Getting started with the pwr package

Clay Ford

2018-03-03

The basic idea of calculating power or sample size with functions in the pwr package is to *leave out* the argument that you want to calculate. If you want to calculate power, then leave the `power` argument out of the function. If you want to calculate sample size, leave `n` out of the function. Whatever parameter you want to calculate is determined from the others.

You select a function based on the statistical test you plan to use to analyze your data. If you plan to use a two-sample t-test to compare two means, you would use the `pwr.t.test` function for estimating sample size or power. All functions for power and sample size analysis in the pwr package begin with `pwr`. Functions are available for the following statistical tests:

- `pwr.p.test`: one-sample proportion test
- `pwr.2p.test`: two-sample proportion test
- `pwr.2p2n.test`: two-sample proportion test (unequal sample sizes)
- `pwr.t.test`: two-sample, one-sample and paired t-tests
- `pwr.t2n.test`: two-sample t-tests (unequal sample sizes)
- `pwr.anova.test`: one-way balanced ANOVA
- `pwr.r.test`: correlation test
- `pwr.chisq.test`: chi-squared test (goodness of fit and association)
- `pwr.f2.test`: test for the general linear model

There are also a few convenience functions for calculating effect size as well as a generic `plot` function for plotting power versus sample size. All of these are demonstrated in the examples below.

A simple example

Let's say we suspect we have a loaded coin that lands heads 75% of the time instead of the expected 50%. We wish to create an experiment to test this. We will flip the coin a certain number of times and observe the proportion of heads. We will then conduct a one-sample proportion test to see if the proportion of heads is significantly different from what we would expect with a fair coin. We will judge significance by our p-value. If our p-value falls below a certain threshold, say 0.05, we will conclude our coin's behavior is inconsistent with that of a fair coin.

- Our null hypothesis is that the coin is fair and lands heads 50% of the time (\(\pi\) = 0.50).
- Our alternative hypothesis is that the coin is loaded to land heads more then 50% of the time (\(\pi\) > 0.50).

How many times should we flip the coin to have a high probability (or *power*), say 0.80, of correctly rejecting the null of \(\pi\) = 0.5 if our coin is indeed loaded to land heads 75% of the time?

Here is how we can determine this using the `pwr.p.test` function.

```
library(pwr)
pwr.p.test(h = ES.h(p1 = 0.75, p2 = 0.50),
           sig.level = 0.05,
           power = 0.80,
           alternative = "greater")
```

```
##
##      proportion power calculation for binomial distribution (arcsine transformation)
##
##               h = 0.5235988
##               n = 22.55126
##       sig.level = 0.05
##           power = 0.8
##     alternative = greater
```

The function tells us we should flip the coin 22.55127 times, which we round up to 23. Always round sample size estimates up. If we're correct that our coin lands heads 75% of the time, we need to flip it at least 23 times to have an 80% chance of correctly rejecting the null hypothesis at the 0.05 significance level.

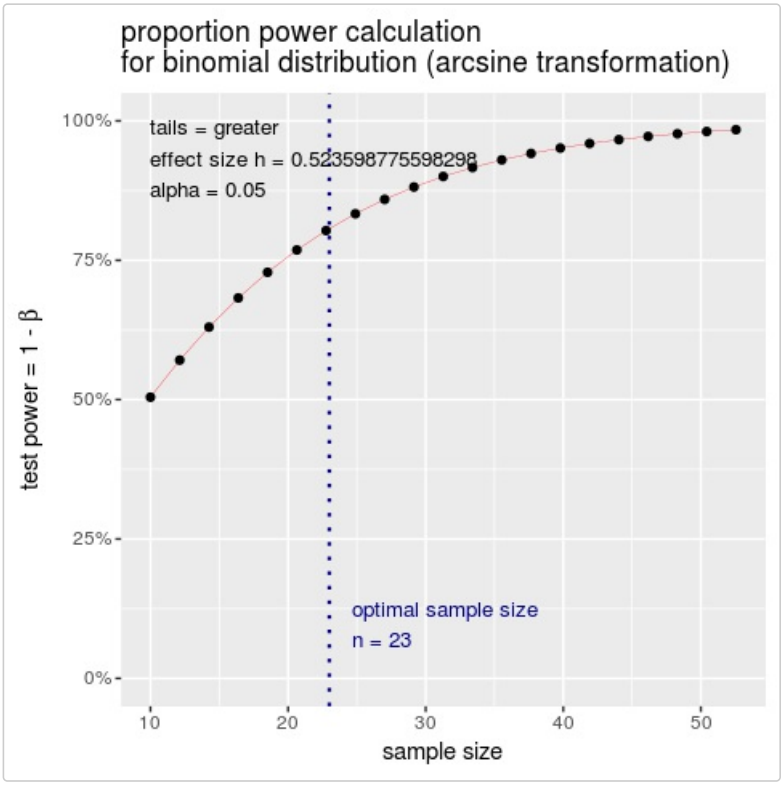
Notice that since we wanted to determine sample size (`n`), we left it out of the function. Our *effect size* is entered in the `h` argument. The label `h` is due to Cohen (1988). The function `ES.h` is used to calculate a unitless effect size using the arcsine transformation. (More on effect size below.) `sig.level` is the argument for our desired significance level. This is also sometimes referred to as our tolerance for a Type I error (\(\alpha\)). `power` is our desired power. It is sometimes referred to as 1 - \(\beta\), where \(\beta\) is Type II error. The `alternative` argument says we think the alternative is "greater" than the null, not just different.

