# An R Survival Guide for EPRS 8530, EPRS 8540 & EPRS 8550

September, 2013


Tianna Floyd

Dr. Chris Oshima

Georgia State University

College of Education

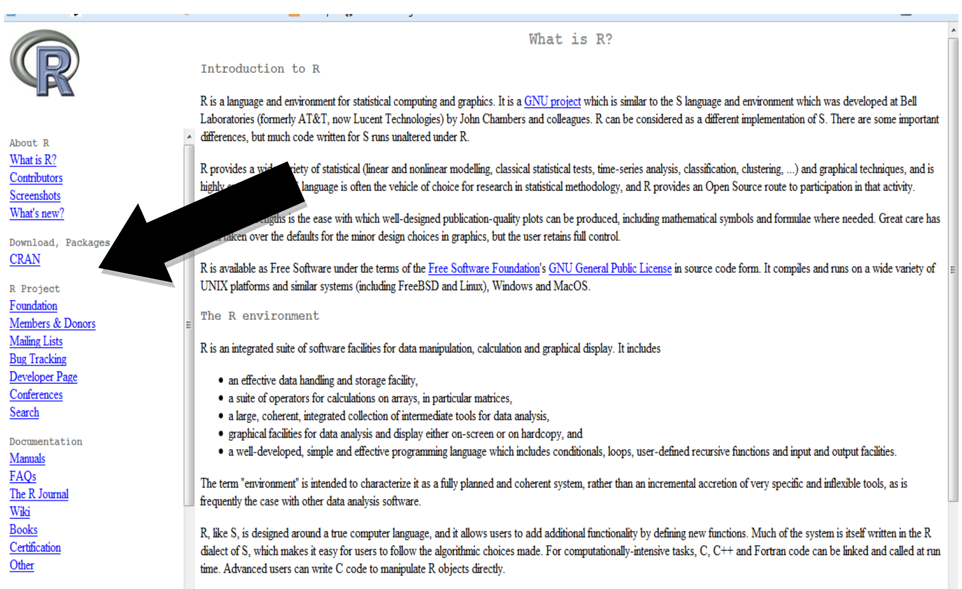
# Getting Started with R

## Overview

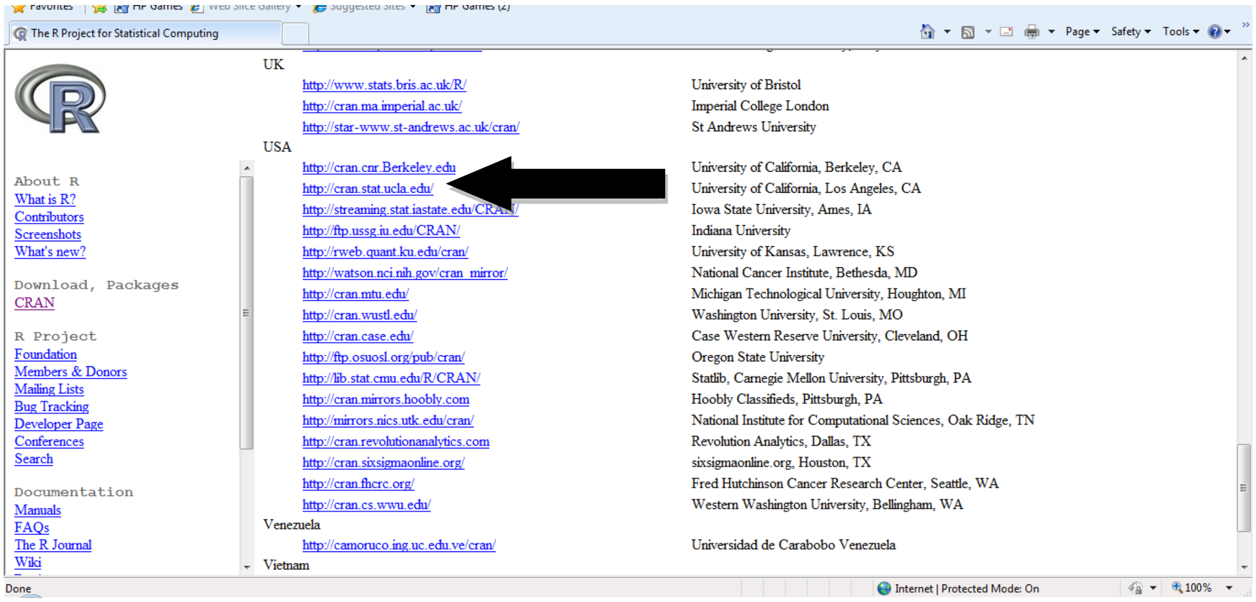
R is a free statistical software that runs on a variety of platforms including UNIX, Windows and MacOS. It was developed in 1991 in New Zealand by Ross Ihaka and Robert Gentleman. This software offers a variety of statistical and graphing techniques as well as producing quality plots that can be used in publications. R is a programming language that is similar to the S language developed in Bell Laboratories. This guide is written for those with limited computer programming knowledge. Some say there is a steep learning curve for R but once you get the hang of it, it is pretty easy.

## Downloading R

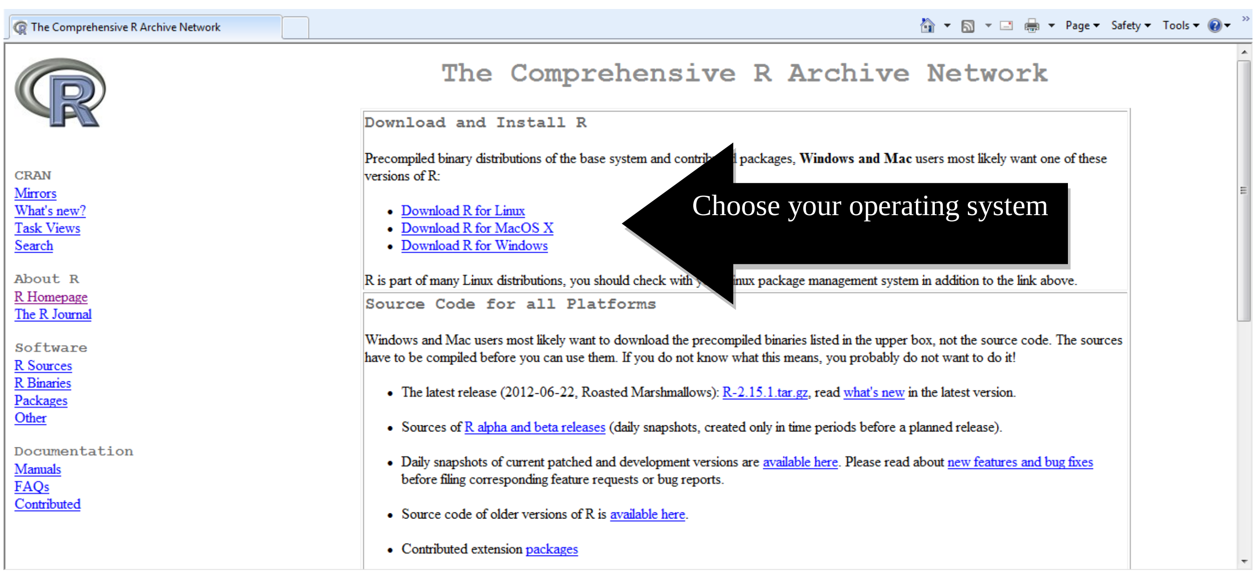
R is available at http://www.r-project.org/. Once you navigate to this website, click on CRAN under download packages.



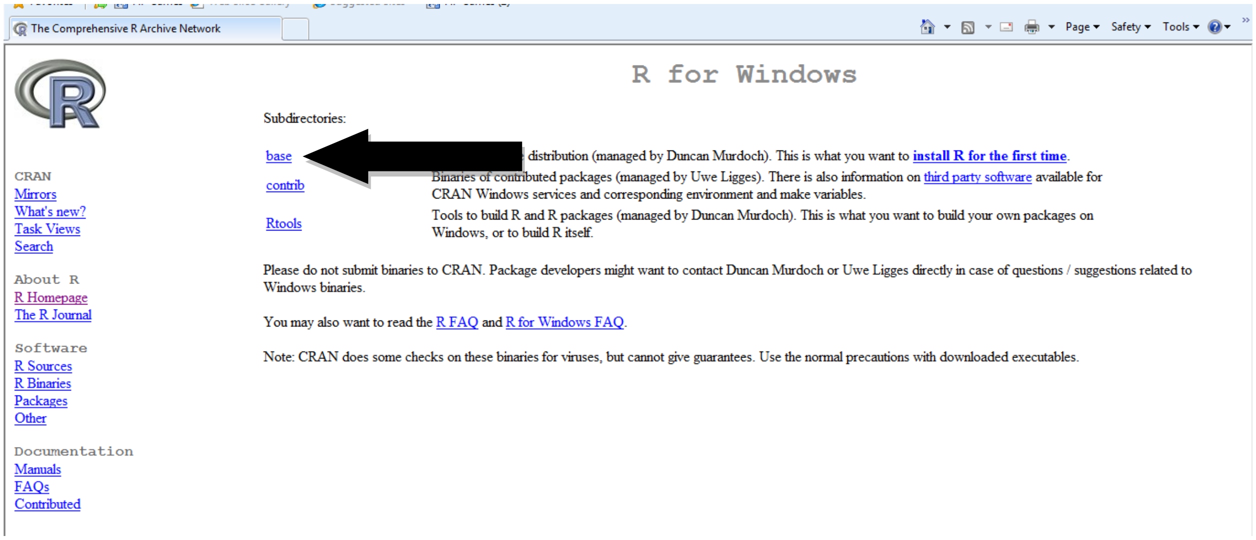

Next scroll down to USA and select a download site. Any should work but for this example we will choose University of California, Los Angeles, CA.




You will now be directed to the UCLA site to download R. Choose your operating system.

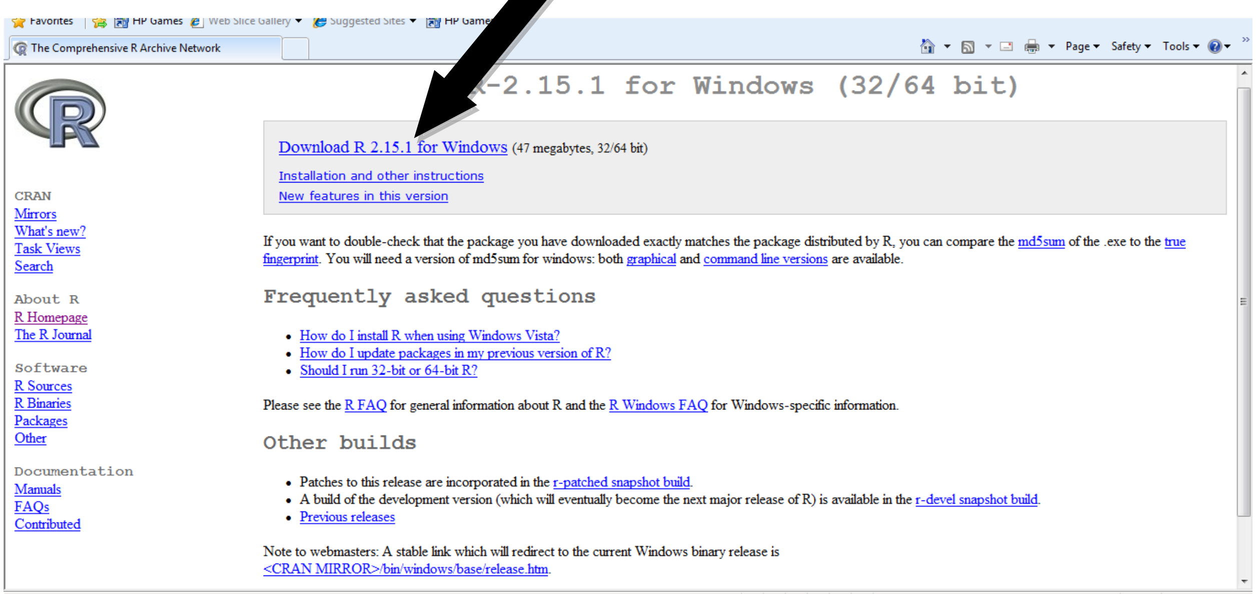



Since this will be your first time downloading R, you want to choose the base subdirectory.

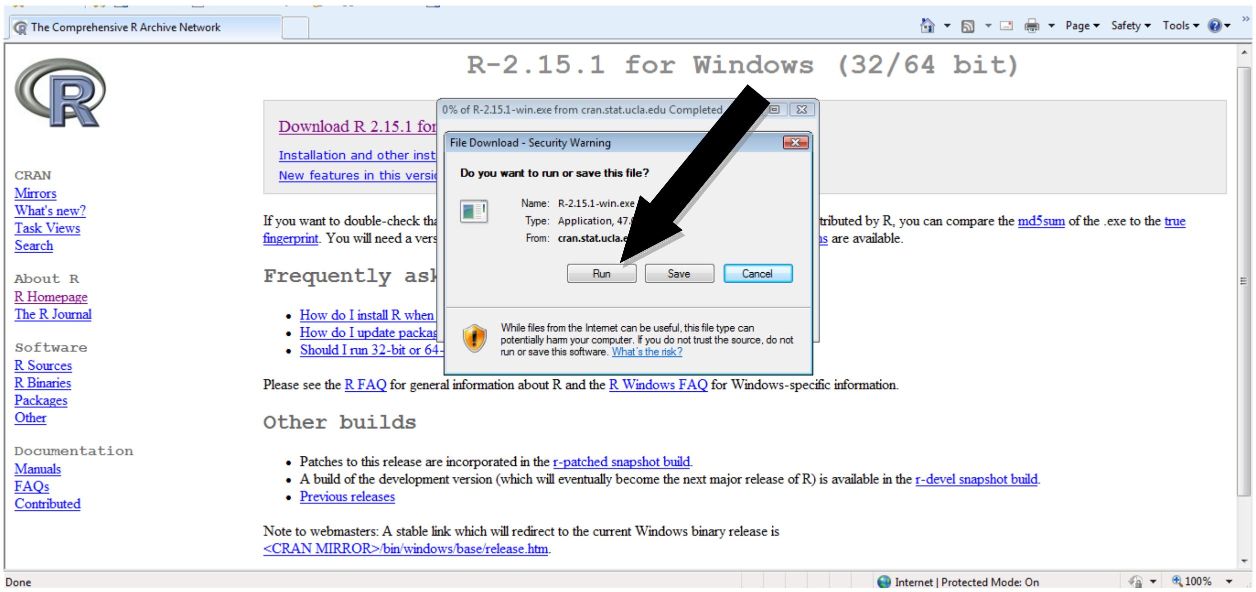



Now you are ready to download R!

