A General Overview of Predictive Modeling with Linear Regression

Harisingh Khedar

December 27, 2016

Hello and Welcome! I'm Harisingh Khedar and I work as Systems Analyst at NTT Data. Are you a data enthusiast? Looking to transition to Data science? We are on the same road then! I started learning about data analysis since few months and what is better than document what one learns? I believe this document will help you in laying the basic foundation around modeling process and predictive analytics.

**What do we learn?**

The primary motive of this document is to let you introduce to life-cycle of statistical modeling. I have come to learn that building models is an iterative and often daunting process divided into various stages - Data collection, data transformation and exploration, model building and evaluation, model validation etc. As part of the exercise, we also learn about regression technique, predictive modeling using R. Without making it too verbose, I will try to explain various terms as we encounter. I use R for this exercise. I encourage you to take some online tutorials on R before starting.

**What is linear regression?**

Linear regression is good place to start if the objective is to predict numerical quantities based on one or more independent variables. The independent variable(s) could be either quantitative or categorical.

**What do we achieve with linear regression?**

Suppose that y[i] is the numeric quantity we want to predict, and x[i,] is a row of inputs that corresponds to output y[i]. Linear regression finds a fit function f(x) such that y[i] ~ f(x[i,]) = b[1] x[i,1] + ... b[n] x[i,n] We want numbers b[1],...,b[n] (called the coefficients or betas) such that f(x[i,]) is as near as possible to y[i] for all (x[i,],y[i]) pairs in our training data. R supplies a one-line command to find these coefficients: lm().

y[i] = b[1] x[i,1] + b[2] x[i,2] + ... b[n] x[i,n] + e[i]

In particular, this means that y is linear in the values of x: a change in the value of x[i,m] by one unit (while holding all the other x[i,k]s constant) will change the value of y[i] by the amount b[m] always, no matter what the starting value of x[i,m] was. This is easier to see in one dimension. If y = 3 + 2\*x, and if we increase x by 1, then y will always increase by 2, no matter what the starting value of x is. This wouldn't be true for, say, y = 3 + 2\*(x^2). The last term, e[i], represents what are called unsystematic errors, or noise. Unsystematic errors average to 0 (so they don't represent a net upward or net downward bias) and are uncorrelated with x[i,] and y[i].

**Dataset Information**

We work with Cars dataset for this exercise. For more information on this dataset, please check the reference manual for 'MASS' package at [<https://cran.r-project.org/>]. We will try to predict gas mileage based on given characteristics.

The typical set of steps involved in data modeling are as follows. We organize the rest of our discussion on linear regression in step-by-step fashion.

* Data collection
* Data exploration and preparation
* Build model
* Evaluate model
* Validate model
* Deploy model

## 1. Data Collection

*Load and attach the dataset to R workspace*

library("MASS")  
Cars93Data <- Cars93  
attach(Cars93Data)

R allows users to upload data from database, text files, web etc. using suitable commands. At any point, if you need detailed information on a given command, please use help(cmd) or ?cmd.

*Getting to know the dataset*

dim(Cars93Data)

# [1] 93 27

str(Cars93Data)

