DSCI 210: Data Science – R Takehome Name(s):_____
Spring 2016
Points: 100 points _____

Consider once again the CitiBike System data provided on the following website.

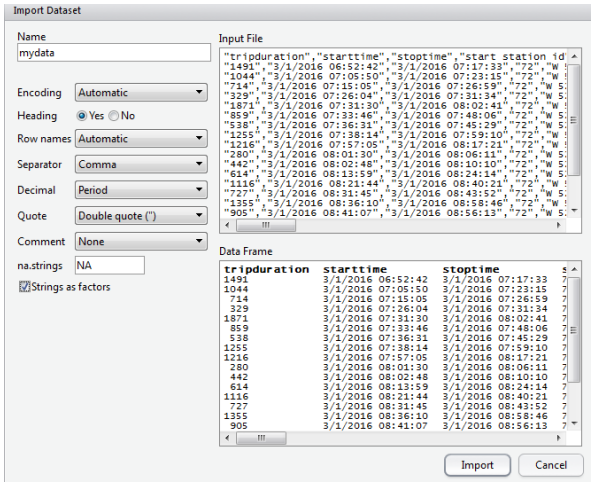
Website: https://s3.amazonaws.com/tripdata/index.html

Every person in class has been assigned a specific year/month dataset for this take-home portion of the exam. Download the following Excel file to determine which year/month you should use. Unless otherwise noted, you should use your assigned year/month for all questions centered on the CitiBike data.

Note: For those choosing to work with a partner, you can choose the most recent year/month dataset.

Assigned Year/Month: http://course1.winona.edu/cmalone/dsci210/datasets/Assigned_BikeData.xlsx

I assigned myself the March 2016 dataset. Use the Import Dataset feature in R to load the dataset into R. The name given to the data.frame in R for March 2016 Citibike data is mydata.



1. What is the dimension of your data.frame?

My Output:	Paste Your Output Here:
> dim(mydata) [1] 919921 15	

Explain what these two numbers are telling you about your dataset. (2 pts)

2. Next, let us understand the structure of mydata. Type str(mydata).

```
> str(mydata)
'data.frame':   919921 obs. of  15 variables:
 $ tripduration           : int  1491 1044 714 329 1871 859 538 1255 1216 280 ...
 $ starttime              : Factor w/ 691366 levels "3/1/2016 00:00:23",..: 1245 1475 1
 $ stoptime               : Factor w/ 690619 levels "3/1/2016 00:02:52",..: 1525 1634 1
 $ start.station.id       : int  72 72 72 72 72 72 72 72 72 72 ...
 $ start.station.name     : Factor w/ 473 levels "1 Ave & E 15 St",..: 437 437 437 437
 $ start.station.latitude : num  40.8 40.8 40.8 40.8 40.8 ...
 $ start.station.longitude: num  -74 -74 -74 -74 -74 ...
 $ end.station.id         : int  427 254 493 478 151 520 533 426 325 500 ...
 $ end.station.name       : Factor w/ 482 levels "1 Ave & E 15 St",..: 86 405 440 10 11
 $ end.station.latitude   : num  40.7 40.7 40.8 40.8 40.7 ...
 $ end.station.longitude  : num  -74 -74 -74 -74 -74 ...
 $ bikeid                 : int  23914 23697 21447 22351 20985 15557 22638 23864 17821
 $ usertype               : Factor w/ 2 levels "Customer","Subscriber": 2 2 2 2 2 2 2 2
 $ birth.year             : int  1982 1978 1960 1986 1978 1975 1993 1988 1982 1982 ...
 $ gender                 : int  1 1 2 1 1 1 1 2 1 1 ...
```

   a. For March 2016, the number of unique stations in which a bike rental originated is 473. How many factor levels does your dataset have for start.station.name? (2 pts)

   b. Use the unique() and length() function in R to verify this count. Provide a copy of the code used to verify below. (3 pts)

My Code:

