

Appendix 1: R Functions

Manuscript: Simulation-Based Power-Analysis for Factorial ANOVA Designs

Daniël Lakens & Aaron R. Caldwell

The goal of the R package ANOVApower is to easily simulate ANOVA designs and calculate the observed power based on a specified pattern of means, standard deviations and correlations. In the main manuscript we reported several simulations, and within this appendix we have reproduced these simulations using the functions in the ANOVApower R package.

Installation

First, install the released version of ANOVApower from GitHub with:

```
devtools::install_github("Lakens/ANOVApower")
```

and load the package.

```
library(ANOVApower)
```

ANOVA__design function

The ANOVA__design function can create designs up three factors, for both within, between, and mixed designs. It requires the following input: design, n, mu, sd, r, and labelnames.

1. design: design that specifies the design (see below).
2. n: the sample size for each between subject condition.
3. mu: a vector with the means for each condition.
4. sd: the population standard deviation. Assumes homogeneity of variances (only one standard deviation can be provided).
5. r: the correlation for within designs (or 0 for between designs).
6. labelnames: This is a n optional vector of words that indicate factor names and level names (see below).
7. A final optional setting is to specify whether you want to output a plot or not (plot = TRUE or FALSE)

ANOVA__power function

The ANOVA__power function is used to perform the simulation. A design must be specified from the ANOVA__design function. It requires the following input: design_result, alpha_level, p_adjust, nsims, seed, and verbose.

1. design_result: output from the ANOVA__design function
2. alpha_level: the alpha level used to determine statistical significance
3. p_adjust: correction for multiple comparisons; set to none is none required or desired
4. nsims: number of simulations to perform
5. seed: set for reproducible results
6. verbose: Set to FALSE to not print results (default = TRUE)

Settings from the manuscript

In order to reproduce the simulations from the manuscript we need to set the number of repetitions in each simulation and the seed number. Although normally simulations will yield slightly different outcomes each

time they are performed, a ‘seed’ in computer simulations initializes the pseudorandom number generator, and setting the seed in replications makes them reproducible.

```
nsims <- 100000  
manuscript_seed = 2019
```

Simple 2 group between-subjects design

The initial study from the manuscript described a study wherein participants interact with an artificial voice assistant who sounds either cheerful or sad, and enjoyment is measured on scale (-5 to +5). In the ANOVA_design function this a two-level between-participant design.

```
design = "2b"
```

In this study, expected mean in the cheerful condition is 1 and in the sad condition is 1 with a standard deviation of 2. We specify the means as a vector (i.e., $c(x,y)$). The standard deviation is assumed to be the same for all conditions (the homogeneity of variances assumption).

```
mu = c(1,0)
```

```
sd = 2
```

We also assumed we will recruit 80 participants in each group.

