

Notes for Stat 430 – Design of Experiments – Lab #7 Oct. 29, 2012.

Agenda:

- ANOVA Power Calculations with Exercise 3.54.
- ANOVA Power and Sample Size with Exercise 3.34.
- Blocking factors and ANOVA, Exercise 4.3.
- (Extra) Non-Centrality Parameter 3.54
- (Extra) Non-Centrality Parameter Simulation

Useful files: http://www.sfu.ca/~jackd/Stat430/Wk07_Code.txt

All lab material found at : <http://www.sfu.ca/~jackd/Stat430/>

Power Calculations, Exercise 3.54

Start with some data from Exercise 3.54, 8th edition hardcover.

```
loom = rep(1:5, each=5)
loom = as.factor(loom)
yield =
c(14,14.1,14.2,14,14.1,13.9,13.8,13.9,14,14,14.1,14.2,14.1,
14,13.9,13.6,13.8,14,13.9,13.7,13.8,13.6,13.9,13.8,14)
data354 = data.frame(loom, yield)
```

We can use `anova()` to get the MS.Error and MS.Treatment

```
mod354 = lm(yield ~ loom, data=data354)
anova(mod354)
```

We can also calculate these values directly for use in other things with a shortened version of some code from Week 6.

```
y = data354$yield
trt = data354$loom
```

Get the basics

```
N = length(y) ## N is the total number of observations
a = length(unique(trt)) ## a is the total number of different treatments
n.pergroup = N / a ## Replications per group
```

Get the group means and grand mean

```
yi._bar = by(y, trt, mean)
y._bar = mean(y)
```

Get the Mean Squared values

```
SS.Total = sum( (y - mean(y))^2 )
SS.Treatment = n.pergroup*sum( (yi._bar - y._bar)^2)
SS.Error = SS.Total - SS.Treatment
```

```
MS.Treatment = SS.Treatment / (a-1)
MS.Error = SS.Error / (N - a)
```

We can find the power of this result with `power.anova.test()`

That is, if this were the true MS.Treatment and MS.Error how often would the result show up as significant?

- groups: number of groups
- n : number of observations per group
- between.var : variance of group means
- within.var : variance within groups (MS error)
- sig.level : type I error chance, alpha
- power : 1 - beta

```
power.anova.test(groups = a, n=n.pergroup,
between.var=var(yi._bar), within.var=MS.Error, sig.level=0.05)
```

(Theory for this continued in extra material on non-centrality parameters)

Power Calculations, Exercise 3.34

Data. Note there is a typo in the title of the data table; it should read "bleach" for the factor and "brightness" for the response.

```
bleach = rep(1:4,each=5)
bleach = as.factor(bleach)
bright = c(77.199,74.466, 92.746, 76.208, 82.876, 80.522,
79.306, 81.914, 80.346, 73.385, 79.417, 78.017, 91.596, 80.802,
80.626, 78.001, 78.358, 77.544, 77.364, 77.386)
data334 = data.frame(bleach, bright)
```

Start with an ANOVA to get the MS, SS, and p values.

```
mod334 = lm(bright ~ bleach, data=data334)
anova(mod334)
```

Now let's see the power

```
y = data334$bright
trt = data334$bleach
yi._bar = by(y, trt, mean)
MS.Error = 23.999
power.anova.test(groups = 4, n=5, between.var=var(yi._bar),
within.var=MS.Error, sig.level=0.05)
```

How big would each group need to be to detect such a difference 90% of the time without sacrificing alpha?

```
power.anova.test(groups = 4, power=0.9,
between.var=var(yi._bar), within.var=MS.Error, sig.level=0.05)
```

Blocking factors and ANOVA (Chapter 4 Material)

A blocking factor is a factor we should account for but either don't care about or can't control. R treats blocking factors and treatment factors (the ones we DO care about) the same way. We simply run an anova with two factors.

