PSYC 6780: Lab Tutorial 4

Chris M. Fiacconi

Planning a Research Study

Power and Sample Size

The planning stage of a research project is a crucial ingredient in determining the success of the project. Such planning often involves consideration of statistical **power** - the probability of correctly rejecting the null hypothesis. In other words, given that an effect exists, what is the probability that you will be able to detect it given the design of your experiment? As you probably know, power is greatly influenced by sample size (N). Power calculations often involve determining the appropriate sample size to achieve a desired level of power. To perform basic power calculations you can use the functions within the *pwr* package.

```
library(pwr) # first install.packages("pwr")
ls("package:pwr")
```

```
##  [1] "cohen.ES"         "ES.h"             "ES.w1"
##  [4] "ES.w2"            "plot.power.htest" "pwr.2p.test"
##  [7] "pwr.2p2n.test"    "pwr.anova.test"   "pwr.chisq.test"
## [10] "pwr.f2.test"      "pwr.norm.test"    "pwr.p.test"
## [13] "pwr.r.test"       "pwr.t.test"       "pwr.t2n.test"
```

```
# Example for paired design with Cohen's d = .2 - get N given power
pwr.t.test(n=NULL,d=.2,sig.level=.05,power=.80,type="paired",alternative="two.sided")
```

```
##
##      Paired t test power calculation
##
##               n = 198.1508
##               d = 0.2
##       sig.level = 0.05
##           power = 0.8
##     alternative = two.sided
##
## NOTE: n is number of *pairs*
```

```
# Example for paired design with Cohen's d = .2 - get power given N
pwr_value<-pwr.t.test(n=40,d=.2,sig.level=.05,power=NULL,type="paired",alternative="two.sided")
print(pwr_value)
```

```
##
##      Paired t test power calculation
##
##               n = 40
##               d = 0.2
##       sig.level = 0.05
##           power = 0.2345965
##     alternative = two.sided
##
## NOTE: n is number of *pairs*
```

```
print(pwr_value$power)
```

```
## [1] 0.2345965
```

Although the concept of power is often taught at the undergraduate level, power calculations can be considerably more complex than they may seem. This complexity stems from the inherent uncertainty in estimating key parameters that are used in power calculations. Part of this uncertainty is a product of *publication bias* - the preferential publication of p-values less than .05 (and by extension effect sizes that likely overestimate the true population effect size). The other element of uncertainty concerns the validity of the effect size estimate that is used to calculate power and/or sample size. Both of these sources of uncertainty can impact power and sample size calculations, and typically lead to under-powered experiments due to required sample sizes being underestimated.

Power and Assurance

Fortunately, methods have been developed that can (at least partially) correct for these biases. To understand these methods, it is important to understand the concept of **assurance** - the long-run probability that the achieved power of an experiment would reach or exceed the desired level (e.g., power = .80). To understand assurance, one must appreciate the fact that estimates of effect size are just that - estimates. To calculate the actual power of a study, we are required to know the population effect size. Of course, we almost never know the true value of δ, and so we estimate it based on the effect size calculated from a representative sample. Keep in mind, however, that the estimated effect size obtained from a sample of data is not a precise estimator of the true population effect size. An obtained Cohen's $d = .8$ may well reflect a true $\delta = .8$, but it is almost equally as likely to be obtained when $\delta = .3$ when it is calculated from a sample of $n = 25$. The precision with which d estimates δ depends on sample size - the number of participants in the sample for which d was calculated. Consider the figure below:

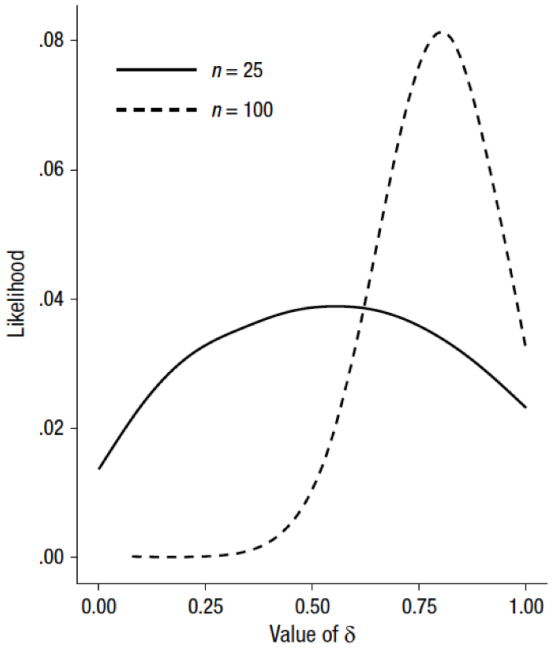


Fig. 2. Relative likelihood of various values of the population effect size, δ, given a sample Cohen's *d* of 0.8 and the presence of publication bias. Separate functions are shown for studies with 25 and 100 participants per group.

