BIOS 545 Introduction to R Programming
2015 Spring Semester Final Exam

(Open notes, slides, lecture material, and your previous homework assignments but no email, texting, or internet chat programs)

INSTRUCTIONS:

Answer all three questions. All work and code must be your own. Save all code into a single text file called BIOS545R_YOURNAME_final.R where YOURNAME should be replaced by your Last and First name. Email the file to dvandom@emory.edu, hao.wu@emory.edu and wsp@emory.edu.

Total is 100 points. Partial credit will be given.

QUESTION #1) (30 points)

In Data Mining it is common to take a data frame and divide it into a training data set and a test dataset. Then you build the model using the training data and then try it out on the test data. Write a function that takes as arguments a data frame and an argument, (a value between 0 and 1), that represents the percentage of row observations to be returned in the training data frame. The remaining records will then be assigned to the test data frame.

A common default for the training set is for 80% of the total number of data frame records to be assigned to the training data frame so the remaining 20% will go into the test data frame. Your function will return a list with the first element being the training data frame and the second being the test data frame.

```
mysampler <- function(df, prop=0.8) {

# df is an input data frame
# prop is the proportion of rows to assign to the training data frame

}
```

As an example:

```
mysets <- mysampler(mtcars,0.8)

# Let's check how many rows in each were returned

str(mysets,1)
List of 2
 $ train:'data.frame':    25 obs. of  11 variables:
 $ test :'data.frame':     7 obs. of  11 variables:

str(mysets,1)
List of 2
 $ train:'data.frame':    16 obs. of  11 variables:
 $ test :'data.frame':    16 obs. of  11 variables:
```

Steps To Take: Figure out how many rows total there are in the input data frame **df** and use **prop** to determine how many of those rows to sample from **df.** Place the resulting number of rows into the training data frame. Then put the remaining rows into the test data frame. These two data frames are mutually exclusive so no records in the training data frame should be in the test data set. Try your function out on mtcars to see if you get answers similar to that above.

QUESTION #2) (40 points)

In this question you will write functions to compute the variance, covariance and correlation of an input vector.

1. Do not use the built in **"var"** (variance function), **"sd"**, or **"cov"** (covariance) functions though you can use the **"mean"** function.
2. Try to vectorize your computations. Using loops is allowed but to receive full credit you should use vectorization where possible.

a) Variance (10 points)

In a sample, the **variance** is the average squared deviation from the sample mean, as defined by the following formula:

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

where x is a vector of length "n" and $\bar{x}$ is the mean of the sample vector. Write a function called **sampvar** that implements this formula. As an example:

```
set.seed(123)
x <- rnorm(30,10)

sampvar(x)
[1] 0.962
```

b) Covariance (15 points)

The **covariance** of two variables, (*x* and *y*), in a data sample measures how the two are linearly related. A positive covariance would indicate a positive linear relationship between the variables, and a negative covariance would indicate the opposite. The **sample covariance** is defined in terms of the sample means as:

$$Q = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$$

$\bar{x}$ and $\bar{y}$ represent the mean of x and y, respectively, and are vectors of length "n". Write a function called "**sampcov**" in R that, given vectors x and y, implements the above formula. Thus, it will return the covariance of two vectors x and y. As an example using the following vectors:

```
set.seed(123)
x <- rnorm(1000)

set.seed(456)
y <- rnorm(1000)

sampcov(x,y)
[1] 0.0134
```

c) Pearson's Correlation Coefficient (15 points)

We can compute the **Pearson's correlation coefficient** by using the formula below where x and y are vectors of length "n". s_x and s_x are the standard deviations of x and y respectively. Note that the standard deviation is the square root of the sample variance.

$$r = \frac{cov(x,y)}{s_x * s_y}$$

where $s_x = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}}$ and $s_y = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \bar{y})^2}{n-1}}$

Write a function called "**mypearson**" that implements the formula for ***r***. Take advantage of the functions you wrote in sections a and b to simplify your work. As an example:

```
set.seed(123)
x <- rnorm(1000)

set.seed(456)
y <- rnorm(1000)

mypearson(x,y)
[1] 0.0137
```

QUESTION #3 (30 points)

Consider the function *f(x)* defined as follows. This type of function is known as a "piecemeal" function.

$$f(x) = \begin{cases} \sin(x), & x \leq 0 \\ x^2, & 0 < x \leq 1 \\ x^3, & > 1 \end{cases}$$

