Experimental power analysis in R

Dr Ed Harris



# Introduction

An experiment with good quality data will not be able to detect a difference, even if one exists, if the sample size is small relative to the magnitude of difference.

When designing experiments, IT IS YOUR RESPONSIBILITY TO ENSURE YOU HAVE REASONABLE EXPERIMENTAL POWER for your experiment.

Today, we'll use R to calculate experimental power for simple experimental designs. In practice, the math is very simple, however the effect size (ES) calculation is unique for most statistical tests. Thus, we'll cover just a few basics.

Power for more complicated designs can be estimated using specialized software like G\*Power (or from scratch using methods in Cohen, J., 1988. *Statistical Power Analysis for the Behavioral Sciences* 2nd ed., Routledge Academic; I consider awareness of Cohen 1988 essential for any working scientist).

G-power is one of my favorites (open source, free, academically precise, relatively easy to use), however we won't have time to learn use it this term.

NB that most applications, and certainly the most powerful applications, of power analysis is at the design stage of an experiment BEFORE you collect any data. However, in some cases you may be constrained in your data collection for practical reasons such as for conservation studies, medical studies, applied research, etc. In this case, the least you are expected to do is to be aware of your experimental power.

# Cohen's power conventions for small, medium and large effects

These conventions should be used with caution. What is a small or even trivial effect in one context may be a large effect in another context? For example, Rosnow and Rosenthal (1989) discussed a 1988 biomedical research study on the effects of taking a small, daily dose of aspirin. Each participant was instructed to take one pill per day. For about half of the participants the pill was aspirin, for the others it was a placebo.

The dependent variable was whether the participant had a heart attack during the study. In terms of a correlation coefficient, the size of the observed effect was *r* = .034. In terms of percentage of variance explained, that is 0.12%. In other contexts this might be considered a small, trivial effect, but it this context the researchers decided it was unethical to continue the study and the contacted all of the participants who were taking the placebo and told them to start taking aspirin every day.

# T-test, difference between two means

|  |  |  |
| --- | --- | --- |
| Size of effect | *d* | % variance |
| small | .2 | 1 |
| medium | .5 | 6 |
| large | .8 | 16 |



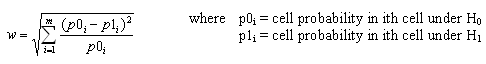
# Pearson Correlation Coefficient

|  |  |  |
| --- | --- | --- |
| Size of effect | *ρ* | % variance |
| small | .1 | 1 |
| medium | .3 | 9 |
| large | .5 | 25 |

We use the population correlation coefficient *p* (normally notated as *r*) as the effect size measure.

# Chi square, contingency tables

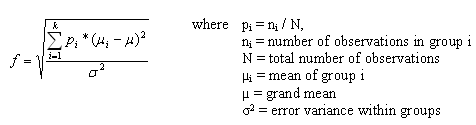
|  |  |  |
| --- | --- | --- |
| Size of effect | *w* =  | odds ratio\* |
| small | .1 | 1.49 |
| medium | .3 | 3.45 |
| large | .5 | 9 |



I.e., p0 is the expected proportion of expected, p1 is the proportion observed for m categories.

# 1-way ANOVA

|  |  |  |
| --- | --- | --- |
| Size of effect | *f* | % of variance |
| small | .1 | 1 |
| medium | .25 | 6 |
| large | .4 | 14 |

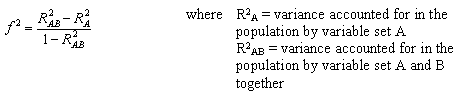


# Multiple *R2*

|  |  |  |
| --- | --- | --- |
| Size of effect | *f2* | % of variance |
| small | .02 | 2 |
| medium | .25 | 13 |
| large | .4 | 26 |



This is used when we are evaluating the impact of a set of predictors on an outcome.



