# BIOS560R - Final

Pittard

Due by 11:59 PM 12/06/2013 - Open Notes and Internet but no email

## 1 Function Writing

Write a function called degs2rads(x) that, given a measurement in degrees, will convert it to radians. The formula to convert from degrees to radians is x $*$ pi \180 where x is a number in degrees. As an example convert 200 degrees to radians.

```
> degs2rads(200)
[1] 3.49
```

## 2 Problem 1 - Aggregation - 35 points

Please read in the following data set as indicated. The resulting data frame contains information on several important characteristics of diamonds. There are 53,940 rows in this data frame. You might want to familiarize yourself with the layout and column names using some of the functions we've discussed in class. In all questions please present the R statements used to arrive at the answer.

```
> myd = read.table("http://www.bimcore.emory.edu/BIOS560R/DATA.DIR/diamonds.csv",header=T,sep=",")
> names(myd)
 [1] "carat"   "cut"     "color"   "clarity" "depth"   "table"   "price"   "x"       "y"       "z"
> nrow(myd)
[1] 53940
> head(myd)
  carat       cut color clarity depth table price    x    y    z
1  0.23     Ideal     E     SI2  61.5    55   326 3.95 3.98 2.43
2  0.21   Premium     E     SI1  59.8    61   326 3.89 3.84 2.31
3  0.23      Good     E     VS1  56.9    65   327 4.05 4.07 2.31
4  0.29   Premium     I     VS2  62.4    58   334 4.20 4.23 2.63
5  0.31      Good     J     SI2  63.3    58   335 4.34 4.35 2.75
6  0.24 Very Good     J    VVS2  62.8    57   336 3.94 3.96 2.48
```

## 2.1 20 points

Summarize the carat size and diamond price in terms of the cut. The summary should present the range, (minimum and maximum), of carat size and price. The resulting output should look like:

```
        cut carat.1 carat.2 price.1 price.2
1      Fair    0.22    5.01     337   18574
2      Good    0.23    3.01     327   18788
3     Ideal    0.20    3.50     326   18806
4   Premium    0.20    4.01     326   18823
5 Very Good    0.20    4.00     336   18818
```

## 2.2 15 points

Present the minimum price of diamonds as organized by color and cut categories. So,for each combination of color and cut present the the minimum diamond price. The correct output will look like:

```
      color       cut price
1      D      Fair   536
2      E      Fair   337
3      F      Fair   496
4      G      Fair   369
5      H      Fair   659
6      I      Fair   735
7      J      Fair   416
8      D      Good   361
9      E      Good   327
10     F      Good   357
11     G      Good   394
12     H      Good   368
13     I      Good   351
14     J      Good   335
15     D     Ideal   367
16     E     Ideal   326
17     F     Ideal   408
18     G     Ideal   361
19     H     Ideal   357
20     I     Ideal   348
21     J     Ideal   340
22     D   Premium   367
23     E   Premium   326
24     F   Premium   342
25     G   Premium   382
26     H   Premium   368
27     I   Premium   334
28     J   Premium   363
29     D Very Good   357
30     E Very Good   352
31     F Very Good   357
32     G Very Good   354
33     H Very Good   337
34     I Very Good   336
35     J Very Good   336
```

# 3 Benchmarking - 40 points

Define the following function within R. Given a matrix it will compute the sum of each row and return those sums in a vector.

```
mysum <- function(mymat)  {
  sumsofrows = vector()
  for (ii in 1:nrow(mymat)) {
    sumsofrows[ii] = sum(mymat[ii,])
  }
  return(sumsofrows)
}

# So given the following matrix

set.seed(123)
mymat = matrix(rnorm(100),10,10)

# We would get the following output:

mysum(mymat)
 [1]  0.4687 -2.4977  1.0609  2.1567  0.1325  3.9579  3.8464 -2.2222  0.0162  2.1210
```

## 3.1 15 points

Use one of the timing functions/methods we learned about in class to see how long it takes to execute the mysum function using the following matrix. Present the R code that accomplishes the timing as well as the results.

```
set.seed(123)
mymat = matrix(rnorm(1000000),1000,1000)
```

## 3.2 25 points

Next, Your job is to find a way to improve the mysum function in a way that is demonstrably faster. You can create an altered version of the function and/or find a preexisting R command/function that would be faster. Check the lecture on matrices for some hints. Use the benchmark function to present the timing of your solution vs. mysum. Refer to the lecture notes for a refresher on how to use the benchmark function. Select 20 replications. Provide a brief explanation (a few sentences) on why your improvement is faster.

# 4 Sweave - 25 points

With respect to your solution for Problem one please create an Sweave document that will dynamically generate the two answers. You can use the template document at http://www.bimcore.emory.edu/BIOS560R/SUPP.DIR/template.Rnw

We should be able to paste your text into RStudio and compile the PDF without errors.