R Handout - Power, Sample Size, Effect Size

Dr. Derek Ogle
Northland College, MTH207 Biometry

Winter 2013

Initialization

```
> library(NCStats)
> setwd("C:/aaaWork/Class Materials/MTH207/Lecture/HOs")
```

1 Initial Concept – 1-Sample Z-Test

1.1 Errors, Power, & Effect Size Definitions

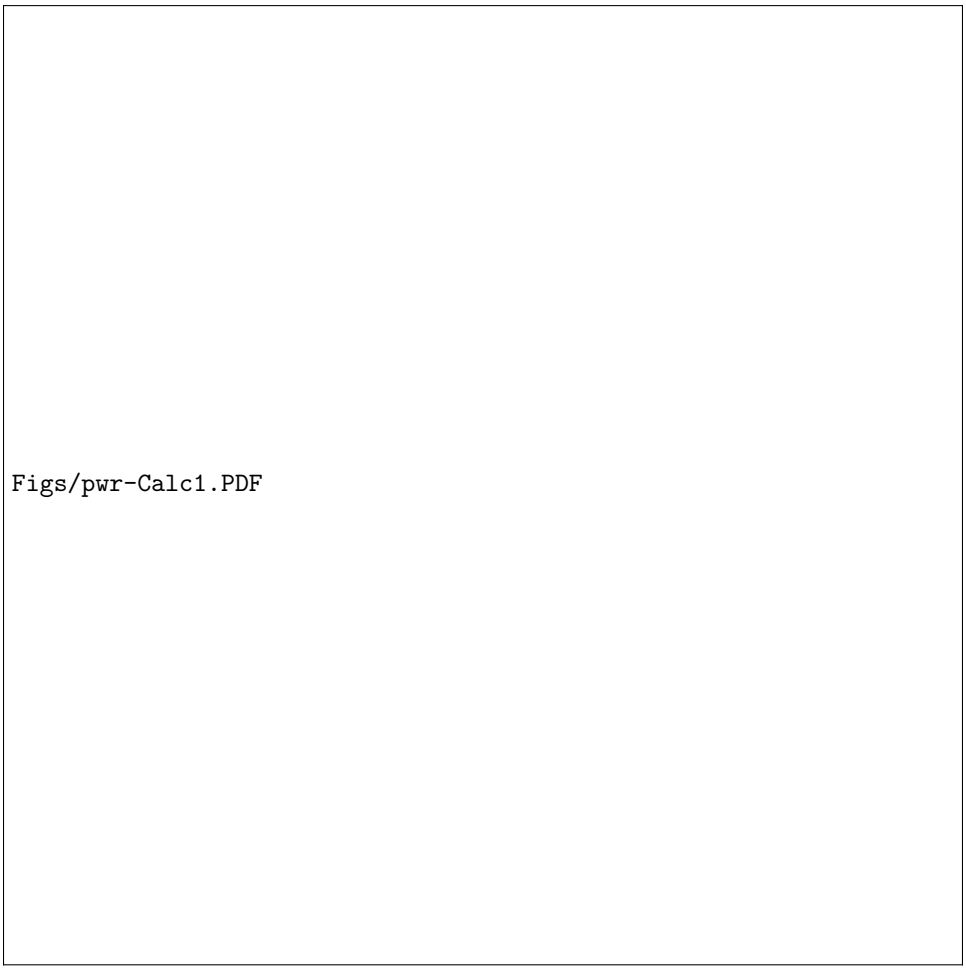
- *Type I error*: Rejecting H_0 when H_0 was actually true. Probability of Type I error is α.
- *Type II error*: Not rejecting H_0 when H_0 was actually false. Probability of Type II error is β.
- *Power*: The probability of correctly rejecting H_0 when H_0 was actually false.

Table 1. Types of decisions that can be made from hypothesis tests.

		Decision from Data	
		Reject	Not Reject
Truth About Population	H_0	Type I	Correct
	H_A	Correct	Type II

- *Crude Effect Size*: The difference between the unknown true parameter and the hypothesized parameter – e.g., $\mu - \mu_0$.
- *Standardized Effect Size*: The crude effect size scaled by the standard deviation – e.g., $\frac{\mu-\mu_0}{\sigma}$
- *Minimum (Crude) Effect Size*: The minimum crude effect size that will be considered significant at a given sample size (n), Type I error rate (α), and power. Sometimes called *minimum detectable difference*.

1.2 Visualization of a Hypothetical Power Calculation



1.3 Relationships between Errors, Power, and Effect Sizes

The `power.sim()` function provides sliders that can be used to determine the effect of changing the sample size (n), Type I error rate (α), and natural variability (σ) on power. Use the default settings of this function to explore the following questions.