Create a function in R that implements this definition. The function will take the following inputs:

1. **xvals**: a single number or a vector for *x*.
2. **plot**: a TRUE/FALSE indicator. (Default is TRUE). If TRUE, a scatter plot of *f(x)* versus *x* will be shown. Use the plots below as a model to emulate.
3. **color:** a color indicator (default is blue)

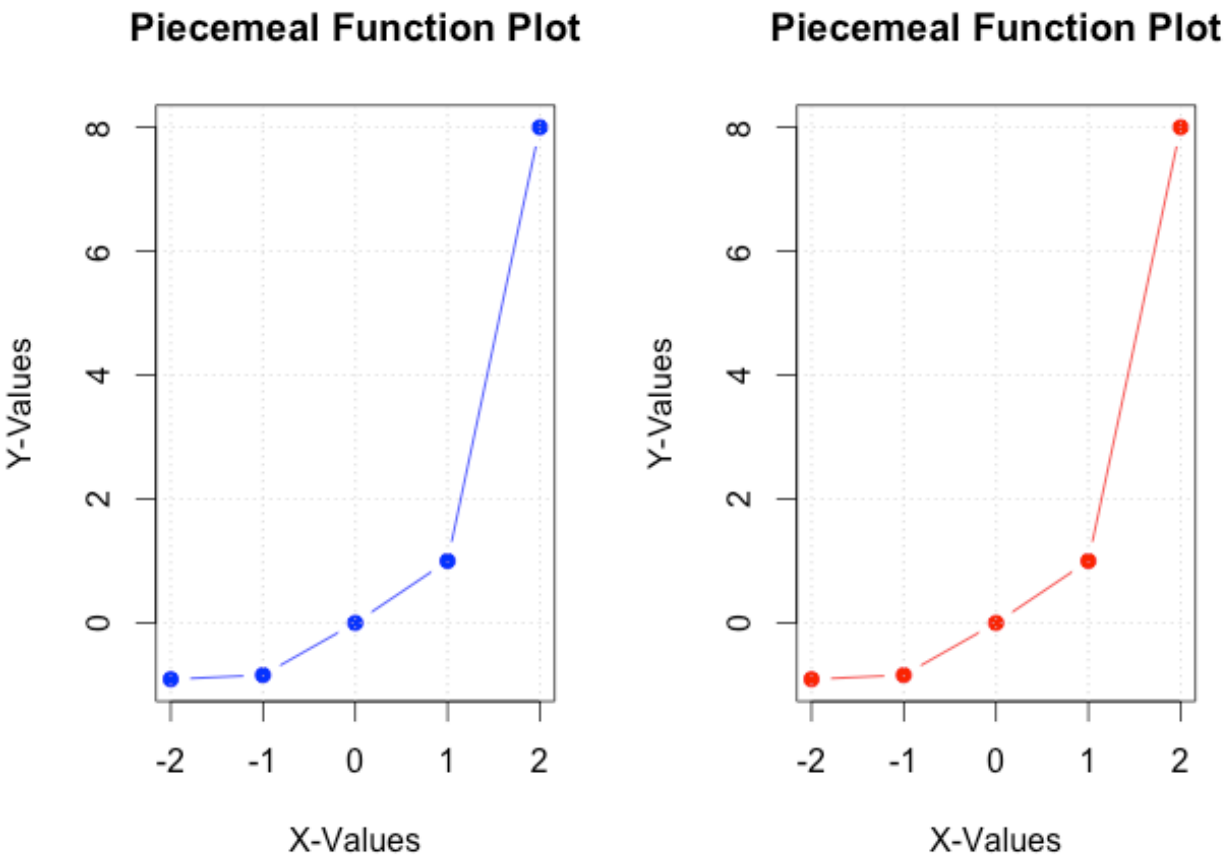
The output of the function is a number or vector for the *f(x)* values. The length of the output should equal to the length of the input. As an example, you should get following results given the following inputs:

```
myfunc(-2:2, F)
[1] -0.9092974 -0.8414710  0.0000000  1.0000000  8.0000000

par(mfrow=c(1,2))

myfunc(-2:2, T)

myfunc(-2:2, T, "red")
```



To get the full credit of the question:

1. Your function has to work with vectors of different lengths and different values.
2. The figure needs to have the same line type, xlab, ylab, and title as shown in the example.