# 'data.frame': 93 obs. of 27 variables:  
# $ Manufacturer : Factor w/ 32 levels "Acura","Audi",..: 1 1 2 2 3 4 4 4 4 5 ...  
# $ Model : Factor w/ 93 levels "100","190E","240",..: 49 56 9 1 6 24 54 74 73 35 ...  
# $ Type : Factor w/ 6 levels "Compact","Large",..: 4 3 1 3 3 3 2 2 3 2 ...  
# $ Min.Price : num 12.9 29.2 25.9 30.8 23.7 14.2 19.9 22.6 26.3 33 ...  
# $ Price : num 15.9 33.9 29.1 37.7 30 15.7 20.8 23.7 26.3 34.7 ...  
# $ Max.Price : num 18.8 38.7 32.3 44.6 36.2 17.3 21.7 24.9 26.3 36.3 ...  
# $ MPG.city : int 25 18 20 19 22 22 19 16 19 16 ...  
# $ MPG.highway : int 31 25 26 26 30 31 28 25 27 25 ...  
# $ AirBags : Factor w/ 3 levels "Driver & Passenger",..: 3 1 2 1 2 2 2 2 2 2 ...  
# $ DriveTrain : Factor w/ 3 levels "4WD","Front",..: 2 2 2 2 3 2 2 3 2 2 ...  
# $ Cylinders : Factor w/ 6 levels "3","4","5","6",..: 2 4 4 4 2 2 4 4 4 5 ...  
# $ EngineSize : num 1.8 3.2 2.8 2.8 3.5 2.2 3.8 5.7 3.8 4.9 ...  
# $ Horsepower : int 140 200 172 172 208 110 170 180 170 200 ...  
# $ RPM : int 6300 5500 5500 5500 5700 5200 4800 4000 4800 4100 ...  
# $ Rev.per.mile : int 2890 2335 2280 2535 2545 2565 1570 1320 1690 1510 ...  
# $ Man.trans.avail : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 1 1 1 1 1 ...  
# $ Fuel.tank.capacity: num 13.2 18 16.9 21.1 21.1 16.4 18 23 18.8 18 ...  
# $ Passengers : int 5 5 5 6 4 6 6 6 5 6 ...  
# $ Length : int 177 195 180 193 186 189 200 216 198 206 ...  
# $ Wheelbase : int 102 115 102 106 109 105 111 116 108 114 ...  
# $ Width : int 68 71 67 70 69 69 74 78 73 73 ...  
# $ Turn.circle : int 37 38 37 37 39 41 42 45 41 43 ...  
# $ Rear.seat.room : num 26.5 30 28 31 27 28 30.5 30.5 26.5 35 ...  
# $ Luggage.room : int 11 15 14 17 13 16 17 21 14 18 ...  
# $ Weight : int 2705 3560 3375 3405 3640 2880 3470 4105 3495 3620 ...  
# $ Origin : Factor w/ 2 levels "USA","non-USA": 2 2 2 2 2 1 1 1 1 1 ...  
# $ Make : Factor w/ 93 levels "Acura Integra",..: 1 2 4 3 5 6 7 9 8 10 ...

head(Cars93Data)

# Manufacturer Model Type Min.Price Price Max.Price MPG.city  
# 1 Acura Integra Small 12.9 15.9 18.8 25  
# 2 Acura Legend Midsize 29.2 33.9 38.7 18  
# 3 Audi 90 Compact 25.9 29.1 32.3 20  
# 4 Audi 100 Midsize 30.8 37.7 44.6 19  
# 5 BMW 535i Midsize 23.7 30.0 36.2 22  
# 6 Buick Century Midsize 14.2 15.7 17.3 22  
# MPG.highway AirBags DriveTrain Cylinders EngineSize  
# 1 31 None Front 4 1.8  
# 2 25 Driver & Passenger Front 6 3.2  
# 3 26 Driver only Front 6 2.8  
# 4 26 Driver & Passenger Front 6 2.8  
# 5 30 Driver only Rear 4 3.5  
# 6 31 Driver only Front 4 2.2  
# Horsepower RPM Rev.per.mile Man.trans.avail Fuel.tank.capacity  
# 1 140 6300 2890 Yes 13.2  
# 2 200 5500 2335 Yes 18.0  
# 3 172 5500 2280 Yes 16.9  
# 4 172 5500 2535 Yes 21.1  
# 5 208 5700 2545 Yes 21.1  
# 6 110 5200 2565 No 16.4  
# Passengers Length Wheelbase Width Turn.circle Rear.seat.room  
# 1 5 177 102 68 37 26.5  
# 2 5 195 115 71 38 30.0  
# 3 5 180 102 67 37 28.0  
# 4 6 193 106 70 37 31.0  
# 5 4 186 109 69 39 27.0  
# 6 6 189 105 69 41 28.0  
# Luggage.room Weight Origin Make  
# 1 11 2705 non-USA Acura Integra  
# 2 15 3560 non-USA Acura Legend  
# 3 14 3375 non-USA Audi 90  
# 4 17 3405 non-USA Audi 100  
# 5 13 3640 non-USA BMW 535i  
# 6 16 2880 USA Buick Century

The above commands help us to know about dataset dimension, its structure, data etc. There are several other commands which help in getting to know more about the dataset.