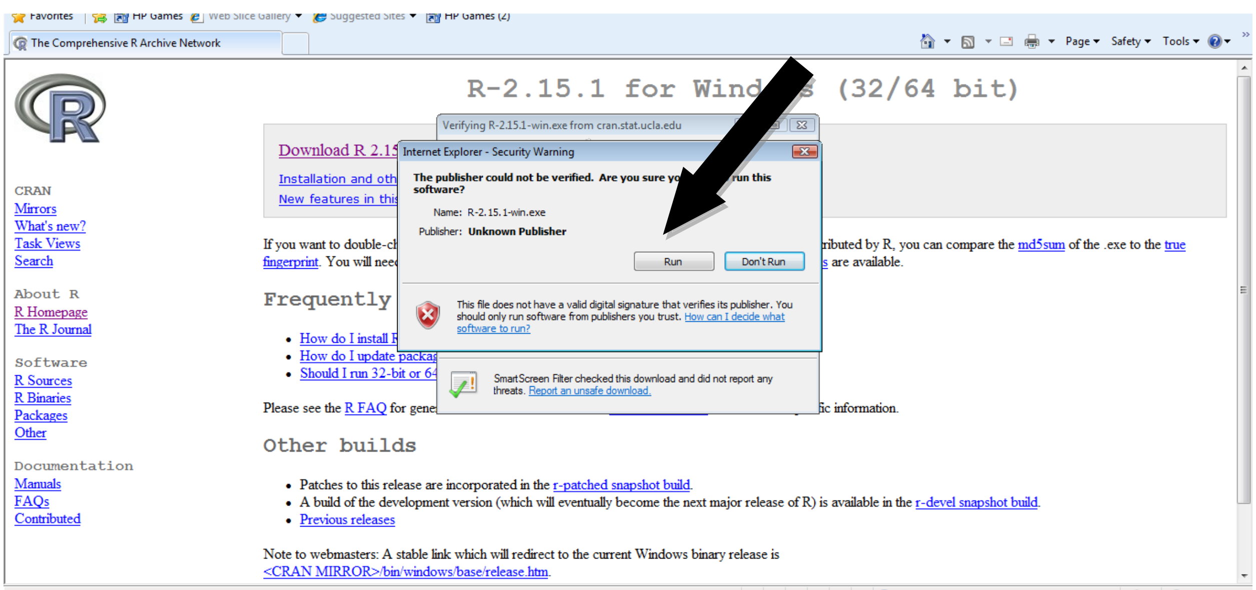
Note: Periodically R is updated and new versions are available. Version 2.15.1 was the newest version available at the time this guide was written. If you are using this guide later and there is a new version available, please select that version.



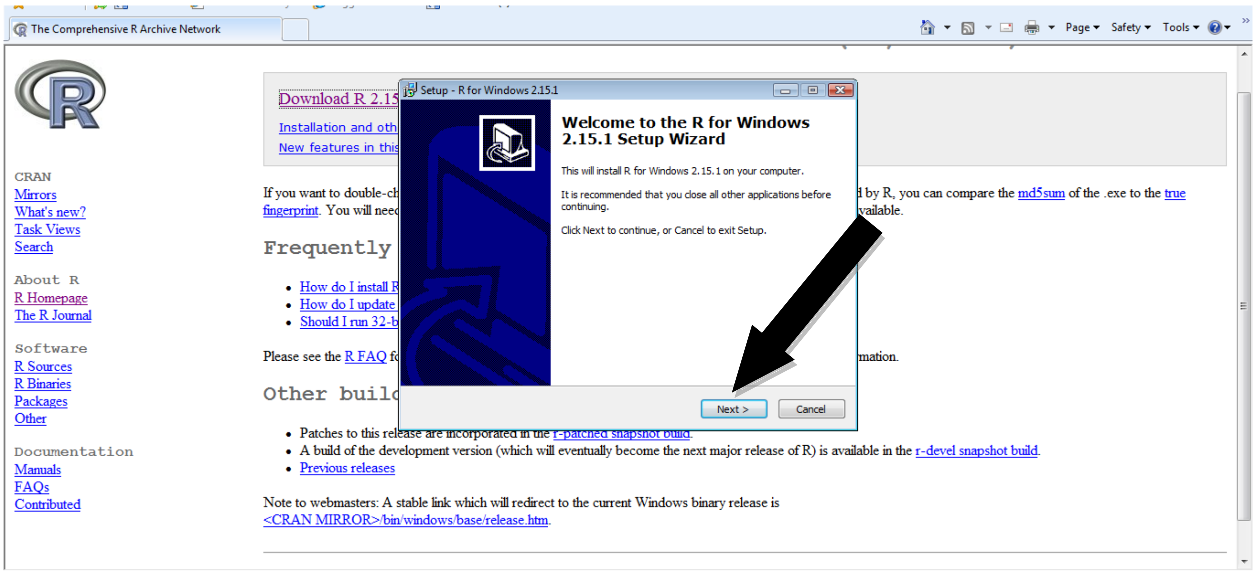

When asked if you want to run or save, select Run



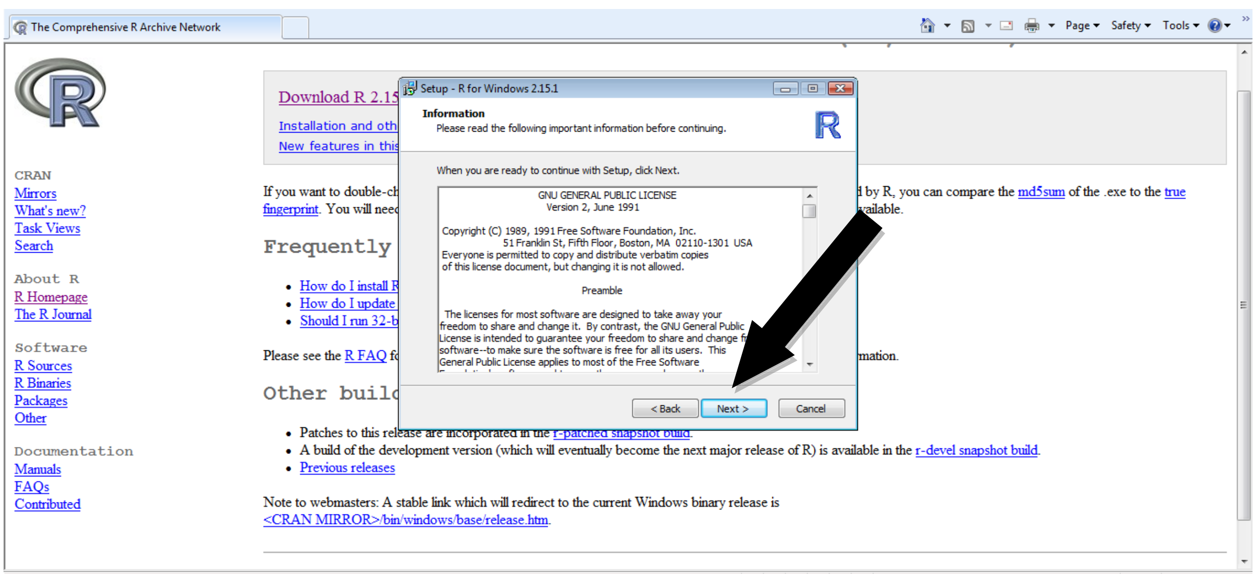

If a security warning comes up, in order to continue select Run once again



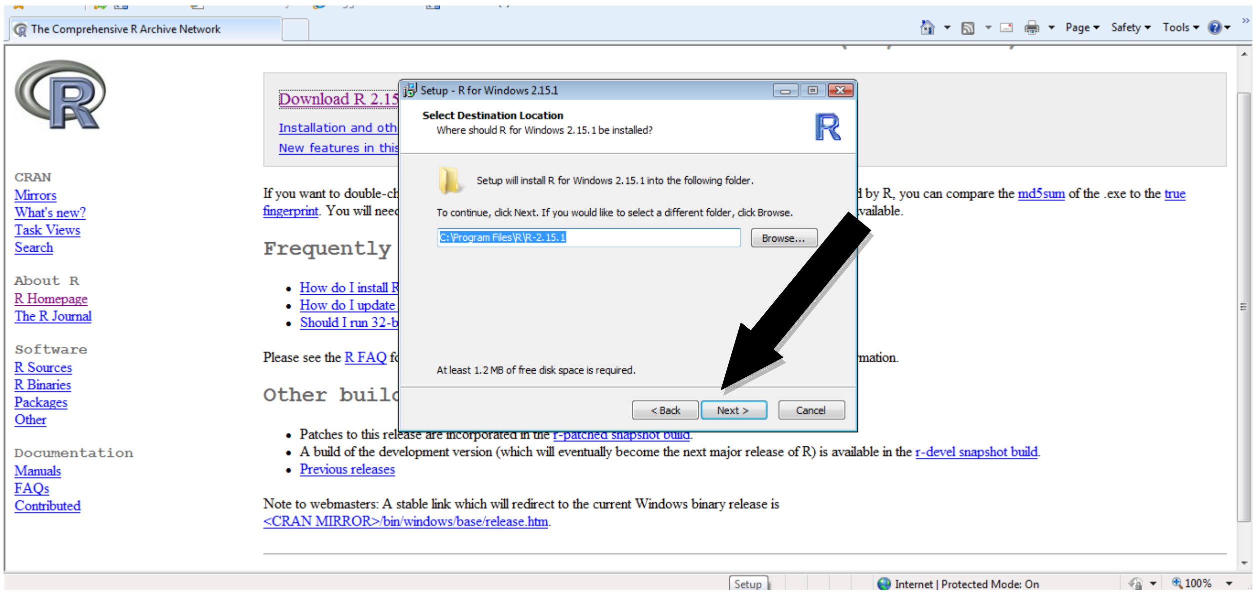

Once download is completed, the R Setup Wizard will now appear, select Next




Read the license and select Next again




Decide where you want R installed and select Next one more time

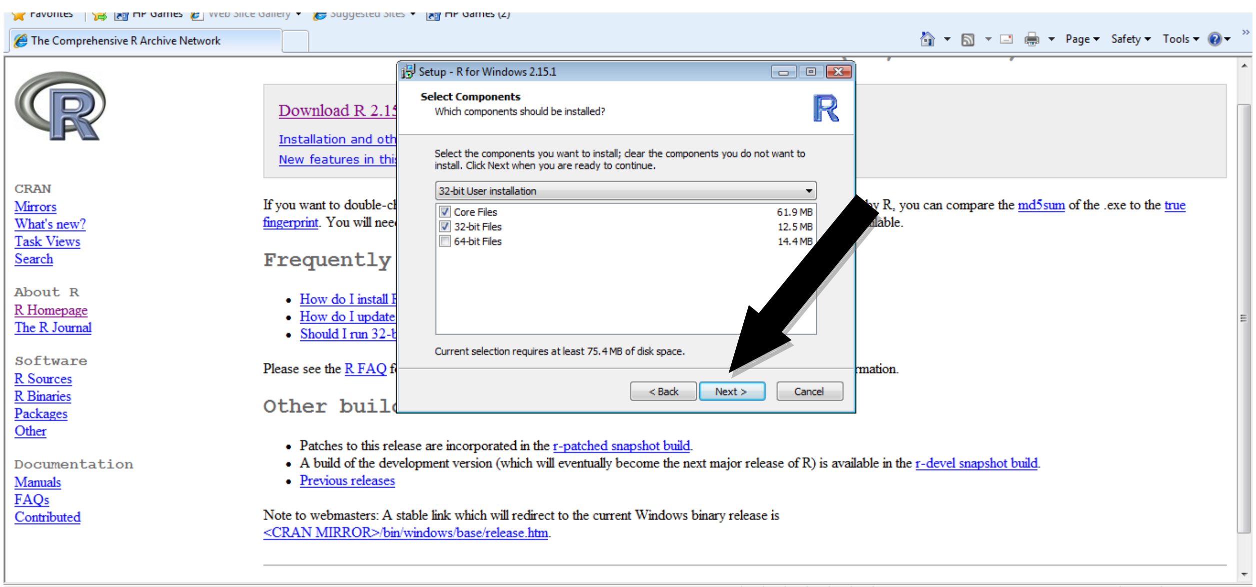



Now you need to decide whether to download the 32 bit or the 64 bit version. If you are running 32 bit version of windows select the 32 bit option but if you are running a 64 bit version of windows select the 64 bit option.  To find out if your computer is running 32-bit or 64-bit Windows, do the following:

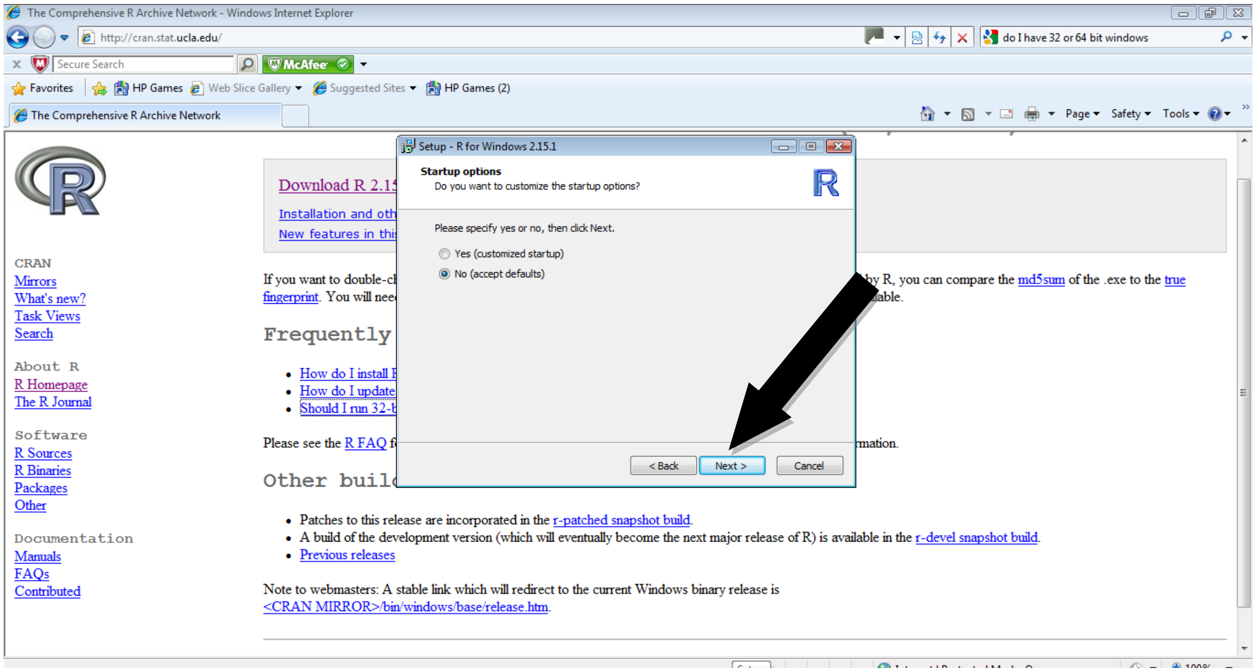
1. Open System by clicking the Start button, clicking Control Panel, clicking System and Maintenance, and then clicking System.