1. What are H_0 and H_A?
2. What effect does increasing n have on power?
3. What effect does decreasing α have on power?
4. What effect does decreasing σ have on power?
5. What effect does increasing the crude effect size have on power?
6. Set the simulator on the following conditions – "Actual mu"=95, "sigma"=10, "n=24", "alpha=0.05" – and take note of the crude effect size and power. Then change to "n=66" and adjust the "Actual mu" value until you get a power value close to what was observed in the initial condition. Take note of the crude effect size. Comment on what effect increasing the sample size but maintaining the same power had on the crude effect size.

2 1-Sample t-Test

The `power.t.test()` function is a powerful function for computing (i) power given a sample size and crude effect size, (ii) sample size given a crude effect size and a power value, and (iii) crude effect size given a sample size and a power value. The `power.t.test()` function has the following arguments:

- `n`: Number of observations (per group)
- `delta`: True difference in means – i.e., crude effect size
- `sd`: Standard deviation – i.e., σ
- `sig.level`: Significance level (Type I error probability) – i.e., α (0.05 is default)
- `power`: Power of test (1 minus Type II error probability)
- `type`: Type of t-Test – either `"two.sample"` (*default*), `"one.sample"`, or `"paired"`.
- `alternative` Type of alternative hypothesise – either `"one-sided"` or `"two-sided"` (*default*).

You must supply `power.t.test()` with four of `n`, `delta`, `power`, `sd`, and `sig.level` with the fifth item being set to `NULL`. The function will then calculate the value of the null-set item given the values of the other four items.

2.1 Power Given a Sample Size and Crude Effect Size

The power to detect a crude effect size of 3 units given a sample size of 10, an α of 0.10, a σ of 10 can be computed with,

```
> power.t.test(n=10,delta=3,power=NULL,sig.level=0.1,sd=10,type="one.sample")

     One-sample t test power calculation

              n = 10
          delta = 3
             sd = 10
      sig.level = 0.1
          power = 0.2217
    alternative = two.sided
```

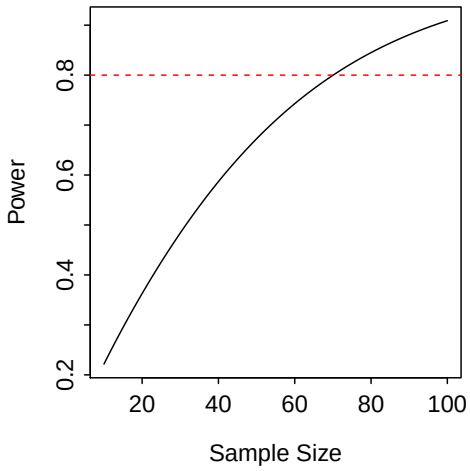
The power for the same situation but with sample sizes of 10, 15, and 20 can be computed with,

```
> power.t.test(n=c(10,15,20),delta=3,power=NULL,sig.level=0.1,sd=10,type="one.sample")

     One-sample t test power calculation

              n = 10, 15, 20
          delta = 3
             sd = 10
      sig.level = 0.1
          power = 0.2217, 0.2950, 0.3628
    alternative = two.sided
```

If one wanted to visually find the sample size required to reach a power of 0.80 they might try this,

```
> ns <- seq(10,100,1)
> p1 <- power.t.test(n=ns,delta=3,power=NULL,sig.level=0.1,sd=10,type="one.sample")
```

```
> plot(p1$power~p1$n,xlab="Sample Size",ylab="Power",type="l")
> abline(h=0.8,col="red",lty=2)
```



2.2 Sample Size Given a Power and Crude Effect Size

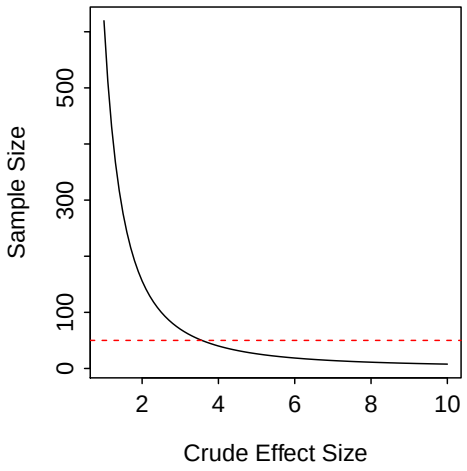
The sample size needed to detect a crude effect size of 3 with a power of 0.8 can be computed with,

```
> power.t.test(n=NULL,delta=3,power=0.8,sig.level=0.1,sd=10,type="one.sample")

     One-sample t test power calculation

              n = 70.07
          delta = 3
             sd = 10
      sig.level = 0.1
          power = 0.8
    alternative = two.sided
```

Compare this to the visual estimate in the previous section.