### Final note on Data collection

Data collection is often tedious task as we need to find out and evaluate suitable data sources. Under real circumstances, you would have to engage with several stakeholders in order to understand and aggregate relevant data in one place.

## 2. Data Exploration and Preparation

### *Fetching the summary statistics*

summary(Cars93Data)

# Manufacturer Model Type Min.Price Price   
# Chevrolet: 8 100 : 1 Compact:16 Min. : 6.70 Min. : 7.40   
# Ford : 8 190E : 1 Large :11 1st Qu.:10.80 1st Qu.:12.20   
# Dodge : 6 240 : 1 Midsize:22 Median :14.70 Median :17.70   
# Mazda : 5 300E : 1 Small :21 Mean :17.13 Mean :19.51   
# Pontiac : 5 323 : 1 Sporty :14 3rd Qu.:20.30 3rd Qu.:23.30   
# Buick : 4 535i : 1 Van : 9 Max. :45.40 Max. :61.90   
# (Other) :57 (Other):87   
# Max.Price MPG.city MPG.highway AirBags   
# Min. : 7.9 Min. :15.00 Min. :20.00 Driver & Passenger:16   
# 1st Qu.:14.7 1st Qu.:18.00 1st Qu.:26.00 Driver only :43   
# Median :19.6 Median :21.00 Median :28.00 None :34   
# Mean :21.9 Mean :22.37 Mean :29.09   
# 3rd Qu.:25.3 3rd Qu.:25.00 3rd Qu.:31.00   
# Max. :80.0 Max. :46.00 Max. :50.00   
#   
# DriveTrain Cylinders EngineSize Horsepower RPM   
# 4WD :10 3 : 3 Min. :1.000 Min. : 55.0 Min. :3800   
# Front:67 4 :49 1st Qu.:1.800 1st Qu.:103.0 1st Qu.:4800   
# Rear :16 5 : 2 Median :2.400 Median :140.0 Median :5200   
# 6 :31 Mean :2.668 Mean :143.8 Mean :5281   
# 8 : 7 3rd Qu.:3.300 3rd Qu.:170.0 3rd Qu.:5750   
# rotary: 1 Max. :5.700 Max. :300.0 Max. :6500   
#   
# Rev.per.mile Man.trans.avail Fuel.tank.capacity Passengers   
# Min. :1320 No :32 Min. : 9.20 Min. :2.000   
# 1st Qu.:1985 Yes:61 1st Qu.:14.50 1st Qu.:4.000   
# Median :2340 Median :16.40 Median :5.000   
# Mean :2332 Mean :16.66 Mean :5.086   
# 3rd Qu.:2565 3rd Qu.:18.80 3rd Qu.:6.000   
# Max. :3755 Max. :27.00 Max. :8.000   
#   
# Length Wheelbase Width Turn.circle   
# Min. :141.0 Min. : 90.0 Min. :60.00 Min. :32.00   
# 1st Qu.:174.0 1st Qu.: 98.0 1st Qu.:67.00 1st Qu.:37.00   
# Median :183.0 Median :103.0 Median :69.00 Median :39.00   
# Mean :183.2 Mean :103.9 Mean :69.38 Mean :38.96   
# 3rd Qu.:192.0 3rd Qu.:110.0 3rd Qu.:72.00 3rd Qu.:41.00   
# Max. :219.0 Max. :119.0 Max. :78.00 Max. :45.00   
#   
# Rear.seat.room Luggage.room Weight Origin   
# Min. :19.00 Min. : 6.00 Min. :1695 USA :48   
# 1st Qu.:26.00 1st Qu.:12.00 1st Qu.:2620 non-USA:45   
# Median :27.50 Median :14.00 Median :3040   
# Mean :27.83 Mean :13.89 Mean :3073   
# 3rd Qu.:30.00 3rd Qu.:15.00 3rd Qu.:3525   
# Max. :36.00 Max. :22.00 Max. :4105   
# NA's :2 NA's :11   
# Make   
# Acura Integra: 1   
# Acura Legend : 1   
# Audi 100 : 1   
# Audi 90 : 1   
# BMW 535i : 1   
# Buick Century: 1   
# (Other) :87