2. Under System, you can view the system type.

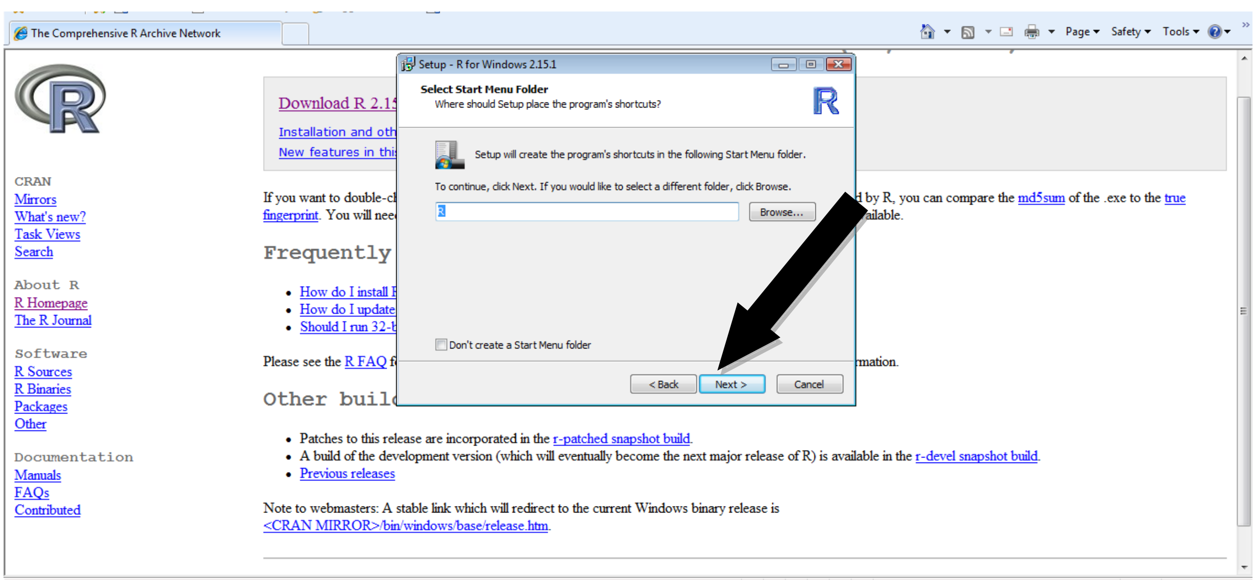
Now select Next



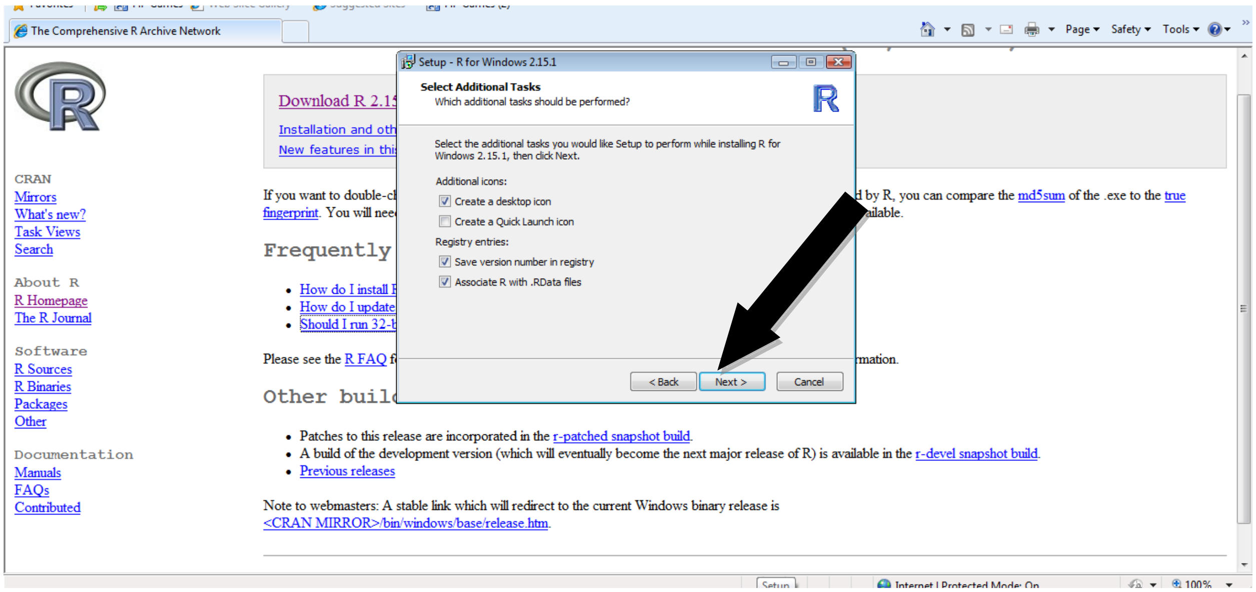

Make sure you have No (accept defaults) selected and hit Next



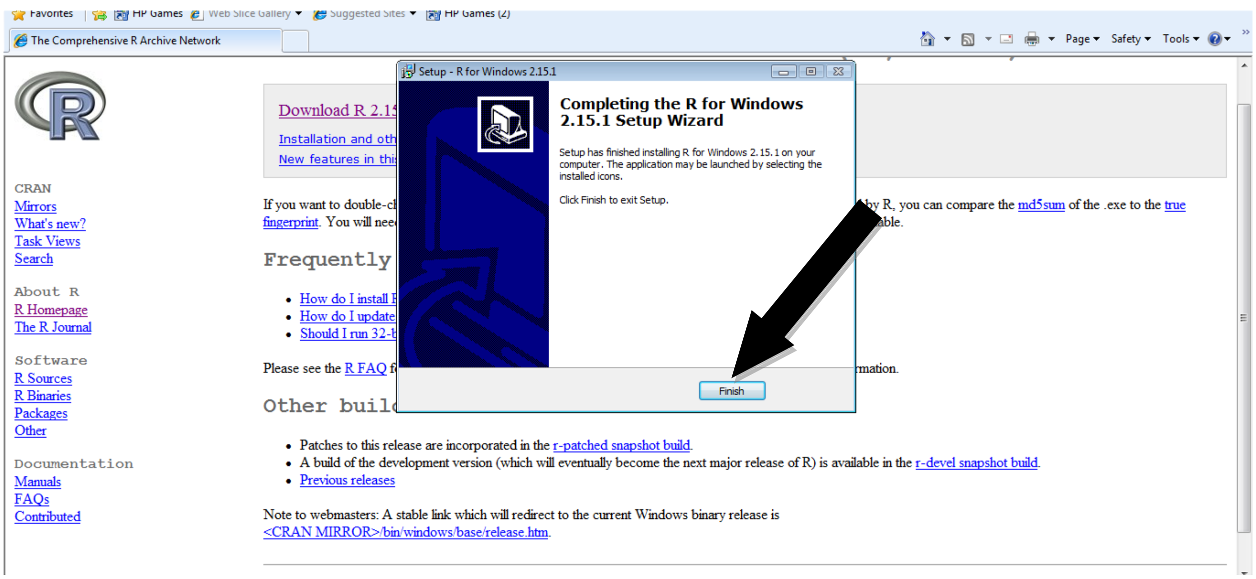

If you would like a shortcut in your start menu, then type in a name for the shortcut in the box, if not, select "Don't create a Start menu folder at the bottom of the box. Then select Next




Select the additional tasks you want (these are all optional) and select Next




R in now ready to be installed. When it has finished installing all components, select Finish



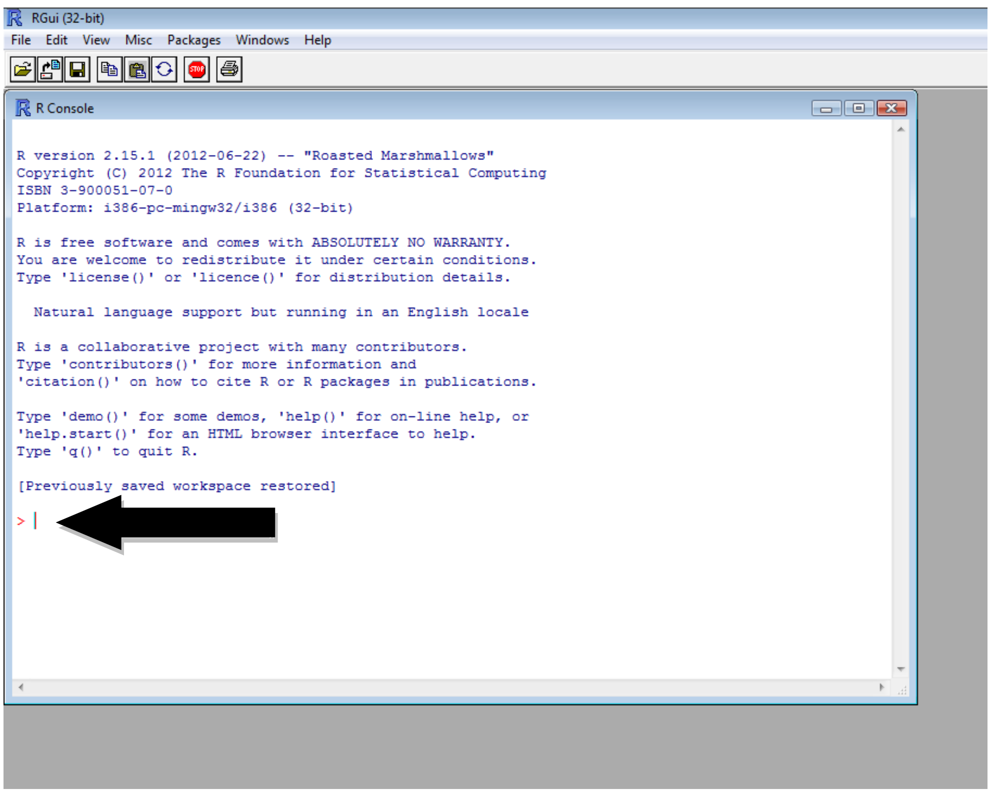

At this point, if you elected to have a shortcut put on your desktop you can open R from there, if not you can select R from your start menu in order to get started. All the functions and scripts that are written in the rest of this survival guide will work in the basic R interface. There is an Integrated Development Environment called R studio that makes working in R easier for some people. It can be found and downloaded at www.rstudio.org. The examples in this handout will show screen shots in R, not R studio

# Using R

R is a dialect of the S language and uses expression to generate outputs. There are five different object types in R;

- Character ex: "Hello"
- Numeric ex: 1.456734, 2.5656565656
- Integer ex: 1,2,3
- Complex ex: (4+3i)
- Logical ex: ( TRUE, FALSE)

R is case sensitive, for instance; School, SCHOOL and school are all different objects. Here is an example of the R interface




You can enter expressions into R at the prompt (>) or run it from a source file. R uses a variety of functions. These functions are words or part of words followed by a set of parenthesis. Some examples are

```
class( )
```

```
hist( )
```

Extra information can be added inside of the parenthesis to give R more direction on what to do.

It is a good idea to add comments to your expressions while working in R so if you ever need to go back to look over what you have done or want to use a certain expression in the future, you can easily identify what you are looking for. Comments always start off with # to differentiate it from the rest of the code or script. Example

```
# This code is used to create a histogram with red bins

hist(math,col="red")
```

```
summary( ) # calculates min, median, mean, max 1st quartile  and
               3rd quartile
```

# Getting Data into R

Now that you have installed R, and understand a little about how R works, the next step is getting data into R so you can start to work with and manipulate it. There are two ways to get data into R. You can use data from an existing file such as a text file, excel worksheet, or a tab delimited file. The other way is to directly input the data into R.

## Data from an existing file

Let's take some example data.

Example: Fifteen middle school students were selected to take a new math and science ability test. Their gender, grade level and score on the math and science section of the test were collected as seen in the table below. Let's call this data schools.

| Math | Science | Gender | Grade |
|---|---|---|---|
| 85 | 94 | 2 | 6 |
| 62 | 83 | 1 | 7 |
| 88 | 85 | 2 | 7 |
| 85 | 83 | 2 | 8 |
| 38 | 78 | 2 | 7 |
| 88 | 82 | 2 | 6 |
| 83 | 31 | 1 | 8 |
| 83 | 86 | 2 | 7 |
| 82 | 66 | 1 | 8 |
| 53 | 75 | 1 | 8 |
| 68 | 86 | 1 | 6 |
| 88 | 81 | 2 | 8 |
| 81 | 71 | 2 | 7 |
| 84 | 50 | 2 | 7 |
| 82 | 88 | 2 | 6 |

Note: Gender is coded 1 for male 2 for female

When using existing data sets in R, it is helpful to have one folder on your computer that contains all the data you use. You want to set the working directory of R to this folder. To find your working directory use the `getwd()` function. Type `getwd()` after the prompt and press enter. For this function you do not have to put anything in the parentheses. After you press enter you will see your working directory. This is where you want to place your data folder because R will only look in this folder for files unless you explicitly specify a new place to look, which will be shown later. If you want to change your working directory to another folder, you can go to the File menu then selection Change dir. This will have to be done each time you start R as the change is not permanent. Make sure the data you want to access in the folder on that your working directory is set to and then you can use the following commands to access data.

```
schools<-read.table("schools.txt", header=TRUE)

# header==TRUE indicates that the data table has a header
```

For a comma separated value (.csv) file use

```
schools<-read.csv("schools.csv", header=TRUE)
```

If you have your data saved elsewhere on your computer other than in your working directory. To call a data file into R, use the one of the read functions. The function you would use depends on the type of data you have. You will need to edit the function and make sure you have the right address to the file .

For instance, the file is saved in the data folder in the C drive as a text (.txt) file on my computer. The function to use to call it into R is

**Remember R is very sensitive, if you mistype even the smallest thing you might receive error messages.**

```
schools <- read.table("c:/data/schools.txt", header=TRUE)
```

For a comma separated value (.csv) file use

```
schools <- read.csv("c:/data/schools.csv", header=TRUE)
```

For tab delimited file (.prn) use

```
schools <- read.delim("c:/data/schools.prn", header=TRUE)
```

If you want to import an excel file, save it as a .csv or .txt file first, then use one of the above mentioned prompts.