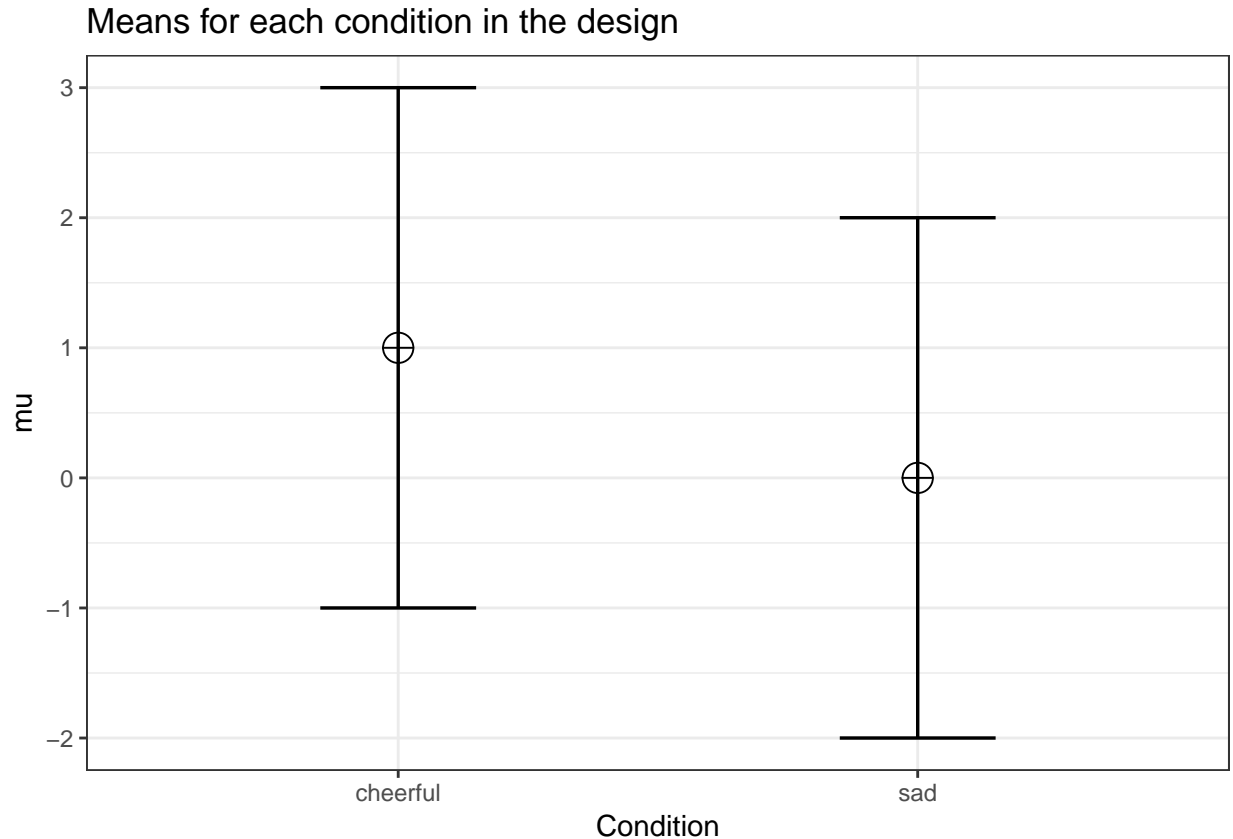
```
n = 80
```

For ease of interpretation the factors and levels are named in our example, factor = “Condition” with levels = “cheerful” and “sad”. It is also possible to not specify the labelnames, in which case they are automatically generated (and would be $c("a", "a1", "a2")$).

```
labelnames = c("Condition", "cheerful", "sad")
```

We now can put this all together to into the ANOVA_design function.

```
design_result <- ANOVA_design(design = "2b",  
                             n = 80,  
                             mu = c(1, 0),  
                             sd = 2,  
                             labelnames = c("Condition", "cheerful", "sad"))
```



Note that we did not define a correlation between the variables. The default is $r = 0$, and since this is a between subjects design, variables should have a correlation of 0. When the default value is fine, you do not need to specify it (but you still can). With the design defined, and stored in the `design_result` object, we can run the simulation with the significance level at 0.05 (`alpha_level = 0.05`) and we will run the previously defined number of simulations (`nsims = nsims`) and set the seed number (`set.seed(manuscript_seed)`).

```
set.seed(manuscript_seed)
power_result <- ANOVA_power(design_result,
                             alpha_level = 0.05,
                             nsims = nsims,
                             verbose = TRUE)

## Power and Effect sizes for ANOVA tests
##           power effect_size
## anova_Condition    90      0.0807
##
## Power and Effect sizes for contrasts
##           power effect_size
## p_Condition_cheerful_Condition_sad    90      -0.5663
```

Note that we did not specify any adjustment for multiple comparisons (`p_adjust`) because there are only two groups to compare, and the default ('none') is what we want. The results of a simulation study will vary each time the simulation is performed, but, in this case the results are reproducible by specifying a `seed`. When 100000 simulations are performed with a seed set to 2019, we see the statistical power (**Power and Effect sizes for ANOVA tests**) is 90% and the average partial η^2 (eta-squared) is 0.0807. In this scenario, the ANOVA results are exactly the same as the power from a simple t-test (**Power and Effect sizes for contrasts**), but the effect size, Cohen's d, is -0.5663.

Extending to 3 conditions

In the next example, we explored what would happen if we extended the design to 3 between-participant conditions.

```
design = "3b"
```

This was accomplished by adding “neutral” condition.

```
labelnames = c("condition", "cheerful", "neutral", "sad")
```