Suppose that you wanted to visualize the effect of crude effect size on required sample size and you could only afford to take a sample of n=50. Try this,

```
> deltas <- seq(1,10,by=0.1)
> p2 <- sapply(deltas,power.t.test,n=NULL,power=0.8,sig.level=0.1,sd=10,type="one.sample")
> p2.n <- as.numeric(p2["n",])
> p2.delta <- as.numeric(p2["delta",])
> plot(p2.n~p2.delta,xlab="Crude Effect Size",ylab="Sample Size",type="l")
> abline(h=50,col="red",lty=2)
```



2.3 Crude Effect Size Given a Sample Size and Power

The minimum detectable crude effect size given a fixed sample size of 50 and a power of 0.8 can be computed with,

```
> power.t.test(n=50,delta=NULL,power=0.8,sig.level=0.1,sd=10,type="one.sample")

     One-sample t test power calculation

              n = 50
          delta = 3.566
             sd = 10
      sig.level = 0.1
          power = 0.8
    alternative = two.sided
```

Compare this to the visual estimate in the previous section.

3 2-Sample t-Test

Computation of the power, sample size, or effect size in a two-sample t-test is accomplished in the same manner as described for a one-sample t-test. In this case, the effect size is the difference in means between the two independent groups and the sample size is the number of individuals in each group. Note that this function requires an equal sample size in both groups. For example, the power to detect a difference in means of four units using a sample size of 20 individuals in each group (using a common σ of 10 and an α of 0.05) is computed with,

```
> power.t.test(n=10,delta=4,power=NULL,sig.level=0.05,sd=10,type="two.sample")

     Two-sample t test power calculation

              n = 10
          delta = 4
             sd = 10
      sig.level = 0.05
          power = 0.133
    alternative = two.sided

 NOTE: n is number in *each* group
```

3.1 Ability to Detect a Significant Decline

Suppose that a population was sampled in 2000 and the abundance of animals was 100 per square mile with a standard deviation of 25 animals per square mile. Now suppose that your are part of a committee that has determined that if the population had declined to less than 80 animals in 2008 that you will have to take strong measures to support this population. As the biometrist on this committee you are asked to determine the minimum sample size that will allow you to detect a population abundance of at least this size with 80% power with no more than a 10% chance of a Type I error. You return to the committee within one minute after having performed the following calculation,

```
> power.t.test(n=NULL,delta=20,power=0.8,sig.level=0.1,sd=25,type="two.sample",alt="one.sided")

     Two-sample t test power calculation

              n = 14.53
          delta = 20
             sd = 25
      sig.level = 0.1
          power = 0.8
    alternative = one.sided

 NOTE: n is number in *each* group
```

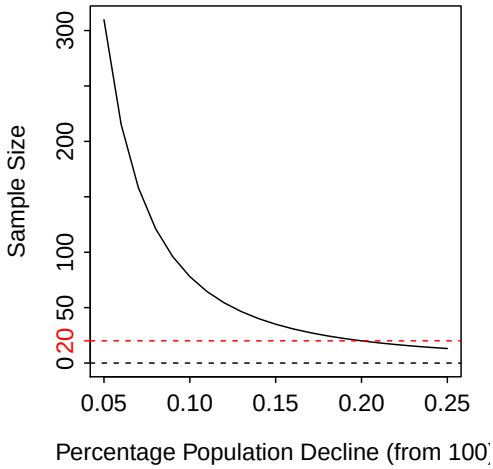
On second thought, suppose the committee wants to detect a 10% drop in population abundance (with all the other conditions remaining the same). You return after making this calculation,

```
> delta1 <- 100*0.10
> power.t.test(n=NULL,delta=delta1,power=0.8,sig.level=0.05,sd=25,type="two.sample",alt="one.sided")

     Two-sample t test power calculation
```

```
              n = 77.97
          delta = 10
             sd = 25
      sig.level = 0.05
          power = 0.8
    alternative = one.sided

 NOTE: n is number in *each* group
```