The summary command is very useful way to learn about data in hand. It provides numerical summaries (Range, Mean/Median, Quantiles etc.) for quantitative variables and frequency distribution for categorical variables. In addition, we also get to know about missing data - variables Rear.seat.room & Luggage.room have missing data. There are no unexpected data values (Ex. negative car price) and data looks reasonably divided for both numerical and categorical variables. We will learn more about data distribution during visualization.

*Why one should care about missing values?*

One should investigate the reasons behind missing data. There are two corrective measures to deal with missing data - Perhaps ignore observations with missing data if they make very small proportion of overall data or treat them according to your investigation. In our case, we will ignore these observations for demonstration purpose but still missing data for Luggage.room makes approximately 12% of dataset which is huge if our dataset were to be very large.

**Removed missing values!**

Cars93Data.v1 <- na.omit(Cars93Data)  
dim(Cars93Data.v1)

# [1] 82 27

*Split numeric and categorical variables*

Variables <- setdiff(colnames(Cars93Data.v1), c("MPG.highway", "Model", "AirBags",   
 "DriveTrain", "Man.trans.avail", "Origin", "Make", "Luggage.room", "Rear.seat.room",   
 "Turn.circle", "Manufacturer"))  
NumVars <- Variables[sapply(Cars93Data.v1[, Variables], class) %in% c("numeric",   
 "integer")]  
CatVars <- Variables[sapply(Cars93Data.v1[, Variables], class) %in% c("factor")]  
rownames(Cars93Data.v1) <- Cars93Data.v1$Make #Reset the row names.

**NOTE** - We removed variables which doesn't seem to have any direct relation with car mileage - Number of air bags, area of origin, luggage room etc.