Type I error, \(\alpha\), is the probability of rejecting the null hypothesis when it is true. This is thinking we have found an effect where none exist. This is considered the more serious error. Our tolerance for Type I error is usually 0.05 or lower.

Type II error, \(\beta\), is the probability of failing to reject the null hypothesis when it is false. This is thinking there is no effect when in fact there is. Our tolerance for Type II error is usually 0.20 or lower. Type II error is 1 - Power. If we desire a power of 0.90, then we implicitly specify a Type II error tolerance of 0.10.

The pwr package provides a generic `plot` function that allows us to see how power changes as we change our sample size. If you have the ggplot2 package installed, it will create a plot using `ggplot`. Otherwise base R graphics are used.

```
p.out <- pwr.p.test(h = ES.h(p1 = 0.75, p2 = 0.50),
                    sig.level = 0.05,
                    power = 0.80,
                    alternative = "greater")
plot(p.out)
```



What is the power of our test if we flip the coin 40 times and lower our Type I error tolerance to 0.01? Notice we leave out the `power` argument, add `n = 40`, and change `sig.level = 0.01`:

```
pwr.p.test(h = ES.h(p1 = 0.75, p2 = 0.50),
           sig.level = 0.01,
           n = 40,
           alternative = "greater")
```

```
##
##      proportion power calculation for binomial distribution (arcsine transformation)
##
##               h = 0.5235988
##               n = 40
##       sig.level = 0.01
##           power = 0.8377325
##     alternative = greater
```

The power of our test is about 84%.

We specified `alternative = "greater"` since we assumed the coin was loaded for more heads (not less). This is a stronger assumption than assuming that the coin is simply unfair in one way or another. In practice, sample size and power calculations will usually make the more conservative “two-sided” assumption. In fact this is the default for pwr functions with an `alternative` argument. If we wish to assume a “two-sided” alternative, we can simply leave it out of the function. Notice how our power estimate drops below 80% when we do this.

```
pwr.p.test(h = ES.h(p1 = 0.75, p2 = 0.50),
           sig.level = 0.01,
           n = 40)
```

```
##
##      proportion power calculation for binomial distribution (arcsine transformation)
##
##              h = 0.5235988
##              n = 40
##       sig.level = 0.01
##           power = 0.7690434
##     alternative = two.sided
```

What if we assume the "loaded" effect is smaller? Maybe the coin lands heads 65% of the time. How many flips do we need to perform to detect this smaller effect at the 0.05 level with 80% power and the more conservative two-sided alternative?

```
pwr.p.test(h = ES.h(p1 = 0.65, p2 = 0.50),
           sig.level = 0.05,
           power = 0.80)
```

```
##
##      proportion power calculation for binomial distribution (arcsine transformation)
##
##              h = 0.3046927
##              n = 84.54397
##       sig.level = 0.05
##           power = 0.8
##     alternative = two.sided
```

About 85 coin flips. Detecting smaller effects require larger sample sizes.

More on effect size

Cohen describes effect size as "the degree to which the null hypothesis is false." In our coin flipping example, this is the difference between 75% and 50%. We could say the effect was 25% but recall we had to transform the absolute difference in proportions to another quantity using the `ES.h` function. This is a crucial part of using the pwr package correctly: *You must provide an effect size on the expected scale.* Doing otherwise will produce wrong sample size and power calculations.

When in doubt, we can use *Conventional Effect Sizes*. These are pre-determined effect sizes for "small", "medium", and "large" effects. The `cohen.ES` function returns a conventional effect size for a given test and size. For example, the medium effect size for the correlation test is 0.3:

```
cohen.ES(test = "r", size = "medium")
```

```
##
##      Conventional effect size from Cohen (1982)
##
##            test = r
##            size = medium
##     effect.size = 0.3
```

For convenience, here are all conventional effect sizes for all tests in the pwr package:

Test	small	medium	large
tests for proportions (`p`)	0.2	0.5	0.8
tests for means (`t`)	0.2	0.5	0.8
chi-square tests (`chisq`)	0.1	0.3	0.5
correlation test (`r`)	0.1	0.3	0.5
anova (`anov`)	0.1	0.25	0.4
general linear model (`f2`)	0.02	0.15	0.35