On third thought, the committee would like to know what type of sample size it would take to detect a variety of percentage drops in population abundance. In particular, though, they are interested in detecting a 20% drop. You then return with the graph produced by the following code,

```
> perc.delta <- seq(0.05,0.25,0.01)
> deltas <- 100*perc.delta
> p3 <- sapply(deltas,power.t.test,n=NULL,power=0.8,sig.level=0.05,sd=25,type="two.sample",alt="one.sid
> p3.n <- as.numeric(p3["n",])
> plot(p3.n~perc.delta,xlab="Percentage Population Decline (from 100)",ylab="Sample Size",type="l",ylim=
> abline(h=0,lty=2)
> abline(h=p3.n[perc.delta==0.2],lty=2,col="red")
> axis(2,p3.n[perc.delta==0.2],round(p3.n[perc.delta==0.2],0),col.axis="red",col="red")
```



4 1-Way ANOVA

Computing the power in a one-way ANOVA is quite different as there is more than two groups, the "nuisance" variability is measured by MS_{within}, and the "effect size" is related to MS_{among}. As with the two-sample t-test it is assumed that the one-way design is "balanced" – i.e., equal numbers of individuals in each group.

The power, sample size, or "effect size" for a one-way ANOVA can be determined with `power.anova.test()`. This function has the following arguments:

- `groups`: Number of groups – i.e., I
- `n`: Number of observations (per group)
- `between.var`: Between group variance – i.e., MS_{among}
- `within.var`: Within group variance – i.e., MS_{among} or s_p^2.
- `sig.level`: Significance level (Type I error probability) – i.e., α (0.05 is default)
- `power`: Power of test (1 minus Type II error probability)

As with `power.t.test()` one of the arguments, excluding `groups`, must be `NULL` such that `power.anova.test()` will solve for that argument.

4.1 Sample Size Given Power and Variances

The following can be used to compute the required sample size to detect a between group variance of 3 units with a within group variance of 1.5 units, a power of 0.8, and four groups.

```
> power.anova.test(groups=4,n=NULL,between.var=3,within.var=1.5,power=0.8,sig.level=0.05)

     Balanced one-way analysis of variance power calculation

         groups = 4
              n = 2.996
    between.var = 3
     within.var = 1.5
      sig.level = 0.05
          power = 0.8

 NOTE: n is number in each group
```

It may be difficult to visualize what a particular MS_{among} value represents. Thus, one can enter in a priori "guesses" at the group means, use `var()` to estimate MS_{among}, and then use this in `power.anova.test`. This is illustrated below,

```
> mns <- c(100,120,120,130)
> msA <- var(mns)
> power.anova.test(groups=length(mns),n=NULL,between.var=msA,within.var=500,power=0.8,sig.level=0.05)

     Balanced one-way analysis of variance power calculation

         groups = 4
              n = 12.5
    between.var = 158.3
     within.var = 500
      sig.level = 0.05
          power = 0.8

 NOTE: n is number in each group
```

4.2 Effect Size Given Power, Sample Size, and MS_{within}

The minimum effect size among four groups that can be detected if power is 0.8, the sample size is 5 per group, and MS_{within} is 500 can be computed with,

```
> power.anova.test(groups=4,n=5,between.var=NULL,within.var=500,power=0.8,sig.level=0.05)

     Balanced one-way analysis of variance power calculation

         groups = 4
              n = 5
```

```
    between.var = 465.1
     within.var = 500
      sig.level = 0.05
          power = 0.8

NOTE: n is number in each group
```

5 Linear Trends

A recent message on the R help page provided a simple function for computing the power of detecting a particular slope in a regression analysis. A modification of this function is shown below,

```
> power.trend <- function(x,y,sd.y,slope,sig.level=0.05){
    ncp <- slope^2 * sum((x - mean(x))^2)/sd.y^2
    return(1-pf(qf(1-sig.level,1,length(x)-2),1,length(x)-2,ncp=ncp))
 }
```

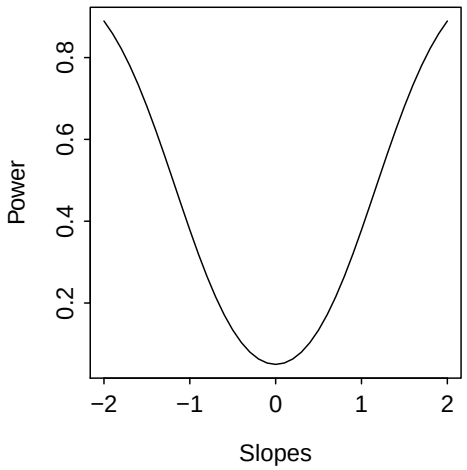
For example, with the following data (d is the response variable)

```
> t <- 1:6
> d <- c(303, 302, 304, 306, 307, 303)
```

the power to detect slopes of 0.5 and 1.75 is computed with,

```
> power.trend(x=t,sd=sd(d),slope=0.5)
[1] 0.134
> power.trend(x=t,sd=sd(d),slope=1.75)
[1] 0.8021
```

```
> slopes <- seq(-2,2.0,0.1)
> p5 <- sapply(slopes,power.trend,x=t,y=d,sd.y=sd(d))
> plot(p5~slopes,xlab="Slopes",ylab="Power",type="l")
```