3. Next, use the table() function in the following way.

```
> table(mydata$start.station.id)
```

   What output does this produce? Why might this output be useful? Discuss in detail. (2 pts)

4. What information does the following provide? Discuss. (2 pts)

```
> max(table(mydata$start.station.id))
```

5. Next, let us consider application of the which() function in R.

```
> which(table(mydata$start.station.id) == max(table(mydata$start.station.id)))
519
284
```

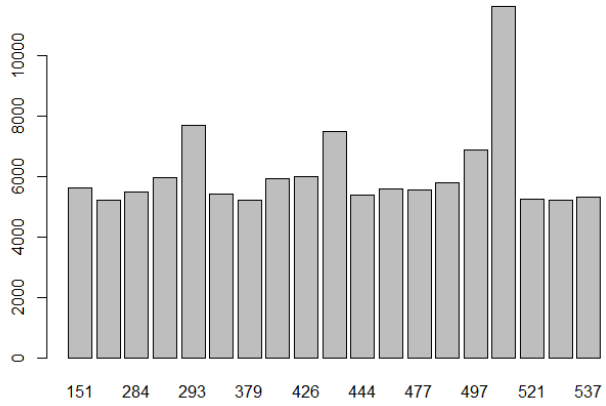
   Notice that two values are returned, i.e. 519 and 284. Your values may or may not be different. What are these values? How might they be useful? (4 pts)

Note: The following code may be useful in trying to answer this question.

```
> counts <- table(mydata$start.station.id)
> counts["519"]
  519
11634
> counts[284]
  519
11634
```

6. The following code is used to identify which starting stations have more than 5000 bike rentals. Make a similar plot for your data. (4 pts)
   Comments
   1) Your plot should indicates the top few start stations. You may have to change the 5000 value – I just want the top 20 or so stations for which bike rentals originate.
   2) My barplot does not show all the station labels. Figure out how to flip the labels so that all station labels appear on your graph.

```
> counts <- table(mydata$start.station.id)
> which_ones <- which(counts > 5000)
> barplot(counts[which_ones])
```



7. Consider the following use of the aggregate() function. What information about the data is gained by looking at this function? (4 pts)

```
> counts<-aggregate(mydata$start.station.id,by=list(mydata$start.station.id,mydata$end.station.id),function(x) { return( length(x) ) } )
```

Note: The following simplified version may be useful in trying to answer this question.

```
> data<-data.frame(x=c(1,1,1,2,2,2,3,3,3),y=c(1,1,3,2,1,2,1,2,3))
> counts<-aggregate(data$x, by = list(data$x , data$y) , function(x) { return( length(x) ) } )
```

8. Run the following on the counts obtained in the problem above.

```
> counts[order(-counts$x),][1:20,]
```

Provide a copy of the output returned. Explain what information is gained by this output. (3 pts)

The Fools Five Road Race is a major regional fundraising event for cancer. Fool's Five 2016 raised over $75,000 for cancer research. The race results for the 8k race will be considered here.

Race results are provided online

URL of dataset: http://course1.winona.edu/cmalone/dsci210/datasets/FoolsFive2016_Final.txt

9. Use the following outline of code to figure out how to read in the Fool's Five dataset directly into R. (5 pts).

```
temp<-tempfile()
  download.file( <web address url>, <local filename> , mode="wb")
  foolsfivedata<-read.csv( <local filename>, <other options>)
unlink(temp)
```

Note: If you cannot figure this out, then download the dataset onto your machine and use Import > Dataset to load the data in the usual way. Of course, you will not get credit for this question if this is done.

