Power and sample size using R

Philip Dixon

2023-01-24

How to do power and sample size computations using R

As usual with R, there are many ways to do something, especially when you look into add-on packages. My explanation documents describe what I do.

Tquantile.r shows how to use R to compute quantiles of T distributions. It also shows you to compute probabilities for T distributions. These can be used for all "by hand" calculations.

The week 2 notes have various versions of power and sample size calculations. Here's how to code those in R.

How big a true difference is needed to get 80% power when n = 20 per group and s = 5? Calculate the df and se, then use the fundamental power formula:

```
df <- 2*(20-1)
se <- 5*sqrt(2/20)

(qt(0.975, df) + qt(0.8, df)) * se
```

```
## [1] 4.546686
```

What n is needed for 80% power to detect a difference of 2 when s = 14.8 This requires an iterative calculation. I start with df = 60, which is n = 31 for a two-sample comparison:

```
delta <- 2  # specified by the question
s <- 14.8   # ditto

n <- 31   # arbitrary "getting started" value
df <- 2*(n-1)

# compute new n
n <- 2*(qt(0.975, df) + qt(0.8, df))^2 * (s/delta)^2
n
```

```
## [1] 888.2974
```

```
# repeat for new n
df <- 2*(n-1)
n <- 2*(qt(0.975, df) + qt(0.8, df))^2 * (s/delta)^2
n
```

```
## [1] 860.5548
```

What is power when n = 400, delta = 2, s = 14.8 Calculate df and se, then calculate the necessary Tquantile for power (Tb in the code). Then calculate the probability associated with Tb:

```r
n <- 400
df <- 2*(n-1)
se <- 14.8*sqrt(2/n)
Tb <- delta/se - qt(0.975, df)
Tb
```

```
## [1] -0.05184178
```

```r
# find P[T < Tb]
pt(Tb, df)
```

```
## [1] 0.4793339
```

Power / sample size using R functions

power.t.test() computes power for 1- or 2-sample T tests. The default is two sample. You specify 4 of the 5 parameters and power.t.test() will solve for the remaining one. The arguments are:

- n = sample size per group
- delta = difference in population means
- sd = population standard deviation
- power = desired power
- sig.level = alpha (type I error rate), default = 0.05

You specify 3 of n, delta, sd, and power and power.t.test() will compute the remaining one. sd has a default of 1, so if you omit sd, delta is Cohen's effect size (don't worry about this if you haven't heard about it. It's used most often in the social sciences).

```r
power.t.test(delta = 0.5, sd = 0.9, power = 0.80)  # gives sample size (n is not specified)
```

```
##
##      Two-sample t test power calculation
##
##               n = 51.83884
##           delta = 0.5
##              sd = 0.9
##       sig.level = 0.05
##           power = 0.8
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

```r
power.t.test(n = 10, delta = 0.5, sd = 0.9)        # gives power
```

```
##
##      Two-sample t test power calculation
##
##               n = 10
##           delta = 0.5
##              sd = 0.9
##       sig.level = 0.05
##           power = 0.2167262
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

```r
power.t.test(n = 10, power = 0.8, sd = 0.9)        # gives "minimum detectable effect"
```

```
##
##      Two-sample t test power calculation
##
##               n = 10
##           delta = 1.192451
##              sd = 0.9
##       sig.level = 0.05
##           power = 0.8
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```