. We click the 'Options' button, and check the radiobutton next to 'As in SPSS'. Click ok, and you will notice that the 'Corr among rep measures' field has disappeared. The correlation does not need to be entered separately, but is incorporated in Cohen's f . The value of Cohen's f , which was 0.25, has changed into 0.7024394. This is the SPSS equivalent. The value is much larger. This value, and its corresponding partial eta-squared, incorporate the correlation between observations.



Chapter 6

Repeated Measures-ANOVA Part 3

We first repeat the simulation by Brysbaert:

```
# give sample size
N = 75
# give effect size d
d1 = .4 #difference between the extremes
d2 = .4 #third condition goes with the highest extreme
# give the correlation between the conditions
r = .5
# give number of simulations
nSim = nsims
# give alpha levels
alpha1 = .05 #alpha level for the omnibus ANOVA
alpha2 = .05 #also adjusted from original by DL
# create progress bar in case it takes a while
#pb <- winProgressBar(title = "progress bar", min = 0, max = nSim, width = 300)
# create vectors to store p-values
p1 <- numeric(nSim) #p-value omnibus ANOVA
p2 <- numeric(nSim) #p-value first post hoc test
p3 <- numeric(nSim) #p-value second post hoc test
p4 <- numeric(nSim) #p-value third post hoc test
# open library MASS
library('MASS')
# define correlation matrix
rho <- cbind(c(1, r, r), c(r, 1, r), c(r, r, 1))
# define participant codes
part <- paste("part", seq(1:N))
```

```

for(i in 1:nSim){ #for each simulated experiment
  # setWinProgressBar(pb, i, title=paste(round(i/nSim*100, 1), "% done"))
  data = mvrnorm(n=N, mu=c(0, 0, 0), Sigma=rho)
  data[,2] = data[,2]+d1
  data[,3] = data[,3]+d2
  datalong = c(data[,1],data[,2],data[,3])
  conds= factor(rep(letters[24:26], each = N))
  partID = factor(rep(part, times = 3))
  output <- data.frame(partID,conds,dalong)
  test <- aov(dalong~conds + Error(partID/conds), data=output)
  tests <- (summary(test))
  p1[i] <- tests$'Error: partID:conds'[[1]]$'Pr(>F)'[[1]]
  p2[i] <- t.test(data[,1],data[,2], paired=TRUE)$p.value
  p3[i] <- t.test(data[,1],data[,3], paired=TRUE)$p.value
  p4[i] <- t.test(data[,2],data[,3], paired=TRUE)$p.value
}
#close(pb)#close progress bar
#printing all unique tests (adjusted code by DL)
sum(p1<alpha1)/nSim

```

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

```
sum(p2<alpha2)/nSim
```

```
## [1] 0.96
```

```
sum(p3<alpha2)/nSim
```

```
## [1] 0.94
```

```
sum(p4<alpha2)/nSim
```

```
## [1] 0.06
```

6.1 Reproducing Brysbaert

We can reproduce the same results as Brysbaert finds with his code:

```

design <- "3w"
n <- 75
mu <- c(0, 0.4, 0.4)
sd <- 1
r <- 0.5
labelnames <- c("speed", "fast", "medium", "slow")

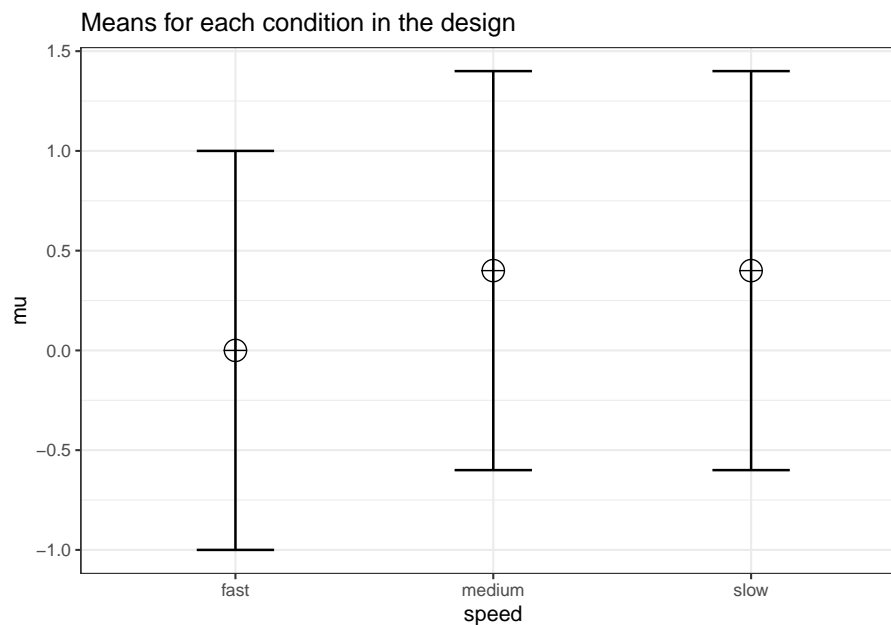
```