10. Run the str() function on the Fool's Five dataset. Provide a copy of the output (2 pts)

11. Run the following command. What does this produce? Explain. (3 pts)

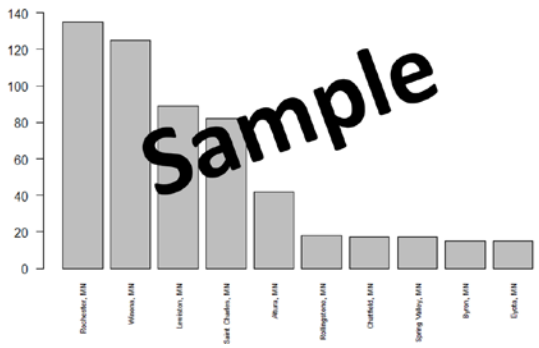
```
> sort(table(foolsfivedata$Hometown),decreasing=TRUE)[1:10]
```

12. Use **a single line of code** to create the following barplot. (5 pts)

My Code:

Your plot should:

1) Have labels for each bar
2) Have a y-axis, i.e. count axis, that exceed the tallest bar



13. Unfortunately, Time is being treated as a Factor in R.

```
> str(FoolsFive2016_Final)
'data.frame':   869 obs. of  8 variables:
 $ Rank     : int  1 2 3 4 5 6 7 8 9 10 ...
 $ Name     : Factor w/ 869 levels "Aaliyah Douangdy",..: 429 52 108 293 152 707 196 663 415 50 ..
 $ Bib      : int  2775 2602 2498 3307 2266 2002 2635 2820 2828 3315 ...
 $ Distance: Factor w/ 1 level "8K": 1 1 1 1 1 1 1 1 1 1 ...
 $ Time     : Factor w/ 745 levels "0:27:29","0:30:32",..: 1 2 3 4 5 6 6 7 8 9 ...
 $ Hometown: Factor w/ 98 levels "Adams, MN","Albert Lea, MN",..: 76 76 64 56 76 97 89 1 76 76 ..
 $ Division: Factor w/ 26 levels "FEMALE 1-14",..: 16 16 15 19 17 19 17 15 17 21 ...
 $ DivRank : int  1 2 1 1 1 2 2 2 3 1 ...
```

The following function would prove helpful in dividing up the time into hours, minutes, and seconds.

```
> strsplit(as.character(FoolsFive2016_Final$Time),":")
```

As is often the case, the object returned in a list. Use R to verify that indeed the object returned from this function is a list. What code did you use for this? (3 pts)

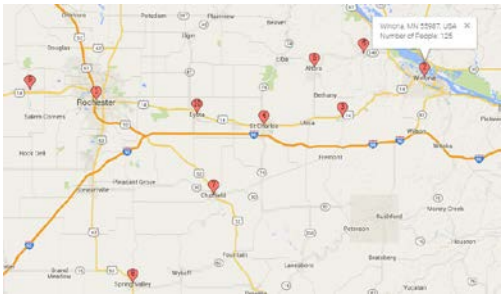
14. The following can be used identify the various components of the list obtained above.

```
> as.numeric(strsplit(as.character(FoolsFive2016_Final$Time),":")[[1]][1])
[1] 0
> as.numeric(strsplit(as.character(FoolsFive2016_Final$Time),":")[[1]][2])
[1] 27
> as.numeric(strsplit(as.character(FoolsFive2016_Final$Time),":")[[1]][3])
[1] 29
```

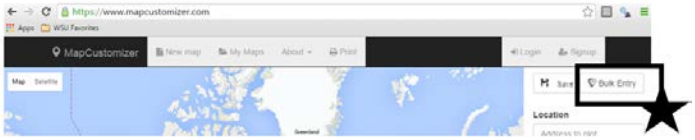
Suppose the desired units of measurement for Time is minutes. How would you convert these values to Time (in minutes)? (2 pts)

Next, create a map of the Hometowns for the top 10 Hometown locations for the Fool's Five 2016 race. This will be done using as the following website as they allow for descriptors to be added to each plotting location. The number of people from each location should be added as a descriptor.