*Correlation matrix for numeric variables*

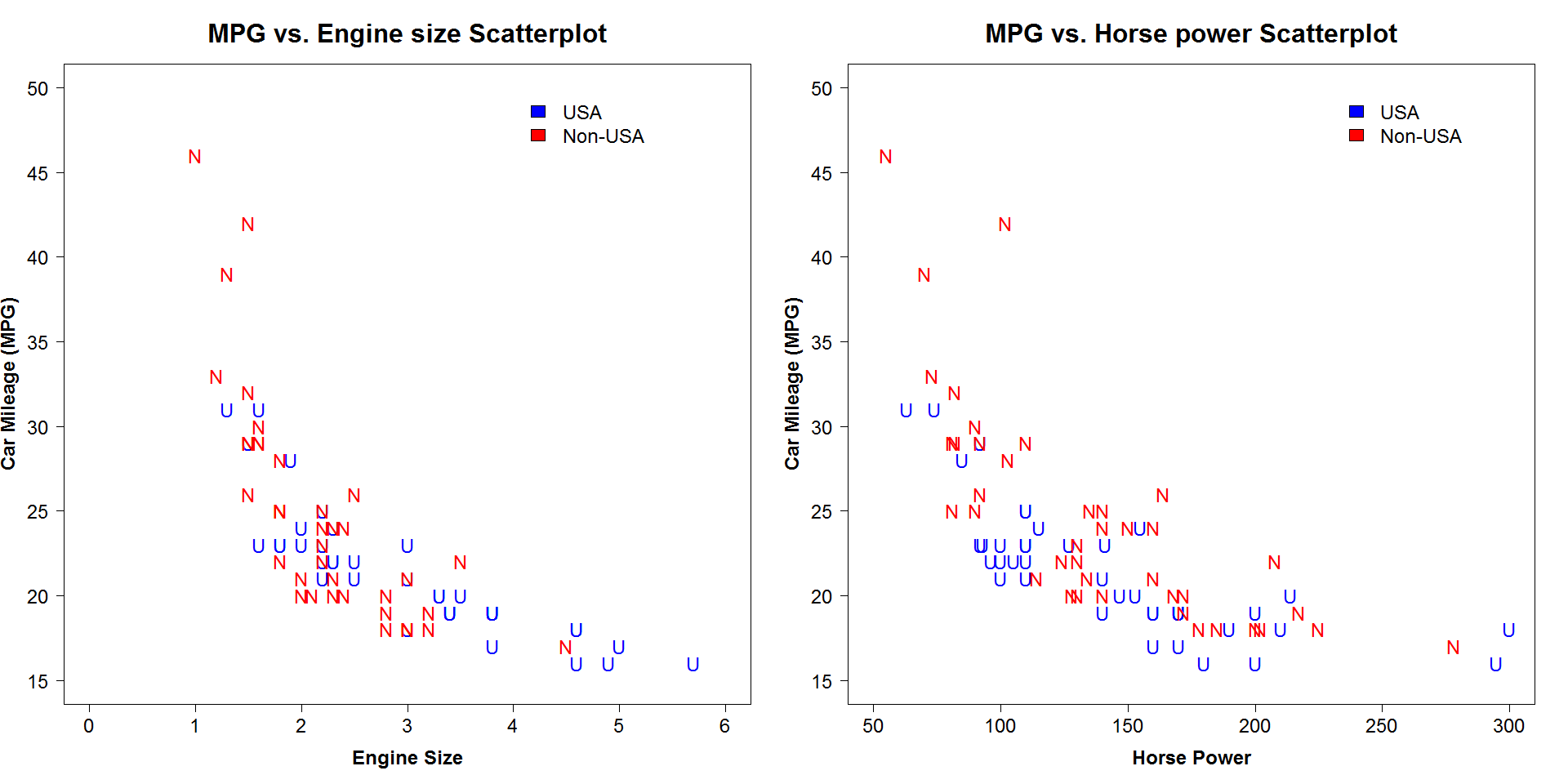
round(cor(Cars93Data.v1[, NumVars]), 3)

