POLI 787 - Power and Sensitivity Analysis

Colin Case

September 8, 2022

Today we are going to focus on two tools: power analysis and sensitivity analysis. Both are important tools you could implement either in the beginning of your research design or after the fact to validate your results. To demonstrate their usefulness, we will be working with simulated data from `fabricate` so you can see how these tools might work both before fielding a study or in validating your results.

Power Analysis

The statistical power of a hypothesis test can be thought of as the probability of detecting an effect, if there is a true effect to be found. In general, the power of a test is determined by the hypothesized true effect, sample size, standard deviation, and significance level desired. There are also a few different variations of this test depending on your quantity of interest and research design. Let's start with a really simple example from the vignette to see how power analysis can work.

Imagine you and a friend are flipping a coin. When it lands heads, you get a dollar, when it lands tails, she gets a dollar. The only problem is, over the first few times you've played this game, the coin has landed tails 60% of the time! You want to conduct an experiment to determine whether or not the coin is fair, but how many times should you flip the coin? Even if the coin almost always comes up heads, you will sometimes flip tails. So in any given sample, you may simply get unlucky and coin will look fair even though it isn't. This is unlikely to be the case if you flip each coin a large number of times, but that takes effort, and you can never guarantee that you didn't get unlucky, so the question is: how comfortable are you with being wrong? What if the coin only comes up heads 55% of the time? Intuitively it seems like you would need a bigger sample, but how much bigger?

These are all questions about statistical power. In layman's terms, power is your ability to detect effects. In less layman-y terms, power is the probability of identifying an effect, conditional on one being present. Power underpins the design of vaccine trials, A/B tests, and is a driver of the "reproducibility crisis" in the social sciences. Ensuring that you have sufficient power means that you can be more certain of what your statistics are telling you. Being underpowered means that any effects you find are the result of chance, and any effects you don't find could just be because you might not have enough data.

Statistical power is the probability of identifying an effect, conditional on one being present. Power is generally a function of your posited effect size, the variability among your observations, your sample size, your confidence threshold. Power analyses are most often used to determine: a) how big of a sample you need for your analysis, and b) what is the minimum effect size that you are able to detect. This is done by holding the other factors constant, and increasing/decreasing the variable of interest (sample size or effect size) until you hit a power threshold. Generally researchers select the arbitrary power threshold of 0.80, meaning that you have a power of 80% and are able to detect true effects of that magnitude, with that sample size, 80% of the time.

Let's turn back to our coin example. Assuming a true proportion for the coin of landing heads at 60%, we can calculate how many coin flips we would need to achieve a power level of 80% with a significance level of .05. How do you think this would change if we thought the true proportion was 75% for the coin landing heads.

```r
# Set Seed
set.seed(100)
# Clear Environment
rm(list = ls())
# Load Packages
library(pwr)

# Conduct One-Sample proportions test (80%)
pwr.p.test(h = ES.h(p1 = 0.60, p2 = 0.5), # Specify difference
           sig.level = 0.05, # Specify significance level
           power = 0.80, # Specify power level
           alternative = 'greater') # Specify hypothesis
```

```
##
##      proportion power calculation for binomial distribution (arcsine transformation)
##
##               h = 0.2013579
##               n = 152.4863
##       sig.level = 0.05
##           power = 0.8
##     alternative = greater
```

```r
# Conduct with higher power level (90%)
pwr.p.test(h = ES.h(p1 = 0.75, p2 = 0.5),
           sig.level = 0.05,
           power = 0.90,
           alternative = 'greater')
```

```
##
##      proportion power calculation for binomial distribution (arcsine transformation)
##
##               h = 0.5235988
##               n = 31.23717
##       sig.level = 0.05
##           power = 0.9
##     alternative = greater
```