From the figure it is clear that the precision with which d estimates δ is heavily dependent on the sample size on which d is calculated. The uncertainty in this estimate poses serious problems for accurate sample size calculations, as it becomes difficult to determine the appropriate value of δ to use in this calcuation.

As a consequence of this uncertainty, the actual power achieved by a given experiment can fall well below the desired level. If we knew that $\delta = .8$, and we use this value to calculate the sample size needed to acheive a power = .80, then 4/5 experiments would correctly reject the null hypothesis. However, because we don't know δ, and we have to use Cohen's $d = .80$ instead, there is no guarantee that 4/5 experiments will correctly reject the null hypothesis. In reality, the proportion of experiments that correctly reject the null hypothesis is likely to be far less than .80, especially when small sample sizes are used.

On the bright side, there are alternative power and sample size calculations that can correct for the bias that is introduced by publication bias and the uncertainty in estimating δ. These calculations allow you to determine the required sample size that is needed to reach the desired level of power with a specified level of assurance. For example, one could ask, "How big does our sample size need to be in order achieve a power = .80 X% of the time?" These modified power and sample size calculations can be easily implemented for a variety of different experimental designs using the *BUCSS* (Bias and Uncertainty Corrected Sample Size) package for *R*.

Using the BUCSS Package

```
library(BUCSS) # first install.packages("BUCSS")
ls("package:BUCSS") # List functions within the BUCSS package
```

```
##  [1] "ss.power.ba"          "ss.power.ba.general"  "ss.power.dt"
##  [4] "ss.power.it"          "ss.power.reg.all"     "ss.power.reg.joint"
##  [7] "ss.power.reg1"        "ss.power.spa"         "ss.power.spa.general"
## [10] "ss.power.wa"          "ss.power.wa.general"
```

These functions allow you to calculate the appropriate sample size needed to achieve the desired level of power after correction for publication bias and paramater (δ) uncertainty. These functions are also available on a web app (created using *R*) and can be found at: https://designingexperiments.com/shiny-r-web-apps

```
# Example for paired design (dependent t-test) - previous study reported t-value = 3.00, N=40
smpl_size_prior<-ss.power.dt(t.observed=3,N=40,alpha.prior=.05,alpha.planned=.05,
                        assurance=.9,power=.8)

print(smpl_size_prior)
```

```
## [1] 8092
```

```
# Example for paired design (dependent t-test) - pilot study reported t-value = 2.40, N=40
smpl_size_pilot<-ss.power.dt(t.observed=3,N=40,alpha.prior=1,alpha.planned=.05,
                        assurance=.9,power=.8) # Set alpha.prior = 1

print(smpl_size_pilot)
```

```
## [1] 121
```

```r
# Example for pilot study using 2 X 2 factorial repeated-measures design using
# RM-ANOVA (F = 4.55, N = 36; for interaction)
w_ANOVA_smpl_size<-ss.power.wa(F.observed=4.55,N=36,levels.A=2,levels.B=2,
                               effect="interaction",alpha.prior=1,alpha.planned=.05,
                               assurance=.9,power=.8)  # Set alpha.prior = 1

print(w_ANOVA_smpl_size)
```

