Notes for Stat 430 – Design of Experiments – Lab #6 Oct. 22, 2012.

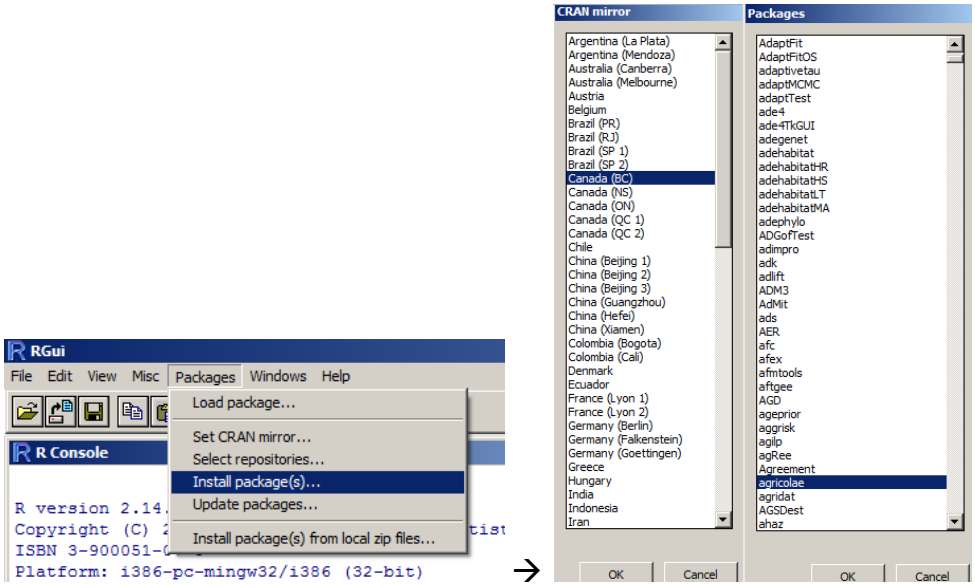
Agenda:

- Installing and using packages
- Multiple Testing with Fisher LSD
- Multiple Testing with Tukey HSD
- Randomization Test for ANOVA

Useful files: http://www.sfu.ca/~jackd/Stat430/Wk06_Code.txt
All lab material found at : http://www.sfu.ca/~jackd/Stat430/

Installing and Using Packages

To install packages, in the top bar menu go to:
Packages --> Install Package(s)
Then Choose a server to download from (Canada BC is at the SFU Burnaby , pick it)
Then choose the packages you want. Today we'll need "**agricolae**", it takes a little scrolling down to find.



Installing only has to be done once (unless you're on a self-restoring computer like those in the library). Loading the package has to be done everytime R is opened. It's generally good form to include the load command in R code you send to someone else. That way they know what package they need to install, and after it's installed they don't have to worry about it being loaded.

To load the package, use this after it's been installed

```
library(agricolae) # adds "agricolae" to the library
```

This is essentially the same thing you do when loading a pre-made dataset

```
data(InsectSprays)  # build-in, not part of agricolae package
```

Mutliple Testing with Fisher's LSD (Page 99-100)

First, we'll need a data set. This is from Exercise 3.27.
It's the concentration of a chemical after beign exposed to one of four catalysts. Note that not every sample is the same size.

```
conc =
c(58.2,57.2,58.4,55.8,54.9,56.3,54.5,57.0,55.3,50.1,54.2,55.4,52
.9,49.9,50.0,51.7)
catl = c(1,1,1,1,1,2,2,2,2,3,3,3,4,4,4,4)
catl = as.factor(catl)
data327 = data.frame(catl,conc)
```

The command for Fisher's LSD (Least Significant Difference) is part of the agricolae package
So make sure agricolae is installed and loaded first.

You need to put a model into the `LSD.test()` function. But either `aov()` or `lm()` works.

```
mod1 = aov(conc ~ catl, data=data327)
LSD.test(mod1,"catl")
```

`lm()` will give you the same results as `aov()`

```
mod2 = lm(conc ~ catl, data=data327)
LSD.test(mod2,"catl")
```

This does a t-test on each possible pair.
It needs an ANOVA or linear model to get the MSE, which it uses as the variance estimate for every pair instead of getting the pooled variance for each one. This way the variance is pooled among ALL groups and you get N - a degrees of freedom.

One issue is that it's doing 4 choose 2, or 6 distinct t-tests.
If each one has a false positive rate of alpha = 0.05, then the chance of at least one of the t-tests giving a false positive is much higher than 0.05.