It is worth noting that pwr functions can take vectors for effect size and n arguments. This allows us to make many power calculations at once, either for multiple effect sizes or multiple sample sizes. For example, let's see how power changes for our coin flipping experiment for the three conventional effect sizes of 0.2, 0.5, and 0.8, assuming a sample size of 20.

```
pwr.p.test(h = c(0.2,0.5,0.8),
           n = 20,
```

```
           sig.level = 0.05)
```

```
##
##      proportion power calculation for binomial distribution (arcsine transformation)
##
##               h = 0.2, 0.5, 0.8
##               n = 20
##       sig.level = 0.05
##           power = 0.1454725, 0.6087795, 0.9471412
##     alternative = two.sided
```

As we demonstrated with the `plot` function above, we can save our results. This produces a list object from which we can extract quantities for further manipulation. For example, we can calculate power for sample sizes ranging from 10 to 100 in steps of 10, with an assumed "medium" effect of 0.5, and output to a data frame with some formatting:

```
n <- seq(10,100,10)
p.out <- pwr.p.test(h = 0.5,
                    n = n,
                    sig.level = 0.05)
data.frame(n, power = sprintf("%.2f%%", p.out$power * 100))
```

```
##       n  power
## 1    10 35.26%
## 2    20 60.88%
## 3    30 78.19%
## 4    40 88.54%
## 5    50 94.24%
## 6    60 97.21%
## 7    70 98.69%
## 8    80 99.40%
## 9    90 99.73%
## 10 100 99.88%
```

We can also directly extract quantities with the `$` function appended to the end of a pwr function. For example,

```
pwr.p.test(h = 0.5, n = n, sig.level = 0.05)$power
```

```
##  [1] 0.3526081 0.6087795 0.7819080 0.8853791 0.9424375 0.9721272 0.9869034
##  [8] 0.9940005 0.9973108 0.9988173
```

More examples

pwr.2p.test - two-sample test for proportions

Let's say we want to randomly sample male and female college undergraduate students and ask them if they consume alcohol at least once a week. Our null hypothesis is no difference in the proportion that answer yes. Our alternative hypothesis is that there is a difference. This is a two-sided alternative; one gender has higher proportion but we don't know which. We would like to detect a difference as small as 5%. How many students do we need to sample in each group if we want 80% power and a significance level of 0.05?

If we think one group proportion is 55% and the other 50%:

```
pwr.2p.test(h = ES.h(p1 = 0.55, p2 = 0.50), sig.level = 0.05, power = .80)
```

```
##
##      Difference of proportion power calculation for binomial distribution (arcsine
transformation)
##
##               h = 0.1001674
##               n = 1564.529
##       sig.level = 0.05
##           power = 0.8
##     alternative = two.sided
##
## NOTE: same sample sizes
```

Notice the sample size is *per group*. We need to sample 1,565 males and 1,565 females to detect the 5% difference with 80% power.

If we think one group proportion is 10% and the other 5%:

```
pwr.2p.test(h = ES.h(p1 = 0.10, p2 = 0.05), sig.level = 0.05, power = .80)
```

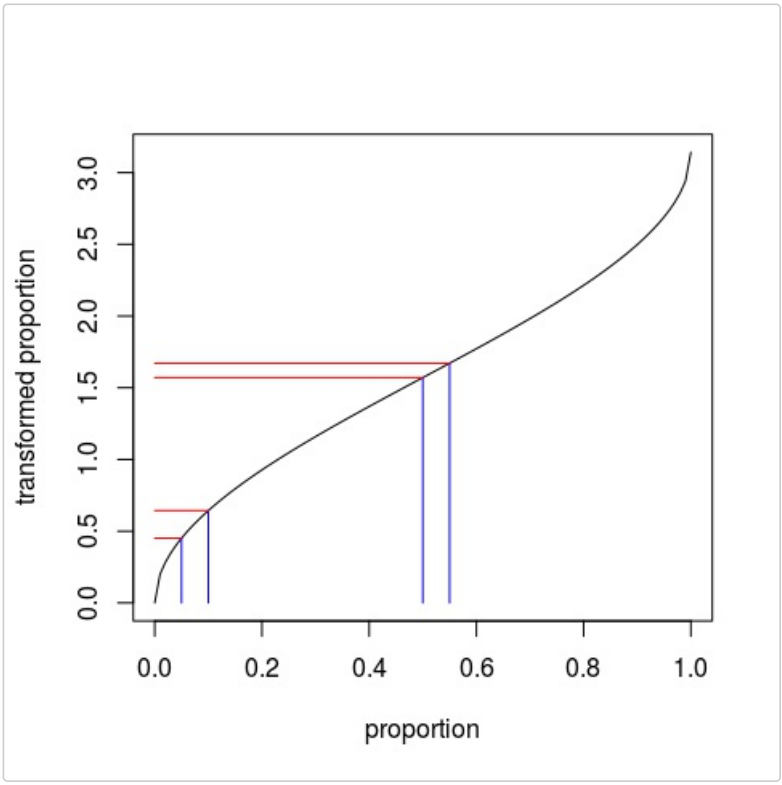
```
##
##      Difference of proportion power calculation for binomial distribution (arcsine
transformation)
##
##               h = 0.1924743
##               n = 423.7319
##       sig.level = 0.05
##           power = 0.8
##     alternative = two.sided
##
## NOTE: same sample sizes
```

Even though the absolute difference between proportions is the same (5%), the optimum sample size is now 424 per group. 10% vs 5% is actually a bigger difference than 55% vs 50%. A heuristic approach for understanding why is to compare the ratios: 55/50 = 1.1 while 10/5 = 2.