We create the within design, and run the simulation

```

design_result <- ANOVA_design(design = design,
                             n = n,
                             mu = mu,
                             sd = sd,
                             r = r,
                             labelnames = labelnames)

```



```
ANOVA_power(design_result, nsims = nsims)
```

```

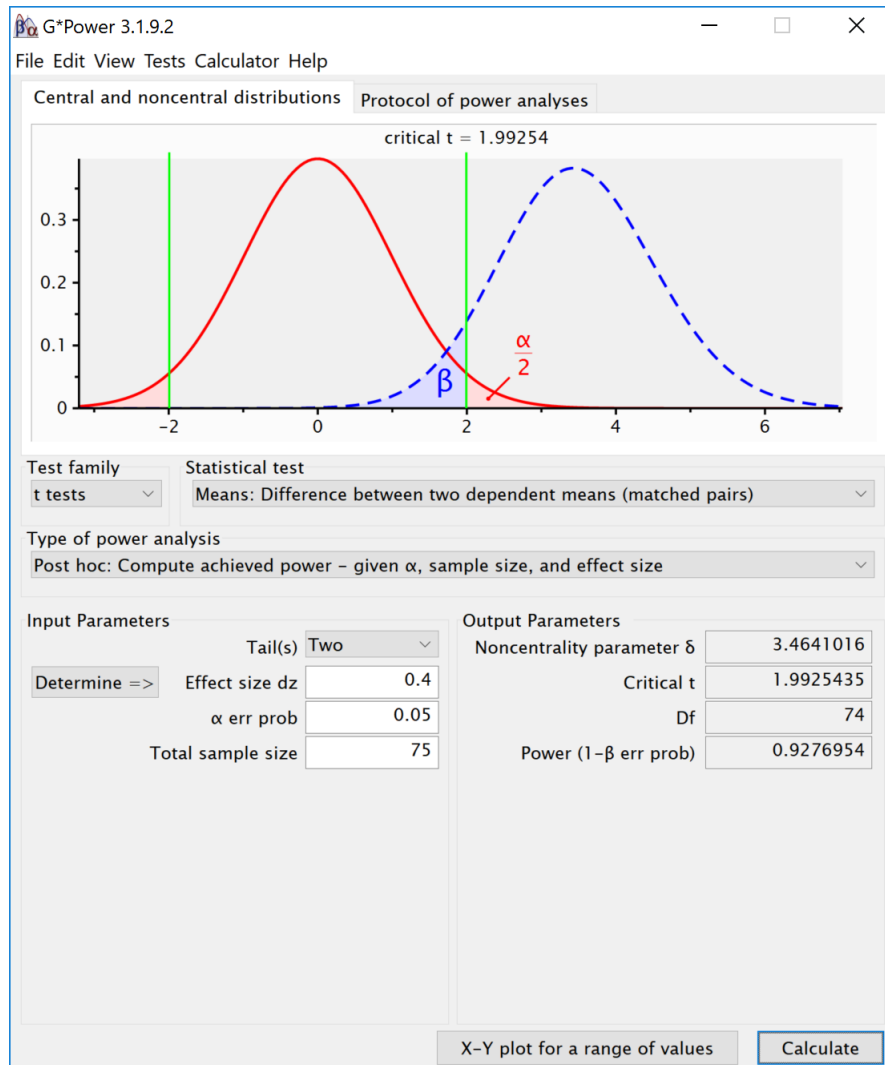
## Power and Effect sizes for ANOVA tests
##           power effect_size
## anova_speed    98    0.1108
##
## Power and Effect sizes for contrasts

```

```
##                                     power effect_size
## p_speed_fast_speed_medium      89    0.411530
## p_speed_fast_speed_slow       95    0.414675
## p_speed_medium_speed_slow      8     0.005365
##
## Within-Subject Factors Included: Check MANOVA Results
```