If you have your data saved under my documents, then your prompt might look something like this

```
schools <- read.csv("c:/documents/data/schools.csv", header=TRUE)
```

Or for a specific class you might use

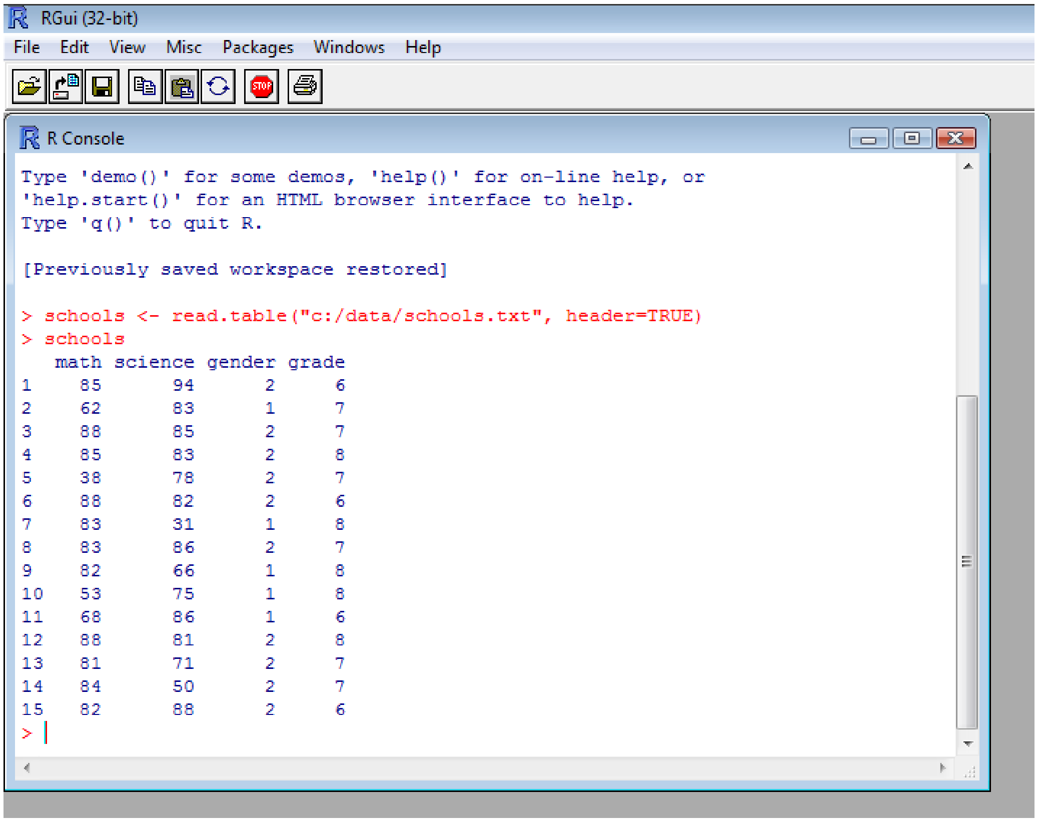
```
schools <- read.table("c:/EPRS8530/schools.txt", header=TRUE)
```

The main thing to remember when calling files from elsewhere on your computer is to correctly specify that path that will take R to the document. After calling your data, you can make sure it was correctly called into R.

```
schools     #shows the data
```

Now you need to attach the data in order to work with it. This is done with `attach()` function.

```
attach(schools) # attaches the data
```



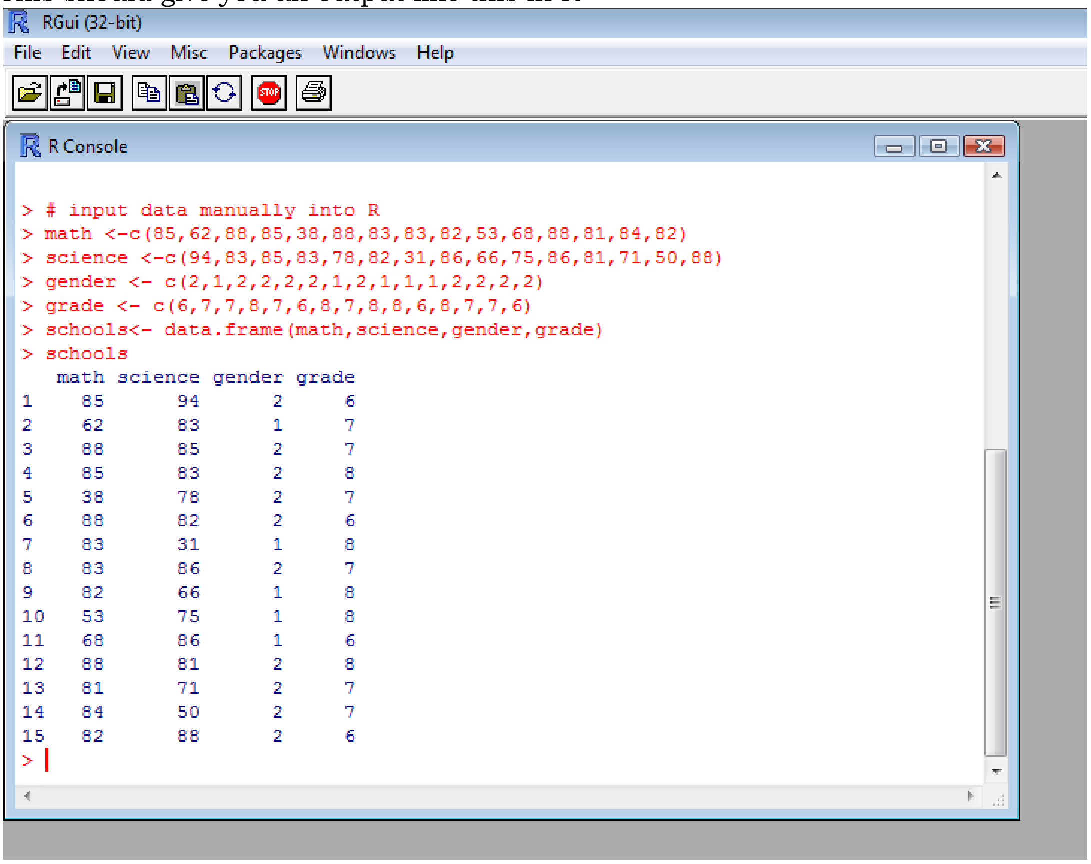

## Inputting Data

It is a little more difficulty to input large amounts of data into R. It might be easier to put it in another format and then call it into R. If you have a small data set, here is an example of how you would input it into R. We will use this data set form the previous example. **Again, remember R is very sensitive**. Make sure there are no spaces when you type and ensure that you type it as it is written to decrease the amount of error messages you receive. When inputting data you will use `<-` which is the assignment operation and `c()` which creates a vector. When used together you are in essence creating a vector and assigning it a name.

```
# input data manually into R
math<-c(85,62,88,85,38,88,83,83,82,53,68,88,81,84,82)
```

```
science<-c(94,83,85,83,78,82,31,86,66,75,86,81,71,50,88)
gender<-c(2,1,2,2,2,2,1,2,1,1,1,2,2,2,2)
grade<-c(6,7,7,8,7,6,8,7,8,8,6,8,7,7,6)
schools<-data.frame(math,science,gender,grade)
schools
```

This should give you an output like this in R

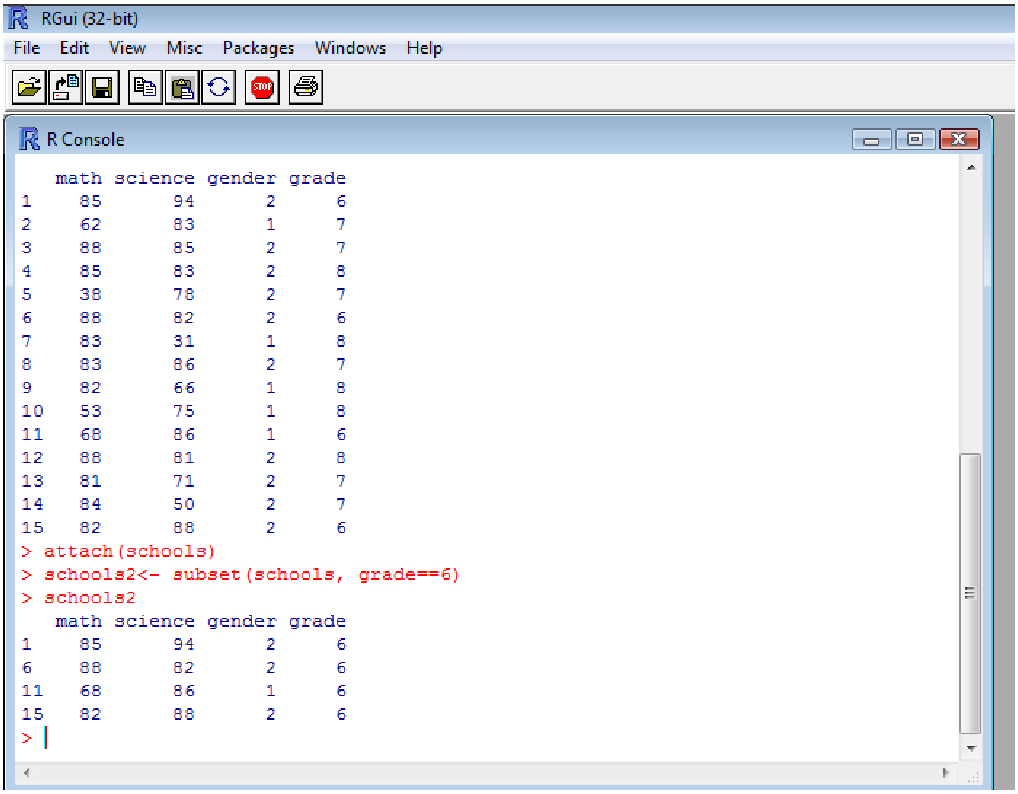



## Using subsets of data

There might be a time that you want to use only a subset of your data. In order to do this use the `subset( )` command. For example, to create a new subset of a data frame called schools that only includes values in which grade is equal to 6 which was coded for 6th grade students

```
schools2<- subset(schools, grade==6)     # subsets data

schools2   #Shows new data table
```

The output in R should look like this




Other examples

```
schools3<- subset(schools, math>=76)     #Selects math scores
                                          greater or equal to 76
```

```
schools4<- subset(schools, science < 86) # Selects science
                                             scores less than 86
```

If you want to work with this new data set, make sure you use the `attach()` function. For example `attach(schools2)` will allow you to work with the new data that only has scores of students in the 6th grade. Operators that you can use are

```
==        # exactly equal to

>         # greater than

>=        #greater than or equal to

<         # less thank
```

```
<=        # less than or equal to

!=        # not equal to
```

## Saving work

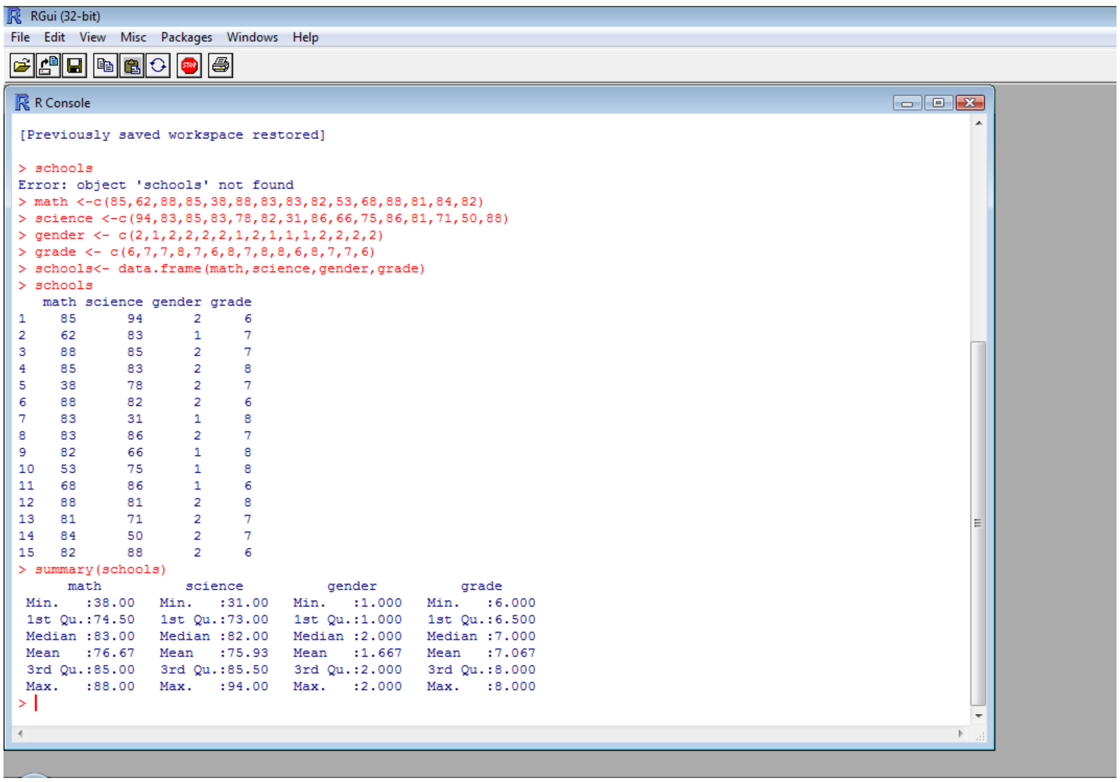
There are different ways to save your work in R.  One way is by going to the File menu and Save to File. This will save what you are currently working on as a text (.txt) file. When you open R again you can go to File, Display files, and select the file you want displayed.  Then you can copy and paste what you want in the command window.  Likewise you can save any graphical output by selecting the window with the graph displayed and going to File, Save As, selecting the file type and naming the file. This guide will go over saving and opening scripts in the Inferential Statistics section.

# Descriptive Statistics

Now that you have data to work with, let's move on to descriptive statistics. Remember in order to work with the data you must attach it first with the `attach()` function.

The following functions are often used to calculate descriptive statistics

For example, the `summary()` function produces  min, median, mean, max, 1st quartile  and 3rd quartile of all variable in schools data set.  Example, `summary(schools)` gives the following output.




Another function that can be used is the describe() function found in the psych library.  This function produces the item name, item number, number of valid entries, mean, standard deviation, median, MAD ( median absolute deviation), minimum value, maximum value, skewness, kurtosis, and se ( standard error)  This can be done using the following commands,

```
library(psych) # loads psych library functions

describe(schools) # produces the aforementioned information
```

To get the means for variables in the schools data frame.

```
sapply(schools, mean, na.rm=TRUE)

# na.rm=TRUE excludes missing values
```

You can also use `sd, var, min, max, med, range`, and `quartile` instead of mean to get those values for variable in the data

```
sapply(schools, var, na.rm=TRUE)

sapply(schools, range, na.rm=TRUE)
```

Other functions to use are as follows. Make sure you put the variable name that you want the information on inside the parentheses.

```
min()     # gives minimum value

max()     # gives maximum value

range()   # gives range

median()  # gives median

mean()    # gives mean

sd()      # gives standard deviation

var()     # gives variance
```

The `table() function` produces frequency of a given variable. Put the variable name inside the parentheses.

To produce crosstabulations of variables gender and grade use

```
table(gender, grade) #gender row, grade column
```

To calculate relative frequency of a given variable, you must make a new variable first then write a formula for how to calculate relative frequency. Example, calculating relative frequency for the grade variable is schools first type;

```
grade2=schools$grade   # makes a new variable grade2 which is
                         the same as the variable grade in
                         schools
```

Next you need to calculate the frequency so type

```
grade2.freq=table(grade2)
```

Next type

```
grade2.relfreq=grade2.freq/nrow(schools)
```

This makes grade2.relfreq equal to grade2.freq divided by the number of rows in the table, thus calculation the relative frequency. Lastly,

```
grade2.relfreq
```

displays values for grade relative frequencies

Cumulative frequency is easy once you have established the formula for relative frequency. To find the cumulative frequency for the grade variable from the previous example use

```
cumsum( )
```

So `cumsum(grade2.relfreq)` will give the cumulative frequency of the grade2.relfreq variable which was established in the above example. This will show the cumulative sum of the grade variable from the schools data.