The `ES.h` function performs an arcsine transformation on both proportions and returns the difference. By setting `p2` to 0, we can see the transformed value for `p1`. We can exploit this to help us visualize how the transformation creates larger effects for two proportions closer to 0 or 1. Below we plot transformed proportions versus untransformed proportions and then compare the distance between pairs of proportions on each axis.

```
addSegs <- function(p1, p2){
  tp1 <- ES.h(p1, 0); tp2 <- ES.h(p2, 0)
  segments(p1,0,p1,tp1, col="blue"); segments(p2,0,p2,tp2,col="blue")
  segments(0, tp1, p1, tp1, col="red"); segments(0, tp2, p2, tp2, col="red")
}

curve(expr = ES.h(p1 = x, p2 = 0), xlim = c(0,1),
      xlab = "proportion", ylab = "transformed proportion")
addSegs(p1 = 0.50, p2 = 0.55) # 50% vs 55%
addSegs(p1 = 0.05, p2 = 0.10) # 5% vs 10%
```



The differences on the x-axis between the two pairs of proportions is the same (0.05), but the difference is larger for 5% vs 10% on the y-axis. The `ES.h` function returns the distance between the red lines.

Base R has a function called `power.prop.test` that allows us to use the raw proportions in the function without a need for a separate effect size function.

```
power.prop.test(p1 = 0.55, p2 = 0.50, sig.level = 0.05, power = .80)
```

```
##
##      Two-sample comparison of proportions power calculation
##
##               n = 1564.672
##              p1 = 0.55
##              p2 = 0.5
##       sig.level = 0.05
##           power = 0.8
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

Notice the results are slightly different. It calculates effect size differently.

If we don't have any preconceived estimates of proportions or don't feel comfortable making estimates, we can use conventional effect sizes of 0.2 (small), 0.5 (medium), or 0.8 (large). The sample size per group needed to detect a "small" effect with 80% power and 0.05 significance is about 393:

```
pwr.2p.test(h = 0.2, sig.level = 0.05, power = .80)
```

```
##
##      Difference of proportion power calculation for binomial distribution (arcsine
transformation)
##
##               h = 0.2
##               n = 392.443
##       sig.level = 0.05
##           power = 0.8
##     alternative = two.sided
##
## NOTE: same sample sizes
```

pwr.2p2n.test - two-sample test for proportions, unequal sample sizes

Let's return to our undergraduate survey of alcohol consumption. It turns out we were able to survey 543 males and 675 females. The power of our test if we're interested in being able to detect a "small" effect size with 0.05 significance is about 93%.

```
cohen.ES(test = "p", size = "small")
```

```
##
##      Conventional effect size from Cohen (1982)
##
##            test = p
##            size = small
##     effect.size = 0.2
```

```
pwr.2p2n.test(h = 0.2, n1 = 543, n2 = 675, sig.level = 0.05)
```

```
##
##      difference of proportion power calculation for binomial distribution (arcsine
transformation)
##
##               h = 0.2
##              n1 = 543
##              n2 = 675
##       sig.level = 0.05
##           power = 0.9344102
##     alternative = two.sided
##
## NOTE: different sample sizes
```

Let's say we previously surveyed 763 female undergraduates and found that *p*% said they consumed alcohol once a week. We would like to survey some males and see if a significantly different proportion respond yes. How many do I need to sample to detect a small effect size (0.2) in either direction with 80% power and a significance level of 0.05?

```
pwr.2p2n.test(h = 0.2, n1 = 763, power = 0.8, sig.level = 0.05)
```

```
##
##      difference of proportion power calculation for binomial distribution (arcsine
transformation)
##
##               h = 0.2
##              n1 = 763
##              n2 = 264.1544
##       sig.level = 0.05
##           power = 0.8
##     alternative = two.sided
##
## NOTE: different sample sizes
```

About 265.

pwr.t.test - one-sample and two-sample t tests for means

We're interested to know if there is a difference in the mean price of what male and female students pay at a library coffee shop. Let's say we randomly observe 30 male and 30 female students check out from the coffee shop and calculate the mean purchase price for each gender. We'll test for a difference in means using a two-sample t-test. How powerful is this experiment if we want to detect a "medium" effect in either direction with a significance level of 0.05?