6 Simulation and Power

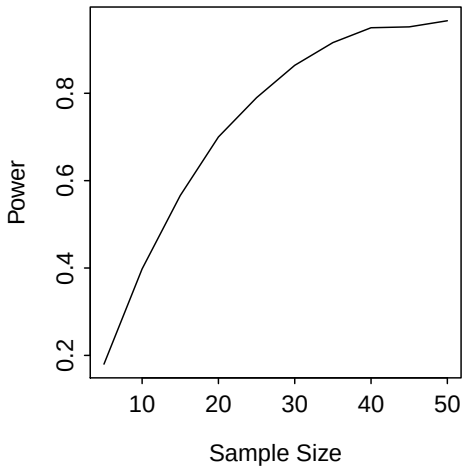
6.1 Two-Sample t-test

```
> alpha <- 0.05
> sdA <- sdB <- 25
> delta <- 20
> muA <- 0
> muB <- muA+delta
> nA <- nB <- 20
> nsims <- 500
> diff.pval <- numeric(nsims)
> for (i in 1:nsims) {
   smplA <- rnorm(nA,muA,sdA)
   smplB <- rnorm(nB,muB,sdB)
   diff.pval[i] <- t.test(smplA,smplB,var.equal=TRUE)$p.value
 }
> power.2t.1 <- sum(diff.pval < alpha)/nsims
> power.2t.1
[1] 0.684
```

Compare this to

```
> power.t.test(n=nA,delta=delta,sd=sdA,power=NULL,sig.level=alpha,type="two.sample",alt="two.sided")

     Two-sample t test power calculation

              n = 20
          delta = 20
             sd = 25
      sig.level = 0.05
          power = 0.6934
    alternative = two.sided

 NOTE: n is number in *each* group
```

```
> ns <- seq(5,50,5)
> power.2t.2 <- numeric(length(ns))
> for (j in 1:length(ns)) {
   nA <- nB <- ns[j]
   diff.pval <- numeric(nsims)
   for (i in 1:nsims) {
     smplA <- rnorm(nA,muA,sdA)
     smplB <- rnorm(nB,muB,sdB)
     diff.pval[i] <- t.test(smplA,smplB,var.equal=TRUE)$p.value
   }
   power.2t.2[j] = sum(diff.pval < alpha)/nsims
 }
> plot(power.2t.2~ns,xlab="Sample Size",ylab="Power",type="l")
```



6.2 Linear Trends

Suppose that you are monitoring a population that had an estimated abundance of 100 individuals in 2000. You plan to monitor this population for the next 10 years and want to determine what type power you can obtain for different potential slopes and variabilities around the line. In particular, you are concerned about being able to detect a decrease of 10 animals per year.

First, declare some variables,

```
> nsims <- 500
> alpha <- 0.05
> x <- 0:10
> a <- 100
```

Second, set the values that will change and prepare the data structures,

```
> bs <- seq(-15,0,1)
> s.yxs <- seq(20,100,10)
> power.1 <- matrix(nrow=length(bs),ncol=length(s.yxs))
> pval <- numeric(nsims)
```

Third, perform the simulations

```
> for (i in 1:length(bs)) {
>   for (j in 1:length(s.yxs)) {
>     for (k in 1:nsims) {
>       y.mn = a+bs[i]*x
>       y <- y.mn + rnorm(length(y.mn),0,s.yxs[j])
>       m <- lm(y~x)
>       pval[k] <- coef(summary(m))["x", "Pr(>|t|)"]
>     }
>     power.1[i,j] <- sum(pval < alpha)/nsims
>   }
> }
```

Fourth, observe the obtained power values as a table,

```
> rownames(power.1) <- bs
> colnames(power.1) <- s.yxs
> round(power.1,2)
```

Fifth, observe the obtained power values as a graphic,

```
> contour(x=bs,y=s.yxs,z=power.1,xlab="Slopes",ylab="Variabilities")
Warning:  no non-missing arguments to min; returning Inf
Warning:  no non-missing arguments to max; returning -Inf
Error:  no contour values
> contour(x=bs,y=s.yxs,z=power.1,levels=0.8,add=TRUE,col="red",lwd=2)
Warning:  all z values are NA
> abline(v=10,col="red",lwd=2,lty=2)
> abline(v=-10,col="red",lwd=2,lty=2)
```