

Experimental Design and Power Analysis

When designing experiments of critical importance is assessing how best to use your resources to test your hypothesis. Part of this includes how many replicates you need to be confident you will detect a biologically significant difference if it exists. It is always a good idea to determine power of your design in advance. If you do not and find no significant differences in analysis, you will then want to do a power analysis to determine what the probability is that you would detect a biologically significant difference.

The base version of R has functions for determining the power of a t-test, anova, or tests of proportion differences.

```
power.t.test(n, delta, sd, sig.level=0.05, power)
```

```
power.anova.test(groups, n, between.var, within.var, sig.level=0.05, power)
```

```
power.prop.test(n, p1, p2, sig.level=0.05, power)
```

In all of these functions, any one of the arguments listed for them above can be passed a value=NULL and it will be calculated based on the other values. Values must be given for each argument except sig.level, which has a default value of 0.05 in all functions; if you want to calculate sig.value based on the other arguments you will need to explicitly set it equal to NULL. For all other arguments, leaving it out of the function call gives it a default value of NULL.

Writing your own code to determine power

If you have a complex or unusual experimental design, it is unlikely that one of the built-in power functions will fit your needs. However, as we have seen it is easy to randomize data in R, and thus it is easy to run a randomization that will let you determine power for any model you can run in R and any data you generate.

The basic concept is simple and no different from the typical randomization tests used to assess p-values for tests with non-normal data. Let's start with a simple example – you want to determine the power of a test to detect a difference between two groups, let's say in mean tibia length, just for a concrete example. You know from previous work that an increase of 2 mm in length increases fitness for this organism, so you want to be able to detect any difference ≥ 2 as a difference of this size will be biologically important. From a preliminary study, you have an estimate of variance in tibia length (0.8 mm²). And finally you hope a sample size of 12 per group will be sufficient.

To determine power under these circumstances, create a data set with two groups and draw the groups at random from a normal distribution (we expect tibia length to be normally distributed) with means that differ by 2 and with variance of 0.8. We run a t-test on the 2 groups to determine if they differ significantly and save the result. We do this many, many times. The proportion of times we find a significant difference is our power (since we KNOW the two groups differ on average by 2 mm). Let's do this.

```
groupA <- rnorm(12, 7, sqrt(0.8))
```

```
groupB <- rnorm(12, 9, sqrt(0.8))
```

Check to see that we are getting what we expected.

```
mean(groupA); var(groupA)
```

```
mean(groupB); var(groupB)
```

Now we run a t-test on the data:

```
res <- t.test(groupA,groupB, var.equal = TRUE)
```

```
str(res)
```

```
res$p.value
```

We'll save the results in a vector, coding the result as 1 if $p < 0.05$ and 0 otherwise.

```
sigs <- vector("numeric")
```

To do this many times, we need to use a loop.

```
for(i in 1:1000) {groupA <- rnorm(12, 18, sqrt(1.8))
```

```
  groupB <- rnorm(12, 19, sqrt(1.8))
```

```
  res <- t.test(groupA,groupB, var.equal = TRUE)
```

```
  sigs[i] <- ifelse(res$p.value<0.05, 1, 0)
```

```
}
```

So, now we can try different sample sizes to see how many we do need for decent power. But copying and pasting this code seems a little inefficient. Let's make it into a function.

```
pow <- function(n) {
```

```
  sigs <- vector("numeric")
```

```
  for(i in 1:1000) {
```

```
    groupA <- rnorm(n, 18, sqrt(1.8))
```

```
    groupB <- rnorm(n, 19, sqrt(1.8))
```

```
    res <- t.test(groupA,groupB, var.equal = TRUE)
```

```
    sigs[i] <- ifelse(res$p.value<0.05, 1, 0)}
```

```
  power <- sum(sigs)/999
```

```
  return(power)
```

```
}
```

Well that seems to suggest we need an unreasonably large sample size, maybe we will have to be content with a detecting a larger difference between means. Let's make our function more general so we can modify the other parameters also.

```
pow <- function(n, xbarA, xbarB, err) {
```

```
  sigs <- vector("numeric")
```

```
  for(i in 1:10000) {
```

```
    groupA <- rnorm(n, xbarA, sqrt(err))
```

```
    groupB <- rnorm(n, xbarB, sqrt(err))
```

```
    res <- t.test(groupA,groupB, var.equal = TRUE)
```

```
  sigs[i] <- ifelse(res$p.value<0.05, 1, 0)}  
  tpow <- sum(sigs)/9999  
  return(tpow)  
}
```

Now we can try different n, different group means, and different variances to see how that affects power. Let's compare our results to those using the built-in power function.

```
pow(18,18,19.5,1.8) # our function
```

```
power.t.test(18,1.5,1.341641) #built in function for power of a t.test
```