To do that, we'll first need some two-factor data, like that of exercise 4.3, 8th ed. hardcover. We're looking at the strength (response) of various chemicals (treatment factor) on cloth. ## However, the cloth doesn't come in a large enough bolt to do all our strength tests together, so we use several bolts and each is considered a block.

```
chemical = rep(1:4, each=5)
chemical = as.factor(chemical)
bolt = rep(1:5, times=4)
bolt = as.factor(bolt)
strength =
c(73, 68, 74, 71, 67, 73, 67, 75, 72, 70, 75, 68, 78, 73, 68, 71, 71, 75, 75, 69)
data403 = data.frame(chemical, bolt, strength)
```

From this data, we'll build a linear model with two factors

The model is `<response> ~ <factor1> + <factor2>`

The order of the factors doesn't matter

```
mod403 = lm(strength ~ chemical + bolt, data=data403)
summary(mod403)
```

The intercept is the fitted value of the strength of cloth from bolt 1 treated with chemical 1. It isn't exactly the same as the one observation that fits this because the linear model does not fit this data perfectly.

Note that the estimated effect of chemicals 2, 3, and 4 (as compared to chemical 1) are 0.8, 1.8, and 1.6 respectively. If we look at the group means, grouping by chemical, we can see that the means are 0.8, 1.8, and 1.6 larger than the first mean. The same pattern would arise from grouping by bolt.

```
by(strength, chemical, mean)
```

Now for the ANOVA

```
anova(mod403)
```

Response: Strength

	DF	Sum Squared	Mean Squared	F	p-value
Chemical	3	10.15	3.383	1.3716	0.2985
Bolt	4	150.80	37.700	15.2838	0.0001174
Error	12	29.60	2.467		
Total	19	190.55			

Some observations specific to this data:

1. Chemical was not significant at $\alpha = 0.05$ because its F-statistic was not larger than the F-critical for 3 and 12 degrees of freedom.
2. Bolt was significant. The variance between bolts of cloth does matter, even if the type of chemical does not. Bolt's F-statistic was compared to the F-critical for 4 and 12 degrees of freedom.
3. The multiple r-squared is $(10.15 + 150.80) / 190.55 = 0.8447$. The remaining 15.53% of the variance is either randomness or some interaction, which we can check graphically (covered in a later lab).

Observations about the ANOVA Table:

1. The total sample was 20, so there are 19 degrees of freedom in total
- There were 4 types of chemical tested, so the chemical factor costs 3 df.

- There were 5 bolts (blocks) used, which cost 4 degrees of freedom.
- The remaining $19 - 4 - 5 = 12$ degrees of freedom go to the residual error.

2. The sum of squared total is total of the other sum-of-squares.

- Most of the variance in the strength ended up being explained by the differences between bolts, and not between the factor we can actually control, chemical.

3. Mean squared is always (sum squared) / (degrees of freedom)

4. The F value is always (mean squared of this) / (mean squared of error)

Finally, the raw amount of variance (Sum of squared) explained by a factor is unaffected by the other factors in the model. To show this, we'll do the ANOVA again but without the bolt factor.

```
mod.simple = lm(strength ~ chemical, data=data403)
anova(mod.simple)
```

The SS.Treatment is the same as before. However, the F-statistic and the p-value have changed. Why?

(Extra) Non-Centrality Parameters – Exercise 3.54

```
power.anova.test(groups = a, n=n.pergroup,
between.var=var(yi._bar), within.var=MS.Error, sig.level=0.05)
```

What `power.anova.test` is calculating is the proportion of the non-central F distribution based around the F statistic we observe...

```
Real.F = MS.Treatment / MS.Error
Real.F
```

... that is higher than the critical value for the alpha and degrees of freedom given

```
critical = qf(.95, df1=(a-1), df2=(N-a))
critical
```

The proportion of the non-central F beyond that is

```
lambda = SS.Treatment / MS.Error
```

```
pf(x=critical, df1=(a-1), df2=(N-a), ncp=lambda,  
lower.tail=FALSE)
```

Where lambda is the *non-centrality parameter*

Note that this area beyond the critical value is the same as the power found in
`power.anova.test()`

The non-central F is like the usual F distribution, but the non-central F assumes there is a difference in the true means based on the between and within variances observed. The central F assumes there is no difference between the group means.

For comparison:

```
curve( df(x, df1=(a-1), df2=(N-a)), lwd=2, to=0, from=15)  
curve( df(x, df1=(a-1), df2=(N-a), ncp=lambda), lwd=2,  
to=0, from=15, col="Red", add=TRUE)  
abline(v = critical, col="Blue", lwd=2) ## Draws a vertical line at the critical  
point calculated before, adds automatically
```

The non-central (using the calculated non-central parameter above) is in red.
Higher F values mean a higher MS.Treatment to MS.Error ratio.

