BIOS560R Introduction to R Programming
2014 Spring Semester Final Exam

(Open notes but no email or texting)

INSTRUCTIONS:

Answer all three questions. All work and code must be your own. Save all code into a single text file called 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) (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 #2 (30 points)

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

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

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**: optional TRUE/FALSE indicator. (This should be equal to TRUE by default). If plot=TRUE, a scatter plot of *f(x)* versus *x* will be shown.

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.

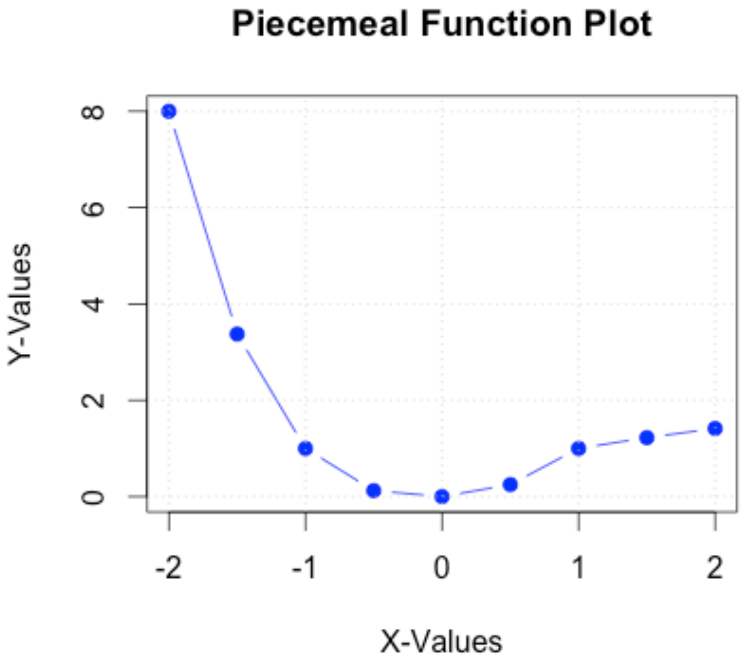
As an example, you should get following results given the following inputs:

```
myfunc(-2:2, plot=FALSE)

[1] 8.000000 1.000000 0.000000 1.000000 1.414214

myfunc(seq(-2,2, by=0.5))  # Note that this will create a plot

[1] 8.000000 3.375000 1.000000 0.125000 0.000000 0.250000 1.000000 1.224745
[9] 1.414214
```



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.

<u>QUESTION #3 (30 points)</u>

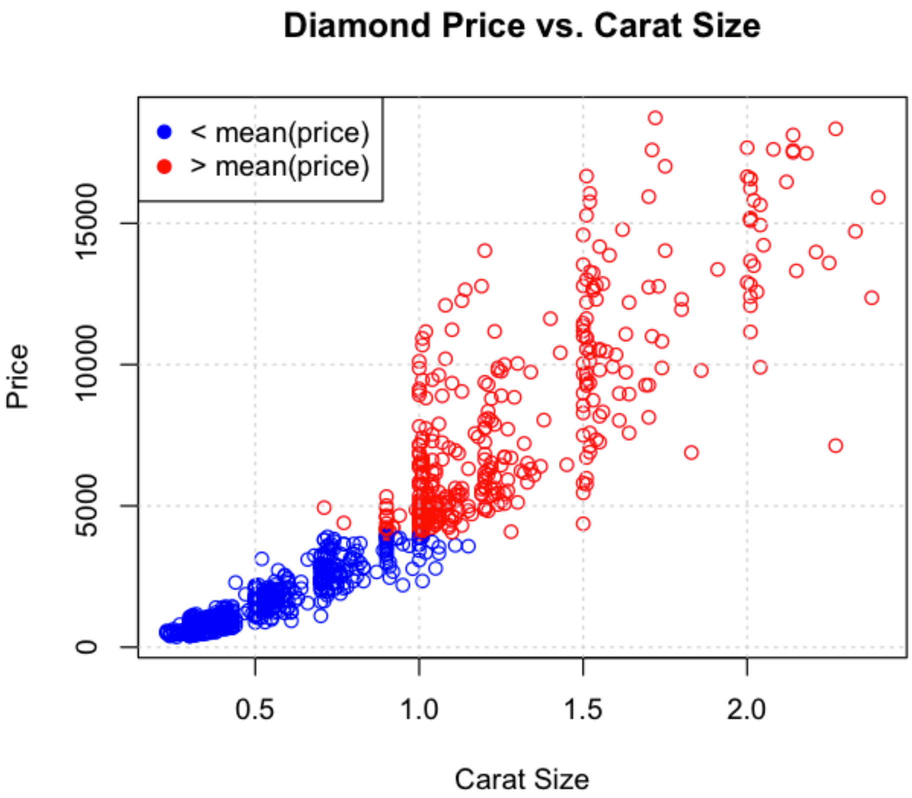
Read the following .csv file in

```
url = "http://steviep42.bitbucket.org/bios560rs2014/DATA.DIR/my.diamonds.csv"
myd = read.csv(url)
```

Create a function that presents a plot of price (y axis) vs. carat size (x axis) such that the points lying below the average price have the color blue and the points lying above the average price are in red. Use base graphics to create this function.

```
myplotter(mydf=myd)
```

To get full credit the figure generated from the function must have similar point type, colors, labels and titles as shown in the example.


Diamond Price vs. Carat Size
● < mean(price)
● > mean(price)
Price
Carat Size