# Comparing proportions, h

|  |  |
| --- | --- |
| Size of effect | *h* |
| small | .2 |
| medium | .5 |
| large | .8 |

p1=First proportion

p2=Second proportion

*h* = 2\*asin(sqrt(p1))-2\*asin(sqrt(p2))

For non-parametric tests, it is an acceptable start to use the parametric equivalent power test for. However, because the statistical power of non-parametric tests tends to be weaker than their parametric counterparts, you should plan for a slightly larger sample size to account for it (e.g., +10%) .

# 01. The R library pwr

If needed, **download and load the library {pwr}** in R. This library is used to calculate power for many of the most common statistical test you are likely to encounter.

You will probably find that liberal use of the help() function is useful.

The goal here for every example below will be to calculate the required sample size to achieve power = 0.80.

To do this, we'll need to estimate an effect size (ES) for each statistical test we plan to evaluate. Doing this is specific to each individual test. There are some functions to help us in {pwr}.

# 02. 1 and 2 proportion test to compare proportions

There are slightly different tests for comparing one proportion to some null proportion, versus comparing 2 observed proportions.

You can use the function ES.h() to look at effect sizes for two proportion values.

First, let's look at the effect size of two proportions: 0.5 and 0.25.

help(ES.h)

ES.h(0.5, 0.25)

Is your effect size small, medium or large?

Now let's compare .25 to a null of .5 using the one proportion pwr test.

help(pwr.p.test)

Basically, you need to provide information to the respective arguments in order to calculate sample size (*a priori*) or power (*a posteriori*). Here given alpha = 0.05, power = 0.80, h = ES.H(0.5,0.25), what sample size do we need for our one sample proportion test?

> pwr.p.test(h = ES.h(0.5,0.25), n = NULL, sig.level = 0.05, power = 0.8)

proportion power calculation for binomial distribution (arcsine transformation)

h = 0.5235988

n = 28.62921

sig.level = 0.05

power = 0.8

alternative = two.sided

We need a sample size of n = 29.

What would our power be if we only have n=25?

> pwr.p.test(h = ES.h(0.5,0.25), n = 25, sig.level = 0.05, power = NULL)

proportion power calculation for binomial distribution (arcsine transformation)

h = 0.5235988

n = 25

sig.level = 0.05

power = 0.7447429

alternative = two.sided

Here, power = 0.74. Is this good enough if you can design your experiment to collect a further 3 data points? Only you can decide (but Asashouryuu is watching...).

The pwr.2p.test() function is similar, but for the case when you are comparing two functions directly to each other (rather than a single proportion to a known population proportion). Use help() to check it out.

help(pwr.2p.test)

What sample size is required for power = 0.80 to compare two measured proportions of .4 and .5. The SPECIFIC question is whether the means are different, thus here we want a two-tailed test.

> pwr.2p.test(h=ES.h(0.5,0.40), power = 0.8, sig.level=0.05, alternative = "two.sided")

Difference of proportion power calculation for binomial distribution (arcsine transformation)

h = 0.2013579

n = 387.1677

sig.level = 0.05

power = 0.8

alternative = two.sided

NOTE: same sample sizes

The required n is 388. Each! Gosh!

What if you specificly want to know the sample size for .80 power to see if your .5 proportion is bigger than .4. This calls for a “greater than one-sided test”. Before trying it, do you think the required sample size will be greater or smaller than that for the two sided test?

> pwr.2p.test(h=ES.h(0.5,0.40), power = 0.8, sig.level=0.05, alternative = "greater")

Yes, good science is JUST THAT EASY!

Try another 2 sample test. This time perhaps the estimates of expected proportions are closer together (i.e., a smaller ES). Say, 0.8 and 0.65. What is the required sample size for power = 0.80 for a two-tailed test?

What happens with this test if you can only get n = 55 subjects per sample (or, say, only have n = 55)? I.e., what is the expected two-tailed power for n=55 ?

# 03. 1 and 2 sample t-test

Here we'll compare one sample variable against a test parameter, or, more typically, we'll compare two means separated by a categorical variable with two levels using the pwr.t.test() function.