```
cohen.ES(test = "t", size = "medium")
```

```
##
##      Conventional effect size from Cohen (1982)
##
##            test = t
##            size = medium
##     effect.size = 0.5
```

```
pwr.t.test(n = 30, d = 0.5, sig.level = 0.05)
```

```
##
##      Two-sample t test power calculation
##
##               n = 30
##               d = 0.5
##       sig.level = 0.05
##           power = 0.4778965
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

Only 48%. Not very powerful. How many students should we observe for a test with 80% power?

```
pwr.t.test(d = 0.5, power = 0.80, sig.level = 0.05)
```

```
##
##      Two-sample t test power calculation
##
##               n = 63.76561
##               d = 0.5
##       sig.level = 0.05
##           power = 0.8
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

About 64 per group.

Let's say we want to be able to detect a difference of at least 75 cents in the mean purchase price. We need to convert that to an effect size using the following formula:

$$d = \frac{m_{1} - m_{2}}{\sigma}$$

where m_{1} and m_{2} are the means of each group, respectively, and σ is the common standard deviation of the two groups. Again, the label *d* is due to Cohen (1988).

We have $m_{1} - m_{2} =$ 0.75. We need to make a guess at the population standard deviation. If we have absolutely no idea, one rule of thumb is to take the difference between the maximum and minimum values

and divide by 4. Let's say the maximum purchase is $10 and the minimum purchase is $1. Our estimated standard deviation is (10 - 1)/4 = 2.25. Therefore our effect size is 0.75/2.25 \(\approx\) 0.333.

```
d <- 0.75/2.25
pwr.t.test(d = d, power = 0.80, sig.level = 0.05)
```

```
##
##      Two-sample t test power calculation
##
##               n = 142.2462
##               d = 0.3333333
##       sig.level = 0.05
##           power = 0.8
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

For a desired power of 80%, Type I error tolerance of 0.05, and a hypothesized effect size of 0.333, we should sample at least 143 per group.

Performing the same analysis with the base R function `power.t.test` is a little easier. The difference \(m_{1} - m_{2} =\) 0.75 is entered in the `delta` argument and the estimated \(\sigma\) = 2.25 is entered in the `sd` argument:

```
power.t.test(delta = 0.75, sd = 2.25, sig.level = 0.05, power = 0.8)
```

```
##
##      Two-sample t test power calculation
##
##               n = 142.2466
##           delta = 0.75
##              sd = 2.25
##       sig.level = 0.05
##           power = 0.8
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

To calculate power and sample size for one-sample t-tests, we need to set the `type` argument to `"one.sample"`. By default it is set to `"two.sample"`.

For example, we think the average purchase price at the Library coffee shop is over $3 per student. Our null is $3 or less; our alternative is greater than $3. We can use a one-sample t-test to investigate this hunch. If the true average purchase price is $3.50, we would like to have 90% power to declare the estimated average purchase price is greater than $3. How many transactions do we need to observe assuming a significance level of 0.05? Let's say the maximum purchase price is $10 and the minimum is $1. So our guess at a standard deviation is 9/4 = 2.25.

```
d <- 0.50/2.25
pwr.t.test(d = d, sig.level = 0.05, power = 0.90, alternative = "greater",
           type = "one.sample")
```

```
##
##      One-sample t test power calculation
##
##               n = 174.7796
##               d = 0.2222222
##       sig.level = 0.05
##           power = 0.9
##     alternative = greater
```

We should plan on observing at least 175 transactions.

To use the `power.t.test` function, set `type = "one.sample"` and `alternative = "one.sided"`:

```
power.t.test(delta = 0.50, sd = 2.25, power = 0.90, sig.level = 0.05,
             alternative = "one.sided", type = "one.sample")
```

```
##
##      One-sample t test power calculation
##
##               n = 174.7796
```

```
##           delta = 0.5
##              sd = 2.25
##       sig.level = 0.05
##           power = 0.9
##     alternative = one.sided
```

"Paired" t-tests are basically the same as one-sample t-tests, except our one sample is usually differences in pairs. The following example should make this clear.

(*From Hogg & Tanis, exercise 6.5-12*) 24 high school boys are put on a ultra-heavy rope-jumping program. Does this decrease their 40-yard dash time (i.e., make them faster)? We'll measure their 40 time in seconds before the program and after. We'll use a paired t-test to see if the difference in times is greater than 0 (before - after). Assume the standard deviation of the differences will be about 0.25 seconds. How powerful is the test to detect a difference of about 0.08 seconds with 0.05 significance?

Notice we set `type = "paired"`:

```
pwr.t.test(n = 24, d = 0.08 / 0.25,
           type = "paired", alternative = "greater")
```

```
##
##      Paired t test power calculation
##
##               n = 24
##               d = 0.32
##       sig.level = 0.05
##           power = 0.4508691
##     alternative = greater
##
## NOTE: n is number of *pairs*
```

Only 45%. Not all that powerful. How many high school boys should we sample for 80% power?