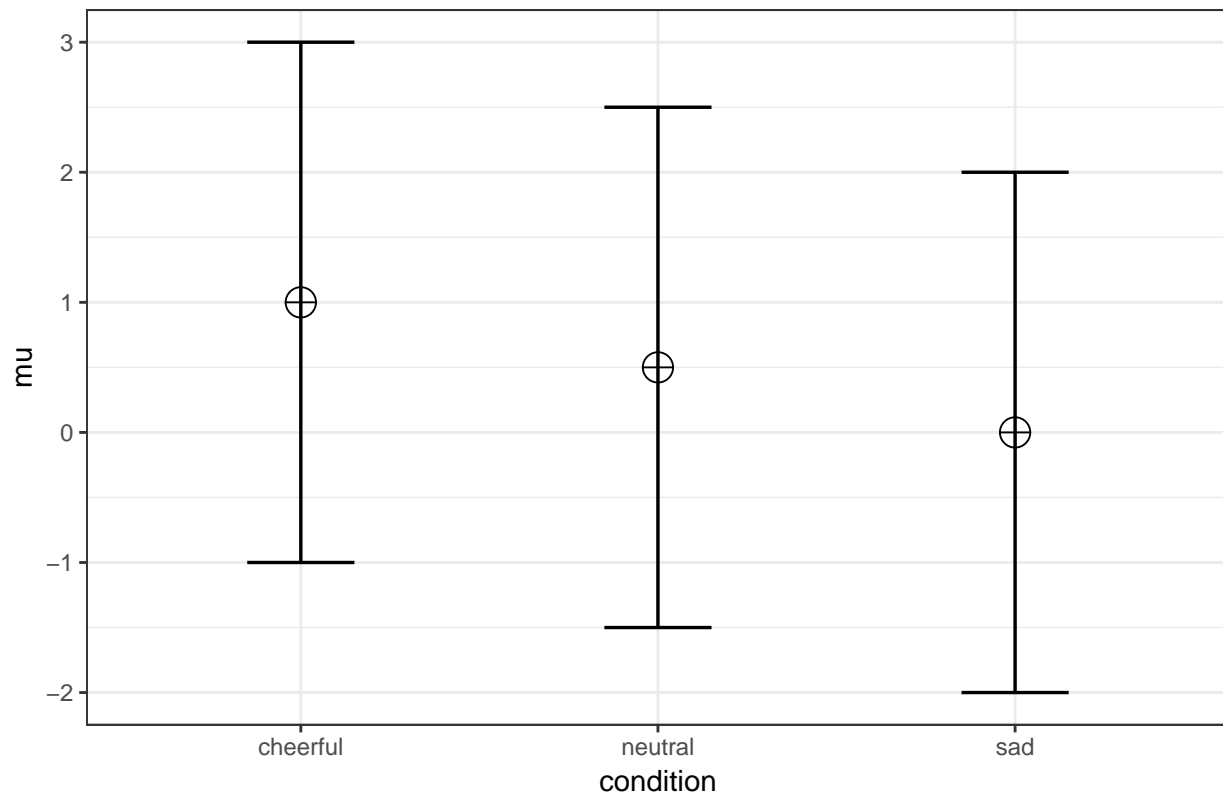
In one scenario, the expected means were 1 (cheerful), 0.5 (neutral), and 0 (sad).

```
mu = c(1, 0.5, 0)
```

Make sure the labels and mean correspond when specifying the design. This full design can be implemented as the following in the `ANOVA_design` function.

```
design_result_1 <- ANOVA_design(design = "3b",  
                               n = 80,  
                               mu = c(1, 0.5, 0),  
                               sd = 2,  
                               labelnames = c("condition", "cheerful", "neutral", "sad"))
```

Means for each condition in the design



Now we can simulate this design with the `ANOVA_power` function.

```
set.seed(manuscript_seed)  
power_result_1 <- ANOVA_power(design_result_1,  
                              alpha_level = 0.05,
```

```
nsims = nsims,  
verbose = TRUE)
```