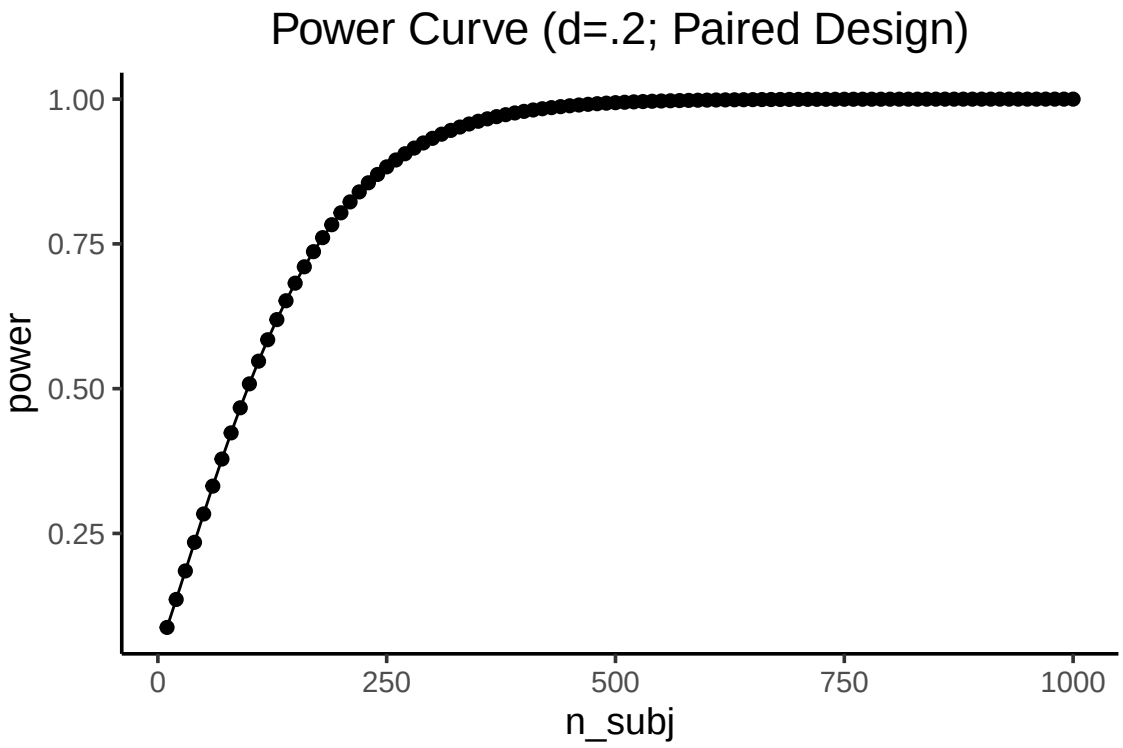
```
## [1] 466
```

Simulating Power and Sample Size

Power Curves

In addition to power calculations that specify a given N required for a desired level of power, it is often useful to know precisely how power increases with sample size for a given effect size. Generating a power curve can reveal this relationship to help you better decide on an appropriate N for your experiment. Here's an example of a power curve for a paired design with $\delta = .2$

```r
n_subj<-seq(10,1000,10) # Create vector of N's

power<-sapply(n_subj,
              FUN=function(x){
                pwr.t.test(n=x,
                d=.2,
                sig.level=.05,
                type="paired",
                alternative="two.sided")$power})

power_curve_data<-data.frame(n_subj,power)

library(tidyverse)
ggplot(data=power_curve_data,aes(x=n_subj,y=power)) +
  geom_point(size=2) +
  geom_line() +
  theme_classic(base_size=14) +
  ggtitle("Power Curve (d=.2; Paired Design)") +
  theme(plot.title = element_text(hjust = 0.5))
```