```
pwr.t.test(d = 0.08 / 0.25, power = 0.8,
           type = "paired", alternative = "greater")
```

```
##
##      Paired t test power calculation
##
##               n = 61.75209
##               d = 0.32
##       sig.level = 0.05
##           power = 0.8
##     alternative = greater
##
## NOTE: n is number of *pairs*
```

About 62.

For paired t-tests we sometimes estimate a standard deviation for *within* pairs instead of for the difference in pairs. In our example, this would mean an estimated standard deviation for each boy's 40-yard dash times. When dealing with this type of estimated standard deviation we need to multiply it by $\sqrt{2}$ in the `pwr.t.test` function. Let's say we estimate the standard deviation of each boy's 40-yard dash time to be about 0.10 seconds. The sample size needed to detect a difference of 0.08 seconds is now calculated as follows:

```
pwr.t.test(d = 0.08 / (0.1 * sqrt(2)), power = 0.8,
           type = "paired", alternative = "greater")
```

```
##
##      Paired t test power calculation
##
##               n = 20.74232
##               d = 0.5656854
##       sig.level = 0.05
##           power = 0.8
##     alternative = greater
##
## NOTE: n is number of *pairs*
```

We need to sample at least 21 students.

pwr.t2n.test - two-sample t test for means, unequal sample sizes

Find power for a two-sample t-test with 28 in one group and 35 in the other group and a medium effect size. (sig.level defaults to 0.05.)

```
pwr.t2n.test(n1 = 28, n2 = 35, d = 0.5)
```

```
##
##      t test power calculation
##
##              n1 = 28
##              n2 = 35
##               d = 0.5
##       sig.level = 0.05
##           power = 0.4924588
##     alternative = two.sided
```

pwr.chisq.test - Goodness of fit test

(*From Cohen, example 7.1*) A market researcher is seeking to determine preference among 4 package designs. He arranges to have a panel of 100 consumers rate their favorite package design. He wants to perform a chi-square goodness of fit test against the null of equal preference (25% for each design) with a significance level of 0.05. What's the power of the test if 3/8 of the population actually prefers one of the designs and the remaining 5/8 are split over the other 3 designs?

We use the `ES.w1` function to calculate effect size. To do so, we need to create vectors of null and alternative proportions:

```
null <- rep(0.25, 4)
alt <- c(3/8, rep((5/8)/3, 3))
ES.w1(null,alt)
```

```
## [1] 0.2886751
```

To calculate power, specify effect size (`w`), sample size (`N`), and degrees of freedom, which is the number of categories minus 1 (`df` = 4 - 1).

```
pwr.chisq.test(w=ES.w1(null,alt), N=100, df=(4-1), sig.level=0.05)
```

```
##
##      Chi squared power calculation
##
##               w = 0.2886751
##               N = 100
##              df = 3
##       sig.level = 0.05
##           power = 0.6739834
##
## NOTE: N is the number of observations
```

If our estimated effect size is correct, we only have about a 67% chance of finding it (i.e., rejecting the null hypothesis of equal preference).

How many subjects do we need to achieve 80% power?

```
pwr.chisq.test(w=ES.w1(null,alt), df=(4-1), power=0.8, sig.level = 0.05)
```

```
##
##      Chi squared power calculation
##
##               w = 0.2886751
##               N = 130.8308
##              df = 3
##       sig.level = 0.05
##           power = 0.8
##
## NOTE: N is the number of observations
```

If our alternative hypothesis is correct then we need to survey at least 131 people to detect it with 80% power.

pwr.chisq.test - test of association

We want to see if there's an association between gender and flossing teeth among college students. We randomly sample 100 students (male and female) and ask whether or not they floss daily. We want to carry out a chi-square test of association to determine if there's an association between these two variables. We set our significance level to 0.01. To determine effect size we need to propose an alternative hypothesis, which in this case is a table of proportions. We propose the following:

gender	Floss	No Floss
Male	0.1	0.4
Female	0.2	0.3

We use the `ES.w2` function to calculate effect size for chi-square tests of association

```
prob <- matrix(c(0.1,0.2,0.4,0.3), ncol=2,
               dimnames = list(c("M","F"),c("Floss","No Floss")))
prob
```

```
##   Floss No Floss
## M   0.1      0.4
## F   0.2      0.3
```

This says we sample even proportions of male and females, but believe 10% more females floss.

Now use the matrix to calculate effect size:

```
ES.w2(prob)
```

```
## [1] 0.2182179
```

We also need degrees of freedom. `df` = (2 - 1) * (2 - 1) = 1

And now to calculate power:

```
pwr.chisq.test(w = ES.w2(prob), N = 100, df = 1, sig.level = 0.01)
```

```
##
##      Chi squared power calculation
##
##               w = 0.2182179
##               N = 100
##              df = 1
##       sig.level = 0.01
##           power = 0.3469206
##
## NOTE: N is the number of observations
```

At only 35% this is not a very powerful experiment. How many students should I survey if I wish to achieve 90% power?