How much higher the collective false positive chance depends on the assumption of the t-tests being independent.

However, we can use a multiple comparison adjustment to reduce the chances of or there being a false positive to close to that of alpha. The simplest way is the Bonferroni method.

```
LSD.test(mod1,"catl",p.adj="bonferroni")
```
Only two distinct groups are found now, instead of three.

The Bonferroni method reduces the p-value of each test by a factor of the number of tests. There are six tests, so the p-value is reduced to 0.05 / 6 here.
Under the assumpting of independent tests, the expected number of false positives is alpha.

There are other methods, but I suggest you read up on them before using them in case you need to justify your decisions later.

```
LSD.test(mod1,"catl",p.adj="fdr")  #False discovery rate
LSD.test(mod1,"catl",p.adj="holm") # Holm
LSD.test(mod1,"catl",p.adj="BH")  # Bonferroni-Holm
```

Multiple Testing with Tukey HSD (Page 98)

HSD stands for Honest Significant Difference. It's similar to the LSD, but has a built it correction for multiple testing that doesn't require an adjustment to the p-value.

```
HSD.test(mod1,"catl")
```

It's set up and interpreted the same way as the LSD method. It doesn't require a package, but the agricolae package does offer its own version of it.

With both the HSD and LSD methods you can change the per-test type 1 error rate with `alpha=`

```
HSD.test(mod1,"catl",alpha=0.01)
LSD.test(mod1,"catl",alpha=0.10)
```

Randomization Test for ANOVA (Page 77-78)

Here we use code from the randomization test in lab 2 and expand it to account for more than 2 groups.

First, we need the means every group.
One fast way to do that is with the by() command
by (response variable, grouping variable, function)

```
real.means = by(data327$conc, data327$catl, mean)
```

The general plan for the randomization test is:

- Get the basic values like sample size, y.. and yi.
- Get the SS_{Total} and $SS_{Treatment}$
  For each of many runs...
  - Randomly assign the treatments to the responses
  - Get the $SS_{Treatment}$ for each response

  (end for loop)
- The proportion of randomizations that have a bigger $SS_{Treatment}$ than the observed $SS_{Treatment}$. *This is equivalent to finding the proportion that have a bigger F-value than the observed F-value, which we can prove or show by graphing.*

Feel free to copy paste this entire section, as it's fairly complex.

```
# Get the basic values
y = data327$conc
trt = data327$catl

# Use the unique() command to make a list of the different values of something.
N = length(y)  ## N is the total number of observations
a = length(unique(trt))  ## a is the total number of different treatments
y.. = sum(y)
Real.SS.Total = sum( (y - mean(y))^2 )
```

Get the SS_{Total} and $SS_{Treatment}$ from (3.8) and (3.9)on page 74 respectively.
Note that formula (3.9) is modified because of the unequal sample sizes

```
yi. = by(y, trt, sum)
ni  = by(y, trt, length)
yi._bar = yi. / ni
y.._bar = mean(y)
Real.SS.Treatment = sum( ni*(yi._bar - y.._bar)^2)
Real.SS.Error = Real.SS.Total - Real.SS.Treatment
```

For each of many runs...

```
rnd.SS.Treatment = rep(NA,10000)
for(run in 1:10000)
{
      #Randomly assign the treatments to the responses
      random.trt = sample(trt)

      # Get the SS.treatment for each response
      yi. = by(y, random.trt, sum)
      yi._bar = yi. / ni
      rnd.SS.Treatment[run] = sum( ni*(yi._bar - y.._bar)^2)
}   # (end for loop)
```

Get the proportion of randomizations that have a bigger SS.treatment
#Make a list of the runs that produced a larger SStreatment than the observed one

```
random.beat.real = which(rnd.SS.Treatment > Real.SS.Treatment)
```

Find how many there are out of 10000

```
p = length(random.beat.real) / 10000
```

p is the approximate p-value, you can compare it to the ANOVA test.

```
anova(mod1)
```

We can also get the F-Stat from each ss.treatment compare to f-dist

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

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

How does our random F compare to the density curve?

```
hist(rnd.F, freq=FALSE, xlim=c(0,6),n=80)
curve(df(x, df1=(a-1), df2=(N-a)),from=0,to=6,add=TRUE,lwd=2)
```

Graphical Evidence that in a one-factor ANOVA, F is is strictly increasing over SS.Treatment

```
plot(rnd.F ~ rnd.SS.Treatment)
```