# Min.Price Price Max.Price MPG.city EngineSize  
# Min.Price 1.000 0.971 0.910 -0.663 0.705  
# Price 0.971 1.000 0.983 -0.622 0.642  
# Max.Price 0.910 0.983 1.000 -0.565 0.567  
# MPG.city -0.663 -0.622 -0.565 1.000 -0.747  
# EngineSize 0.705 0.642 0.567 -0.747 1.000  
# Horsepower 0.795 0.787 0.750 -0.704 0.772  
# RPM -0.092 -0.035 0.011 0.327 -0.524  
# Rev.per.mile -0.487 -0.437 -0.381 0.712 -0.824  
# Fuel.tank.capacity 0.736 0.707 0.656 -0.797 0.821  
# Passengers 0.327 0.274 0.222 -0.463 0.526  
# Length 0.611 0.552 0.485 -0.725 0.852  
# Wheelbase 0.676 0.628 0.566 -0.671 0.859  
# Width 0.544 0.495 0.438 -0.694 0.890  
# Weight 0.777 0.737 0.677 -0.836 0.917  
# Horsepower RPM Rev.per.mile Fuel.tank.capacity  
# Min.Price 0.795 -0.092 -0.487 0.736  
# Price 0.787 -0.035 -0.437 0.707  
# Max.Price 0.750 0.011 -0.381 0.656  
# MPG.city -0.704 0.327 0.712 -0.797  
# EngineSize 0.772 -0.524 -0.824 0.821  
# Horsepower 1.000 0.024 -0.592 0.796  
# RPM 0.024 1.000 0.507 -0.244  
# Rev.per.mile -0.592 0.507 1.000 -0.631  
# Fuel.tank.capacity 0.796 -0.244 -0.631 1.000  
# Passengers 0.241 -0.378 -0.482 0.451  
# Length 0.643 -0.440 -0.747 0.794  
# Wheelbase 0.663 -0.405 -0.731 0.793  
# Width 0.689 -0.489 -0.789 0.772  
# Weight 0.860 -0.348 -0.784 0.900  
# Passengers Length Wheelbase Width Weight  
# Min.Price 0.327 0.611 0.676 0.544 0.777  
# Price 0.274 0.552 0.628 0.495 0.737  
# Max.Price 0.222 0.485 0.566 0.438 0.677  
# MPG.city -0.463 -0.725 -0.671 -0.694 -0.836  
# EngineSize 0.526 0.852 0.859 0.890 0.917  
# Horsepower 0.241 0.643 0.663 0.689 0.860  
# RPM -0.378 -0.440 -0.405 -0.489 -0.348  
# Rev.per.mile -0.482 -0.747 -0.731 -0.789 -0.784  
# Fuel.tank.capacity 0.451 0.794 0.793 0.772 0.900  
# Passengers 1.000 0.646 0.649 0.516 0.510  
# Length 0.646 1.000 0.911 0.885 0.880  
# Wheelbase 0.649 0.911 1.000 0.850 0.879  
# Width 0.516 0.885 0.850 1.000 0.869  
# Weight 0.510 0.880 0.879 0.869 1.000

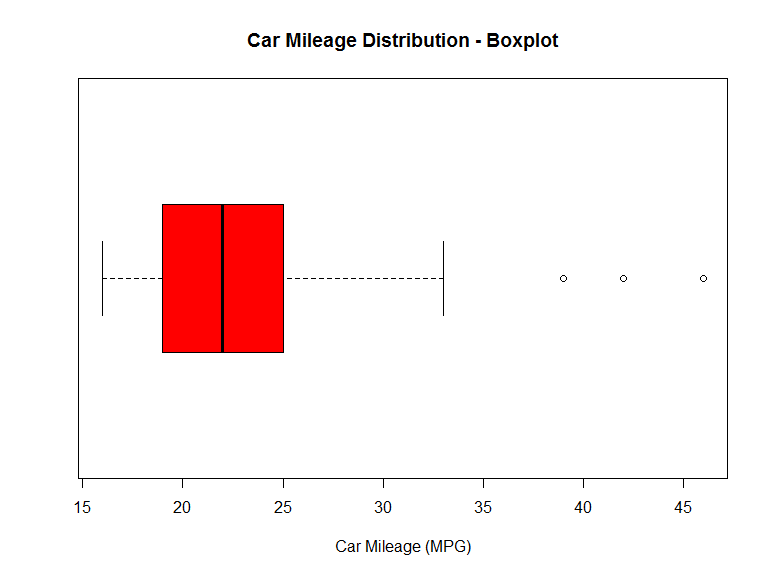
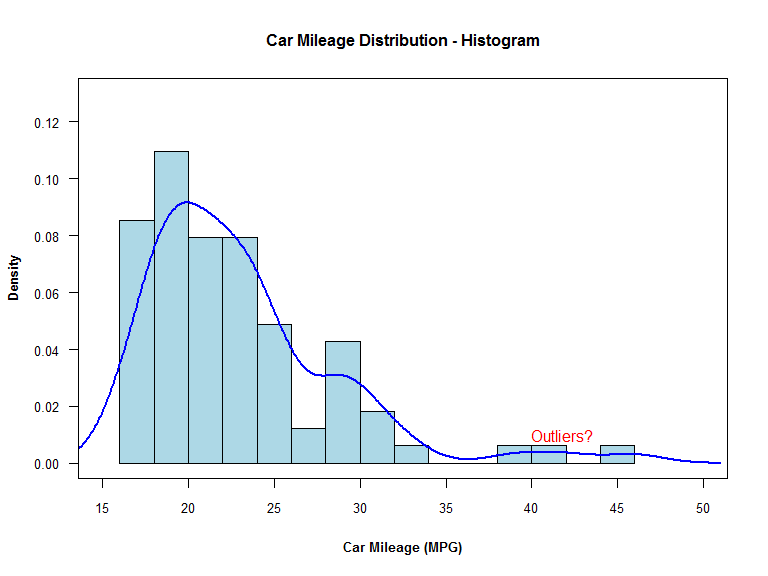
### *Data Visualization*