As you can see, the power test gives us the desired sample size to calculate an experiment with the desired outputs and hypothesized effects. We can also consider solving for the effect size we could uncover given a specific power threshold, significance level, and sample size as well as the power level of a given study.

```r
# Conduct One-Sample proportions test (80%)
pwr.p.test(n = 200,
           sig.level = 0.05,
           power = 0.80,
           alternative = 'greater')
```

```
##
##      proportion power calculation for binomial distribution (arcsine transformation)
##
##               h = 0.1758177
##               n = 200
##       sig.level = 0.05
##           power = 0.8
##     alternative = greater
```

```r
# Conduct with higher power level (90%)
pwr.p.test(h = ES.h(p1 = 0.6, p2 = 0.5),
           sig.level = 0.05,
           n = 200,
           alternative = 'greater')
```

```
##
##      proportion power calculation for binomial distribution (arcsine transformation)
##
##               h = 0.2013579
##               n = 200
##       sig.level = 0.05
##           power = 0.8854688
##     alternative = greater
```

Exercise: Power Analysis Using Simulated Data

For our exercise, we are going to see the effect of a treatment, Z, on number of school days attended by children. To do this, we are going to write a function where we can change the effect size and the sample size in our simulated data set.

```r
# Load Packages
library(fabricatr)

data.gen.func <- function(effect_size, sample_size){ # Specify function where can change
                                                     # effect_size and sample_size
    fabricate( # Call fabricate
    N = sample_size, # Specify sample size argument in fabricate
    school_n = sample(0:3, N, replace = TRUE), # Varying Intercept by School
    Z = sample(0:1, N, replace = TRUE), # Randomly assign treatment assignment
    days_attended = round(150 + effect_size*Z + school_n*2 + rnorm(N, mean = 0, sd = 20))
    # Specify outcome variable
  )
}
```

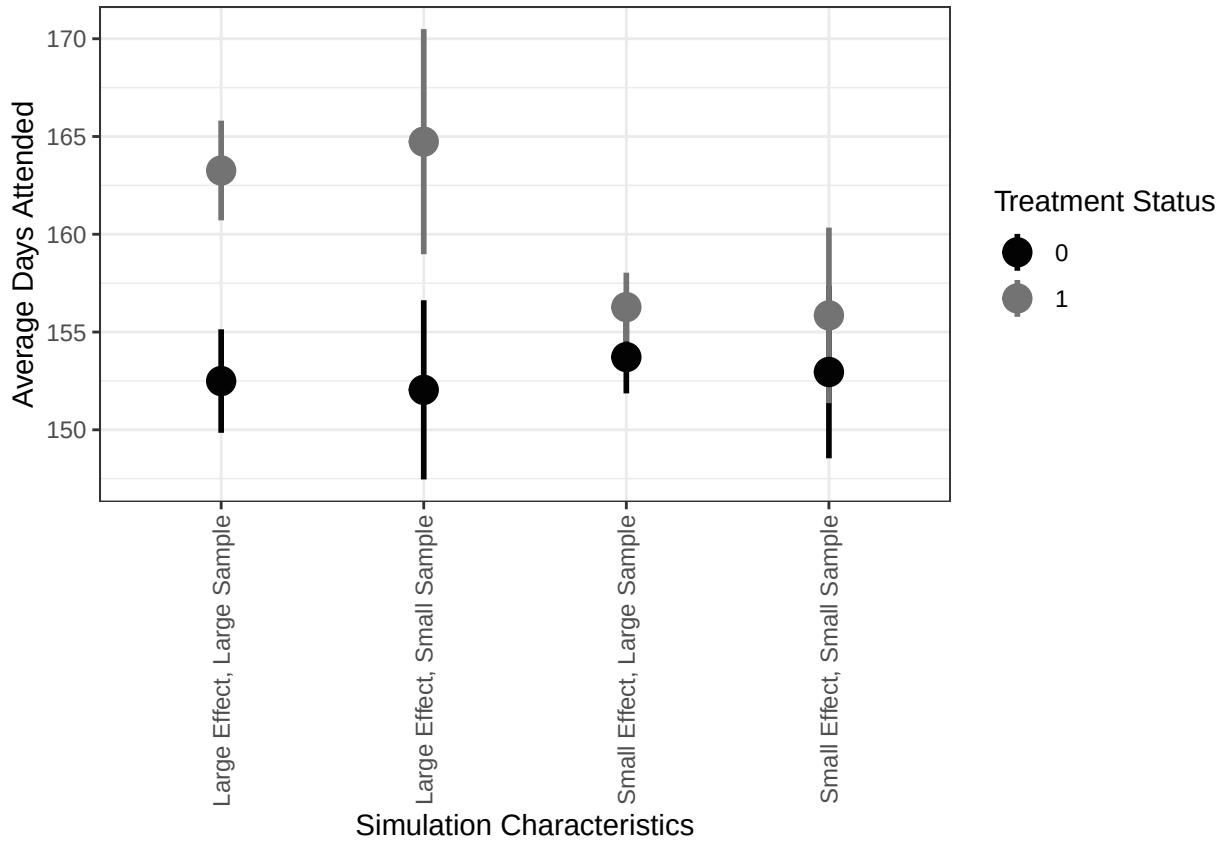
We are going to use four examples to see how these dataframes look: small effect, small sample; small effect, large sample, large effect, small sample, large effect, large sample

```r
# Small Effect Small Sample (1, 100)
ss <- data.gen.func(2, 100)

# Small Effect Large Sample (1, 500)
sl <- data.gen.func(2, 500)

# Large Effect Small Sample (10, 100)
ls <- data.gen.func(10, 100)

# Large Effect Large Sample (10, 500)
ll <- data.gen.func(10, 500)
```