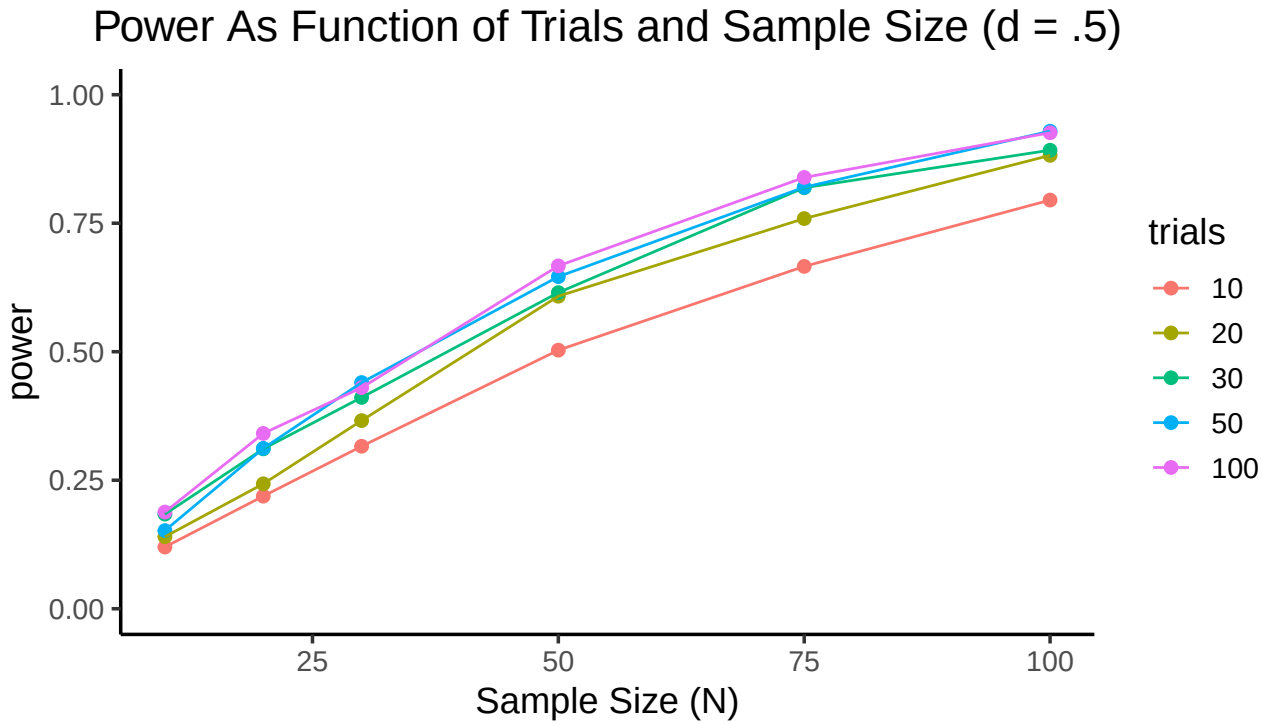


In experimental psychology, we typically measure participants' behaviour more than once per condition. In other words, we include multiple trials within a condition. This procedure ensures that we are obtaining an accurate estimate of how that individual performs in any given condition of the experiment. It turns out that the number of trials per condition can also impact power, and is therefore an important variable to consider when planning an experiment. Unfortunately, most power calculations do not take into account how power can be affected by the number of trials in a given condition.

However, we can obtain a rough estimate of how trial number affects power by running a *simulation*. Simulations allow us to create a world in which we "know" the relevant parameters, and allow us to see what would happen theoretically if we were to systematically vary those parameters. In our case, we can simulate power while varying both number of trials and sample size. Let's walk through an example of such a simulation for an independent groups design with a single factor comprised of two levels.

```
# Function to simulated power based on no. trials per condition and no. subjects

# nsubs sets number of subjects
# ntrials to change number of trials
# d sets effect size - Cohen's d
# this simulation is for a independent-groups design using a t-test
# assumes homogeneity of variance


sim_power <- function(nsubs,ntrials,d,sdev_b,sdev_w){
  A <- c()
  B <- c()
  for (i in 1:nsubs) {
  # Get distribution of subject means in Cond A
  Asub_means <- rnorm(n=nsubs,mean=1800,sd=sdev_b)
  # Sample ntrials from each subject mean
  A[i] <- mean(rnorm(n=ntrials,mean=sample(Asub_means,size=1),sd=sdev_w))
```

```
  # Convert standardized mean difference (d) into raw mean difference
  raw_diff <- sdev_b*d
  # Get distribution of subject means in Cond B
  Bsub_means <- rnorm(n=nsubs,mean=1800+raw_diff,sd=sdev_b)
  # Sample ntrials from each subject mean
  B[i] <- mean(rnorm(n=ntrials,mean=sample(Bsub_means,size=1),sd=sdev_w))
  }
  return(t.test(A,B,var.equal=TRUE)$p.value) # Get p-value from independent groups t-test
}

# vectors for number of subjects and trials
n_subs_vector <- c(10,20,30,50,75,100)
n_trials_vector <- c(10,20,30,50,100)

# a loop to run all simulations
power <- c()
subjects <- c()
trials <- c()

d <- .5 # medium effect size
sdev_b <- 200 # pooled between-subjects SD
sdev_w <- 500 # average within-subjects SD
n_reps <- 1000 # number of simulations to run

i <- 0 # use this as a counter for indexing
for(s in n_subs_vector){
  for(t in n_trials_vector){
    i <- i+1
    sims <- replicate(n_reps,sim_power(s,t,d,sdev_b,sdev_w))
    power[i] <- length(sims[sims<.05])/length(sims)
    subjects[i] <- s
    trials[i] <- t
  }
}

# combine into dataframe
plot_df <- data.frame(power,subjects,trials)
plot_df$trials<-as.factor(plot_df$trials)

# plot the power curve
ggplot(plot_df, aes(x=subjects,y=power,group=trials,color=trials)) +
  geom_point(size=2) +
  geom_line() +
  labs(x="Sample Size (N)") +
  ggtitle("Power As Function of Trials and Sample Size (d = .5)") +
  theme_classic(base_size=14) +
  theme(plot.title = element_text(hjust = 0.5)) +
  scale_y_continuous(limits=c(0,1))
```


Power As Function of Trials and Sample Size (d = .5)
power
1.00
0.75
0.50
0.25
0.00
25
50
75
100
Sample Size (N)
trials
10
20
30
50
100