Using data visualization, we can learn more about data - its shape, distribution, outliers, range, relationships etc. It is another way to learn about data and its shortcomings.

The below scatter plot tell us same story as indicated by correlation matrix - the car mileage is linearly related to engine size, horse power etc. But they provide us more elaborated story.



A typical car has its mileage between 15 - 30 MPG. The below graphs indicate **extreme values, possible outliers?**.

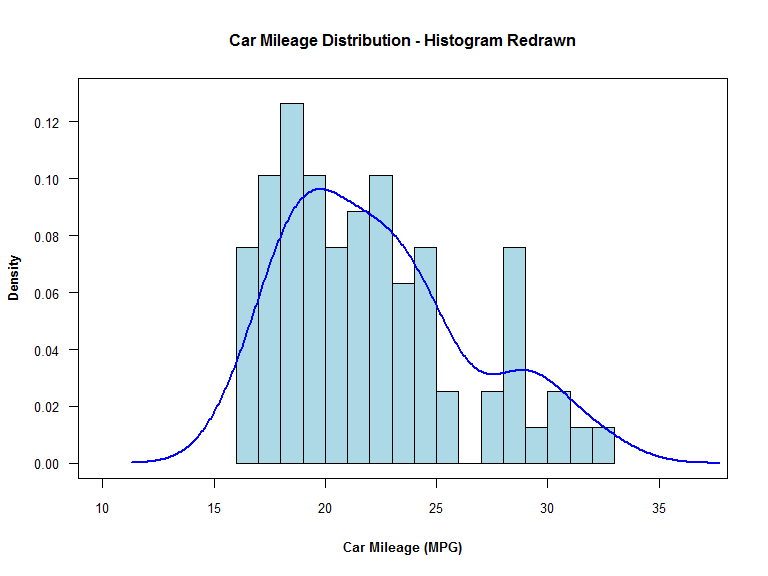


At this point, we just ignore the extreme outliers and redraw the histogram. **Much better!** We could apply transformations to approximate it more to normal curve.

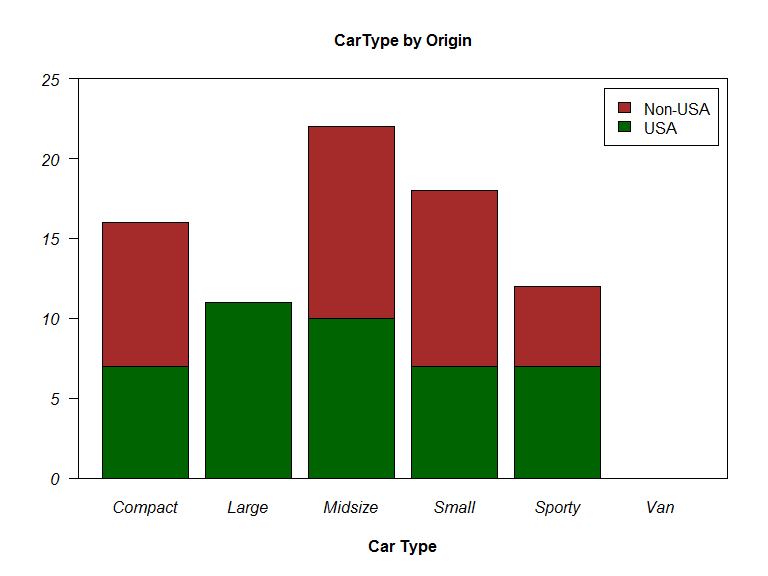
Cars93Data.v2 <- Cars93Data.v1[MPG.city < 35, ]  
dim(Cars93Data.v2)

# [1] 79 27

attach(Cars93Data.v2)



The bar chart reveals more information about data in hand - there is no observation for car type "Van". I missed noticing from summary statistics but this chart helps to spot such anomalies. We don't have any large cars from outside USA. The small and medium size cars are best sold.

