A priori power analysis for ANOVA interaction effects with the anovapowersim R package: a short introduction

Shaheed Azaad
University of Muenster, Germany

Although power analysis has become increasingly accessible over time, researchers still find it challenging to compute power for complex ANOVA interactions. *anovapowersim* is an *R* package that enables users to easily specify between, within, and mixed interactions of varying complexity and simulate power based on a target effect size. This tutorial provides a short introduction to the package's core functionality, along with worked examples and sample *R* code.

Keywords: power, ANOVA, interactions, R

A critical early step in designing studies is determining how many participants to recruit. Typically, researchers want to know the sample size needed to achieve a desired level of *statistical power* (Cohen, 1988), i.e., the probability that a study will detect an effect of a specified size if that effect truly exists. Thanks to free and user-friendly software such as Faul et al.'s G*Power (2007), a priori power calculations for most designs are as simple as specifying a design, an effect size, the desired power, and an alpha level. However, calculating power for one common scenario - tests for more complex ANOVA interaction effects - is often more challenging (Lakens & Caldwell, 2021).

One solution is to use the *Superpower* package for *R* (Lakens & Caldwell, 2021), which provides simulation-based power. However, this package requires the user to enter condition means and standard deviations rather than a target effect size. While this approach may have its advantages (see Lakens & Caldwell, 2021 for a discussion), some researchers might still find it more intuitive to calculate power based on effect sizes.

An introduction to *anovapowersim*

The goal of *anovapowersim* is to simplify a priori power analysis for ANOVA interaction effects by allowing users to easily specify complex designs and to estimate power based on a target effect size in η_p^2. The package provides both simulation-based power estimates and analytical power calculations, allowing users to choose their preferred approach, although these should generally yield the same results (but see Lakens & Caldwell, 2021 for a discussion of the merits of simulation-based power).

anovapowersim can be installed from CRAN with the following command:

```
install.packages("anovapowersim")
```

And loaded as follows:

```
library("anovapowersim")
```

The simplest way to obtain a priori power is to use the `power_n()` function. Listing 1 contains an example of how we can obtain the required sample size for the main effect of *condition* in a fully between-subjects 2 (group) × 4 (condition) design. `between = c(group = 2, condition = 4)` indicates that our two factors are between-subjects and how many levels each has. `term = "condition"` indicates that we are interested in power for the main effect of condition, while `target_pes = .05` is the target effect size in η_p^2. Here, we have set `power = .90`, and `alpha = .05`. We need to run a substantial number of simulations to obtain accurate power estimates - here I set `n_sims = 5000` for illustrative purposes, but 10,000 or more is recommended. Setting `parallel = TRUE` speeds up the process by using `n-1` available cores by default, and setting `seed` makes the simulation reproducible.

The output in Listing 1 indicates that we need 288 participants, with 36 in each between-subjects (2 × 4) cell. If we were instead estimating power for a within-subjects ANOVA, with a three-level factor `time`, we would replace `between = c(group = 2, condition = 4)` with `within = c(time = 3)` and `term = "condition"` with `term = "time"`.

The corresponding within-subjects analysis is shown in Listing 2 This time, the total n needed (124) and the n per

Shaheed Azaad

The author has no conflicts of interest to declare. The author has no specific funding to declare.

Correspondence concerning this article should be addressed to Shaheed Azaad, Email: sazaad@uni-muenster.de

Listing 1: A priori power analysis for a between-subjects main effect.

```
power_n(
  between = c(group = 2, condition = 4), # group has 2 levels, condition has 4
  term = "condition", # we are interested in the main effect of condition
  target_pes = .05, # target effect size
  power = .90,
  alpha = .05,
  n_sims = 5000, # increase to 10000+ for more precise estimates
  parallel = TRUE, # simulations will be run in parallel for speed
  seed = 123 # for reproducibility
)
```

```
<anovapowersim_curve>
  term:          'condition'
  target power:  0.900
  alpha:         0.05
  effect size:   pes = 0.05
  n values:      6 per-cell sample sizes visited
  sims per cell size: 5000
  simulated test: uncorrected (auto)
  means pattern: default linear/Kronecker
  SS type:       III
  n needed for between-subjects cell: 36
  total N needed: 288

 n_per_cell total_n n_sims num_df den_df    ncp power_calc power_sim
         28     224   5000      3    216 11.368      0.810     0.811
         35     280   5000      3    272 14.316      0.899     0.899
         36     288   5000      3    280 14.737      0.908     0.904
         38     304   5000      3    296 15.579      0.924     0.923
         43     344   5000      3    336 17.684      0.954     0.950
         56     448   5000      3    440 23.158      0.988     0.988
```

between-subjects cell are the same, because there are no between-subjects factors. Of course, we could also get power for mixed designs by including both within- and between-subjects factors as in Listing 3.

Interactions

We can easily obtain power for interaction effects by entering the term of interest. Interactions can be specified by separating factors with a colon `:`. For example, if we are interested in the interaction between *group* and *time*, we set `term = "group:time"`. Listing 4 shows the code and results for this interaction. Obtaining the required sample size for the three-way interaction is as simple as changing `term = "group:time"` to `term = "group:time:condition"`.

A note on ordinal versus crossover interactions

anovapowersim does not differentiate between ordinal and crossover interactions (see Figure Figure 1). This is because two interactions with the same effect size require the same power, regardless of shape. To illustrate, below I simulate data for a 2 × 2 between-subjects design, in which case the interaction is the difference between two simple differences. The data simulation code is omitted here for brevity. The standard deviations for each cell are 1. In the crossover interaction, the effect of factor B flips between levels of A, from 0.3 to -0.3. In the ordinal interaction, the effect of B is amplified across levels of A, from 0.2 to 0.8. The cell summaries are shown in Listing 5 and Listing 6.

Importantly, both interactions are of the same magnitude. That is, the difference in B's effect across levels of A is 0.6 in both cases. As a result, both yield the same intereaction F statistic and η_p^2 (see Listing 7).

Listing 2: A priori power analysis for a within-subjects main effect.

```r
power_n(
  within = c(time = 3),
  term = "time",
  target_pes = .05,
  power = .90,
  alpha = .05,
  n_sims = 5000,
  parallel = TRUE,
  seed = 123
)
```

```
<anovapowersim_curve>
  term:          'time'
  target power:  0.900
  alpha:         0.05
  effect size:   pes = 0.05
  n values:      7 per-cell sample sizes visited
  sims per cell size: 5000
  simulated test: uncorrected (auto)
  means pattern: default linear/Kronecker
  SS type:       III
  n needed for between-subjects cell: 124
  total N needed: 124

 n_per_cell total_n n_sims num_df den_df    ncp power_calc power_sim
         98      98   5000      2    194 10.211      0.818     0.816
        123     123   5000      2    244 12.842      0.901     0.895
        124     124   5000      2    246 12.947      0.903     0.905
        125     125   5000      2    248 13.053      0.906     0.902
        129     129   5000      2    256 13.474      0.915     0.914
        146     146   5000      2    290 15.263      0.945     0.948
        196     196   5000      2    390 20.526      0.987     0.988
```

Listing 3: A priori power analysis for a mixed-design main effect.

```r
power_n(
  between = c(group = 2, condition = 4),
  within = c(time = 3),
  term = "time",
  target_pes = .05,
  power = .90,
  alpha = .05,
  n_sims = 5000,
  parallel = TRUE,
  seed = 123
)
```

Listing 4: A priori power analysis for a mixed-design interaction.

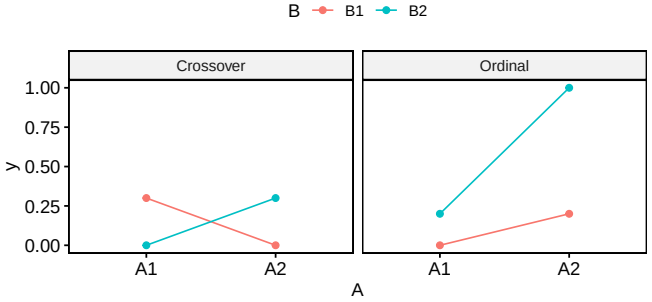
```
power_n(
  between = c(group = 2, condition = 4),
  within = c(time = 3),
  term = "group:time",
  target_pes = .05,
  power = .90,
  alpha = .05,
  n_sims = 5000,
  parallel = TRUE,
  seed = 123
)
```

```
<anovapowersim_curve>
  term:          'group:time'
  target power:  0.900
  alpha:         0.05
  effect size:   pes = 0.05
  n values:      6 per-cell sample sizes visited
  sims per cell size: 5000
  simulated test: uncorrected (auto)
  means pattern: default linear/Kronecker
  SS type:       III
  n needed for between-subjects cell: 17
  total N needed: 136

 n_per_cell total_n n_sims num_df den_df    ncp power_calc power_sim
         13     104   5000      2    192 10.105      0.813     0.813
         16     128   5000      2    240 12.632      0.896     0.894
         17     136   5000      2    256 13.474      0.915     0.918
         18     144   5000      2    272 14.316      0.931     0.933
         20     160   5000      2    304 16.000      0.955     0.953
         26     208   5000      2    400 21.053      0.988     0.986
```

Figure 1

Ordinal and crossover interaction patterns with the same difference between differences.



While it is true that crossover interactions are likely larger in *practice*, this difference is reflected in effect sizes. Therefore, when a smaller, ordinal interaction is expected, the solution is to enter a smaller effect size when estimating power.

Specifying an effect size

Cohen (1988) suggested that researchers conduct power analyses based on field-specific effect distributions. Research on effect size distributions in psychology has indeed found substantial variability between fields (Azaad, 2026; Gignac & Szodorai, 2016; Lovakov & Agadullina, 2021). For this reason, it is best to specify an effect size based on related research rather than general small/medium/large thresholds.

It is important to note that η_p^2 from between-subjects and within-subjects designs are computed differently, and therefore not directly comparable (Lakens & Caldwell, 2021). The

Listing 5: Cell means and standard deviations for the ordinal interaction.

```
ordinal_data %>%
  group_by(A, B) %>%
  summarise(
    mean_y = round(mean(y), 2),
    sd_y = round(sd(y), 2),
    .groups = "drop"
  )
```

```
# A tibble: 4 x 4
  A     B     mean_y  sd_y
  <chr> <chr>  <dbl> <dbl>
1 A1    B1       0       1
2 A1    B2       0.2     1
3 A2    B1       0.2     1
4 A2    B2       1       1
```

Listing 6: Cell means and standard deviations for the crossover interaction.

```
crossover_data %>%
  group_by(A, B) %>%
  summarise(
    mean_y = round(mean(y), 2),
    sd_y = round(sd(y), 2),
    .groups = "drop"
  )
```

```
# A tibble: 4 x 4
  A     B     mean_y  sd_y
  <chr> <chr>  <dbl> <dbl>
1 A1    B1       0.3     1
2 A1    B2       0       1
3 A2    B1       0       1
4 A2    B2       0.3     1
```

same pattern of means and standard deviations can produce considerably larger effects in within-subjects designs when observations are positively correlated across conditions. Using a η_p^2 from a within-subjects design for a between-subjects power calculation therefore risks underpowering one's study. Because η_p^2 increases with the repeated-measures correlation, researchers should also consider whether they expect their design to produce smaller (or larger) correlations than the studies used as a reference for the expected effect size.

Additional options and features

Although *anovapowersim* is primarily for a priori power calculations, it also offers additional features that researchers may find useful. I describe some of these below, but a thorough overview (and future updates to the feature set) can be found in the package's documentation: https://shaheedazaad.github.io/anovapowersim/

Sensitivity and achieved power analyses

Users can also obtain estimates of sensitivity (the minimum detectable effect size given power and alpha; Listing 8) and achieved power (given an effect size and sample size; Listing 9) using the `power_sensitivity()` and `power_achieved()` functions, respectively.

Listing 7: ANOVA results for ordinal and crossover interactions.

```
library(afex)

aov_car(
  y ~ A * B + Error(1 / id),
  data = ordinal_data,
  anova_table = list(correction = "none", es = "pes")
)
```

```
Anova Table (Type 3 tests)

Response: y
  Effect     df  MSE         F  pes p.value
1      A 1, 196 1.00 12.50 *** .060   <.001
2      B 1, 196 1.00 12.50 *** .060   <.001
3    A:B 1, 196 1.00    4.50 * .022    .035
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '+' 0.1 ' ' 1
```

```
aov_car(
  y ~ A * B + Error(1 / id),
  data = crossover_data,
  anova_table = list(correction = "none", es = "pes")
)
```

```
Anova Table (Type 3 tests)

Response: y
  Effect     df  MSE      F   pes p.value
1      A 1, 196 1.00   0.00 <.001   >.999
2      B 1, 196 1.00   0.00 <.001   >.999
3    A:B 1, 196 1.00 4.50 *  .022    .035
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '+' 0.1 ' ' 1
```

Calculations without simulation

For those interested only in calculated power, the `power_n_calc()` function runs the same calculation as `power_n()`, but without the simulations (Listing 10). In this example, we also use the `f_to_pes()` helper to calculate based on Cohen's *f*. Nonsphericity can also be accounted for by setting `epsilon` to a value between 0 and 1. Analogous calculation-only functions for sensitivity (`power_sensitivity_calc()`) and achieved power (`power_achieved_calc()`) are also available.

These calculation functions are also available through a graphical *R* Shiny interface hosted at https://shaheedazaad.github.io/anovapowersimshiny/.

Power curves

anovapowersim also has the option to plot power curves using the `power_curve()` function (Listing 11). This is useful if one wants to obtain power at specific sample sizes. Here, I have set `n_range = c(16, 20, 23, 28)` to estimate power at specific sample sizes.

The results of a power curve can also be plotted using `plot_power_curve()` (Listing 12).

Comparison with G*Power

anovapowersim and G*Power (Faul et al., 2007) translate the requested η_p^2 into power using slightly different approaches. Where G*Power uses the total sample size to compute the non-centrality parameter, *anovapowersim* uses

Listing 8: Sensitivity analysis.

```
power_sensitivity(
  within = c(time = 3),
  term = "time",
  n = 32, # per between-subjects cell
  power = .90,
  alpha = .05,
  n_sims = 5000,
  parallel = TRUE,
  seed = 123
)
```

```
<anovapowersim_sensitivity>
  term:             'time'
  fixed total N:    32
  target power:     0.900
  alpha:            0.05
  detectable pes:   0.176492
  search interval:  [1e-06, 0.99]
  requested width:  0.001
  simulated points: 10
  final width:      0.0006894198
  converged:        yes
  simulations/point: 5000
  simulated test:   uncorrected (auto)
  means pattern:    default linear/Kronecker
  SS type:          III

 target_pes n_per_cell total_n n_sims num_df den_df    ncp power_calc power_sim
      0.000         32      32   5000      2     62  0.000      0.050     0.048
      0.088         32      32   5000      2     62  6.001      0.562     0.567
      0.132         32      32   5000      2     62  9.459      0.771     0.767
      0.154         32      32   5000      2     62 11.323      0.845     0.845
      0.165         32      32   5000      2     62 12.293      0.875     0.876
      0.171         32      32   5000      2     62 12.787      0.888     0.897
      0.174         32      32   5000      2     62 13.036      0.894     0.893
      0.175         32      32   5000      2     62 13.162      0.897     0.893
      0.176         32      32   5000      2     62 13.225      0.899     0.898
      0.176         32      32   5000      2     62 13.288      0.900     0.900
```

degrees of freedom by default. These can produce different power estimates. Setting `gpower = TRUE` will produce power estimates comparable to those from G*Power. Sample sizes may nevertheless differ because *anovapowersim* searches over balanced per-cell sample sizes, whereas G*Power returns a sample size that is not necessarily divisible between cells. In the examples below, I have chosen effect sizes where G*Power returns a balanced sample size for comparison.

Listing 13 contains the G*Power calculation for a 2 × 2 between-subjects interaction with $\eta_p^2 = .0478$:

With `gpower=TRUE`, *anovapowersim* produces the same required n of 212 (Listing 14).

For within-subjects or mixed calculations, *anovapowersim* produces power estimates corresponding to G*Power's 'as in Cohen (1988)' option based on $\eta_p^2 = .06$ (Listing 15).

In this case, *anovapowersim* recommends a slightly smaller sample size because the simulations estimated power to be .90 at $n = 166$. But the `power_calc` column shows the same power of .902 at $n = 168$ as G*Power (Listing 16).

Listing 9: Achieved-power analysis.

```
power_achieved(
  within = c(time = 3),
  term = "time",
  n = 32, # per between-subjects cell
  target_pes = .176492,
  alpha = .05,
  n_sims = 5000,
  parallel = TRUE,
  seed = 123
)
```

```
<anovapowersim_achieved_power>
  term:             'time'
  fixed total N:    32
  target pes:       0.1765
  alpha:            0.05
  simulations:      5000
  simulated test:   uncorrected (auto)
  means pattern:    default linear/Kronecker
  achieved power:   0.900
  calculated power: 0.900
  SS type:          III

 n_per_cell total_n n_sims num_df den_df    ncp power_calc power_sim
         32      32   5000      2     62 13.288      0.900     0.900
```

Conclusion

A priori power analysis for ANOVA interaction effects can be challenging. *anovapowersim* simplifies this by enabling researchers to easily specify complex designs and estimate power intuitively based on effect sizes. However, *anovapowersim* has a few limitations. Specifically, it does not readily support unbalanced designs, nor does it provide power for post-hoc comparisons. *Superpower* (Lakens & Caldwell, 2021) offers these features, but it requires entering means and standard deviations rather than effect sizes.

References

Azaad, S. (2026). *Empirically derived effect size guidelines for social, individual differences, and cognitive psychology*. http://dx.doi.org/10.31234/osf.io/r4xwb_v1

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Academic Press.

Faul, F., Erdfelder, E., Lang, A.-G., & Buchner, A. (2007). G*power 3: A flexible statistical power analysis program for the social, behavioral, and biomedical sciences. *Behavior Research Methods*, *39*(2), 175–191. https://doi.org/10.3758/BF03193146

Gignac, G. E., & Szodorai, E. T. (2016). Effect size guidelines for individual differences researchers. *Personality and Individual Differences*, *102*, 74–78. https://doi.org/10.1016/j.paid.2016.06.069

Lakens, D., & Caldwell, A. R. (2021). Simulation-based power analysis for factorial analysis of variance designs. *Advances in Methods and Practices in Psychological Science*, *4*(1), 2515245920951503. https://doi.org/10.1177/2515245920951503

Lovakov, A., & Agadullina, E. R. (2021). Empirically derived guidelines for effect size interpretation in social psychology. *European Journal of Social Psychology*, *51*(3), 485–504. https://doi.org/10.1002/ejsp.2752

Listing 10: Calculation-only a priori power analysis.

```r
power_n_calc(
  within = c(condition = 4),
  term = "condition",
  target_pes = f_to_pes(0.25),
  power = .90,
  alpha = .05,
  epsilon = 1
)
```

```
<anovapowersim_curve>
  term:          'condition'
  target power:  0.900
  alpha:         0.05
  effect size:   pes = 0.0588
  n values:      13 per-cell sample sizes visited
  calculation:   calculated power only
  n needed for between-subjects cell: 78
  total N needed: 78

 n_per_cell total_n n_sims num_df den_df    ncp power_calc power_sim
          2       2     NA      3      3  0.188      0.054      <NA>
          4       4     NA      3      9  0.562      0.073      <NA>
          8       8     NA      3     21  1.312      0.123      <NA>
         16      16     NA      3     45  2.812      0.240      <NA>
         32      32     NA      3     93  5.812      0.486      <NA>
         64      64     NA      3    189 11.812      0.825      <NA>
         72      72     NA      3    213 13.312      0.873      <NA>
         76      76     NA      3    225 14.062      0.892      <NA>
         77      77     NA      3    228 14.250      0.896      <NA>
         78      78     NA      3    231 14.438      0.901      <NA>
         80      80     NA      3    237 14.812      0.909      <NA>
         96      96     NA      3    285 17.812      0.955      <NA>
        128     128     NA      3    381 23.812      0.990      <NA>
```

Listing 11: Estimating a power curve at specified sample sizes.

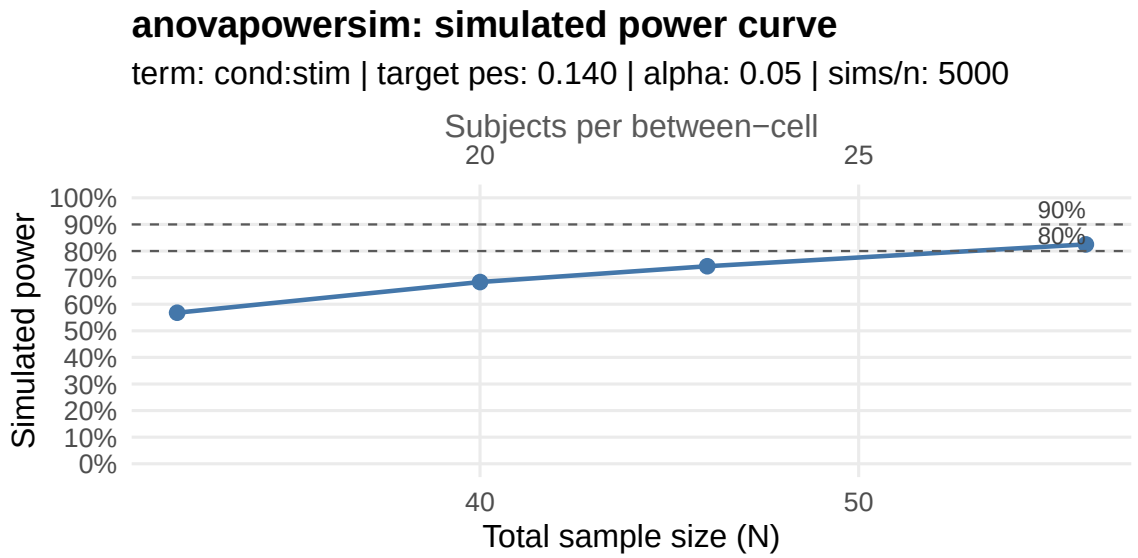
```
pc <- power_curve(
  between = c(cond = 2),
  within = c(stim = 2),
  term = "cond:stim",
  target_pes = 0.14,
  n_range = c(16, 20, 23, 28), # n per between-subject cell
  n_sims = 5000,
  seed = 123,
  parallel = TRUE
)
pc
```

```
<anovapowersim_curve>
  term:          'cond:stim'
  target power:  <not specified>
  alpha:         0.05
  effect size:   pes = 0.14
  n values:      4 per-cell sample sizes visited
  sims per cell size: 5000
  simulated test: uncorrected (auto)
  means pattern: default linear/Kronecker
  SS type:       III
  n needed for between-subjects cell: <not reached>
  total N needed: <not reached>

 n_per_cell total_n n_sims num_df den_df   ncp power_calc power_sim
         16      32   5000      1     30 4.884      0.571     0.568
         20      40   5000      1     38 6.186      0.679     0.683
         23      46   5000      1     44 7.163      0.745     0.743
         28      56   5000      1     54 8.791      0.829     0.825
```

Listing 12: Plotting the power curve.

```
plot_power_curve(
  pc,
  power_lines = c(.80, .90) # adds horizontal lines at 80% and 90% power
)
```



Listing 13: G*Power output for the between-subjects calculation.

```
F tests - ANOVA: Fixed effects, special, main effects and interactions

Analysis:   A priori: Compute required sample size
Input:      Effect size f                   =   0,2240525
            α err prob                      =   0,05
            Power (1-β err prob)            =   0,9
            Numerator df                    =   1
            Number of groups                =   4
Output:     Noncentrality parameter λ       =   10,6422988
            Critical F                      =   3,8865546
            Denominator df                  =   208
            Total sample size               =   212
            Actual power                    =   0,9009862
```

Listing 14: G*Power-compatible between-subjects analysis.

```
power_n(
  between = c(group = 2, cond = 2),
  term = "cond:group",
  target_pes = .0478,
  n_sims = 5000,
  power = .90,
  alpha = .05,
  gpower = TRUE,
  parallel = TRUE,
  seed = 123
)
```

```
<anovapowersim_curve>
  term:          'group:cond'
  target power:  0.900
  alpha:         0.05
  effect size:   pes = 0.0478
  n values:      8 per-cell sample sizes visited
  sims per cell size: 5000
  simulated test: uncorrected (auto)
  means pattern: default linear/Kronecker
  G*Power convention: TRUE
  SS type:       III
  n needed for between-subjects cell: 53
  total N needed: 212

 n_per_cell total_n n_sims num_df den_df    ncp power_calc power_sim
         42     168   5000      1    164  8.434      0.823     0.819
         52     208   5000      1    204 10.442      0.895     0.892
         53     212   5000      1    208 10.642      0.901     0.904
         54     216   5000      1    212 10.843      0.906     0.911
         55     220   5000      1    216 11.044      0.911     0.912
         56     224   5000      1    220 11.245      0.916     0.911
         63     252   5000      1    248 12.650      0.943     0.944
         84     336   5000      1    332 16.867      0.984     0.983
```

Listing 15: G*Power output for the within-subjects calculation.

```
F tests - ANOVA: Repeated measures, within factors

Analysis:   A priori: Compute required sample size
Input:      Effect size f(V)               =   0,2526456
            α err prob                     =   0,05
            Power (1-β err prob)           =   0,9
            Number of groups               =   2
            Number of measurements         =   2
            Nonsphericity correction ε     =   1
Output:     Noncentrality parameter λ      =   10,7234063
            Critical F                     =   3,8980886
            Numerator df                   =   1,0000000
            Denominator df                 =   166
            Total sample size              =   168
            Actual power                   =   0,9024596
```

Listing 16: G*Power-compatible mixed-design analysis.

```r
power_n(
  between = c(group = 2),
  within = c(cond = 2),
  term = "cond:group",
  target_pes = .06,
  n_sims = 5000,
  power = .90,
  alpha = .05,
  gpower = TRUE,
  parallel = TRUE,
  seed = 123
)
```

```
<anovapowersim_curve>
  term:          'group:cond'
  target power:  0.900
  alpha:         0.05
  effect size:   pes = 0.06
  n values:      8 per-cell sample sizes visited
  sims per cell size: 5000
  simulated test: uncorrected (auto)
  means pattern: default linear/Kronecker
  G*Power convention: TRUE
  SS type:       III
  n needed for between-subjects cell: 84
  total N needed: 168

 n_per_cell total_n n_sims num_df den_df    ncp power_calc power_sim
         67     134   5000      1    132  8.553      0.827     0.827
         82     164   5000      1    162 10.468      0.895     0.890
         83     166   5000      1    164 10.596      0.899     0.900
         84     168   5000      1    166 10.723      0.902     0.915
         87     174   5000      1    172 11.106      0.912     0.916
         88     176   5000      1    174 11.234      0.915     0.905
         99     198   5000      1    196 12.638      0.943     0.943
        134     268   5000      1    266 17.106      0.985     0.983
```