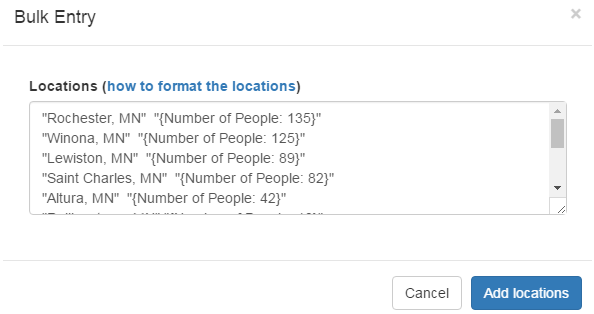
Website: http://www.mapcustomizer.com



Use Bulk Entry near the upper-right corner of their site to map multiple locations.



The Bulk Entry form requires one location per line. The descriptor should be placed inside curly brackets, i.e. "{ }".



15. Use the following example code to figure out how to create a data.frame in R that looks like the one provided below.

Code that should help you in creating the data.frame to the right.

```
> counts<-sort(table(mydata$Hometown),decreasing=TRUE)[1:10]
> names(counts)
 [1] "Rochester, MN"     "Winona, MN"        "Lewiston, MN"
 [6] "Rollingstone, MN"  "Chatfield, MN"     "Spring Valley, MN
> paste("{Number of People: ",as.character(counts),"}",sep="")
 [1] "{Number of People: 135}" "{Number of People: 125}" "{Numb
 [5] "{Number of People: 42}"  "{Number of People: 18}"  "{Numb
 [9] "{Number of People: 15}"  "{Number of People: 15}"
```

The required format for the output data.frame in R

	City	Counts
1	Rochester, MN	{Number of People: 135}
2	Winona, MN	{Number of People: 125}
3	Lewiston, MN	{Number of
4	Saint Charles, MN	{Numb
5	Altura, MN	People: 42}
6	Ro	{Number of People: 18}
7	Cha d, MN	{Number of People: 17}
8	Spring Valley, MN	{Number of People: 17}
9	Byron, MN	{Number of People: 15}
10	Eyota, MN	{Number of People: 15}

Provide a screen shot of the data.frame you've created in R. (4 pts)

16. Try to copy and paste the data.frame from R into the Bulk Entry form on the http://www.mapcustomizer.com website. Why happens? (2 pts)

17. Use the following write.table() function to write the output data.frame you've obtain above into a *.txt file.

```
> write.table(output,file="C:/DSCI210/FoolsFive_CitiestoMap.txt",row.names=FALSE,col.names=FALSE)
```

Copy and paste the contents of this file into the Bulk Entry form on the http://www.mapcustomizer.com website. Provide a screen shot of your final map from the mapcustomizer.com website. (3 pts)

The goal of this problem is to understand the relationship between price of a used car as a function of its year and number of miles. The data for this example was pulled from FindCars.com and only cars from 2000-2010 were included and is available on the course website.

Note: There was an issue with the original file having hidden characters. This appears to be an issue only for MAC machines. An alternative version has been provided for those with a MAC.

Response: Price

Predictors: 1) Year
2) Number of Miles

Modeling Approaches

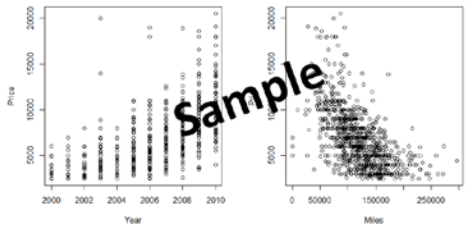
1) Regression Models
2) Loess Models

18. Use the following code to create the plot provided below.

```
> par(mfrow=c(1,2))
> par(pty="s")
> plot(FindCars$Year,FindCars$Price,xlab="Year",ylab ="Price")
> plot(FindCars$Miles,FindCars$Price,xlab="Miles",ylab ="Price")
```

Note: par(mfrow=c(1,1) and par(pty="m") will reset the graphics window to a single plot of maximal size.