More on the calculation of lambda, the non-centrality parameter, is available at
<http://digitalcommons.calpoly.edu/cgi/viewcontent.cgi?article=1002&context=statsp>

(Extra) Non-Centrality Parameter - Simulation

Done via simulation (See Casella and Berger for a proof of the relationship between normal, chi-sq, and F distributions)

Power calculations in ANOVA use the non-central F-distribution.
The non-centrality comes from the chi-squared distribution.

The chi-squared distribution is the sum on independent standard normals squared. We can generate the chi-squared by randomly generating standard normals, squaring them and adding them.

```

chi2 = rep(NA,10000)
for(k in 1:10000)
{
  z = rnorm(n=10, mean=0, sd=1)
  chi2[k] = sum(z^2)
}

```

We summed 10 squared standard normals for each chi-squared, so the degrees of freedom is 10. We can demonstrate this assertion with the curve() function.

```

hist(chi2,prob=TRUE)
curve(dchisq(x,df=10),from=0,to=max(chi2), add=T, lwd=2)

```

Now let's bring the non-centrality parameter in. Instead of squaring the standard normal, let's square the normal with standard deviation of 1 but a mean of some value other than 0.

```

chi2.zmu1 = rep(NA, 10000)
chi2.zmu2 = rep(NA, 10000)
chi2.zmu3 = rep(NA, 10000)
for(k in 1:10000)
{
  z.mu1 = rnorm(n=10, mean=1, sd=1)
  z.mu2 = rnorm(n=10, mean=2, sd=1)
  z.mu3 = rnorm(n=10, mean=3, sd=1)

  chi2.zmu1[k] = sum(z.mu1^2)
  chi2.zmu2[k] = sum(z.mu2^2)
  chi2.zmu3[k] = sum(z.mu3^2)
}

```

The mean of the central chi-squared is its degrees of freedom

```
summary(chi2)
```

The mean of the non-central chi-squared is $(1 + \mu)^2 * (\text{degrees of freedom})$

Where μ is the mean of the underlying normals behind the non-central chi-squared.

```

summary(chi2.zmu1)
summary(chi2.zmu2)
summary(chi2.zmu3)

```

For the sake of later use, the ncp, non-centrality parameter, is written in terms of how much the non-zero mu values increase the mean of the chi-sq distribution.

So the sum of 10 normals with mu = 1 produce a value from the chi-squared, df=10, non-centrality = 10

```
hist(chi2.zmu1,prob=TRUE)
curve(dchisq(x,df=10,ncp=10),from=0,to=max(chi2.zmu1), add=T,
lwd=2)
```

The sum of 10 normals with mu = 2 produce a value from the chi-squared df=10, ncp=40

```
hist(chi2.zmu2,prob=TRUE)
curve(dchisq(x,df=10,ncp=40),from=0,to=max(chi2.zmu2), add=T,
lwd=2)
```

The sum of 10 normals with mu = 3 produce a value from the chi-squared df=10, ncp=90

```
hist(chi2.zmu3,prob=TRUE)
curve(dchisq(x,df=10,ncp=90),from=0,to=max(chi2.zmu3), add=T,
lwd=2)
```

Anything that uses the chi-squared statistic then also uses this non-centrality parameter, but in cases you're familiar with it's set to its default of zero.

Recall that the F distribution is ratio of two chi-squares, each divided by their degrees of freedom. Including a non-centrality parameter changes the numerator chi-squared distribution.

Since including a non-centrality parameter of lambda increases the mean of the chi-squared by one by spreading the distribution farther to the right, including an NCP of lambda in an F distribution increases the mean of the F-distribution by lambda / (Numerator df)

Although it isn't considered here, the T-Distribution also uses a chi-square, so it also has a non-centrality parameter.

```
curve(df(x, df1=3, df2=10), lwd=2, from=0, to=12)
curve(df(x, df1=3, df2=(N-a), ncp=2), from=0, to=12, add=TRUE)
curve(df(x, df1=3, df2=(N-a), ncp=5), from=0, to=12, add=TRUE)
curve(df(x, df1=3, df2=(N-a), ncp=10), from=0, to=12, add=TRUE)
curve(df(x, df1=3, df2=(N-a), ncp=20), from=0, to=12, add=TRUE)
```