## Graphing

This section will go over some common graphs used for descriptive statistics. These graphs will be displayed in a separate graphics window in R. You can resize this window with the `windows()` function. For example, if you want a window with a width of 15 and height of 2 you would use this expression

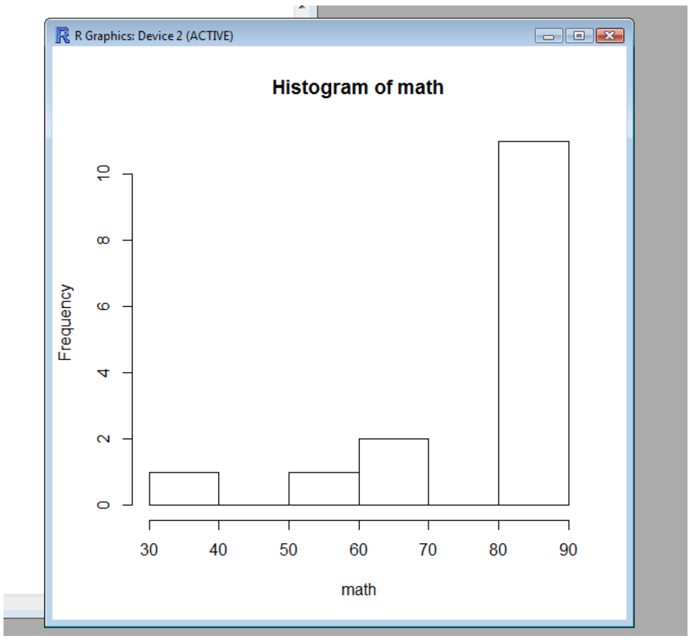
```
windows(15,2)
```

However this will make a long skinny graph. A good size to go with would be a width and a height of 7. To so this you would type

```
windows(7,7)
```

## Histogram

Creating a histogram in R is relatively easy. A basic histogram can be produced with the `hist()` function where the variable name goes into the parenthesis. This will produce a histogram in the graphics window. Again make sure your data is attached using the `attach()` function. For example `hist(math)` gives the following output

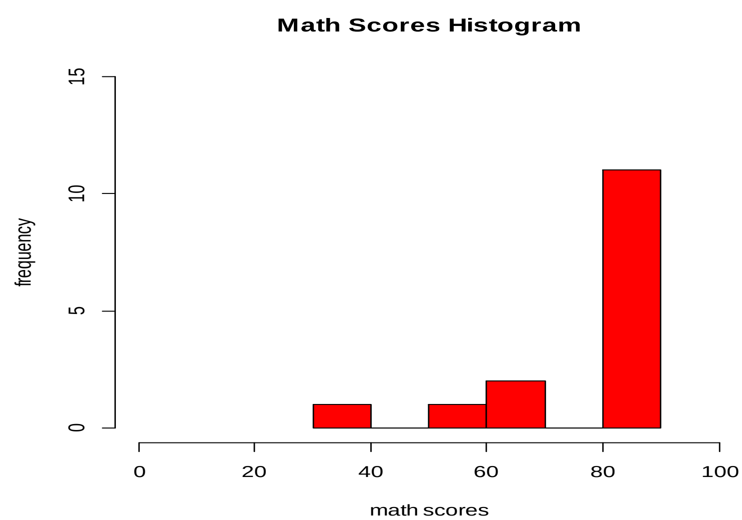



We can improve this histogram with added expressions

```
xlab= " "        # labels x axis

ylab=" "         # labels y axis

main= " "        # give the graph a title

col= " "         # colors the histogram bins Available colors
                 include red, blue, green, yellow, orange, purple

breaks=          # tells the sections to break the x axis into

xlim=c()         # sets the range of the x axis, values go inside
                 the parenthesis separated by comma

ylim=c()         # sets the range of the y axis, values go inside
                 the parenthesis separated by comma
```

Example

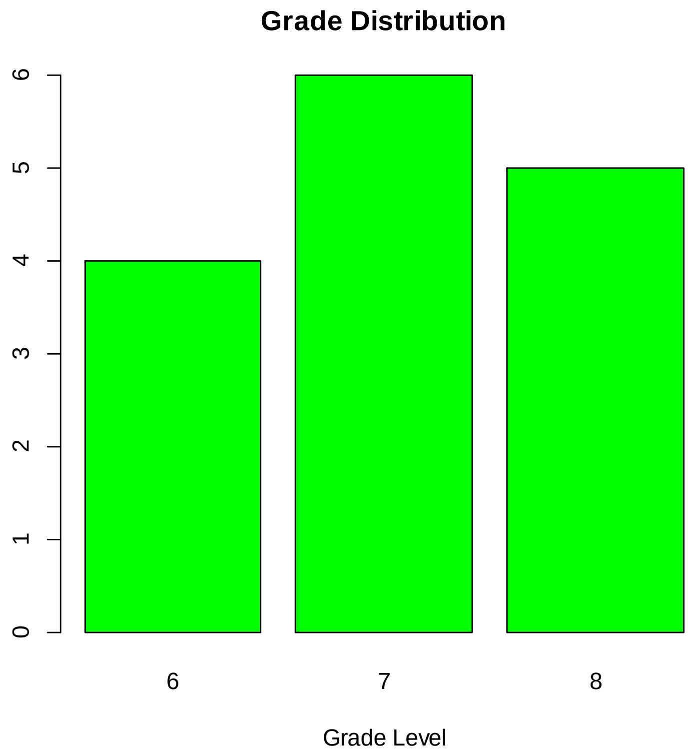
```
hist(math, xlab="math scores", ylab="frequency", main="Math
Scores Histogram", col="red", breaks=5, xlim=c(0,100),
ylim=c(0,15))

#creates a histogram with red bins, labels x axis "math
#scores",labels y axis "frequency",labels graph "math score
#histogram", x axis range 1-100, y axis range 0-15, breaks x
#axis into 5 sections
```




## Bar Graph

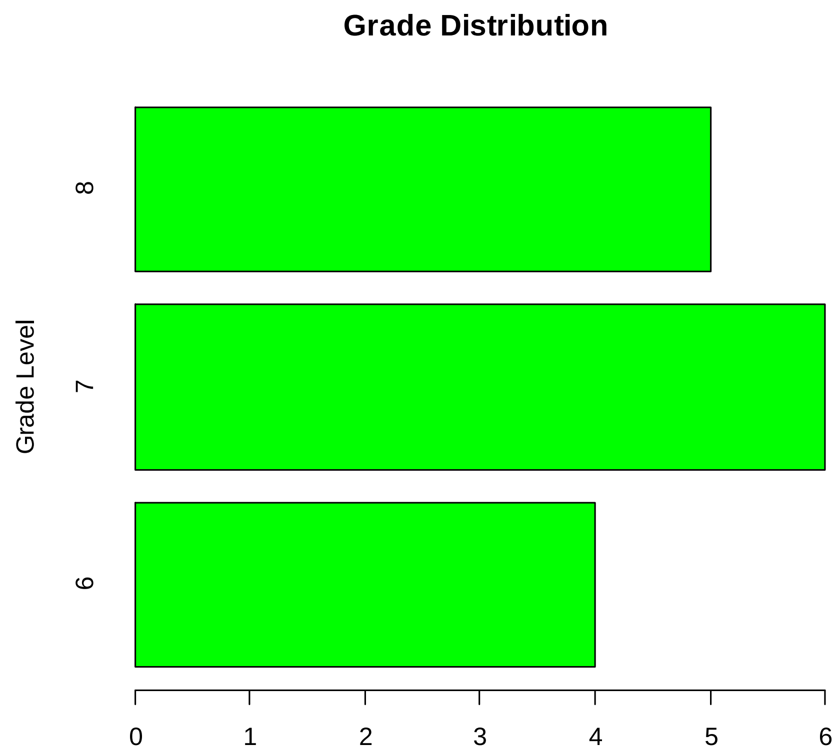
Bar graphs are also easy to create in R. First, however, the number of observations must be counted. To plot a bar graph of the number of students in each grade, the following expression would be used

```
counts <- table(grade) #counts the number of students in each
grade

barplot(counts, main= "Grade Distribution", xlab="Grade level",
col="green")
```




To convert to a horizontal bar graph use the expression `horiz=TRUE`

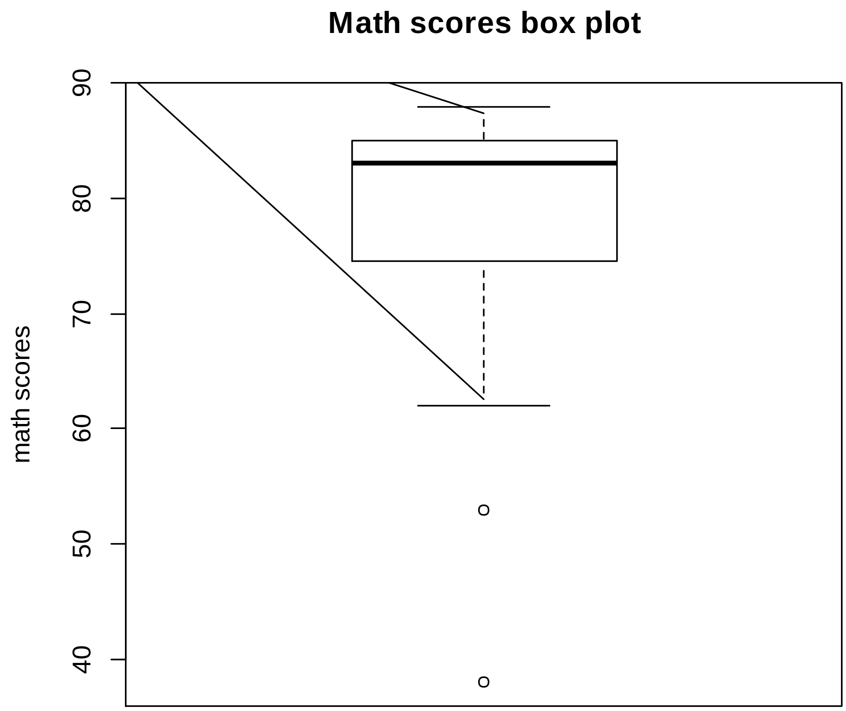
```
barplot(counts, main="Grade Distribution", ylab="Grade
Level",col="green", horiz=TRUE)
```




## Box plot

Using the `boxplot()` function will produce a box plot in R. You can continue to use the `xlab, ylab, main, xlim, ylim` expressions to improve the look of your box plot. Example

```
boxplot(math, ylab="math scores", main="Math scores box plot")
```

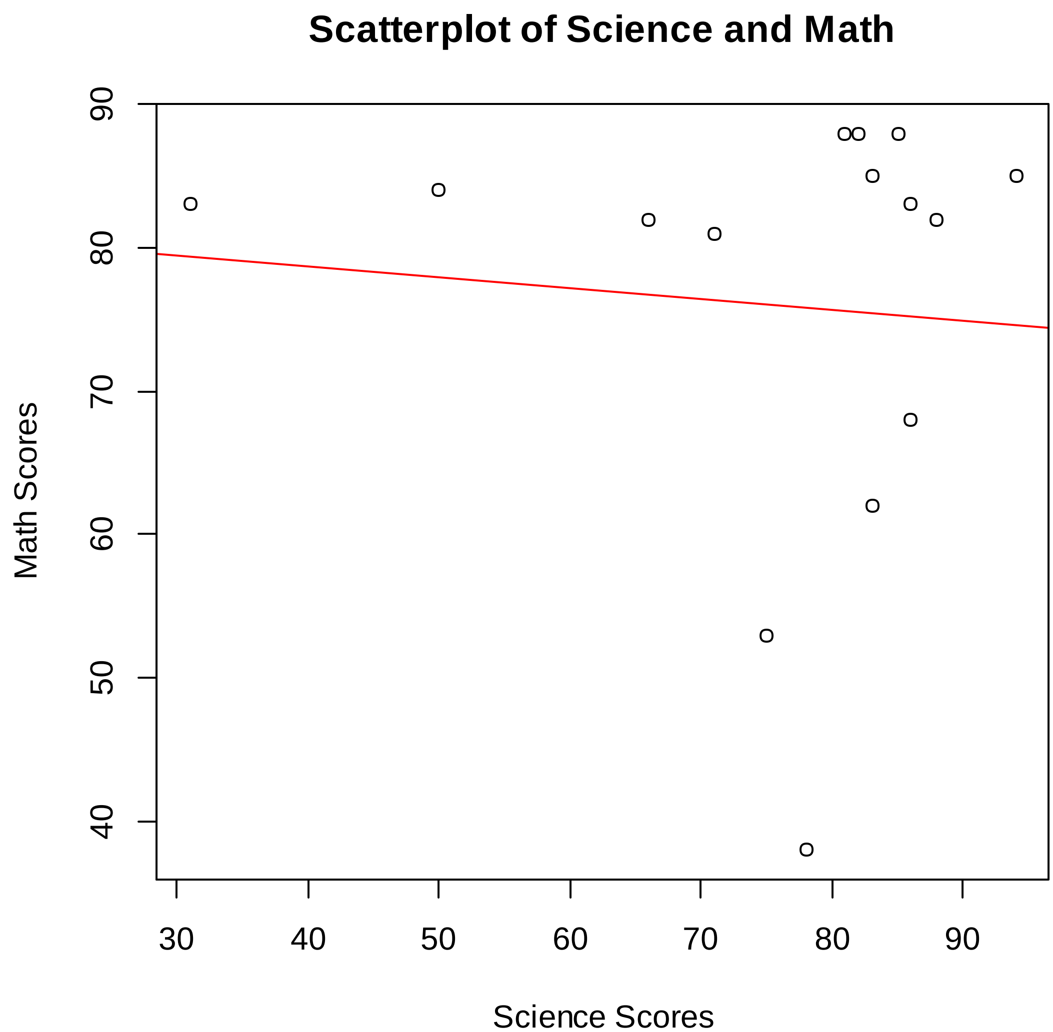



## Scatterplot

The `plot(x,y)` function will produce a simple scatter plot where x is the variable on the x axis and y is the variable on the y axis. The following expressions can also be used to improve the graph as with the histogram, and box plot `xlab, ylab, main, xlim, ylim.` The expression `abline(lm(x~y))` will produce a line of best fit on the same graph as the scatter plot.

Example:

```
plot(science,math, main="Scatterplot of Science and Math",
xlab="Science Scores", ylab="Math Scores",
abline(lm(science~math), col="red"))
```




The function `cor(x,y)` will give the correlation coefficient between two variables x and y but it will not do a significance test and give and value.

For example `cor(science,math)` gives `-0.0681767.`

## Linear Regression

To get the linear regression line, multiple steps are needed and the function `lm()` is used. For example, if you wanted to find the regression equation between the science and math variable you would take the following steps

```
regression.science.math=lm(science~math)
```

# this names the regression line and identifies the variables used

```
summary(regression.science.math)
```

# produces information on the regression line

The following output is produced (minus the highlights)

```
Call:

lm(formula = science ~ math)

Residuals:

    Min       1Q   Median       3Q      Max

-44.458  -3.660   5.965   9.667  18.693

Coefficients:

            Estimate Std. Error t value Pr(>|t|)

(Intercept) 81.69288   23.78064   3.435  0.00443 **

math        -0.07512    0.30490  -0.246  0.80923

---

Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 16.92 on 13 degrees of freedom

Multiple R-squared: 0.004648,   Adjusted R-squared: -0.07192

F-statistic: 0.06071 on 1 and 13 DF,  p-value: 0.8092
```

This is a lot of information but from this we can see the slope of the line is -0.07512 and the intercept is 81.69288. We can also see the two stars correspond to a significance value of 0.001.

# Inferential Statistics

We will now move on to inferential statistics. The first test we see is the one sample t-test.