Let's take some time to visualize this result. How do the confidence intervals appear relative to other sample sizes and effect sizes?

```r
# Load Packages
library(dplyr)
library(ggplot2)
```

```r
# Create Summary Data
ss.plot <- ss %>%
  group_by(Z) %>%
  summarise(mean = mean(days_attended), sd = sd(days_attended), n = nrow(ss)/2)

# Create Label for Plot
ss.plot$example <- 'Small Effect, Small Sample'

# Create Mean, SD and N for plot
sl.plot <- sl %>%
  group_by(Z) %>%
  summarise(mean = mean(days_attended), sd = sd(days_attended), n = nrow(sl))

sl.plot$example <- 'Small Effect, Large Sample'

ls.plot <- ls %>%
  group_by(Z) %>%
  summarise(mean = mean(days_attended), sd = sd(days_attended), n = nrow(ls)/2)

ls.plot$example <- 'Large Effect, Small Sample'

ll.plot <- ll %>%
  group_by(Z) %>%
  summarise(mean = mean(days_attended), sd = sd(days_attended), n = nrow(ll)/2)

ll.plot$example <- 'Large Effect, Large Sample'

# Create Plot Data
plot.data <- rbind(ss.plot, sl.plot, ls.plot, ll.plot)

# Plot Data
ggplot(plot.data, aes(x = as.factor(example), y = mean,
                colour = as.factor(Z))) +
  geom_pointrange(aes(ymin=mean - (1.96*sd)/sqrt(n),
                      ymax=mean + (1.96*sd)/sqrt(n)), size = 1) +
  theme_bw() +
  labs(x = 'Simulation Characteristics', y = 'Average Days Attended') +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) +
  guides(colour = guide_legend(title = 'Treatment Status')) +
  scale_color_manual(values=c("Gray1", "Gray45"))
```



As you can see, our ability to actually observe the simulated effect is a function of both the effect size and the sample size. Let's go ahead and conduct a power analysis for the small effect, small sample DF and the large effect, large sample DF.

```
power.small.small <- pwr.t.test(n = nrow(ss)/2, # Specify number in each treatment
                                d = (mean(ss$days_attended[ss$Z == 1]) - # Call mean using data
             mean(ss$days_attended[ss$Z == 0]))/sd(ss$days_attended), # divide by SD
           sig.level = 0.05) # Specify significance level

power.small.small # Call object
```

```
##
##      Two-sample t test power calculation
##
##               n = 50
##               d = 0.1803085
##       sig.level = 0.05
##           power = 0.1451005
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

```
power.large.large <- pwr.t.test(n = nrow(ll)/2, d = (mean(ll$days_attended[ll$Z == 1]) -
             mean(ll$days_attended[ll$Z == 0]))/sd(ss$days_attended),
           sig.level = 0.05)
power.large.large
```

```
##
```

```
##      Two-sample t test power calculation
##
##               n = 250
##               d = 0.6706122
##       sig.level = 0.05
##           power = 1
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

As you can see, our first example is pretty under-powered while our second example is pretty over-powered. We can use this to think about how many people we need in certain treatment conditions or sample.