We need to know the effect size for a t-test. The effect size is the parameter d. The definition for a two sample t-test for d is:

d = |(mean.01 - mean.02)| / (standard deviation)

note: |stuff| is the symbol for absolute value of stuff - no negatives. You can do this in r with the abs() function.

Again, Cohen’s rules of thumb state that 0.2 is a small ES, 0.5 is medium ES and 0.8 is a large ES for a t-test. You need pilot data, estimates from literature or an educated guess here (i.e., how SURE are you that there is a difference?).

Let's do some tests.

> ?pwr.t.test

First, what is the power for a one sample t-test with n=60 subjects, and effect size of d = 0.2 (small) and alpha = 0.05?

> pwr.t.test(d=0.2,n=60,sig.level=0.05,

+ type="one.sample",alternative="two.sided")

One-sample t test power calculation

n = 60

d = 0.2

sig.level = 0.05

power = 0.3316786

alternative = two.sided

Not good for power huh? How many subjects needed for 0.80 power?

Second, what is the power of a (type=) "paired" t-test with medium ES and n=40. I hope you said power=0.87. You did, didn't you?

Third, what is the power of an experiment for a two-sample, two-sided t-test with n = 30, alpha = 0.05, absolute mean difference of 2, and pooled standard deviation of 2.8?

Is this good enough to do the experiment? How might you change the experiment to your liking?

# 04. correlation

Here, we'll be using pwr.r.test().

?pwr.r.test

Conveniently, the correlation coefficient, r, is itself the ES.

If r = .3 and n = 50, what is the power for a 2-sided (i.e., positive or negative) correlation?

pwr.r.test(r=0.3,n=50,sig.level=0.05,alternative="two.sided")

What about an experiment where your alternative hypothesis is (for justifiable biological reasons) that the correlation coefficient is “greater” than 0?

How many subjects required to detect a (two.sided) correlation with .80 power if the ES is 0.30? 0.50? 0.10?

Surprised?

# 05. chi square

?pwr.chisq.test

Here Cohen recommends the ES parameter, w, to be 0.1, 0.3 and 0.5, when power is small, medium and large respectively.

Let's say you have and experiment looking at the proportion of offspring expected from 3 phenotypes. You expect a large effect size. If your sample size is 200 offspring, what is your power expected to be?

pwr.chisq.test(w=0.5, df=(3-1), N=200)

What sample size do you need for power = 0.8?

Calculate w for an experiment where you observe 40 and 60 in categories A and B, where you expect 50 and 50.

|  |
| --- |
|  |

# 06. General linear model

Calculating power for a general linear model is straightforward, but there are a few extra bits of information required.

?pwr.f2.test

pwr.f2.test(u =, v = , f2 = , sig.level = , power = )

where u and v are the numerator and denominator degrees of freedom. We use f2 as the effect size measure. Cohen suggests f2 values of 0.02, 0.15, and 0.35 represent small, medium, and large effect sizes.

For a simple, one factor model (analogous to one-way ANOVA): The numerator degrees of freedom is usually the number of levels of a factor minus one (may be more complicated in more complicated models). The denominator is the sample size minus the numerator degrees of freedom.

E.g., in an experiment with one factor that has 4 levels (e.g., population north, south, east, west) measuring body size in 100 individuals (25 from each population), u = 4 - 1 and v = 100 - u.

Let's calculate power for the experiment above if the expected ES is medium

> pwr.f2.test(u = 3, v = 97, f2 = .15, sig.level = 0.5)

Multiple regression power calculation

u = 3

v = 97

f2 = 0.15

sig.level = 0.5

power = 0.9969326

How many subjects do we need for 80% power if the ES is small? The answer is <you tell me> per group...!

# 07. Data for power curves

Graphing “power curves” across different effect sizes and sample sizes is a popular way to explore the dimensional space for a particular experimental design and is a convenient, visual tool.

Let's do this for a correlation experiment.