**Example problem**

The principal of Anywhere Middle School wanted to see if the mean science scores of her students differed from the population mean of 75. She used the scores of 15 students in grades 6-8.

| Student | Math | Science | Gender | Grade |
|---|---|---|---|---|
| 1 | 85 | 94 | 2 | 6 |
| 2 | 62 | 83 | 1 | 7 |
| 3 | 88 | 85 | 2 | 7 |
| 4 | 85 | 83 | 2 | 8 |
| 5 | 38 | 78 | 2 | 7 |
| 6 | 88 | 82 | 2 | 6 |
| 7 | 83 | 31 | 1 | 8 |
| 8 | 83 | 86 | 2 | 7 |
| 9 | 82 | 66 | 1 | 8 |
| 10 | 53 | 75 | 1 | 8 |
| 11 | 68 | 86 | 1 | 6 |
| 12 | 88 | 81 | 2 | 8 |
| 13 | 81 | 71 | 2 | 7 |
| 14 | 84 | 50 | 2 | 7 |
| 15 | 82 | 88 | 2 | 6 |

Note: Gender is coded 1 for male 2 for female

This is the same data set, `schools`, which we used previously. Please refer to the section Getting Started with R to see how to input or recall the data to work with it. Remember to attach the data set before continuing.

The one sample test function is `t.test()`. Our input will be

```
# mu is the population mean & the confidence level ( conf.level)
#is at 95%

t.test(science, mu=75, conf.level=.95)
```

We then get the following output

```
One Sample t-test

data:  science

t = 0.2212, df = 14, p-value = 0.8281

alternative hypothesis: true mean is not equal to 75

95 percent confidence interval:

 66.88334 84.98333

sample estimates:

mean of x

 75.93333
```

From this we can see $t_{(.05,\ 14)} = .22$, $p = .83$ and we can be 95% sure that confidence interval is 66.88 to 84.98. We fail to reject the null hypothesis and concluded that there is no significant difference between the population mean score and the mean scores of the students.

**Independent t-test**

The principal of Anywhere Middle School wanted to see if there was a difference in the mean math scores of boys and girls at the school. She used the scores of 15 students in grades 6-8.

The independent t-test function is relatively easy to use in R. There are two possibilities

```
# independent 2-group t-test

t.test(y~x) # where y is numeric and x is coded into 2 variables


# independent 2-group t-test

t.test(y1,y2) # where  both variables ,y1 and y2, are numeric
```

First we need to test for equal variance with an F test.  The function for this is var.test()

```
#  We have to compare the means of math score if the gender is 1
# and the math scores if gender is 2

var.test(math[gender==1], math[gender==2])
```

We then get the following output

```
F test to compare two variances

data:  math[gender == 1] and math[gender == 2]

F = 0.7397, num df = 4, denom df = 9, p-value = 0.8237

alternative hypothesis: true ratio of variances is not equal to
1

95 percent confidence interval:

 0.1567765 6.5866472

sample estimates:

ratio of variances

         0.7396836
```

From this we can conclude that the variances is the same for both scores ($p$ = 0.82, $F$ = .7397).  So we can continue with the t test.

Since gender is already coded into two variables we will use the t.test(x~y)

```
# t test with equal variance ( var.equal=T)
#and with α=.05 (conf.level=.95)


t.test(math~gender, var.equal=T, conf.level=.95)
```

We then receive the following output

```
        Two Sample t-test

data:  math by gender
t = -1.3417, df = 13, p-value = 0.2027
alternative hypothesis: true difference in means is not equal to
0
95 percent confidence interval:
 -27.668069   6.468069
sample estimates:
mean in group 1 mean in group 2
           69.6            80.2
```

From the output, we fail to reject the null hypothesis and concluded that there is no significant difference between the mean math scores of the boys and girls in Anywhere Middle school.

**Dependent T-test**

Using the same data, the principal at Anywhere Middle wants to see if the mean science scores differ from the mean math scores.

In order to do this we will use the dependent t test. In R this is very similar to the independent t-test with an added part to let the function know that data is paired.

```
# dependent t test with α=.05
t.test(math, science, paired=TRUE, conf.level=.95)
```

Here is the output

```
Paired t-test

data:  math and science

t = 0.1245, df = 14, p-value = 0.9027

alternative hypothesis: true difference in means is not equal to
0

95 percent confidence interval:

 -11.89564  13.36230

sample estimates:

mean of the differences

              0.7333333
```

From the output we fail to reject the null hypothesis and concluded that there is no significant difference between the mean scores of the science and math test.

**Effect size**

The best way to calculate effect size is by writing a script. It is beneficial to use a script because we can type in the different calculations together and run it at one time instead of typing in one line at a time, waiting for the output and using that in the next line. We can also save this script for later use.

To write a script we will go to File→New script. This will open a new blank window. In this window is where you type the script. Here is an example of an effect size calculator

```
# Effect Size calculator

s1=sd(math[gender==1])

# calculates standard deviation of boys math scores

s2=sd(math[gender==2])
```

```
 # calculates standard deviation of girls math scores

n1=5 #number of boys

n2=10 # number of girls

xbar1=mean(math[gender==1])

#calculates mean of boys math scores

xbar2=mean(math[gender==2])

#calculates mean of girls math scores

es=((xbar1-xbar2)/sqrt((((s1^2)*(n1-1))+((s2^2)*(n2-1)))/(n1+n2-
2)))

#calculates effect size
```

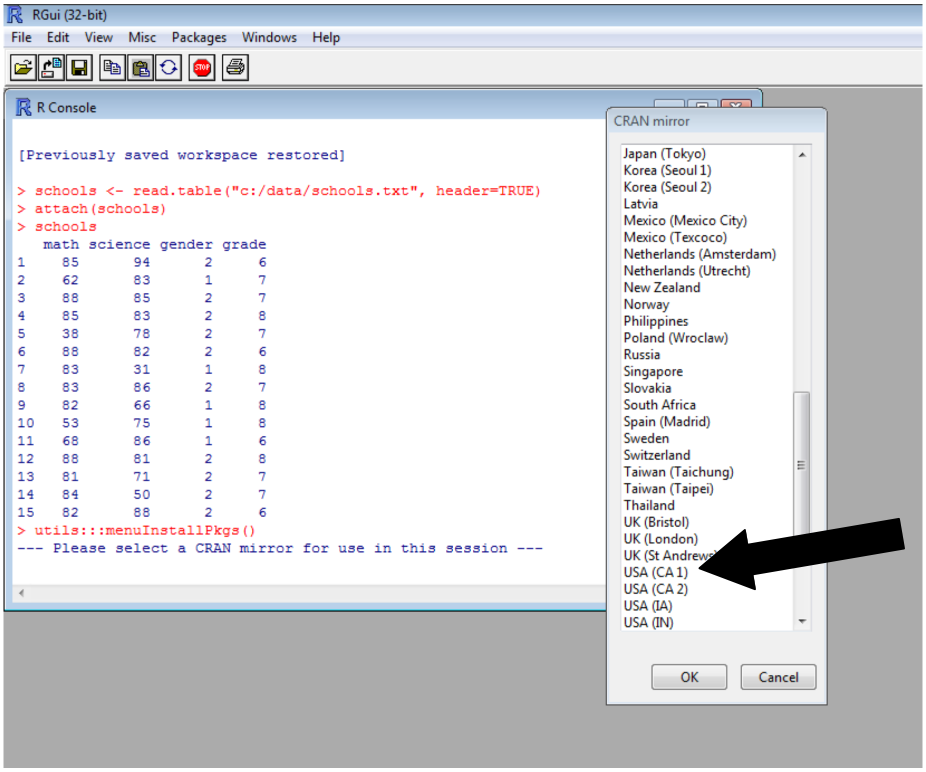
While the script window is open, go to Edit→Run all, you will see the script run in the other R window. When it is complete type `es` ( In order to output the results of the Effect Size script) and press enter and you will get

```
[1] -0.7348691
```

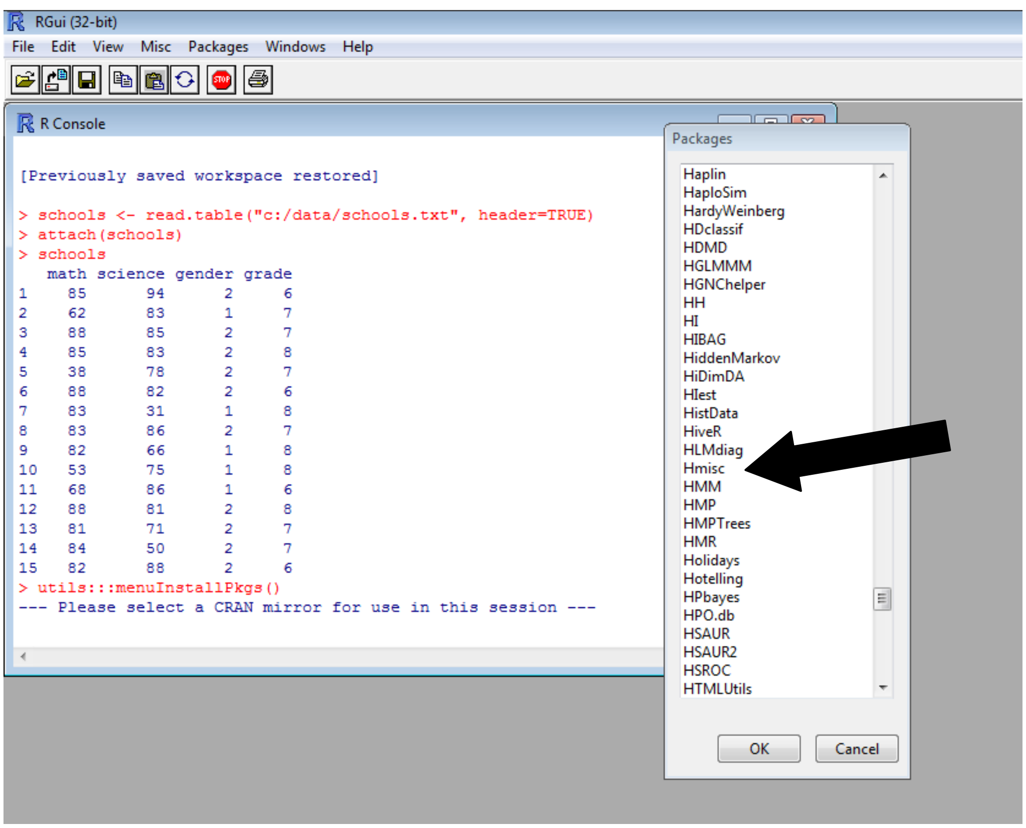
So your effect size is -0.735.

**Correlation**

Now let's test to see if there is a correlation between the math and science scores of the 15 students at Anywhere Middle Schools. In order to find the significance levels related to correlations we will use the `rcorr (x,y)` command. This command, however, is not in the initial commands downloaded so we must download the Hmisc package. Here are the steps to do this. Go to Packages →Install Package(s). Then select USA(CA 1)




A list of available packages will show up. Scroll down and select the `Hmsic` package.



The package will now be loaded. As you can see there are a plethora of packages available. As you progress with R, you can use the following website to identify packages that might be useful for you

http://cran.r-project.org/web/packages/available_packages_by_name.html

Now we need to load the `Hmisc` package and run the correlation

```
library( Hmisc)

# loads the Hmisc package

rcorr(math, science, type= "pearson")

# runs a correlation with significance levels as part of the
#output x is math and y is #science type can be pearson or
#spearman
```

We then get the following output

```
  x     y
x  1.00 -0.07
y -0.07  1.00

n= 15


P
  x      y
x        0.8092
y 0.8092
```

We can see the correlation coefficient is -0.07 and the p = .809. From this we can conclude that there is no significant correlation between the math and science scores of 15 students at Anywhere Middle.

# ANOVA

Now let's move onto ANOVA.  We are going to use a new data set for these tests.  A researcher wants to examine the effectiveness on three types of professional development on teachers. Twelve teachers from two schools were given one of three types of professional development, online, in person or a hybrid model.  Teachers were tested at the beginning and at the end of the professional development. We will call this data "profdev"

| teacher | pre | post | school | PD | Gender |
|---|---|---|---|---|---|
| 1 | 70 | 72 | A | O | M |
| 2 | 76 | 79 | A | O | F |
| 3 | 80 | 80 | B | O | F |
| 4 | 84 | 88 | B | O | M |
| 5 | 78 | 76 | A | P | M |
| 6 | 98 | 95 | A | P | M |
| 7 | 80 | 84 | B | P | F |
| 8 | 86 | 87 | B | P | F |
| 9 | 86 | 88 | A | H | F |
| 10 | 70 | 75 | A | H | M |
| 11 | 87 | 91 | B | H | F |
| 12 | 75 | 89 | B | H | M |

O: online   P: in person     H: hybrid

Since this is a new dataset, please refer to the section Getting Started with R to see how to input or recall the data to work with it.  Remember to attach the data set before continuing.

* Note: The following functions will use sequential sum of squares if your data is not balanced (the same number in each group) it will produce a different output than in other programs such as SPSS.

**Problem**

The researcher wants to see if there is a difference in the mean scores of the three types of professional development.

We will start with Levene's test of Equal variance.   In order to conduct Levene's test we must fist load the car package.  On the R tool bar go to packages →Load Packages →car. Now we are ready to use Levene's Test.

```
leveneTest(y=post, group=PD)
```

We get the following output

```
Levene's Test for Homogeneity of Variance (center = median)
      Df F value Pr(>F)
group  2  0.0789 0.9247
       9
```

So we can assume equal variance and continue with the ANOVA.

There are a few commands for ANOVA one of which is `aov(y~x)` where y is the dependent variable and x is the group factor. When using the aov command, R assumes the x variable is categorical. If you have used numbers for this variable, i.e. 1= online, 2= in person 3= hybrid, R will treat your entries as numerical and produce the incorrect results.

```
aov(post~PD)
```

We get the following output.

```
Call:
   aov(formula = post ~ PD)
Terms:
                      PD Residuals
Sum of Squares   92.1667  472.5000
Deg. of Freedom        2         9
Residual standard error: 7.245688
Estimated effects may be unbalanced
```

By using the `summary()` function we can get more information on the ANOVA model

```
summary(aov(post~PD))

# this summarizes and displays the ANOVA calculations
```

Now we get this output

```
            Df Sum Sq Mean Sq F value Pr(>F)
PD           2   92.2   46.08   0.878  0.448
Residuals    9  472.5   52.50
```

From here we can see the degrees of freedom, Sum of Squares, Mean Squares and F value. We can conclude that there is not a statistically significant difference between the scores in the three types of professional development, $F(2,9) = .88$, $p = .45$.

**Tukey's Post Hoc Test**

Let's say we found a statistically significant difference and needed to do a post hoc test to see where the difference lies. We would use the `TukeyHSD ()` command.

```
TukeyHSD(aov(post~PD))
```

We get the following,

```
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = post ~ PD)

$PD
               diff        lwr       upr     p adj
online-hybrid -6.00 -20.304772  8.304772 0.4980835
person-hybrid -0.25 -14.554772 14.054772 0.9986884
person-online  5.75  -8.554772 20.054772 0.5251277
```

This produces the difference in the means, the lower and upper bounds and the p value. We can see that none on the interactions are statistically significant which is what we expected.