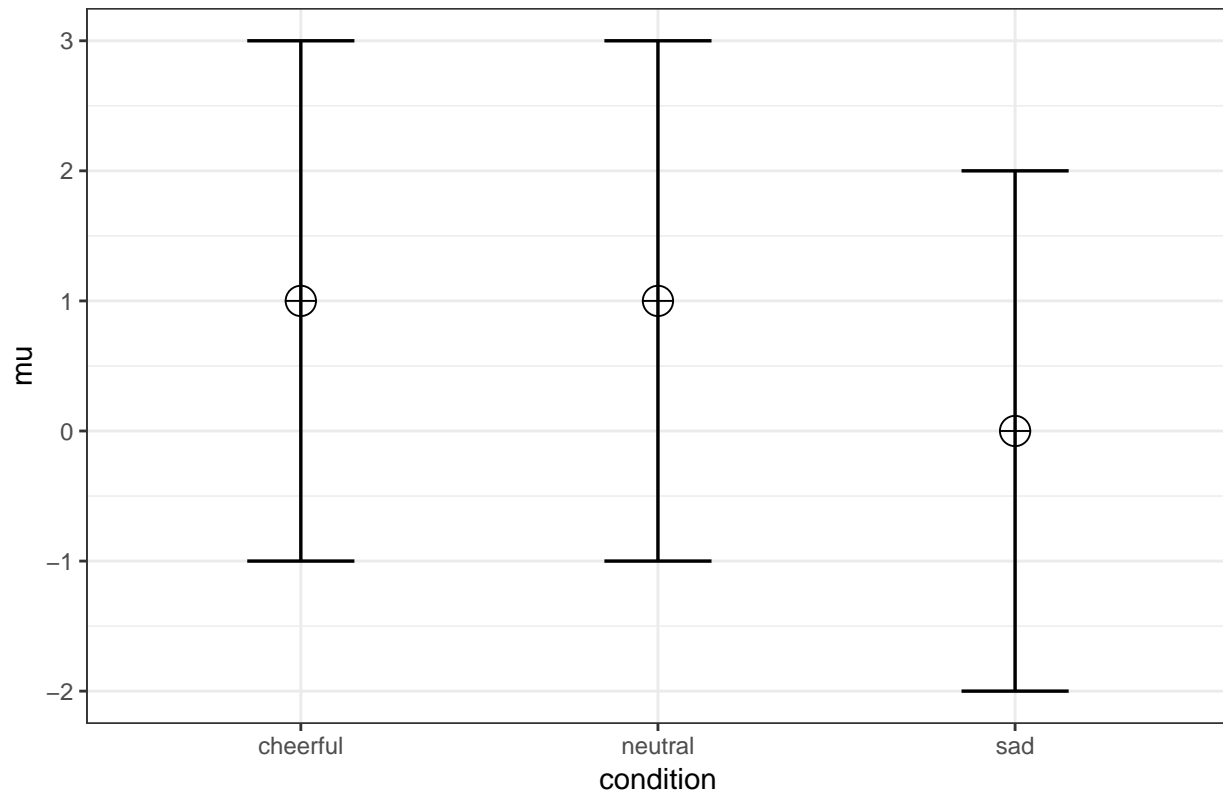
```
## Power and Effect sizes for ANOVA tests  
##           power effect_size  
## anova_condition    70      0.0328  
##  
## Power and Effect sizes for contrasts  
##                               power effect_size  
## p_condition_cheerful_condition_neutral    40    -0.2757  
## p_condition_cheerful_condition_sad        80    -0.4754  
## p_condition_neutral_condition_sad        30    -0.2090
```

In the second scenario, we assume that the enjoyment is the same (mean = 1) in the cheerful and neutral condition.

```
mu = c(1, 1, 0)
```

```
design_result_2 <- ANOVA_design(design = "3b",  
                               n = 80,  
                               mu = c(1, 1, 0),  
                               sd = 2,  
                               labelnames = c("condition", "cheerful", "neutral", "sad"))
```

Means for each condition in the design



Again, we perform simulations based on this new design.

```
set.seed(manuscript_seed)  
power_result_2 <- ANOVA_power(design_result_2,
```

```
alpha_level = 0.05,
nsims = nsims,
verbose = TRUE)
```

```
## Power and Effect sizes for ANOVA tests
##           power effect_size
## anova_condition    90    0.0513
##
## Power and Effect sizes for contrasts
##           power effect_size
## p_condition_cheerful_condition_neutral    10    -0.0205
## p_condition_cheerful_condition_sad        80    -0.4754
## p_condition_neutral_condition_sad        80    -0.4598
```

As we can see power for the ANOVA in scenario 1 was 70% compared to 90% in scenario 2. Further, the effect size, partial η^2 , increased from 0.0328 to 0.0513. As we can see, changing the design from 2b to 3b has an effect on power dependent upon the expected pattern of the means.

Changing to a within-subjects design

Now we modify the design by changing it to a within-subjects (i.e., repeated measures) design. Therefore, the first modification we must make in the `ANOVA_design` function is to the design definition. The number indicated the number of levels for each factor (in this case, a single factor with 3 levels) and the letter specifies if the factor is manipulated within or between participants (in this case within).

```
design = "3w"
```

