HYDROINFORMATICS FALL 2015:

PRE-CLASS EXERCISE

LEARNING OBJECTIVES

- Explain the role of R in the Hydroinformatics course and in the broader field of water resources
- Use R to load, examine, and visualize data
- Identify resources for becoming familiar with R and future reference

R is a free software language and environment that is commonly used for statistical computing and data analysis. In addition to Python, we will be using R to explore, analyze, and visualize hydrologic data during this course. R can also be used for building models, image analysis, and a wide variety of statistical tools.

For an introduction to and background of R, please visit: https://www.r-project.org/about.html

INSTALLATION

Prior to class, please install R on your personal laptop by choosing the version corresponding with your operating system: https://cran.cnr.berkeley.edu/

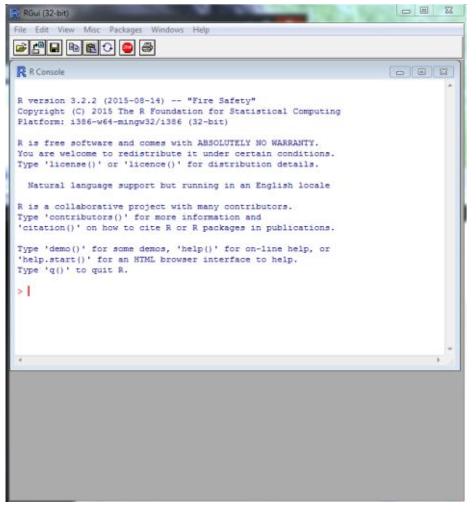
Windows:

- Download R for Windows > install R for the first time
- Run the installation wizard, and accept all defaults

Mac:

- Download R for (Mac) OS X > R-3.2.2.pkg or R-3.2.1-snowleopard.pkg depending on the version of your OS.

Upon successful installation, you will see the following home screen (console) when you open R.



Getting Started

Individual lines of code can be written directly in the R console one at a time, while multiple lines can be written, saved in a script, and run sequentially. If you need to repeat or modify a previous command in the console, use the up arrow to display recently typed lines instead of retyping it.

To create a new script:

- File>New Script

To run a script:

- Edit>Run All or "Run line or selection"

To assign a variable (e.g., to set the value of x to be 10), use the following syntax:

- x<-10

IMPORTANT: Commands and variables are case sensitive. This means that "x" is not the same as "X" and the command "scan" is not the same as "Scan."

To use a command, use the following syntax:

- samplevariable <-command(argument1, argument2, argument3)

Many commands have arguments that you may need to or want to specify. A list of resources with useful commands is included at the end of the exercise.

OTHER IMPORTANT NOTES:

To end a statement in your script, simply close the bracket on your command, and do not end a line with an operator. For example if you have the script:

```
a<-1+2
+4

b<-1+2+
3
```

Variable “a” will equal 3. (The +4 on the next line is not associated with the assignment of a, because the first line does not end with an operator). Variable “b” will equal 6 because the first line ends with an operator.

To add a comment to your script, simply type “#,” and everything on that line will be read as a comment, instead of a command or a variable. Comments will be very useful in making your code more readable, debugging code, when working with others who are unfamiliar with your code, or if you need to make changes to code later.

EXERCISE– FAMILIARIZING YOURSELF WITH R

This exercise will help you become familiar with the environment of the R and a handful of its abilities. This exercise explores and plots a time series of the lake levels of the Great Salt Lake near Saltair.

Data can be loaded into the R environment using commands such as read.csv(), read.table(), or a number of other methods (see Input and Output section of the R Reference Card). You can also add data manually. This exercise will use data from the USGS gauge near Saltair to examine changes in annual lake elevation.

First, use the command read.csv() to read data provided in the .csv titled “GSL_Elevation” and assign the data in the column “Year” to the variable “Year.”

```
> loadedData<-read.csv("E://University of Utah - Coursework//FALL - 2015//Hydroinformatics//GSL_Elevation.csv",header=T)
> Year<-loadedData$Year
> Year
 [1] 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000
[12] 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011
[23] 2012 2013 2014
> |
```

To demonstrate inputting data manually, load the elevation data from the .csv file: Type scan(), hit return,

```
> Elevation<-scan()
1: |
```

then copy(1) and paste(2) data from a spreadsheet, and hit return again(3) to finish.