Finally, we are going to do something you can do in one of your research designs by simulating different effect sizes and sample sizes to see how the power changes. This time, we conducting the power analysis, we'll specify our desired level of power (80%) and not the sample size.

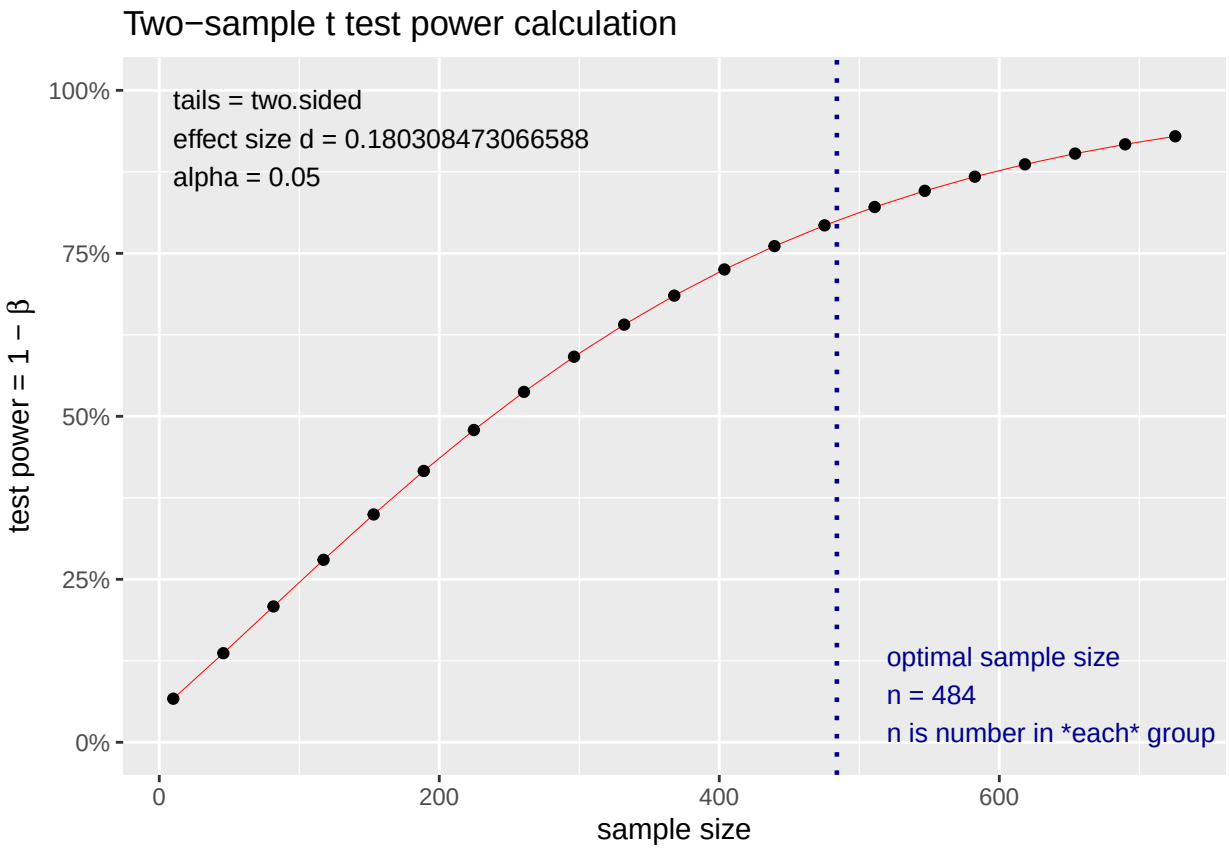
```
power.small.small <- pwr.t.test(power = 0.8 , d = (mean(ss$days_attended[ss$Z == 1]) -
            mean(ss$days_attended[ss$Z == 0]))/sd(ss$days_attended),
          sig.level = 0.05) # DO the same as above w/o n and specify power = 0.8

power.small.small
```

```
##
##      Two-sample t test power calculation
##
##               n = 483.8035
##               d = 0.1803085
##       sig.level = 0.05
##           power = 0.8
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