```
pwr.chisq.test(w = ES.w2(prob), power = 0.9, df = 1, sig.level = 0.01)
```

```
##
##      Chi squared power calculation
##
##               w = 0.2182179
##               N = 312.4671
##              df = 1
##       sig.level = 0.01
##           power = 0.9
##
## NOTE: N is the number of observations
```

About 313.

If you don't suspect association in either direction, or you don't feel like building a matrix in R, you can try a conventional effect size. For example, how many students should we sample to detect a small effect?

```
cohen.ES(test = "chisq", size = "small")
```

```
##
```

```
##      Conventional effect size from Cohen (1982)
##
##            test = chisq
##            size = small
##     effect.size = 0.1
```

```
pwr.chisq.test(w = 0.1, power = 0.9, df = 1, sig.level = 0.01)
```

```
##
##      Chi squared power calculation
##
##               w = 0.1
##               N = 1487.939
##              df = 1
##       sig.level = 0.01
##           power = 0.9
##
## NOTE: N is the number of observations
```

1,488 students. Perhaps more than we thought we might need.

We could consider reframing the question as a two-sample proportion test. What sample size do we need to detect a "small" effect in gender on the proportion of students who floss with 90% power and a significance level of 0.01?

```
pwr.2p.test(h = 0.2, sig.level = 0.01, power = 0.9)
```

```
##
##      Difference of proportion power calculation for binomial distribution (arcsine
transformation)
##
##               h = 0.2
##               n = 743.9694
##       sig.level = 0.01
##           power = 0.9
##     alternative = two.sided
##
## NOTE: same sample sizes
```

About 744 per group. Notice that 744 \(\times\) 2 = 1,488, the sample size returned previously by `pwr.chisq.test`. In fact the test statistic for a two-sample proportion test and chi-square test of association are one and the same.

pwr.r.test - correlation test

(*From Hogg & Tanis, exercise 8.9-12*) A graduate student is investigating the effectiveness of a fitness program. She wants to see if there is a correlation between the weight of a participant at the beginning of the program and the participant's weight change after 6 months. She suspects there is a "small" positive linear relationship between these two quantities. She will measure this relationship with correlation, *r*, and conduct a correlation test to determine if the estimated correlation is statistically greater than 0. How many subjects does she need to sample to detect this small positive (i.e., $r > 0$) relationship with 80% power and 0.01 significance level?

There is nothing tricky about the effect size argument, `r`. It is simply the hypothesized correlation. It can take values ranging from -1 to 1.

```
cohen.ES(test = "r", size = "small")
```

```
##
##      Conventional effect size from Cohen (1982)
##
##            test = r
##            size = small
##     effect.size = 0.1
```

```
pwr.r.test(r = 0.1, sig.level = 0.01, power = 0.8, alternative = "greater")
```

```
##
##      approximate correlation power calculation (arctangh transformation)
##
```

```
##               n = 999.2054
##               r = 0.1
##       sig.level = 0.01
##           power = 0.8
##     alternative = greater
```

She needs to observe about a 1000 students.

The default is a two-sided test. We specify `alternative = "greater"` since we believe there is small positive effect.

If she just wants to detect a small effect in either direction (positive or negative correlation), use the default settings of "two.sided", which we can do by removing the `alternative` argument from the function.

```
pwr.r.test(r = 0.1, sig.level = 0.01, power = 0.8)
```

```
##
##      approximate correlation power calculation (arctangh transformation)
##
##               n = 1162.564
##               r = 0.1
##       sig.level = 0.01
##           power = 0.8
##     alternative = two.sided
```

Now she needs to observe 1163 students. Detecting small effects requires large sample sizes.

pwr.anova.test - balanced one-way analysis of variance tests

(*From Hogg & Tanis, exercise 8.7-11*) The driver of a diesel-powered car decides to test the quality of three types of fuel sold in his area based on the miles per gallon (mpg) his car gets on each fuel. He will use a balanced one-way ANOVA to test the null that the mean mpg is the same for each fuel versus the alternative that the means are different. ("balanced" means equal sample size in each group; "one-way" means one grouping variable.) How many times does he need to try each fuel to have 90% power to detect a "medium" effect with a significance of 0.01?

We use `cohen.ES` to get learn the "medium" effect value is 0.25. We put that in the `f` argument of `pwr.anova.test`. We also need to specify the number of groups using the `k` argument.

```
cohen.ES(test = "anov", size = "medium")
```

```
##
##      Conventional effect size from Cohen (1982)
##
##            test = anov
##            size = medium
##     effect.size = 0.25
```

```
pwr.anova.test(k = 3, f = 0.25, sig.level = 0.01, power = 0.9)
```

```
##
##      Balanced one-way analysis of variance power calculation
##
##               k = 3
##               n = 94.48714
##               f = 0.25
##       sig.level = 0.01
##           power = 0.9
##
## NOTE: n is number in each group
```

He would need to measure mpg 95 times for each type of fuel. His experiment may take a while to complete.

The effect size `f` is calculated as follows:

$$f = \frac{\sigma_{means}}{\sigma_{pop'n}}$$

where σ_{means} is the standard deviation of the *k* means and $\sigma_{pop'n}$ is the common standard deviation of the *k* groups. These two quantities are also known as the *between-group* and *within-group* standard deviations. If our driver suspects the between-group standard deviation is 5 mpg and the within-group standard deviation is 3 mpg, `f` = 5/3.