# Two Way ANOVA

Using the same Data set, 'profdev", we can perform a Two Way ANOVA analysis. This time the researcher wants to look at the effects of school and professional development type of scores. We will use the same aov() command but add extra syntax inside

```
> aov(post~school*PD)

# school*PD will give the interaction effect
```

We get the following output

```
Call:
   aov(formula = post ~ school * PD)

Terms:
                  school       PD school:PD Residuals
Sum of Squares   96.3333  92.1667   48.1667  328.0000
Deg. of Freedom        1        2         2         6
Residual standard error: 7.393691
Estimated effects may be unbalanced
```

In order to get the full details we need to use the `summary()` command. Inside of the summary command is the same syntax from `aov()`

```
> summary(aov(post~school*PD))
            Df  Sum Sq  Mean Sq  F value  Pr(>F)
school       1    96.3    96.33    1.762   0.233
PD           2    92.2    46.08    0.843   0.476
school:PD    2    48.2    24.08    0.441   0.663
Residuals    6   328.0    54.67
```

From the $F$ statistics, we can see that neither school, $F(1,6)=1.76$, professional development, $F(2,6) = .84$or the interaction of the two, $F(2,6) = .44$ are statistically significant.

# Factorial ANOVA

We can continue to use the same data and look at more interactions. This time we want to see the effect of school, professional development and gender on scores. This will be a 2X3X2 ANOVA with three independent variables and one dependent variable. The syntax is very similar to that of the two way ANOVA

```
aov(post~school*PD*gender)

# this test the interaction between school, PD and gender

# it will produce all two way and three way interactions
```

Here is the output

```
Call:
   aov(formula = post ~ school * PD * gender)

Terms:
                   school        PD    gender school:PD
school:gender PD:gender
Sum of Squares   96.33333  92.16667   0.66667  72.00000
84.50000  32.00000
Deg. of Freedom         1         2         1         2
1         1
                school:PD:gender Residuals
Sum of Squares           2.00000 185.00000
Deg. of Freedom                1         2

Residual standard error: 9.617692
2 out of 12 effects not estimable
Estimated effects may be unbalanced
```

In order to get the full details we need to use the `summary()` command again.

```
> summary(aov(post~school*PD*gender))

                 Df  Sum Sq Mean Sq F value  Pr(>F)
school            1   96.33   96.33   1.041   0.415
PD                2   92.17   46.08   0.498   0.667
gender            1    0.67    0.67   0.007   0.940
school:PD         2   72.00   36.00   0.389   0.720
school:gender     1   84.50   84.50   0.914   0.440
PD:gender         1   32.00   32.00   0.346   0.616
school:PD:gender  1    2.00    2.00   0.022   0.897
Residuals         2  185.00   92.50
```

From these results we can see that all the factors and all the interactions produce non significant results.

# ANCOVA

In order to do an ANCOVA, we are going to use the professional development pre test scores as a covariate.  We are going to see if there is a difference in the mean post test scores given the type of professional development while using the pre test score as a covariate.

Before we can do ANCOVA we have to test the regression of slopes.   One way to do this is to see if the interaction between the covariate and the treatment (group) is significant.  If it is not significant then we can continue with the ANCOVA (Tabachnick & Fidell 2001).

Test of regression slopes

```
slopes<-aov(post~PD*pre, data=profdev)
```

```
> summary(slopes)
```

Here is the output

```
            Df Sum Sq Mean Sq F value Pr(>F)
PD           2   92.2    46.1   2.962 0.1274
pre          1  368.7   368.7  23.702 0.0028 **
PD:pre       2   10.4     5.2   0.336 0.7276
Residuals    6   93.3    15.6
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Looking at the interaction of PD and Pretest we can see that it is not significant and therefore we can assume equal regression slopes and continue with the ANCOVA.

We will use the same `aov()` command.  Make sure that the covariate it after the grouping variable.  If it is typed in the reverse order, it will produce incorrect results.

Input

```
ancova<-aov(post~PD + pre)
```

```
summary(ancova)
```

Output

```
              Df Sum Sq Mean Sq F value    Pr(>F)    
PD             2   92.2    46.1   3.552 0.078683 .  
pre            1  368.7   368.7  28.424 0.000701 ***
Residuals      8  103.8    13.0                     
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The results show that professional development is not statistically significant given the pre test scores as a covariate.

At this point there does not appear to be a function in R that will easily conduct a Bryant-Paulson Post Hoc test. However, there is a great Bryant Paulson Post Hoc calculator by Dr. T. Chris Oshima that can be found at

http://education.gsu.edu/coshima/statistics_2.htm

## Repeated Measure ANOVA

Test scores were collected at three different times for two different groups, online class and traditional class. For this example, a repeated measure ANOVA will be used which has one between factor, group and one within factor, time. The dependent variable is test scores. This data set will be called "repanova"

| id | group | score | time |
|---|---|---|---|
| 1 | 1 | 71 | 1 |
| 1 | 1 | 51 | 2 |
| 1 | 1 | 33 | 3 |
| 2 | 1 | 65 | 1 |
| 2 | 1 | 47 | 2 |
| 2 | 1 | 25 | 3 |
| 3 | 1 | 73 | 1 |
| 3 | 1 | 45 | 2 |
| 3 | 1 | 29 | 3 |
| 4 | 1 | 69 | 1 |
| 4 | 1 | 43 | 2 |
| 4 | 1 | 27 | 3 |

| id | group | score | time |
|---|---|---|---|
| 5 | 2 | 57 | 1 |
| 5 | 2 | 87 | 2 |
| 5 | 2 | 45 | 3 |
| 6 | 2 | 54 | 1 |
| 6 | 2 | 93 | 2 |
| 6 | 2 | 53 | 3 |
| 7 | 2 | 100 | 1 |
| 7 | 2 | 93 | 2 |
| 7 | 2 | 27 | 3 |
| 8 | 2 | 60 | 1 |
| 8 | 2 | 95 | 2 |
| 8 | 2 | 51 | 3 |

```
# calls up and attaches the data file

repanova<-read.csv("repanova.csv", header=TRUE)

attach(repanova)


#changes variables to factors (categorical variables) in order
to use repeated measure ANOVA

> repanova<-within(repanova, {

 group<-factor(group)

 time<-factor(time)

 id<-factor(id)

})
```

We know that one difference between the a one way ANOVA and repeated measures ANOVA is the error so this must be accounted for when we use the `aov()` command.

```
repanova.aov <- aov(score ~ group * time + Error(id), data =
repanova)

> summary(repanova.aov)
```

And here is the output

```
Error: id
          Df Sum Sq Mean Sq F value   Pr(>F)    
group      1 2340.4  2340.4   59.86 0.000245 ***
Residuals  6  234.6    39.1                     
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Error: Within
           Df Sum Sq Mean Sq F value  Pr(>F)    
time        2   5700  2850.0  19.756 0.00016 ***
group:time  2   2287  1143.4   7.926 0.00640 ** 
Residuals  12   1731   144.3                    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can see that time, group and the interaction between time and group are all significant so we must do a Post Hoc test to see where the significance lies. However at the writing of this guide there does not appear to be a script for a Post Hoc test for repeated measure ANOVA. There are many calculators online. The one that is used in this example is from Graphpad and can be found at http://graphpad.com/quickcalcs/posttest1.cfm.

This is the output from the site.

Confidence intervals

| Comparison | Mean1 - Mean2 | 95% CI of difference |
|---|---|---|
| 1: test1-2 | - 0.625 | - 10.903 to + 9.653 |
| 2: test 1-3 | + 32.375 | + 22.097 to + 42.653 |
| 3: test 2-3 | + 33.000 | + 22.722 to + 43.278 |

Statistical Significance

| Comparison | Significant? (P <0.05?) | t |
|---|---|---|
| 1: test1-2 | No | 0.200 |
| 2: test 1-3 | Yes | 10.355 |
| 3: test 2-3 | Yes | 10.555 |

It can be concluded that along with a statistical significance between the groups and a statistically significance interaction, there is also significance between tests 1 and 3 and test 2 and 3.

# Linear Regression

We will now move onto regression. We are going to use a new data set for these tests. A researcher wanted to examine how well certain variables predicted the score on a math final. This data set is called "mathfinal"

| ID | math.final | math.midterm | hours | SAT.math |
|---|---|---|---|---|
| 1 | 73.76 | 77.16 | 1 | 525.17 |
| 2 | 74.83 | 73.45 | 2 | 464.47 |
| 3 | 88.05 | 78.43 | 3 | 216.68 |
| 4 | 92.16 | 86.33 | 6 | 542.42 |
| 5 | 75.08 | 75.16 | 6 | 512.81 |
| 6 | 88.52 | 73.46 | 8 | 496.01 |
| 7 | 74.84 | 70.13 | 2 | 529.1 |
| 8 | 75.47 | 75.91 | 1 | 541.82 |
| 9 | 75.15 | 75.09 | 2 | 479.18 |
| 10 | 74.93 | 75.6 | 2 | 560.94 |
| 11 | 83.52 | 75.27 | 5 | 461.19 |
| 12 | 92.46 | 70.33 | 10 | 464.04 |
| 13 | 82.9 | 72.25 | 5 | 481.39 |
| 14 | 82.61 | 76.58 | 2 | 528.97 |
| 15 | 74.4 | 74.18 | 2 | 483.09 |
| 16 | 85.36 | 74.28 | 8 | 520.95 |
| 17 | 73.94 | 72.33 | 2 | 519.25 |
| 18 | 75.579 | 76.98 | 1 | 510.4 |
| 19 | 80.76 | 75.6 | 6 | 468.89 |
| 20 | 67.61 | 74.51 | 1 | 488.22 |
| 21 | 74.71 | 72.88 | 1 | 539.81 |
| 22 | 88.12 | 74.78 | 7 | 494.07 |
| 23 | 75.66 | 75.58 | 0 | 510.38 |
| 24 | 77.72 | 73.68 | 5 | 467.93 |
| 25 | 78.38 | 73.74 | 4 | 544.23 |

You can refer to the section Getting Started with R to see how to input or recall the data to work with it. Remember to attach the data set before continuing.

# Simple Regression

We will start with a simple linear regression and see how well the number of hours studied predicts the math final grade.

Here are the commands

```
> regmathfinal<-lm(math.final~hours)

> summary(regmathfinal)
```

Here is the output

```
Call:
lm(formula = math.final ~ hours)

Residuals:
    Min      1Q  Median      3Q     Max 
-8.7377 -1.9057 -0.6677  1.1513  9.8663 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  72.5497     1.4264  50.863  < 2e-16 ***
hours         1.8780     0.3134   5.993 4.12e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.198 on 23 degrees of freedom
Multiple R-squared:  0.6096,    Adjusted R-squared:  0.5927 
F-statistic: 35.92 on 1 and 23 DF,  p-value: 4.124e-06
```

The bottom of the output shows us that the overall regression model is significant, $F(1,23) = 35.92$, $p = .00$. The intercept is 72.5497 and the regression coefficient is 1.8780. The output also produces the $t$ values and the $p$ values. From the $F$ statistic and its associated $p$ value, we can see that the number of study hours is a statistically significant predictor of the math final grade.

# Multiple Regression

We can simply add on more predictors to go from simple to multiple regression. This time we will see how well math midterm grade and number of hours studied predicts the math final grade.

Command

```
regmathfinal2<-lm(math.final~math.midterm + hours)
```

```
> summary(regmathfinal2)
```

Here is the output

```
Call:
lm(formula = math.final ~ math.midterm + hours)

Residuals:
    Min      1Q  Median      3Q     Max 
-9.0259 -1.7290 -0.1035  1.9939  7.4799 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)   20.0849    18.3849   1.092   0.2864    
math.midterm   0.6971     0.2437   2.860   0.0091 ** 
hours          1.9384     0.2744   7.065 4.36e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.664 on 22 degrees of freedom
Multiple R-squared:  0.7154,    Adjusted R-squared:  0.6896 
F-statistic: 27.66 on 2 and 22 DF,  p-value: 9.905e-07
```

The overall regression equation is significant ($F$(2,22) = 27.66, $p$ = .00) and both math midterm grade and number of hours studied is significant ($t$ = 2.86, $p$ = .01, $t$ = 7.07, $p$ = .00). Note: the adjusted $R^2$ that is given is according to Wherry's formula.

Here are some more useful commands for simple and multiple regression

```
coefficients(regmathfinal) # model coefficients
confint(regmathfinal, level=0.95) # CIs for model parameters
fitted(regmathfinal) # predicted values
residuals(regmathfinal) # residuals
anova(regmathfinal) # anova table
```

If you would like to get the beta coefficients, you can load the QuantPsyc package and use the `lm.beta()` function. For example `lm.beta(regmathfinal2)` produces the following beta coefficients

```
math.midterm        hours 
   0.3262640    0.8058974 
```

## Stepwise Regression

R uses the AIC when conducting forward, backward or stepwise regression. Remember, these forms of regression come with their own flaws. First make sure you have the MASS package loaded by going to packages → load package→ MASS. The commands for forward, backward and stepwise are the same just type in either `forward`, `backward` or `both` after `direction` to indicate which you will use

Command

```
library(MASS)
> regmathfinal3<-lm(math.final~math.midterm + hours + SAT.math)
# Stepwise Regression
> step<-stepAIC(regmathfinal3, direction="both")
```

Here is the output

```
Start:  AIC=65.9
math.final ~ math.midterm + hours + SAT.math

               Df Sum of Sq    RSS    AIC
<none>                      253.42 65.904
- SAT.math      1     41.98 295.41 67.737
- math.midterm  1     98.86 352.28 72.139
- hours         1    624.20 877.62 94.959
```

Now to see what the final model is, we will use the following command

```
> step$anova
```

Here is the output

```
Stepwise Model Path
Analysis of Deviance Table

Initial Model:
math.final ~ math.midterm + hours + SAT.math

Final Model:
math.final ~ math.midterm + hours + SAT.math


  Step Df Deviance Resid. Df Resid. Dev      AIC
1                         21   253.4215 65.90446
```

## Comparing models

If we would like to compare the model with study hours and math mideterm grade as the predictors, `regmathfinal2,` with the output we just received from the step wise regression `regmathfinal3.` This can be done using the `anova()` command

Command

```
anova(regmathfinal2, regmathfinal3)
```

Output

```
Analysis of Variance Table

Model 1: math.final ~ math.midterm + hours
Model 2: math.final ~ math.midterm + hours + SAT.math
  Res.Df    RSS Df Sum of Sq      F  Pr(>F)
1     22 295.41
2     21 253.42  1    41.985 3.4791 0.07618 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can see that the difference between the models is not statistically significant, $F = 3.48$, $p = .08$.

## Part (semipartial) and Partial Correlation

The `lm.sumSquares()` function in the lmSupport package will give the delta $R^2$ (dR-sqr) which is the semipartial or part correlation squared and the pEta-sqr which is the partial correlation squared. For example,

```
#this function gives more information on the regression model
#and we can use it to obtain part and partial correlations
> lm.sumSquares(regmathfinal3)
```

Output

```
                       SS dR-sqr pEta-sqr df       F p-value
(Intercept)      37.33764 0.0360   0.1284  1  3.0940  0.0931
math.midterm     98.86261 0.0952   0.2806  1  8.1923  0.0093
hours           624.20135 0.6013   0.7112  1 51.7250  0.0000
SAT.math         41.98470 0.0404   0.1421  1  3.4791  0.0762
Error (SSE)     253.42150     NA       NA 21      NA      NA
Total (SST)    1038.12495     NA       NA NA      NA      NA
```

We can use this to calculate the part and partial correlation. For instance, the part or semipartial correlation of hours is $\sqrt{.6013} = .7754$ and the partial correlation for the same variable is $\sqrt{.7112} = .8433$.