(1)

390	4203.422
991	4201.678
392	4200.503
393	4199.718
394	4199.46
395	4198.819
396	4199.308
397	4199.953
398	4201.596
399	4202.96
000	4202.517
001	4200.876
002	4198.971
003	4197.144
004	4195.402
005	4195.65
006	4196.667
007	4196.703
008	4195.363
009	4194.89
010	4194.825
2011	4196.052
012	4197.836
013	4196.106
014	4194.479

(2)

```
> Elevation<-scan()
1: 4203.422
2: 4201.678
3: 4200.503
4: 4199.718
5: 4199.46
6: 4198.819
7: 4199.308
8: 4199.953
9: 4201.596
10: 4202.96
11: 4202.517
12: 4200.876
13: 4198.971
14: 4197.144
15: 4195.402
16: 4195.65
17: 4196.667
18: 4196.703
19: 4195.363
20: 4194.89
21: 4194.825
22: 4196.052
23: 4197.836
24: 4196.106
25: 4194.479
26:
```

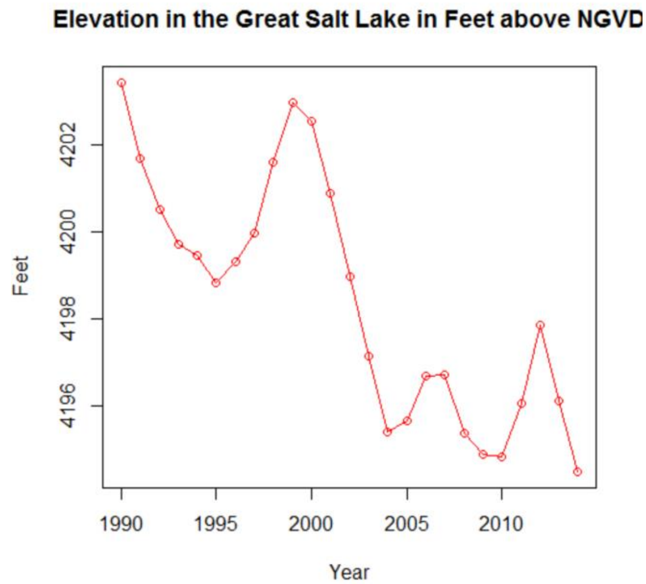
(3)

```
Read 25 items
>
```

```
> Elevation
 [1] 4203.422 4201.678 4200.503 4199.718 4199.460 4198.819 4199.308 4199.953 4201.596 4202.960 4202.517 4200.876 4198.971 4197.144
[15] 4195.402 4195.650 4196.667 4196.703 4195.363 4194.890 4194.825 4196.052 4197.836 4196.106 4194.479
>
```

Now, plot the elevation versus time, and use the optional arguments (described in the resources below) to customize. An example of a plot with red lines and markers is shown below.

```
#Plot the Elevation on the y axis and Year on the x axis
#Optional arguments include the plot type, color, and labels of the axes
plot(Year,Elevation, type="o", col="red", xlab="Year", ylab="Feet")
#Declare the plot title
title(main="Elevation in the Great Salt Lake in Feet above NGVD")
```



Voila! A plot of a time series of lake levels in the Great Salt Lake. Feel free to explore the options available for customizing your plot.

Calculate some basic statistics using the mean, max, and min commands:

```
> mean(Elevation)
[1] 4198.436
> max(Elevation)
[1] 4203.422
> min(Elevation)
[1] 4194.479
>
```

You may want to obtain information about a variable. For example, if you would like to know the number of items in the Great Salt Lake Elevation vector, use the "length" command:

```
> length(Elevation)
[1] 25
>
```

Useful Commands And Resources

Short list of useful commands with examples:

http://www.personality-project.org/r/r.commands.html

R Reference Card:

https://cran.r-project.org/doc/contrib/Short-refcard.pdf

Exercises that Practice and Extend Skills with R:

http://maths-people.anu.edu.au/~johnm/courses/r/exercises/pdf/r-exercises.pdf

Producing Simple Plots in R

http://www.harding.edu/fmccown/r/

There are also example datasets that are available through several libraries (e.g., the MASS (Modern Applied Statistics with S)) that can help you to get acquainted with R.