Lab 1. Introduction to R

`www.nmt.edu/~olegm/382labs/Lab1r.pdf`

Note: the menus and other things you will read or type on the computer are in italics. All the files mentioned can be found at `www.nmt.edu/~olegm/382labs/`

In this Lab, we will learn how to use `R` for working with data and conducting some simple statistical analyses.

1 General description

`R` is an advanced language designed for statistical analysis, covering the wide range of statistical analyses and high-resolution graphics. It is popular among statisticians and data analysts. It uses a command-line interface, or you can also run scripts.

A free copy of `R` can be downloaded from `https://www.r-project.org/`

Your lab instructor will show you how to start `R`.

`R` environment consists of *Console* window and zero or more *Graphics* windows. You can save your `R` workspace by going to *File* menu[1]. You can export graphics by using *File* → *Save As* when you're in a Graphics window.

Pressing an *up arrow* key will recall previous commands.
To get help on an `R` function, for example, `sd`, type `?sd`.
To get the list of all current variables in your workspace, type `ls()`.
To quit `R`, type `q()`.

It's a good idea to keep all your code in one file. For Windows, we recommend `Notepad++` editor, as it allows multiple tabs and highlights `R` syntax.

2 Data management

2.1 Hand entry

The lab instructor will pass around a data sheet to record some personal variables of interest. You will then enter them into `R`.

In `R`, the data are stored in *objects*, which can be single variables, vectors, matrices and so on. A typical way to store statistical data for analysis is a

[1]These instructions are primarily geared towards Windows machines, you might need to refer to online sources for Mac or Linux.

data frame. A data frame contains individual variables.

The variables are primarily of these types:

- *Numerical* (or quantitative),
- *Text* (or categorical), and
- *Logical* or Boolean.

Here's an example of entering some data into `R`:

```
GPA1 = c(2.5, 3, 3.8, 2.7, 3.2)         # this is a vector of length 5
eyec1 = c("Green", "Brown", "Hazel", "Brown", "Gray")
   # note that variable names and text values are case-sensitive!
interest1 = c(2, 5, 4, 4, 3)
mydata1 = data.frame(GPA = GPA1, eye.color = eyec1, interest = interest1)
mydata1
```

You do not have to put all the variables into the data frame, but many analyses become more convenient with data frames. To use a variable from the data frame, you can use `$` notation:

```
mydata1$eye.color
mydata1$GPA < 3     # this produces a vector of Boolean (TRUE/FALSE) values
```

Missing data (if there are any) can be entered as `NA`.
Now, create a data frame with all the data from the personal data sheet. Enter eye color and gender as text and the rest of the variables as numerical.

2.2 Input/output

Very frequently, we'll be using the pre-fab data sets, or import data sets from other sources. It is convenient to import *csv* (comma-separated), or tab-separated text data.

Load the Earthquakes data set `earthq.csv` (see, for example, `https://earthquake.usgs.gov/earthquakes/browse/`)
We will use this data set for some analysis below.

```
setwd('C:/local/classes/382/Labs')
    # sets up the working directory, yours will be different!
    # make sure to use forward slash / instead of backward slash
eq = read.csv("earthq.csv")
head(eq)
dim(eq)
```

You will get a data frame called `eq` with 766 cases and 9 variables.

For output, you can use

```
write.csv(mydata1, "mydata.csv")
```

2.3 Data manipulation

It's easy to apply various calculations to vectors.

```
logM = log(eq$Mag)   # natural log.
                     # Notice you can shortcut the name "Magnitude"
sL = sin(eq$Lat*pi/180)
one = sin(eq$Lat*pi/180)^2 + cos(eq$Lat*pi/180)^2
```

You can also subset your dataset according to some criteria.

```
var1 = eq[,5]       # this is the 5th column of your dataset
eq1 = eq[1:50,]     # these are the first 50 rows
bigOnes = eq[eq$Mag > 5,]
 # this selects the rows according to a given criterion

rep(1,10)                 # this will create a vector repeating a given value
1:5
seq(1,5, by = 0.2)        # this will create a sequence of numbers
```

3 Graphics

R offers a variety of graphs. One commonly used type is a **histogram**, which shows the groups on the x-axis and frequencies (counts), or percentages on the y-axis. A graph popular for describing relationships between **two** variables is a **scatterplot**.

```
attach(mydata1)
   # we will use this so we don't have to type mydata1$... every time
hist(GPA)                       # histogram
hist(GPA, breaks=seq(1,4,0.5))    # this controls the bins of the histogram
plot(GPA, calc.grade)           # scatterplot
boxplot(GPA)
boxplot(GPA ~ gender)           # boxplots
```

4 Descriptive statistics

Descriptive statistics produce numerical summaries of a data set.

```
mean(GPA)   # mean or average
median(GPA)
var(GPA)    # variance
sd(GPA)     # standard deviation = sqrt(variance)
```

```
mean(log(GPA))
detach(mydata1)
 # now we will close the shortcut access to "mydata1",
 # so we can work on a different dataset
```

Also, find average GPA for each gender. Do males or females have higher average GPA?

Problem 1. Use the earthquakes data set.

a. Make graphical and numerical summaries for variable `Magnitude`. Describe in words what they tell you. Are strong (Mag. > 6) quakes frequent?

b. Make a scatteplot of Magnitude vs Depth of earthquakes. Do these variables appear to be connected in some way?
To make the graph more readable, make a scatterplot of log(Magnitude) vs log(Depth).

c. Try and produce a "map" (of course, it has to be a flat representation of the Earth's surface) of the quakes' locations. Describe what you see. What are the two geographical regions with a lot of earthquake activity?

Problem 2. The file `ttucson.csv` contains average daily temperatures (in Celcius) for May-August 2007 at Tucson, AZ.

a. Convert temperatures to Fahrenheit ($^oF = {^oC} \times 1.8 + 32$).

b. Are there any unusually high/low observations?

c. Find the mean temperature and standard deviation (both Celcius and Fahrenheit). Do they satisfy the conversion formula in (a.)?

d. Compare with another location: Eugene, OR (`teugene.csv`).
Compare the temperatures both graphically and using descriptive stats.
[**Hint:** Put histograms one above the other using `par(mfrow=c(2,1)); hist(var1); hist(var2)`, also make sure that the scales are the same.]