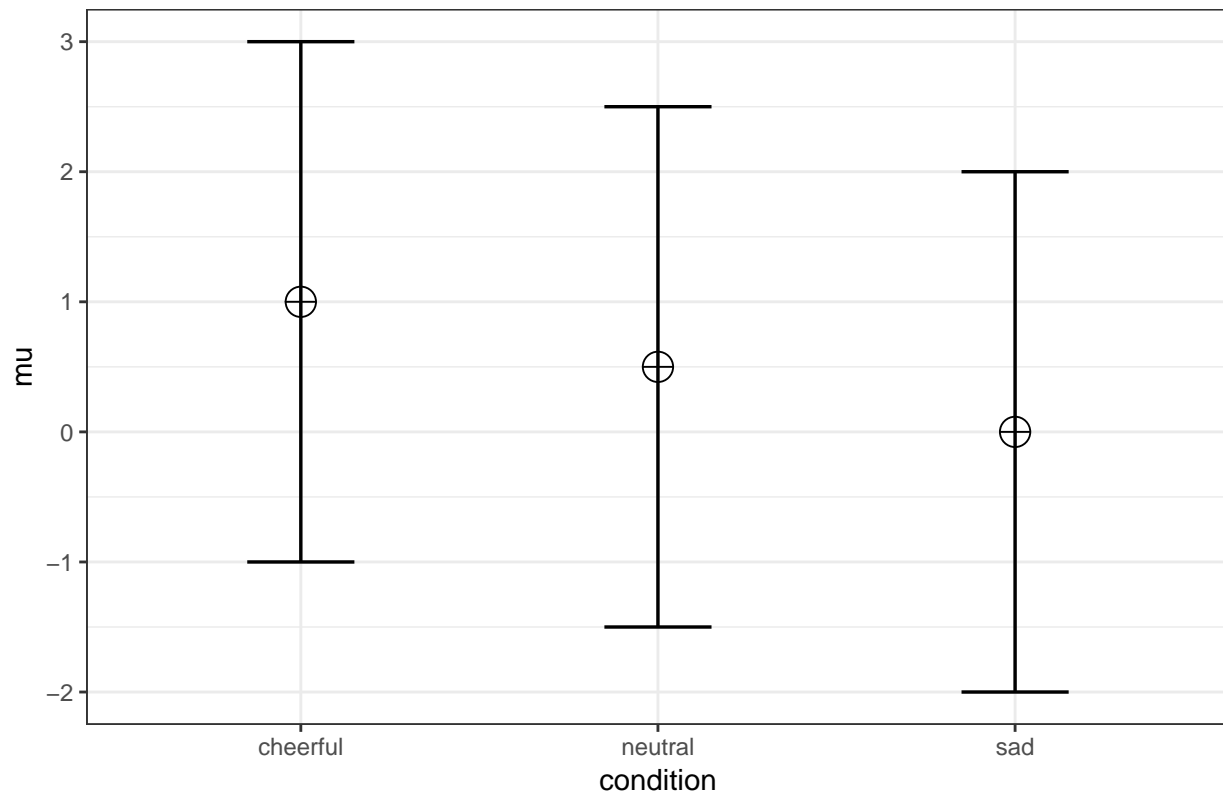
We must also specify the correlation between dependent measures. Here, we assume the correlations between all three pairs of variables is 0.5 (it is also possible to specify different correlations for each pair by entering a vector of correlations).

```
r = 0.5
```

Now we have the information to define the ANOVA design.

```
design_result_within_1 <- ANOVA_design(design = "3w",
                                     n = 80,
                                     mu = c(1, 0.5, 0),
                                     sd = 2,
                                     r = 0.5,
                                     labelnames = c("condition", "cheerful",
                                                    "neutral", "sad"))
```

Means for each condition in the design



Now, visually speaking, nothing about the pattern of means seems to have changed. However, we can see by looking at the correlation matrix from the 3b design that the correlation was 0 between all measurements:

```
design_result_1$cor_mat
```

```
##           cheerful neutral sad
## cheerful         1      0  0
## neutral          0      1  0
## sad              0      0  1
```

whereas we see when we look at the correlation matrix for the 3w design that the variables *are* correlated.

```
design_result_within_1$cor_mat
```

```
##           cheerful neutral sad
## cheerful         1.0     0.5 0.5
## neutral          0.5     1.0 0.5
## sad              0.5     0.5 1.0
```

The within-subjects design has been specified; now we run the simulation.

```
set.seed(manuscript_seed)
power_result_within_1 <- ANOVA_power(design_result_within_1,
                                     alpha_level = 0.05,
                                     nsims = nsims,
                                     verbose = TRUE)
```

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

```
## anova_condition    100      0.1208
##
## Power and Effect sizes for contrasts
##
##               power effect_size
## p_condition_cheerful_condition_neutral    40    -0.2011
## p_condition_cheerful_condition_sad        100    -0.5003
## p_condition_neutral_condition_sad         90     -0.3138
```

Power has now increased to 100% from the 3b design (70%). The total number of participants is lower in the within-subjects design ($N = 240$ in the between design, $N = 80$ in the within design), but there are now three observations per participant ($80 \times 3 = 240$).

Power for Interactions

In addition to simple one-way effects, in the manuscript we demonstrate the power analysis for 2x2 between-subjects design. In this scenario there two factors “condition” and “voice”. Again, participants are exposed to a “sad” or “cheerful” condition. However, the voice is either human-like (“human”) or robotic (“robot”). Note how we first specify the factor name for the first factor, then the names for all levels of the factor, and then specify the second factor label, and the labels for each level.