#Results

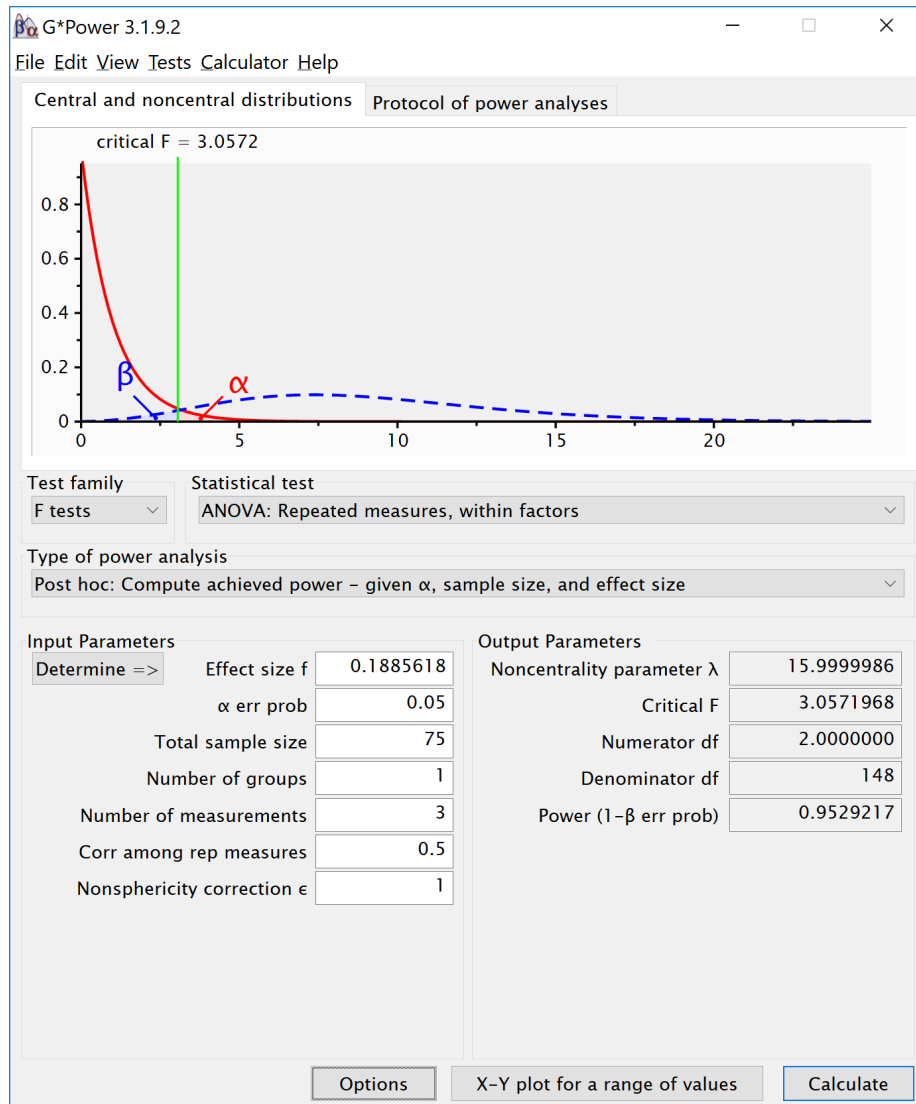
The results of the simulation are very similar. Power for the ANOVA F -test is around 95.2%. For the three paired t -tests, power is around 92.7. This is in line with the a-priori power analysis when using *g*power*:



We can perform an post-hoc power analysis in G*power. We can calculate Cohen's f based on the means and sd, using our own custom formula.

```
# Our simulation is based on the following means and sd:
# mu <- c(0, 0.4, 0.4)
# sd <- 1
f <- sqrt(sum((mu-mean(mu))^2)/length(mu))/sd #Cohen, 1988, formula 8.2.1 and 8.2.2
# We can see why f = 0.5*d.
# Imagine 2 group, mu = 1 and 2
# Grand mean is 1.5, we have sqrt(sum(0.5^2 + 0.5^2)/2), or sqrt(0.5/2), = 0.5.
# For Cohen's d we use the difference, 2-1 = 1.
```

The Cohen's f is 0.1885618. We can enter the f (using the default 'as in G*Power 3.0' in the option window) and enter a sample size of 75, number of groups as 1, number of measurements as 3, correlation as 0.5. This yields:



6.2 Reproducing Brysbaert Variation 1 Changing Correlation

```
# give sample size
N = 75
# give effect size d
d1 = .4 #difference between the extremes
```