```
pwr.anova.test(k = 3, f = 5/3, sig.level = 0.01, power = 0.9)
```

```
##
##      Balanced one-way analysis of variance power calculation
##
##               k = 3
##               n = 3.842228
##               f = 1.666667
##       sig.level = 0.01
##           power = 0.9
##
## NOTE: n is number in each group
```

In this case he only needs to try each fuel 4 times. Clearly the hypothesized effect has important consequences in estimating an optimum effect size.

We can also use the `power.anova.test` function that comes with base R. It requires between-group and within-group *variances*. To get the same result as `pwr.anova.test` we need to square the standard deviations to get variances and multiply the between-group variance by \(\frac{k}{k-1}\). This is because the effect size formula for the ANOVA test assumes the between-group variance has a denominator of *k* instead of *k - 1*.

```
power.anova.test(groups = 3,
                 within.var = 3^2,
                 between.var = 5^2 * (3/2),
                 sig.level = 0.01, power = 0.90)
```

```
##
##      Balanced one-way analysis of variance power calculation
##
##          groups = 3
##               n = 3.842225
##     between.var = 37.5
##      within.var = 9
##       sig.level = 0.01
##           power = 0.9
##
## NOTE: n is number in each group
```

pwr.f2.test - test for the general linear model

(*From Kutner, et al, exercise 8.43*) A director of admissions at a university wants to determine how accurately students' grade-point averages (gpa) at the end of their first year can be predicted or explained by SAT scores and high school class rank. A common approach to answering this kind of question is to model gpa as a function of SAT score and class rank. Or to put another way, we can perform a multiple regression with gpa as the dependent variable and SAT and class rank as independent variables.

The null hypothesis is that none of the independent variables explain any of the variability in gpa. This would mean their regression coefficients are statistically indistinguishable from 0. The alternative is that at least one of the coefficients is not 0. This is tested with an F test. We can estimate power and sample size for this test using the `pwr.f2.test` function.

The F test has numerator and denominator degrees of freedom. The numerator degrees of freedom, `u`, is the number of coefficients you'll have in your model (minus the intercept). In our example, `u = 2`. The denominator degrees of freedom, `v`, is the number of error degrees of freedom: \(v = n - u - 1\). This implies \(n = v + u + 1\).

The effect size, `f2`, is \(R^{2}/(1 - R^{2})\), where \(R^{2}\) is the coefficient of determination, aka the "proportion of variance explained". To determine effect size you hypothesize the proportion of variance your model explains, or the \(R^{2}\). For example, if I think my model explains 45% of the variance in my dependent variable, the effect size is 0.45/(1 - 0.45) \(\approx\) 0.81.

Returning to our example, let's say the director of admissions hypothesizes his model explains about 30% of the variability in gpa. How large of a sample does he need to take to detect this effect with 80% power at a 0.001 significance level?

```
pwr.f2.test(u = 2, f2 = 0.3/(1 - 0.3), sig.level = 0.001, power = 0.8)
```

```
##
##      Multiple regression power calculation
##
##               u = 2
##               v = 49.88971
```

```
##              f2 = 0.4285714
##       sig.level = 0.001
##           power = 0.8
```

Recall \(n = v + u + 1\). Therefore he needs 50 + 2 + 1 = 53 student records.

What is the power of the test with 40 subjects and a significance level of 0.01? Recall \(v = n - u - 1\).

```
pwr.f2.test(u = 2, v = 40 - 2 - 1, f2 = 0.3/(1 - 0.3), sig.level = 0.01)
```

```
##
##      Multiple regression power calculation
##
##               u = 2
##               v = 37
##              f2 = 0.4285714
##       sig.level = 0.01
##           power = 0.8406124
```

Power is about 84%.

References and Further Reading

Cohen, J. (1988). *Statistical Power Analysis for the Behavioral Sciences (2nd ed.)*. LEA.
Dalgaard, P. (2002). *Introductory Statistics with R*. Springer. (Ch. 2)
Hogg, R and Tanis, E. (2006). *Probability and Statistical Inference (7th ed.)*. Pearson. (Ch. 9)
Kabacoff, R. (2011). *R in Action*. Manning. (Ch. 10)
Kutner, et al. (2005). *Applied Linear Statistical Models*. McGraw-Hill. (Ch. 16)
Ryan, T. (2013). *Sample Size Determination and Power*. Wiley.

The CRAN Task View for Clinical Trial Design, Monitoring, and Analysis lists various R packages that also perform sample size and power calculations.