The design now has multiple factors, and is best described as a ‘two between by two between’ design. This is entered as 2b*2b. Different factors are separated by a * (not by a x or any other symbol).

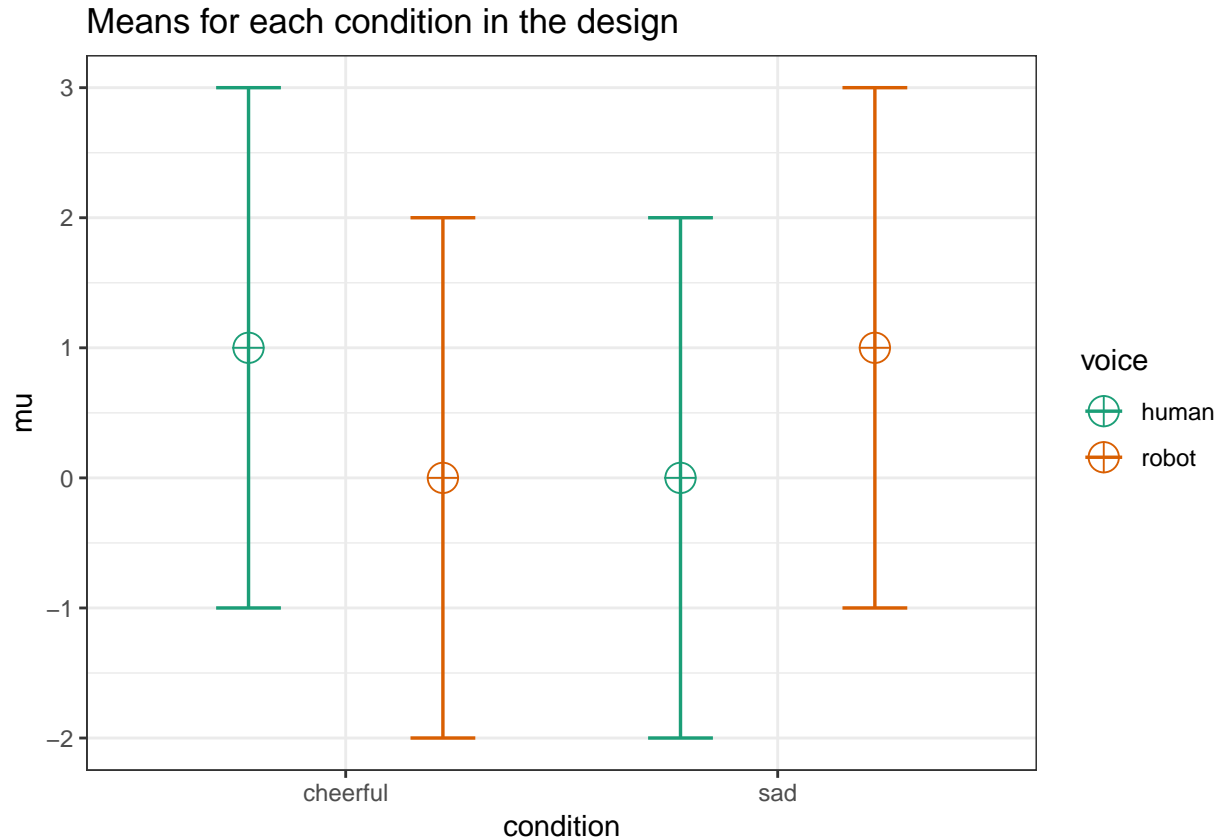
We must also define a new vector of means for the given design. Because we have a 2*2 design, we need to enter 4 means. In this case we expect a cross-over interaction wherein the mean response to cheerful-human will be the same as sad-robot (mean = 1) and cheerful-robot is the same as sad-human (mean = 0). Furthermore, the standard deviation in each condition is equal to 2.

```
mu = c(1, 0, 0, 1)
sd = 2
```

Now that the basics are laid out we can set up the design. We performed the simulation twice (to show the same power for a cross-over interaction as for the simple effect in the 2b design requires half as many participants in each cell), once with 40 participants per condition, once with 80 participants per condition.

```
design_result_cross_40 <- ANOVA_design(design = "2b*2b",
                                     n = 40,
                                     mu = c(1, 0, 0, 1),
                                     sd = 2,
                                     labelnames = c("condition", "cheerful", "sad",
                                                    "voice", "human", "robot"),
                                     plot = FALSE)

design_result_cross_80 <- ANOVA_design(design = "2b*2b",
                                     n = 80,
                                     mu = c(1, 0, 0, 1),
                                     sd = 2,
                                     labelnames = c("condition", "cheerful", "sad",
                                                    "voice", "human", "robot"))
```

Only one means plot is produced because the vector of means and standard deviation are the same, so we used the `plot = FALSE` argument to suppress one of the plots.