Make a vector of correlation coefficients starting at 0.1, and ending at 0.5 and incremented by 0.01 using the seq() function (hence, you want a vector of 0.10, 0.11, 0.12... 0.49, 0.50).

Make a variable with the length of your correlation coefficient vector.

> r <- seq(.1,.5,.01)

> r

[1] 0.10 0.11 0.12 0.13 0.14 0.15 0.16 0.17 0.18 0.19 0.20 0.21 0.22 0.23 0.24 0.25 0.26 0.27 0.28 0.29 0.30 0.31 0.32 0.33 0.34 0.35 0.36

[28] 0.37 0.38 0.39 0.40 0.41 0.42 0.43 0.44 0.45 0.46 0.47 0.48 0.49 0.50

> nr <- length(r)

Similarly make a vector of power values from 0.40 to 0.90 in increments of 0.10 and another variable containing the length of the power vector.

Use the array() function to set up and array containing a number of rows equal to the number of r values and a number of columns equal to the number of power values. This array will hold your sample sizes.

> samsize <- array(data = numeric(nr\*np), dim=c(nr,np))

Now, we are going to use nest for() loops to fill the array. The outer for() loop is for the power values and the inner array is for the correlation values.

My first two for() loop lines looks like this:

> for (i in 1:np){

+ for (j in 1:nr){

Inside the loops, for every respective value of power, np[i], and correlation effect size, nr[i], you want to use pwr.r.test() to calculate the required sample size for this combination of power and effect size. Put the resulting sample size in a place holder variable. You can call this placeholder variable anything you want, even “Foelix”!

Let's late a break for a second and look at the structure of the r object that is the result of pwr.r.test(). Run a correlation power test to calculate the required sample size to achieve .5 power for r = .5 for a two.sided test and put it into an R object (called Foelix, or if you insist, something different).

Print Foelix. What is the class() of Foelix? What are the names() of Foelix. Notice the names attribute called “n”. What is inside Foelix$n?

Okay, back to the for() loops. We were saying that for each value of i and j in the loops you want to calculate the required sample size to achieve each respective level of power for each respective correlation coefficient and put it into Foelix.

Now, Foelix should only hold one number at a time and so at each step, we want a second command that takes the value in Foelix, rounds it up to the nearest whole integer (you can use the ceiling() function to do this) and puts it into our array for that combination of i and j. Something like this:

> samsize[j,i] <- ceiling(Foelix$n)

So far you should have your array set up, your loops set up, your command to perform power and put it into Foelix and your command to round Foelix up and populate our sample size array. Close off the for loops ( }} ) and run it.

I highly recommend you try to understand this process yourself futrther to just copying and pasting my code.

My array looks like this:

> head(samsize)

[,1] [,2] [,3] [,4] [,5] [,6]

[1,] 293 385 490 616 783 1047

[2,] 242 319 405 509 647 865

[3,] 204 268 340 428 543 726

[4,] 174 228 290 364 462 618

[5,] 150 197 250 314 399 532

[6,] 131 172 218 274 347 463

# 08. Setting up a Graph of Power ™

Pseudocode for the graph:

-put the range() of your r variable into an r object (this is the range of the x axis for our power graph)

-put the sample size range() of the sample size array into an R object (the range of our y axis

-set a vector of 6 colours, one for each power value

-use plot() with x and y set to your respective range values and the type argument set to “n”. The y axis should be labelled something like “Sample size” and the x axis is your different correlation coefficient values.

-set up a for() loop from 1 to the number of power levels

-inside the for loop for each power level, you want to draw a lines() setting x to your correlation coefficient values, y to your sample size array for all rows (the rows are the coefficient values) and the columns to the looping variable... Set the type argument to “l” for lines, the lwd argument to 2 for thincker lines and the col argument to your colors variable. The Mine is like this: lines(r, samsize[,i], type="l", lwd=2, col=colors[i])

Can you make yours look exactly like mine (with the abline(), title(), and legend() functions)?

