Lab 1: Plotting in R; Regression [1]

First Name: _____ **Last Name**: _____ **NetID**: _____

In this lab, you will learn how to create simple plots in R. We will also review the basics of regression we learned in lectures using the `ToyotaCorolla` data set. Specifically, we will predict the price of a car model based on some of its features such as mileage, fuel type, weight, age and horse power, etc.

Lab 1 has two parts, and will expand this week and the week after. Lab 1 will be graded. Hand in your work directly to your TA in Lab on the week of 02/09/2015.

1 Plotting in R

Recall the non-linear function we saw on Friday's lecture:

$$y = 0.1x^3 - 2x^2 + 0.1x$$

How do we plot this function in R?
First create a vector `x` using the `seq` funciton and create `y` accordingly:

```
> seq(from = -10, to = 20, by = 0.01)
```

Use the `plot` function to plot `x-y`. Call `help(plot)` and `help(plot.default)` to see the parameter settings. Specifically, pay attention to what the `type, xlim, ylim, main, xlab, ylab` arguments mean.
For now we would like to create a line rather than scatter plot of `y` vs. `x`. We will give a title "`x-y`" to the plot, set the range of the `x`-label to be exactly $[-10, 20]$, and name the `x`-label "`x_axis`" and the `y`-label "`y_axis`". What should the above arguments be in this case?

Suppose you want to plot another function and put the two plots you created in a single page, use this command:

```
> par(mfrow = c(2, 1))
```

[1] The content of this lab is largely based on the course materials Prof. Dawn Woodard used in her Spring 2014 ORIE 4740 course at Cornell University.

2 Simple Regression

Download the `ToyotaCorolla.xls` data set from blackboard. This data set contains the information of more than 1400 trade-in vehicles. Our goal is to predict the price of a vehicle for resell in the market using the variables known like age and mileage of it.

The dataset will need to be in .csv (comma-separated values) text format before R can read it in. Open the Excel file, and navigate to the worksheet that has the data (the other worksheet has the metadata). Save this sheet as a .csv file by going to "File $->$ Save as" and change the file type to be .csv.

Read the .csv file into R using `read.table`. Remember you need to store the data read in a data frame object in R. Note you need to set the `sep` argument to be "," since this is a .csv file. What should the `header` argument be?

Check the number of rows of the dataset. There are 1443 rows in the data set in R, but only 1436 cars in the original Excel file! When Excel exported the data as a .csv file, it added empty rows at the end. Remove these rows by:

```
corollas <- corollas[1:1436,]
```

Check for missing values. How many missing values are there in the data set? You may need to recall what functions we used to check for missing values in Lab 0.

We will not include all variables as predictors in our regression model. For now, we're interested only in the following variables of interest: `Price`(the outcome), `Age_08_04, KM, Fuel_Type, HP, Met_Color, Doors, Quarterly_Tax` and `Weight`. Check the metadata file for what these variables are.
Choose the variables we are interested in to a new object:

```
> corollas2 <- corollas[, c("Price","Age_08_04", "KM", "Fuel_Type", "HP",
"Met_Color", "Doors", "Quarterly_Tax", "Weight") ]
```

Check whether the categorical predictors have been read in correctly as factor type:

```
> is.factor(corollas2[,"Fuel_Type"])
```

Or equivalently,

```
> is.factor(corollas2$Fuel_Type)
```

Make sure the `Fuel_Type` column takes the same 3 values indicated in the metadata.
Check whether the other categorical predictor `Met_Color` has been read in as factor type, and if not convert it to factor type.

Fit the linear regression model:

```
> corollasLM = lm( formula = Price ~ ., data = corollas2 )
> summary(corollasLM)
```

This syntax means that the `Price` variable is the outcome and all the other variables in `corollas2` should be used as predictors.
Look at the output of the `summary` function. For now you need only pay attention to the first column of the table. These are the ω coefficients in the model.
Note we have 9 predictors instead of 8. The `Fuel_Type` variable becomes two predictors: `Fuel_TypeDiesel` and `Fuel_TypePetrol`. Why?

Why don't we have another predictor `Fuel_TypeCNG`?

Are we done? Note we use a linear regression model. But does the linearity assumption hold? Create pairwise scatter plots of the continuous variables in the data set to see this:

```
> pairs(corollas2[, c("Price", "Age_08_04", "KM", "HP", "Doors",
"Quarterly_Tax", "Weight")])
```

Transform several variables so that the assumptions of the linear model are more reasonable. Which predictors should be transformed and how?

Recheck the pairwise scatter plots after transformation.

Rerun your regrssion model and report your new ω coefficients.

3 Take Home Questions

1. In our regression model, what are the units for the different components of ω?
2. How much would the price be affected by a unit change in the weight of a car? Is this rate (dollar/kg) affected by other factors such as color of the car, age of the car, etc.?

3. In our regression model, we are using multiple predictors like the number of doors, age and weight to predict the resale value of a Toyota Corolla. Can the model we are using capture the situation where a 4 door car that weighs more than 1200 kilograms sells at a lower price than a 2 door car that weighs more than 1200 kilograms? If yes, explain why. If no, what do we need to change in order to capture this situation?

4. How many parameters are there in a (standard) linear regression model with 4 continuous predictors? Include all unknown quantities that characterize the model and have to be estimated.

5. How many parameters would there be if we included all pairwise interaction terms, i.e terms of the form $\beta_{j,k}X_{i,j}X_{i,k}$. Terms of these form are called interaction terms. Do not count $\beta_{j,k}X_{i,j}X_{i,k}$ and $\beta_{k,j}X_{i,k}X_{i,j}$ as different terms as they are redundant.