We then perform the simulation to see what the power is if we have a sample size of 40 in each condition.

```
set.seed(manuscript_seed)
power_result_cross_40 <- ANOVA_power(design_result_cross_40,
                                     alpha_level = 0.05,
                                     nsims = nsims,
                                     verbose = TRUE)
```

```
## Power and Effect sizes for ANOVA tests
##           power effect_size
## anova_condition      10    0.0017
## anova_voice           0    0.0071
## anova_condition:voice  90    0.0442
##
## Power and Effect sizes for contrasts
##
##                                     power
## p_condition_cheerful_voice_human_condition_cheerful_voice_robot  60
## p_condition_cheerful_voice_human_condition_sad_voice_human        50
## p_condition_cheerful_voice_human_condition_sad_voice_robot         0
## p_condition_cheerful_voice_robot_condition_sad_voice_human        10
## p_condition_cheerful_voice_robot_condition_sad_voice_robot        40
## p_condition_sad_voice_human_condition_sad_voice_robot             40
##                                     effect_size
## p_condition_cheerful_voice_human_condition_cheerful_voice_robot  -0.4799
```

```
## p_condition_cheerful_voice_human_condition_sad_voice_human      -0.4397
## p_condition_cheerful_voice_human_condition_sad_voice_robot      -0.0043
## p_condition_cheerful_voice_robot_condition_sad_voice_human      0.0497
## p_condition_cheerful_voice_robot_condition_sad_voice_robot      0.4623
## p_condition_sad_voice_human_condition_sad_voice_robot          0.4250
```

And once again for a sample size of 80 per condition.

```
set.seed(manuscript_seed)
power_result_cross_80 <- ANOVA_power(design_result_cross_80,
                                     alpha_level = 0.05,
                                     nsims = nsims,
                                     verbose = TRUE)
```

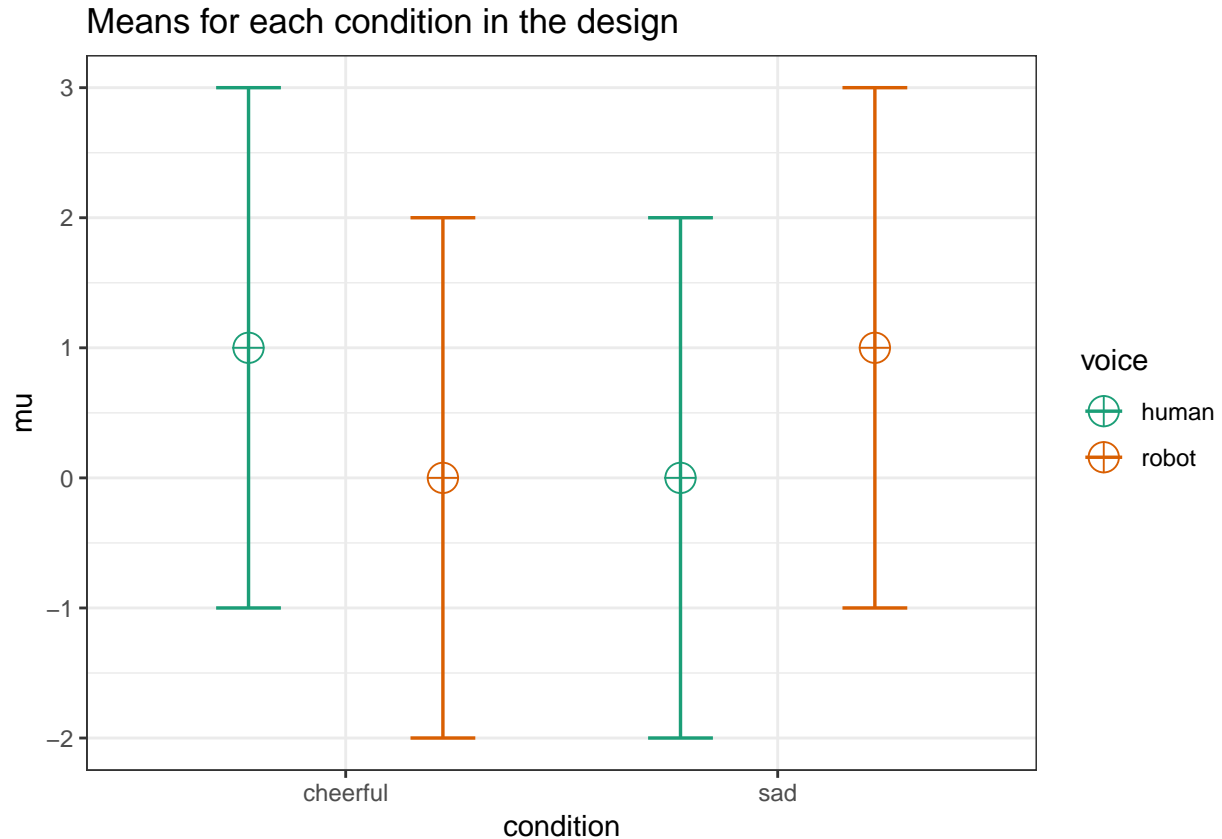
```
## Power and Effect sizes for ANOVA tests
##           power effect_size
## anova_condition      0      0.0017
## anova_voice          10      0.0029
## anova_condition:voice 100      0.0519
##
## Power and Effect sizes for contrasts
##                                     power
## p_condition_cheerful_voice_human_condition_cheerful_voice_robot    70
## p_condition_cheerful_voice_human_condition_sad_voice_human          80
## p_condition_cheerful_voice_human_condition_sad_voice_robot          0
## p_condition_cheerful_voice_robot_condition_sad_voice_human          10
## p_condition_cheerful_voice_robot_condition_sad_voice_robot          100
## p_condition_sad_voice_human_condition_sad_voice_robot              90
##                                     effect_size
## p_condition_cheerful_voice_human_condition_cheerful_voice_robot    -0.4565
## p_condition_cheerful_voice_human_condition_sad_voice_human          -0.4685
## p_condition_cheerful_voice_human_condition_sad_voice_robot          0.0550
## p_condition_cheerful_voice_robot_condition_sad_voice_human          -0.0033
## p_condition_cheerful_voice_robot_condition_sad_voice_robot          0.5110
## p_condition_sad_voice_human_condition_sad_voice_robot              0.5235
```