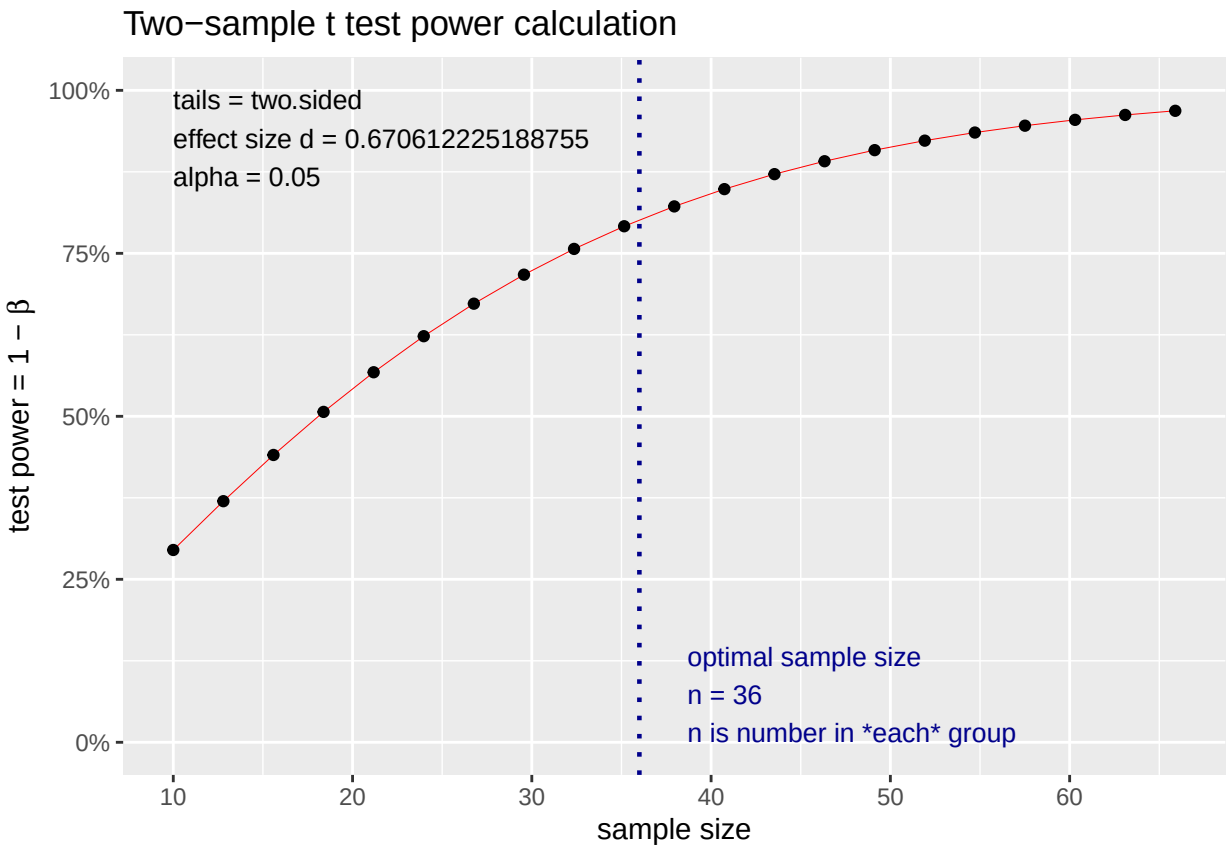
```
power.large.large <- pwr.t.test(power = 0.8, d = (mean(ll$days_attended[ll$Z == 1]) -
            mean(ll$days_attended[ll$Z == 0]))/sd(ss$days_attended),
          sig.level = 0.05) # Do the same as above w/o n and specify power = 0.8

power.large.large
```

```
##
##      Two-sample t test power calculation
##
##               n = 35.8916
##               d = 0.6706122
##       sig.level = 0.05
##           power = 0.8
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

```
# Plot object from power analysis
plot(power.small.small)
```




```
plot(power.large.large)
```



As you can see, our sample is probably way too small for the small effect size and way too big for the large effect size. We should probably reconsider a better design!

