Class Assignment 1

Anurag Nagar

CS 6301

Getting and Analyzing Data

First of all, we will get some data. \ Use the code below to download the Car Worth dataset. Answer the remaining questions.

```
car_worth_train <- read.csv("http://www.utdallas.edu/~axn112530/R/datasets/CarWorth_Train.csv")

# Find out the dimensions of this dataset

# Find out the column names

# Use the View() function to look at the dataset


# Output column with the name "Model"


# Display the summary of the "Price"" column


# Create a histogram of the "Price" column
```

How to search for values within a list/data frame

R is famous for vectorized operations. You can apply a function to an entire column treated as a vector. Suppose we want to find out how many values in the cylinder column that are greater than 4.

```
sum(car_worth_train[,"Cylinder"] > 4)
```

```
## [1] 342
```

Note that we are filtering those rows that have value of "Cylinder" attribute greater than 4. The inner condition returns TRUE for values that satisfy this condition and FALSE otherwise. The *sum* function sums up the TRUE cases. Fill in the answers below:

```
# Find count of rows where the \textit{Doors} value is greater than 2

# Find count of rows where the \textit{Cruise} value is equal to 1

# Are there any rows where the \textit{Cruise} value is null. Hint: use is.na function
```

Working with a raw dataset

Let's look at another dataset - the Ames Housing Dataset. This is an example of a raw dataset that requires pre-processing. I have helped you by giving a code snippet that chooses those columns where the percent of NAs (null values) are greater than 50%.

```
ames_housing <- read.csv("http://www.utdallas.edu/~axn112530/R/datasets/AmesHousing.csv")

# Find percent of nulls in each column
for(i in 1:ncol(ames_housing)) {
  colName <- colnames(ames_housing[i])
  pctNull <- sum(is.na(ames_housing[,i]))/length(ames_housing[,i])
  if (pctNull > 0.50) {
    print(paste("Column ", colName, " has ", round(pctNull*100, 3), "% of nulls"))
  }
}
```

```
## [1] "Column  Alley  has  93.242 % of nulls"
## [1] "Column  Pool.QC  has  99.556 % of nulls"
## [1] "Column  Fence  has  80.478 % of nulls"
## [1] "Column  Misc.Feature  has  96.382 % of nulls"
```

You can drop a column from a data frame by looing at the example below:

```
# create a data frame
df <- data.frame(
  a = c(1, 2, 3),
  b = c(4, 5, 6),
  c= c(7, 8, 9)
)
# let's drop column c
df$c <- NULL
df
```

```
##   a b
## 1 1 4
## 2 2 5
## 3 3 6
```

Drop all columns where percent of null $> 50\%$

```
# Drop columns as specified above
```

Dropping rows that have null values

After dropping with a large fraction of null values, we need to examine rest of the data. If you find a row with NA values, you have several choices e.g. na.omit, na.exclude, etc. You can look up their details in help documentation.

```
# Get rid of rows with null values and call the clean dataset as ames_housing_clean
```

Plotting output vs features

Use the plot function to create a plot having SalePrice on Y-axis and Garage.Area on the X-axis of the **cleaned** data

```
# Create a plot as required above
```

Working with built-in R datasets

R comes with a nice set of built-in datasets. You can explore them by the following command

```
data()
```

Choose one of the datasets, pre-process it by getting rid of null values as explained previously.

Choose 5 numeric attributes (features) and create a histogram of their distribution.

Plot the output variable as Y-axis and the attributes chosen above as X-axis one at a time i.e. separate plots for each attribute.