6.2. REPRODUCING BRYLSBAERT VARIATION 1 CHANGING CORRELATION59

```
d2 = .4 #third condition goes with the highest extreme
# give the correlation between the conditions
r = .6 #increased correlation
# give number of simulations
nSim = nsims
# give alpha levels
alpha1 = .05 #alpha level for the omnibus ANOVA
alpha2 = .05 #also adjusted from original by DL
# create progress bar in case it takes a while
#pb <- winProgressBar(title = "progress bar", min = 0, max = nSim, width = 300)
# create vectors to store p-values
p1 <- numeric(nSim) #p-value omnibus ANOVA
p2 <- numeric(nSim) #p-value first post hoc test
p3 <- numeric(nSim) #p-value second post hoc test
p4 <- numeric(nSim) #p-value third post hoc test
# open library MASS
library('MASS')
# define correlation matrix
rho <- cbind(c(1, r, r), c(r, 1, r), c(r, r, 1))
# define participant codes
part <- paste("part", seq(1:N))
for(i in 1:nSim){ #for each simulated experiment
  # setWinProgressBar(pb, i, title=paste(round(i/nSim*100, 1), "% done"))
  data = mvrnorm(n=N, mu=c(0, 0, 0), Sigma=rho)
  data[,2] = data[,2]+d1
  data[,3] = data[,3]+d2
  datalong = c(data[,1], data[,2], data[,3])
  conds = factor(rep(letters[24:26], each = N))
  partID = factor(rep(part, times = 3))
  output <- data.frame(partID, conds, datalong)
  test <- aov(datalong~conds + Error(partID/conds), data=output)
  tests <- (summary(test))
  p1[i] <- tests$'Error: partID:conds'[[1]]$'Pr(>F)'[[1]]
  p2[i] <- t.test(data[,1], data[,2], paired=TRUE)$p.value
  p3[i] <- t.test(data[,1], data[,3], paired=TRUE)$p.value
  p4[i] <- t.test(data[,2], data[,3], paired=TRUE)$p.value
}
#close(pb)#close progress bar
#printing all unique tests (adjusted code by DL)
sum(p1<alpha1)/nSim
```

```
## [1] 0.98
```

```
sum(p2<alpha2)/nSim
```

```
## [1] 0.97
```

```
sum(p3<alpha2)/nSim
```

```
## [1] 0.95
```

```
sum(p4<alpha2)/nSim
```

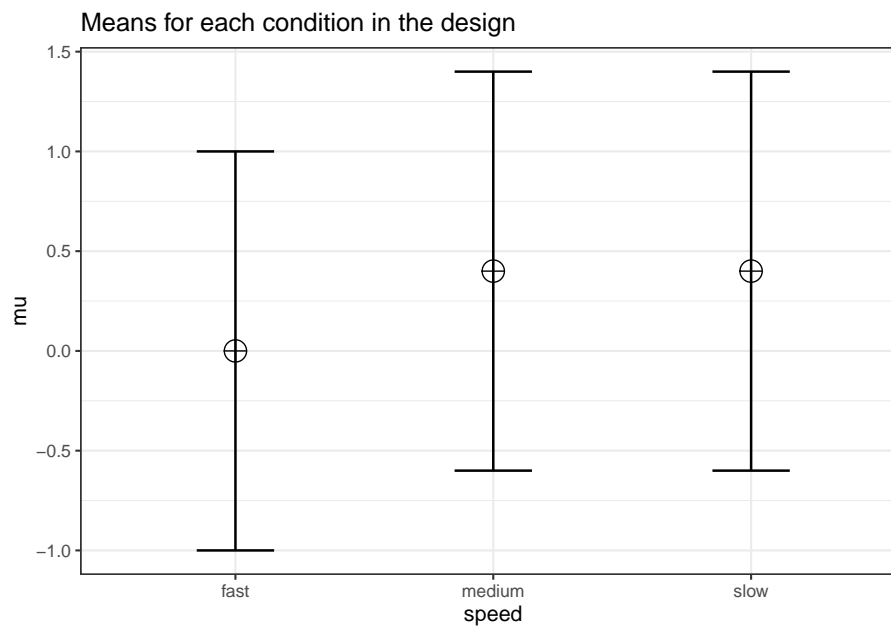
```
## [1] 0.05
```

```
design <- "3w"  
n <- 75  
mu <- c(0, 0.4, 0.4)  
sd <- 1  
r <- 0.6  
labelnames <- c("speed", "fast", "medium", "slow")
```

We create the within design, and run the simulation

```
design_result <- ANOVA_design(design = design,  
                             n = n,  
                             mu = mu,  
                             sd = sd,  
                             r = r,  
                             labelnames = labelnames)
```

6.2. REPRODUCING BRYLSBAERT VARIATION 1 CHANGING CORRELATION61



```
ANOVA_power(design_result, nsims = nsims)
```

```
## Power and Effect sizes for ANOVA tests
##           power effect_size
## anova_speed    99    0.1307
##
## Power and Effect sizes for contrasts
##           power effect_size
## p_speed_fast_speed_medium    99    0.46715
## p_speed_fast_speed_slow      95    0.44372
## p_speed_medium_speed_slow     3   -0.01717
##
## Within-Subject Factors Included: Check MANOVA Results
```

Again, this is similar to `g*power` for the ANOVA:

