pwrcalc Documentation

Release 0.0.0.9000

Vikram Jambulapati

Oct 04, 2018

Contents

1 Introduction **3**
- 1.1 Why Power Analysis? . . . 3
- 1.2 Why Another Power Analysis Package for R? . . . 3

2 Installation **5**
- 2.1 ghit . . . 5
- 2.2 devtools . . . 5
- 2.3 Manual installation . . . 5

3 Reference **7**
- 3.1 twomeans . . . 7
- 3.2 clustered . . . 8

4 Practical Examples **9**
- 4.1 Two sample t-test . . . 9
- 4.2 Two sample t-test with group clusters . . . 10

5 Indices and tables **13**

Contents:

CHAPTER 1

Introduction

1.1 Why Power Analysis?

Power influences many design aspects, including what research questions to pursue, how many treatment arms to employ, and even more fundamentally, whether or not to proceed with a potential research project. For example, it may be that a remedial education program boosts tests scores by 20 percent when comparing treatment and control groups, but due to limited power, the RCT is unable to detect this true effect (with 95% confidence). However, we can estimate whether a given design is likely to be able to detect a reasonable effect size ex ante, allowing us to properly manage partner expectations and make the most of limited research resources.

1.2 Why Another Power Analysis Package for R?

There exists a fine number of options for power calculations in R. From the default power command to pwr to samplesize. However, economists love Stata. For users new to both R and power calculations, replicating the examples used by economists in power calculation lectures can be a bit tricky. This is the impetus for pwrcalc.

CHAPTER 2

Installation

Currently, *pwrcalc* has not been released on CRAN, so you must rely upon either an external package for installing *pwrcalc* or you can install the package manually.

2.1 ghit

ghit is a user-written package to install packages from GitHub, which is were *pwrcalc* lives. Run the following two-lines of code and you should be up and running with *pwrcalc*.

```
install.packages('ghit')
ghit::install_github('vikjam/pwrcalc')
```

2.2 devtools

Alternatively, *devtools* is another user-written package that allows you to install packages from GitHub. *devtools* mainly helps users create R packages, so you'll get a lot of other tools that come along with *devtools*.

```
install.packages('devtools')
devtools::install_github('vikjam/pwrcalc')
```

2.3 Manual installation

Finally, if neither of the previous installations options work for you. You can download the latest release from GitHub. Download the file with the extensions *.tar.gz*. And then use these instructions to install the downloaded file.

CHAPTER 3

Reference

3.1 twomeans

twomeans (*m1 = NULL, m2 = NULL, n1 = NULL, n2 = NULL, nratio = NULL, sd = NULL, sd1 = NULL, sd2 = NULL, sig.level = 0.05, power = 0.80*)

Two-sample t-test power calculation

param m1 Mean of a group 1 (e.g., the control-group)

param m2 Mean of a group 2 (e.g., the experimental-group)

param n1 Number of obs. in group 1 (e.g., the control-group)

param n2 Number of obs. in group 2 (e.g., the control-group)

param nratio Specify the ratio of group 1 to group 2.

param sd Standard deviation of each group, i.e., sd = sd1 = sd2

param sd1 Standard deviation of a group 1 (e.g., the control-group)

param sd2 Standard deviation of a group 2 (e.g., the experimental-group)

param sig.level significance level; default is sig.level = 0.05

param power one minus the probability of type II error, default is power = 0.8

rtype A power.htest object with results in a structured list

Example

```
> twomeans(m1 = 12, m2 = 16, sd = 5)

     Two-sample t-test power calculation

              m1 = 12
              m2 = 16
              n1 = 25
```

(continues on next page)

(continued from previous page)

```
             n2 = 25
      sig.level = 0.05
          power = 0.8
    alternative = two.sided

NOTE:
m1 and m2 are the means of group 1 and 2, respectively.
n1 and n2 are the obs. of group 1 and 2, respectively.
```

3.2 clustered

clustered(*unclustered*, *rho*, *obsclus = NULL*, *numclus = NULL)*)

Power calculations in an experiment with group clusters

params unclustered Results from twomeans not adjusting for clusters

params rho Specifies the intraclass correlation coefficient

params obsclus Number of observations in each cluster

params numclus Maximum number of clusters

rtype A power.htest object with results in a structured list

Example

```
> twomeans(m1 = 12, m2 = 16, sd = 5) %>% clustered(obsclus = 10, rho = 0.3)

     Two-sample t-test power calculation

                      m1 = 12
                      m2 = 16
          n1 (unadjusted) = 25
          n2 (unadjusted) = 25
                     rho = 0.3
       Average per cluster = 10
minimum number of clusters = 19
             n1 (adjusted) = 93
             n2 (adjusted) = 93
                 sig.level = 0.05
                     power = 0.8
               alternative = two.sided

NOTE: m1 and m2 are the means of group 1 and 2, respectively.
n1 and n2 are the obs. of group 1 and 2, respectively.
```

CHAPTER 4

Practical Examples

4.1 Two sample t-test

Load the included Balsakhi data set, which we'll use to estimate the control mean.

```
library(pwrcalc)
data(balsakhi)
control_data <- balsakhi[which(balsakhi$bal == 0), ]
control_mean <- mean(control_data$post_totnorm, na.rm = TRUE)
control_sd   <- sd(control_data$post_totnorm, na.rm = TRUE)
```

Let's inspect the results to make sure we're all on the same page.

```
> print(control_mean)
[1] 0.4288781
> print(control_sd)
[1] 1.15142
```

Let's say, based on other studies, that we expect an effect size of a tenth of a standard deviation. Now let's calculate the sample size for our anticipated effect size.

```
expected_effect <- control_sd / 10
treated_mean    <- control_mean + expected_effect
```

We can now calculate the sample size needed to test that hypothesis at the significance level of 0.05 and power of 0.8.

```
> twomeans(m1 = control_mean, m2 = treated_mean, sd = control_sd)

     Two-sample t-test power calculation

              m1 = 0.4288781
              m2 = 0.5440201
              n1 = 1570
              n2 = 1570
```

(continues on next page)

(continued from previous page)

```
      sig.level = 0.05
          power = 0.8
    alternative = two.sided

NOTE:
m1 and m2 are the means of group 1 and 2, respectively.
n1 and n2 are the obs. of group 1 and 2, respectively.
```

Now imagine we anticipate an effect half as large as the previous example. In particular, we now expect 1/20 of a standard deviation.

```
smaller_expected_effect <- control_sd / 20
smaller_treated_mean    <- control_mean + smaller_expected_effect
```

```
> twomeans(m1 = control_mean, m2 = smaller_treated_mean, sd = control_sd)

     Two-sample t-test power calculation

              m1 = 0.4288781
              m2 = 0.4864491
              n1 = 6280
              n2 = 6280
       sig.level = 0.05
           power = 0.8
    alternative = two.sided

NOTE:
m1 and m2 are the means of group 1 and 2, respectively.
n1 and n2 are the obs. of group 1 and 2, respectively.
```

Notice now we need four times as many observations as the previous example.

4.2 Two sample t-test with group clusters

Many designs randomize at the group level instead of at the individual level. For such designs, we need to adjust our power calculations so that they incorporate the fact that individuals within the same group may be subject to similar shocks, and thereby have correlated outcomes. Duflo et al. presents a modified parametric approach, which takes into account the intra-cluster correlation (ICC) that arises from randomization at the group level.

```
library(ICC)
icc_sample <- control_data[!is.na(divid) & !is.na(post_totnorm), ]
control_subset$divid = as.factor(control_subset$divid)
icc <- ICCest(divid, post_totnorm, data = control_subset)
rho <- icc$ICC
```

```
> twomeans(m1 = control_mean, m2 = treated_mean, sd = control_sd) %>%␣
↪clustered(obsclus = 10, rho = 0.3)

     Two-sample t-test power calculation

                      m1 = 0.4288781
                      m2 = 0.5440201
           n1 (unadjusted) = 1570
```

(continues on next page)

(continued from previous page)

```
           n2 (unadjusted) = 1570
                       rho = 0.3
       Average per cluster = 10
minimum number of clusters = 1162
             n1 (adjusted) = 5809
             n2 (adjusted) = 5809
                 sig.level = 0.05
                     power = 0.8
               alternative = two.sided

NOTE:
- m1 and m2 are the means of group 1 and 2, respectively.
- n1 (unadjusted) and n2 (unadjusted) are the obs. of group 1 and 2 ignoring␣
↪clustering.
- n1 (adjusted) and n2 (adjusted) are the obs. of group 1 and 2 adjusting for␣
↪clustering.
```

CHAPTER 5

Indices and tables

- genindex
- modindex
- search

Index

C

clustered() (built-in function), 8

T

twomeans() (built-in function), 7