Delete my sample plot and provide a screen shot of your plots in its place. (3 pts)



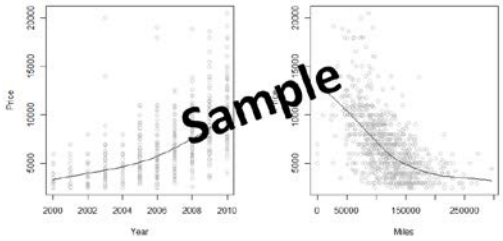
19. The following code can be used to add a regression line to the Price vs. Year scatterplot. Recreate the plots above and add a regression line to each plot. Paste a screen-shot the graphs with the regression lines added below here. (3 pts)

```
> plot(FindCars$Year,FindCars$Price,xlab="Year",ylab="Price",col="grey")
> abline(lm(FindCars$Price~FindCars$Year))
```

20. In what ways might a regression model fail us in predicting used car prices using age of car and mileage of car? Discuss. (3 pts)

21. The following code can be used to fit a loess smooth when you have one predictor. The span value in the scatter.smooth() function controls the amount of smoothing. The span value should be between 0 and 1 -- a value near 1 is similar to a regression model. You should tweak the span value to find a reasonable value for each plot. Provide a screen-shot of your final plots. Specify the values you decided upon for the span parameter. (4 pts)

```
> scatter.smooth(FindCars$Year,FindCars$Price,xlab="Year",ylab="Price",span=0.5,col="grey")
```



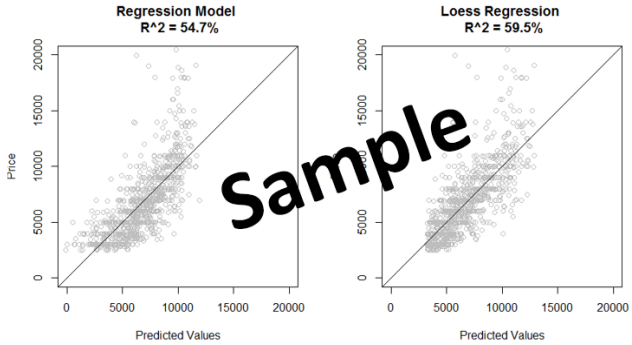
The following code can be used to obtain to fit

1) a regression model, named regression.model here
2) a loess model, named loess.model here

```
> #Fitting a regression model in R
> regression.model<-lm(Price~Year+Miles,data=FindCars)
> predicted.regression.model<-predict(regression.model,newdata=FindCars)
>
> #Fitting a lowess model in R
> loess.model<-loess(Price~Year+Miles,data=FindCars)
> predicted.loess.model<-predict(loess.model,newdata=FindCars)
```

22. The following code can be used to obtain the R^2 value for the regression model, and plot the predicted values versus the actual values for the regression model. Use this code (and modify it as needed to create the plots below. Remove my sample and place a screen-shot of your plot in its place. (4 pts)

```
> #Getting R^2 for each model
> cor(predicted.regression.model,FindCars$Price)^2
> #Making the plot
> plot(predicted.regression.model,FindCars$Price,xlim=c(0,20000),ylim=c(0,20000),xlab="Predicted Values",ylab="Price",col="grey")
> abline(0,1)
> title("Regression Model \n R^2 = 54.7%")
```



23. Which model, regression or loess, is better at predicting used car prices? Discuss. (3 pts)

24. Recall, the formula for a residual. This is simple a measure of the inaccuracy of our predictions. (5 pts)

Residual Value = (Actual Value – Predicted Value)

a. What is the average absolute residual value for the regression model?
b. What is the average absolute residual value for the loess model?
c. Compare these two values. Explain how these values support your answer from the previous problem.