Sensitivity Analysis

As we have discussed throughout the semester, an important component of making causal claims in observation data is the assumption that, conditional on some covariates, our treatment status independent of the outcome variable. In most cases, that assumption is unlikely to hold and it is difficult, if not impossible, to discuss the universe of potential unobserved confounders and how they might bias the causal estimate. Sensitivity analysis is a way of quantitatively discussing the fragility of a result when our central assumption may be violated.

Cinelli and Hazlett (2019) develop a number of tools in their `sensemakr` package for dealing with potential issues of omitted variable bias. To do so, we are going to again be using simulated data to see how stable the relationship we have actually is.

Exercise: Sensitivity Analysis

Let's return to our simulated example from above. For this, we will use a more reasonable sample with an effect size of 4 and a sample size of 550 To see our SATE, let's run a quick OLS regression predicting days attended with treatment as the primary covariate. We will also control for school number (as a factor).

```
# Load Package
library(sensemakr)

# Simulate Data
df <- data.gen.func(4, 550)
```

```r
# Run Model
m1 <- lm(days_attended ~ Z + as.factor(school_n), data = df)

# See Output
summary(m1)
```

```
##
## Call:
## lm(formula = days_attended ~ Z + as.factor(school_n), data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -51.988 -13.358  -1.074  13.676  55.674
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)            148.369      1.802  82.349  < 2e-16 ***
## Z                        3.850      1.669   2.307 0.021438 *
## as.factor(school_n)1     2.769      2.345   1.181 0.238279
## as.factor(school_n)2     5.941      2.289   2.595 0.009701 **
## as.factor(school_n)3     7.958      2.404   3.310 0.000995 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 19.54 on 545 degrees of freedom
## Multiple R-squared:  0.03357,    Adjusted R-squared:  0.02647
## F-statistic: 4.732 on 4 and 545 DF,  p-value: 0.0009243
```

Significant effects! We are well on our way to publication! However, so far we have made the assumption of no unobserved confounders for unbiasedness. However, we've come to find out that our treatment was not randomly assigned, and parents could opt their children in. As you can imagine, that leaves us with a lot of potential confounders. We'll focus on one – parents' level of education. It is likely that this effects both our treatment status and our outcome variable. Let's start conducting an analysis to see how large of a problem this might be.

Sensitivity analysis works by taking a covariate in the model (in our case, School number 3), and seeing how sensitive results are to a potentially unobserved confounder some magnitude larger than the potential confounder.

Begin the analysis by applying `sensemakr` to the original regression model, `m1`.

The arguments are:

- model: the lm object with the outcome regression.
- treatment: the name of the treatment variable.
- benchmark_covariates: the names of covariates that will be used to bound the plausible strength of the unobserved confounders.
- kd and ky: these arguments parameterize how many times stronger the confounder is related to the treatment (kd) and to the outcome (ky) in comparison to the observed benchmark covariate. We will specify kd to 1:3 (i.e. once, twice and three times the size of school 3).
- q: this allows the user to specify what fraction of the effect estimate would have to be explained away to be problematic. Setting q = 1, as we do here, means that a reduction of 100% of the current effect estimate, that is, a true effect of zero, would be deemed problematic. The default is 1.

- alpha: significance level of interest for making statistical inferences. The default is 0.05.
- reduce: should we consider confounders acting towards increasing or reducing the absolute value of the estimate? The default is reduce = TRUE, which means we are considering confounders that pull the estimate towards (or through) zero.

```
# Conduct Sensitivity Analysis
model.sensitivity <- sensemakr(model = m1,  # Specify model
                               treatment = "Z", # Specify treatment
                               benchmark_covariates = "as.factor(school_n)3", # Covariate
                               kd = 1:10) # Size of Effect


summary(model.sensitivity) # Summary(object)
```