## Model Validation

R gives you Wherry's adjusted $R^2$ and you can easily calculate Steins adjusted $R^2$. R will calculate the PRESS statistics which you can use to get the PRESS $R^2$. In order to do this you need to install the MPV package. Within this package is the `PRESS()` function which will calculate the PRESS statistic.

```
library (MPV)
PRESS(regmathfinal)
```

Output

```
[1] 464.0054
```

Use this to calculate the PRESS $R^2$

## t test and ANOVA using Linear Regression

Since and ANOVA is a special type of a linear regression, we can conduct the same analysis using the `lm()` function instead of the `aov()` function. First we have to dummy code the variables in order for it to work. We will use the same data as that is in the "profdev" file but it will be dummy coded. This was the old data set

| teacher | pre | post | school | PD | Gender |
|---|---|---|---|---|---|
| 1 | 70 | 72 | A | O | M |
| 2 | 76 | 79 | A | O | F |
| 3 | 80 | 80 | B | O | F |
| 4 | 84 | 88 | B | O | M |
| 5 | 78 | 76 | A | P | M |
| 6 | 98 | 95 | A | P | M |
| 7 | 80 | 84 | B | P | F |
| 8 | 86 | 87 | B | P | F |
| 9 | 86 | 88 | A | H | F |
| 10 | 70 | 75 | A | H | M |
| 11 | 87 | 91 | B | H | F |
| 12 | 75 | 89 | B | H | M |

We will use school and PD in the analysis so these will be the two variables that will be dummy coded.

New data set “profdev2” dummy coded

| teacher | pre | post | school | PD1 | PD2 |
|---|---|---|---|---|---|
| 1 | 70 | 72 | 0 | 1 | 0 |
| 2 | 76 | 79 | 0 | 1 | 0 |
| 3 | 80 | 80 | 1 | 1 | 0 |
| 4 | 84 | 88 | 1 | 1 | 0 |
| 5 | 78 | 76 | 0 | 0 | 1 |
| 6 | 98 | 95 | 0 | 0 | 1 |
| 7 | 80 | 84 | 1 | 0 | 1 |
| 8 | 86 | 87 | 1 | 0 | 1 |
| 9 | 86 | 88 | 0 | 0 | 0 |
| 10 | 70 | 75 | 0 | 0 | 0 |
| 11 | 87 | 91 | 1 | 0 | 0 |
| 12 | 75 | 89 | 1 | 0 | 0 |

We will use the `lm()` function for the *t*-test. This will look at the difference in the mean scores for the two different schools.

```
> model1<-lm(post~school)
> summary(model1)
```

Here is the output

```
Call:
lm(formula = post ~ school)

Residuals:
    Min      1Q  Median      3Q     Max 
-8.8333 -5.0833 -0.6667  3.0000 14.1667 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   80.833      2.794  28.933 5.67e-11 ***
school         5.667      3.951   1.434    0.182    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.843 on 10 degrees of freedom
Multiple R-squared:  0.1706,    Adjusted R-squared:  0.08766 
F-statistic: 2.057 on 1 and 10 DF,  p-value: 0.182
```

We can see from the output that the mean value for group 2 is 80.833 and the difference between the groups is 5.667. The *t*-value is 1.434 and the *F* statistic is 2.057 which is equal to 1.434*1.434. In this example we would fail to reject $H_0$.

We will also use the `lm()` function for ANOVA. This test will look at the difference between the three types of professional development.

```
model2<-lm(post~PD1+PD2)
> summary(model2)
```

Here is the output

```
Call:
lm(formula = post ~ PD1 + PD2)

Residuals:
    Min      1Q  Median      3Q     Max 
-10.750  -3.062   0.875   3.750   9.500 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   85.750      3.623  23.669 2.05e-09 ***
PD1           -6.000      5.123  -1.171    0.272    
PD2           -0.250      5.123  -0.049    0.962    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 

Residual standard error: 7.246 on 9 degrees of freedom
Multiple R-squared:  0.1632,    Adjusted R-squared:  -0.02273 
F-statistic: 0.8778 on 2 and 9 DF,  p-value: 0.4485
```

The output tells us that 85.750 is the mean of group 3, there is a -6.0 point difference between group 1 and group 3 and an -0.250 point difference between group 2 and group 3. The *F* statistic is 0.878 and just as in the ANOVA example we fail to reject $H_o$.

## ANCOVA using Linear Regression

We will continue to use “profdev2” but in order to do the ANCOVA we must first test the assumption of equal lopes. In order to do this we need to come up with interaction variables between the pretest and the grouping variable. In this case, prePD1 will be the pretest score multiplied by the PD1 variable and prePD2 will be the pretest score multiplied by the PD2 variable. We will then compare the full model with the reduced model

```
> prePD1<-pre*PD1 # creates new variable prePD1
> prePD2<-pre*PD2 # creates new variable prePD2
```

```
#Full model Regression
> fullmodel<-lm(post~pre+PD1+PD2+prePD1+prePD2)
> summary(fullmodel)
```

Here is the output for the full model:

```
Call:
lm(formula = post ~ pre + PD1 + PD2 + prePD1 + prePD2)

Residuals:
    Min      1Q  Median      3Q     Max 
-4.3182 -2.2135  0.1889  1.1567  6.2967 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)  
(Intercept)  31.9258    21.7789   1.466   0.1930  
pre           0.6770     0.2728   2.482   0.0477 *
PD1         -34.3838    36.7618  -0.935   0.3857  
PD2         -16.4444    30.7602  -0.535   0.6121  
prePD1        0.3837     0.4688   0.818   0.4444  
prePD2        0.1419     0.3721   0.381   0.7161  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 

Residual standard error: 3.944 on 6 degrees of freedom
Multiple R-squared: 0.8347,     Adjusted R-squared: 0.697 
F-statistic:  6.06 on 5 and 6 DF,  p-value: 0.02431 

#reduced model
> redmodel<-lm(post~pre+PD1+PD2)
> summary(redmodel)
```

Here is the output for the reduced model:

```
Call:
lm(formula = post ~ pre + PD1 + PD2)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.4088 -2.0926 -0.7466  1.5622  6.9047 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  21.1829    12.2438   1.730 0.121865    
pre           0.8122     0.1523   5.331 0.000701 ***
PD1          -4.3757     2.5649  -1.706 0.126408    
PD2          -5.1230     2.7058  -1.893 0.094940 .  
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.602 on 8 degrees of freedom
Multiple R-squared: 0.8162,     Adjusted R-squared: 0.7473
F-statistic: 11.84 on 3 and 8 DF,  p-value: 0.002593
```

We will now compare the full and reduced models to see if the interaction term is significant.

```
# compares the two models
> anova(fullmodel, redmodel)
Analysis of Variance Table

Model 1: post ~ pre + PD1 + PD2 + prePD1 + prePD2
Model 2: post ~ pre + PD1 + PD2
  Res.Df     RSS Df Sum of Sq      F Pr(>F)
1      6  93.338
2      8 103.777 -2    -10.44 0.3355 0.7276
```

From this formula, we see that the difference is not significant, $F = 0.34$. Now we can move on with the ANCOVA in which we test the full model with pre, PD1 and PD2 with the reduced model which only had PD1 and PD2.

```
> fullmodel2<-lm(post~pre+PD1+PD2) #creates full model
> reducedmodel2<-lm(post~PD1+PD2)  #creates reduced model
> anova(fullmodel2,reducedmodel2)  #compares the two models
```

Output

```
Analysis of Variance Table

Model 1: post ~ pre + PD1 + PD2
Model 2: post ~ PD1 + PD2
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1      8 103.78
2      9 472.50 -1   -368.72 28.424 0.0007013 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We get $F = 28.424$, which is significant. Since we were testing the pre part of the model we can say that the pretest is significant. Lastly we need to test the group difference. The full model in this test will have pre, PD1 and PD2 (which we have already done). The reduced model will only have pre. This will allow us to test the group difference.

```
> reducedmodel3<-lm(post~pre) #creates the new reduced model
> anova(fullmodel2, reducedmodel3) # compares the two models
```

Output

```
Analysis of Variance Table

Model 1: post ~ pre + PD1 + PD2
Model 2: post ~ pre
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1      8 103.78
2     10 163.05 -2   -59.274 2.2847 0.1641
>
```

Comparing these models gives us an $F$ = 2.287 which is not significant, so the group difference is not significant.

## ANCOVA using Johnson Neyman Technique

The following data contains pretest and posttest scores for 20 students who received either an intervention, group 1 or were the control group, group 0. We will call this data “intervention”

| ID | Group | Pre | Post |
|---|---|---|---|
| 1 | 1 | 45 | 34 |
| 2 | 1 | 21 | 10 |
| 3 | 1 | 38 | 26 |
| 4 | 1 | 38 | 22 |
| 5 | 1 | 49 | 31 |
| 6 | 1 | 41 | 27 |
| 7 | 1 | 39 | 24 |
| 8 | 1 | 44 | 22 |
| 9 | 1 | 47 | 20 |
| 10 | 1 | 49 | 24 |
| 11 | 0 | 38 | 42 |
| 12 | 0 | 35 | 48 |
| 13 | 0 | 41 | 29 |
| 14 | 0 | 44 | 34 |
| 15 | 0 | 24 | 47 |
| 16 | 0 | 26 | 42 |
| 17 | 0 | 38 | 45 |
| 18 | 0 | 34 | 46 |
| 19 | 0 | 27 | 44 |
| 20 | 0 | 44 | 45 |

We would like to analyze this data by using the ANCOVA model since we have one qualitative independent variable (group) and one quantitative independent variable (pretest). First we need to test the assumption of equal variance. In this case we want to see if the interaction term (pregroup) is significant. As before, in order to do this we need to test the full model against the reduced model. The full model will have pre, group and pregroup as predictors and the reduced model will have pre and group as predictors. We will then compare the models to see if there is a significant difference between the two. Remember first call and attach the new file “intervention”.

```
#creates the new interaction variable preGroup
> preGroup<-Group*Pre
# creates the full model
> fullmodel<-lm(Post~Group+Pre+preGroup)
# creates the reduced model
> redmodel<-lm(Post~Group+Pre)
> anova(fullmodel, redmodel) # Compares the full and reduced model
```

Output of the comparison test

```
Analysis of Variance Table

Model 1: Post ~ Group + Pre + preGroup
Model 2: Post ~ Group + Pre
  Res.Df    RSS Df Sum of Sq      F   Pr(>F)
1     16 443.50
2     17 691.12 -1   -247.62 8.9331 0.008679 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
>
```

From the *F* statistic and the subsequent *p* value, we can see that the interaction term is significant and therefor the slopes are not homogeneous. Due to this fact, we will need to use the Johnson Neyman Technique. Take note of the residual Sum of Squares (443.50). This will be used in the Johnson Neyman calculations. Now we need to look at the regression model for the two groups separately

```
#subsets the data to include only Group 1 and attaches the file
> int2<-subset (intervention, Group==1)
> attach(int2)
```

Informational output

```
The following object is masked from int:

    Group, ID, Post, Pre
```

```
# gets the pretest mean which we will use later
> mean(Pre)
```
Output of the mean

```
[1] 41.1

# gets the pretest standard deviation which we will use later
> sd(Pre)
```

Output of the standard deviation

```
[1] 8.238797

# regression model for group 1
> modelgroup1<-lm(Post~Pre)
> summary(modelgroup1)
```

Output of the regression model for group 1

```
Call:
lm(formula = Post ~ Pre)

Residuals:
    Min      1Q  Median      3Q     Max 
-7.3416 -3.3858  0.4726  2.9239  7.7911 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)  
(Intercept)   0.7219     8.1625   0.088   0.9317  
Pre           0.5664     0.1951   2.903   0.0198 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.822 on 8 degrees of freedom
Multiple R-squared:  0.513,     Adjusted R-squared:  0.4521 
F-statistic: 8.427 on 1 and 8 DF,  p-value: 0.0198
```

Now we will do the same for the control group, group 0.

```
# subsets data to only unclude group 0 and attaches it
> int3<-subset(intervention, Group==0)
> attach(int3)
```

Informational output

```
The following object is masked from int2:

    Group, ID, Post, Pre
The following object is masked from int:
```

```
    Group, ID, Post, Pre

# gets the pretest mean which we will use later
> mean(Pre)
```

Output of the mean

```
[1] 35.1

# gets the pretest stand dev which we will use later
> sd(Pre)
```

Output of the standard deviation

```
[1] 7.324995

#regression model for group 0
> modelgroup0<-lm(Post~Pre)
> summary(modelgroup0)
```

Output of the regression model for group 0

```
Call:
lm(formula = Post ~ Pre)

Residuals:
     Min       1Q   Median       3Q      Max
-10.8884  -3.1674   0.6936   3.7944   6.2870

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  55.9522     9.2373   6.057 0.000303 ***
Pre          -0.3918     0.2582  -1.518 0.167579
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.673 on 8 degrees of freedom
Multiple R-squared:  0.2235,    Adjusted R-squared:  0.1265
F-statistic: 2.303 on 1 and 8 DF,  p-value: 0.1676
```

In order to calculate to the Johnson Neyman calculations, we need the following information: the mean X value, standard deviation of the X value, slope and intercept for both groups, the sum of squares residual for the interaction and the critical F value.  Once you have that, there is a great Johnson-Neyman Calculator created by Dr. T. Chris Oshima which can be found at http://education.gsu.edu/coshima/stat3.htm.

From our output we can collect the needed information to perform the calculations.

| | Group 0 | Group 1 |
|---|---|---|
| n | 10 | 10 |
| X mean | 35.1 | 41.1 |
| SD for X | 7.32 | 8.24 |
| Slope | -.3918 | .5664 |
| intercept | 55.9522 | .7219 |

Residual Sum of Squares: 443.50
Critical F: 3.633

From the Johnson Neyman calculator we obtain the following
XL:- 0.59546302   XU: 0.61173388

We can conclude that for individuals having a pretest score of less than -0.595, the intervention was not effective and for those with a pretest score greater than .612 the intervention was effective. There was no statistical difference between the intervention and control for students with pretest scores between -0.595 and .612.

# Conclusion

R is a free program that is useful when going a variety of statistical analysis. R might seem overwhelming and have a rather steep learning curve if you have no programming background. However, once you get the hang of it, it can be pretty a pretty easy yet powerful program. As with anything, it takes practice to become comfortable with using it. A good suggestion would be to try to use it on small problems and exercises along with a program that you are familiar with. This will allow you to ease into the program and give you confidence in your skills when your answers are verified with the other software. Don't get frustrated; often mistakes are due to improperly typing in a command or variable name. There are also a lot of websites to help you with basic commands in R as well as take you deeper into what R can do.

## References

Available CRAN packages by name (n.d). Retrieved from http://cran.r-project.org/web/packages/available_packages_by_name.html

Graphpad Software, QuickCal (2004, December) Retrieved from http://graphpad.com/quickcalcs/posttest1.cfm.

Kabacoff, R.I. (2012) Quick R. Retrieved from http://www.statmethods.net/

Oshima, T.C (n.d) Bryant Paulson calculator. Available from http://education.gsu.edu/coshima/statistics_2.htm

Oshima, T.C (n.d) Johnson Neyman calculator. Available from http://education.gsu.edu/coshima/statistics_2.htm

R Studio (2013) Retrieved from www.rstudio.org

Tabachnick, B. G., & Fidell, L. S. (2001). Multivariate statistics. *Needham Heights, MA: Allyn* &.Bacon Boston

The R project for Statistical Computing (2013, May). Retrieved from http://www.r-project.org/.