25. The following code can be used to run 10 iterations of cross-validation for our regression model. Run this code.

```
#Cross-Validation for Regression Model, using 10 iterations
output<-data.frame(R.Squared = rep(NA,10),Error = rep(NA,10))
for(i in 1:10){
  test<-sample(1:dim(FindCars)[1],0.7*dim(FindCars)[1])
  regression.model<-lm(Price~Year+Miles,data=FindCars[test,])
  predicted.regression.model<-predict(regression.model,newdata=FindCars[-test,])
  output[i,1] = cor(FindCars$Price[-test],predicted.regression.model)^2
  output[i,2] = mean(abs(FindCars$Price[-test] - predicted.regression.model))
}
```

Provide a screen-shot of the output data.frame before and after the loop. Use summary(output) to obtain basic summary statistics for the output dataframe. Is the R.Squared and Error values consistent across the 10 cross-validation samples? Discuss briefly. (4 pts)

26. The following code is used to run a 10 iterations of cross-validation for our loess model. Run this code.

Note: Precautions are needed for possible NA values returned by the predict() function for the loess model. Thus, the cor() and mean() functions need to be tweaked to handle possible NA values.

```
#Cross-Validation for Loess Model, using 10 iterations
output<-data.frame(R.Squared = rep(NA,10),Error = rep(NA,10))
for(i in 1:10){
  test<-sample(1:dim(FindCars)[1],0.7*dim(FindCars)[1])
  loess.model<-loess(Price~Year+Miles,data=FindCars[test,])
  predicted.loess.model<-predict(loess.model,newdata=FindCars[-test,])
  output[i,1] = cor(FindCars$Price[-test],predicted.loess.model,use="na.or.complete")^2
  output[i,2] = mean(abs(FindCars$Price[-test] - predicted.loess.model),na.rm=TRUE)
}
```

Provide a screen-shot of the output data.frame before and after the loop. Use summary(output) to obtain basic summary statistics for the output dataframe. Is the R.Squared and Error values consistent across the 10 cross-validation samples? Discuss briefly. (3 pts)

27. Which model, regression or loess, tends to perform better over the 10 iterations of cross-validation? Discuss. (3 pts)

Learning to use a new package…

The following code and plot are provided on page 8 of a contributed article in *The R Journal*.

Link: http://course1.winona.edu/cmalone/dsci210/exams/ggmap_article.pdf

```
houston <- get_map('houston", zoom = 14)
HoustonMap <- ggmap("houston",
  extent = "device", legend = "topleft")

HoustonMap +
  stat_density2d(
    aes(x = lon, y = lat, fill = ..level..,
      alpha = ..level..),
    size = 2, bins = 4, data = violent_crimes,
    geom = "polygon")
```

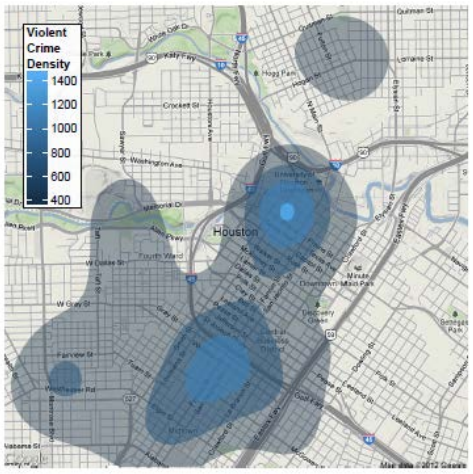


Figure 11: Filled contour plot of violent crimes

28. Create a similar plot for the bike rental intensities using the longitude and latitude measurements for assigned CitiBike dataset.

    Some hints:

    1) Instead of using my complete CitiBike dataset, I randomly selected 10000 rows and plotted only these rows.

```
#Subsetting the data a bit
dim(mydata)
samplerows<-sample(1:dim(mydata)[1],size=10000)
sampledata<-mydata[samplerows,]
```

    2) I found it difficult to get the New York map to center correctly, so instead I specified the longitude and latitude values for the lower-left and upper-right boundary box around the area to be plotted. This can be done using location =c( ) in the get_map() function.

```
#Mapping bike usage for March 2016 via Citi Bike
#newyork <- get_map(location="new york", maptype = "roadmap", zoom=13)
newyork <- get_map(location=c(left=-74.05,bottom=40.68,right=-73.95,top=40.8))
```

Delete my plot for March 2016 and replace with a screen-shot of your map. (10 points)