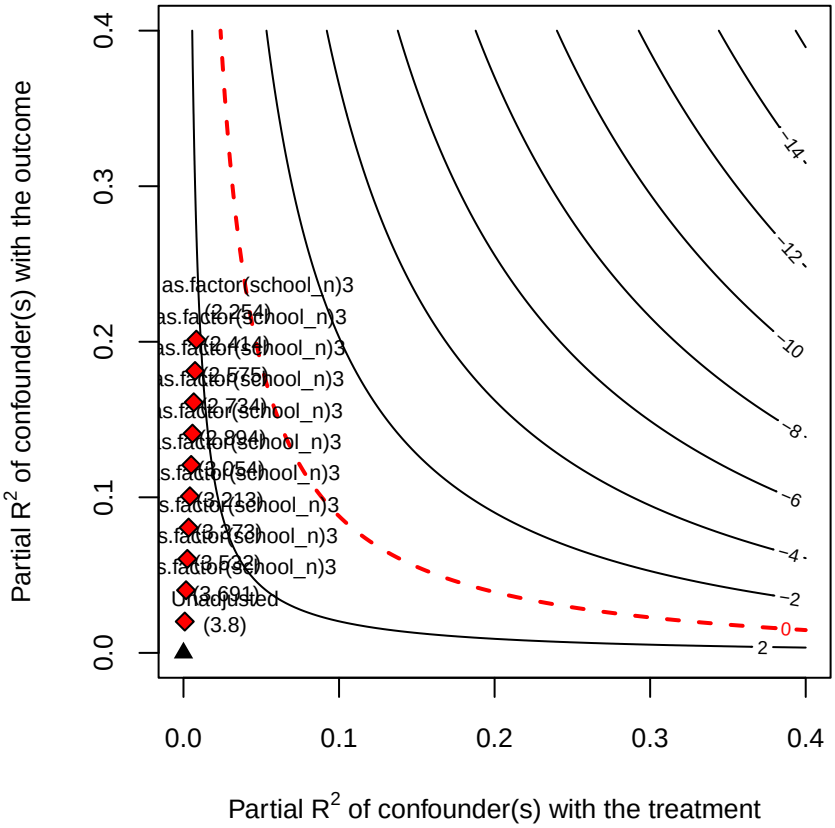
```
## Sensitivity Analysis to Unobserved Confounding
##
## Model Formula: days_attended ~ Z + as.factor(school_n)
##
## Null hypothesis: q = 1 and reduce = TRUE
## -- This means we are considering biases that reduce the absolute value of the current estimate.
## -- The null hypothesis deemed problematic is H0:tau = 0
##
## Unadjusted Estimates of 'Z':
##   Coef. estimate: 3.85
##   Standard Error: 1.6689
##   t-value (H0:tau = 0): 2.3068
##
## Sensitivity Statistics:
##   Partial R2 of treatment with outcome: 0.0097
##   Robustness Value, q = 1: 0.0941
##   Robustness Value, q = 1, alpha = 0.05: 0.0145
##
## Verbal interpretation of sensitivity statistics:
##
## -- Partial R2 of the treatment with the outcome: an extreme confounder (orthogonal to the covariates)
##
## -- Robustness Value, q = 1: unobserved confounders (orthogonal to the covariates) that explain more
##
## -- Robustness Value, q = 1, alpha = 0.05: unobserved confounders (orthogonal to the covariates) that
##
## Bounds on omitted variable bias:
##
## --The table below shows the maximum strength of unobserved confounders with association with the trea
##
##               Bound Label R2dz.x R2yz.dx Treatment Adjusted Estimate
##   1x as.factor(school_n)3 0.0008  0.0201         Z            3.6910
##   2x as.factor(school_n)3 0.0017  0.0403         Z            3.5319
##   3x as.factor(school_n)3 0.0025  0.0604         Z            3.3727
##   4x as.factor(school_n)3 0.0033  0.0805         Z            3.2133
##   5x as.factor(school_n)3 0.0041  0.1007         Z            3.0538
##   6x as.factor(school_n)3 0.0050  0.1208         Z            2.8942
##   7x as.factor(school_n)3 0.0058  0.1409         Z            2.7344
##   8x as.factor(school_n)3 0.0066  0.1611         Z            2.5745
##   9x as.factor(school_n)3 0.0074  0.1812         Z            2.4145
```

```
##  10x as.factor(school_n)3 0.0083  0.2013         Z            2.2543
##  Adjusted Se Adjusted T Adjusted Lower CI Adjusted Upper CI
##       1.6543     2.2312            0.4415            6.9405
##       1.6379     2.1564            0.3146            6.7492
##       1.6213     2.0803            0.1880            6.5573
##       1.6045     2.0027            0.0616            6.3650
##       1.5874     1.9237           -0.0645            6.1721
##       1.5702     1.8432           -0.1903            5.9786
##       1.5528     1.7610           -0.3158            5.7846
##       1.5351     1.6771           -0.4410            5.5900
##       1.5172     1.5914           -0.5658            5.3948
##       1.4991     1.5038           -0.6904            5.1990
```