The simulations show we have 100% power when 80 participants are collected per condition ($N = 320$). If the sample size per cell is reduced from 80 to 40, power to detect the interaction is 90%.

Adjusting for Multiple Comparisons

As mentioned in the manuscript, the number of pairwise comparisons, if left unadjusted, can increase the Type I error rate. So in the manuscript, we revisited the 40 person per group study from the cross-over interaction example. The design below is the exactly same as `design_result_cross_40`.

```
design_result_holm_correction <- ANOVA_design(design = "2b*2b",
                                              n = 40,
                                              mu = c(1, 0, 0, 1),
                                              sd = 2,
                                              labelnames = c("condition",
                                                             "cheerful", "sad",
                                                             "voice",
                                                             "human", "robot"))
```



Now, when we run the simulation we can adjust for multiple comparisons using the `p_adjust` argument. In this case we use the Holm-Bonferroni correction. This feature is derived from `stats` package (base R), and number of different adjustments for multiple comparisons can be applied in the simulation (see `?p.adjust` for options).

```
set.seed(manuscript_seed)
power_result_holm_correction <- ANOVA_power(design_result_holm_correction,
                                             alpha_level = 0.05,
                                             p_adjust = "holm",
                                             nsims = nsims,
                                             verbose = TRUE)
```

```
## Power and Effect sizes for ANOVA tests
##               power effect_size
## anova_condition      10      0.0017
## anova_voice           0      0.0071
## anova_condition:voice  70      0.0442
##
## Power and Effect sizes for contrasts
##                                     power
## p_condition_cheerful_voice_human_condition_cheerful_voice_robot  40
## p_condition_cheerful_voice_human_condition_sad_voice_human        30
## p_condition_cheerful_voice_human_condition_sad_voice_robot         0
## p_condition_cheerful_voice_robot_condition_sad_voice_human        10
## p_condition_cheerful_voice_robot_condition_sad_voice_robot        30
## p_condition_sad_voice_human_condition_sad_voice_robot             40
##                                     effect_size
```

```
## p_condition_cheerful_voice_human_condition_cheerful_voice_robot    -0.4799
## p_condition_cheerful_voice_human_condition_sad_voice_human          -0.4397
## p_condition_cheerful_voice_human_condition_sad_voice_robot          -0.0043
## p_condition_cheerful_voice_robot_condition_sad_voice_human          0.0497
## p_condition_cheerful_voice_robot_condition_sad_voice_robot          0.4623
## p_condition_sad_voice_human_condition_sad_voice_robot              0.4250
```

As we can see from this simulation, power for the first pairwise comparison (cheerful-human vs cheerful-robot) decreases from 60% without any corrections for multiple comparisons, to 40% when the Holm-Bonferroni correction is applied.