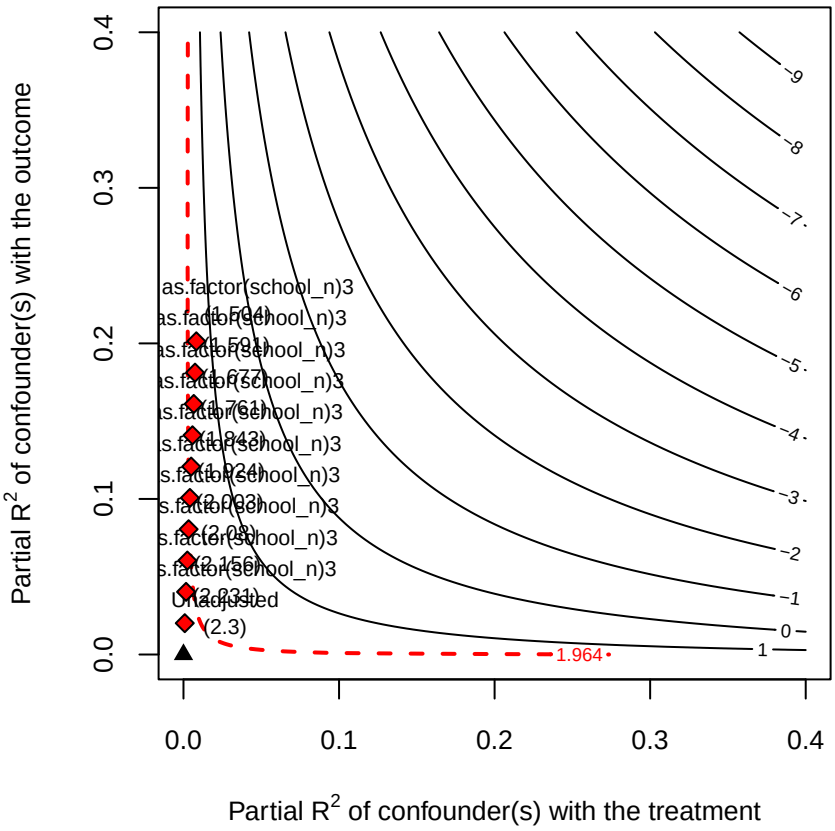
As you can see, it would take a pretty big effect size for our result to become not significant, in this case 5x the effect size of one of the factors we deemed to be substanatively important (school number 3). There is some art to this sort of justification, and it will rely on substantive expertise to justify. But 5x an effect is a pretty large justification.

We can also plot the results using contours plots.

```
plot(model.sensitivity) # Plot(object)
```




```
plot(model.sensitivity, sensitivity.of = "t-value") # Specify sensitivity.of = 't-value'
```



In the first plot, the horizontal axis shows the hypothetical residual share of variation of the treatment that unobserved confounding explains. The vertical axis shows the hypothetical partial R^2 of unobserved confounding with the outcome. The contours represent the effect size at various values. As you can see, where we would be with a confounder (represented by the red dots) of various sizes does not cross either the 0 line (effect size). It crosses the 1.96 contour line when looking at the t-statistic, but this is only after considering a confounder 5x the size of the one we tested.