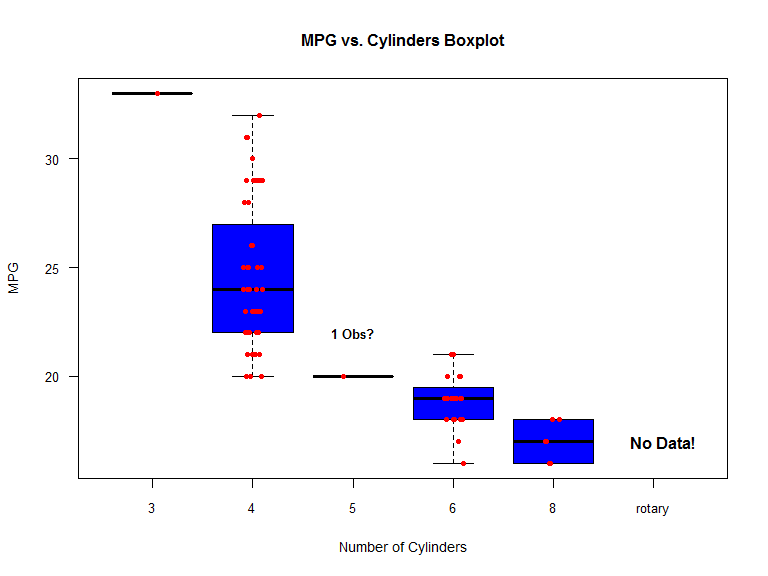


*Reorganize levels for* ***type*** *variable*

Cars93Data.v2$Type1 <- factor(Cars93Data.v2$Type)  
levels(Cars93Data.v2$Type1)

# [1] "Compact" "Large" "Midsize" "Small" "Sporty"

The below graph indicates that cars with fewer cylinders seem to provide higher gas mileage. Interestingly, the data is not equally divided across categories.



*Reorganize levels for* ***Cylinders*** *variable*

Cars93Data.v2$Cylinders <- factor(Cars93Data.v2$Cylinders)  
summary(Cars93Data.v2$Cylinders)

# 3 4 5 6 8   
# 1 47 1 24 6

### Final note on Data Exploration and Preparation

Data exploration & preparation is often very tedious and time-taking task. It helps to prepare the data for more sophisticated analysis in later steps.

## 3. Build and Evaluate Model

After initial analysis of data, we are now ready to build a multiple linear regression model. Before we go any further, lets split the data into training set and test set. The training dataset is used to build the actual model, whereas the test dataset will be used to evaluate the model.

Cars93Data.v2$Sampling <- runif(dim(Cars93Data.v2)[1])  
TrainSet <- subset(Cars93Data.v2, Cars93Data.v2$Sampling > 0.1)  
TestSet <- subset(Cars93Data.v2, Cars93Data.v2$Sampling < 0.1)

### Build the Model

We have decided to work with subset of variables, based on our analysis from the exploration phase.

Predictors <- paste(c("EngineSize", "Horsepower", "Weight", "Origin", "Cylinders",   
 "Fuel.tank.capacity"), collapse = "+")  
Formula <- paste("MPG.city", Predictors, sep = "~")  
Formula

# [1] "MPG.city~EngineSize+Horsepower+Weight+Origin+Cylinders+Fuel.tank.capacity"

Model <- lm(Formula, data = TrainSet)  
summary(Model)

#   
# Call:  
# lm(formula = Formula, data = TrainSet)  
#   
# Residuals:  
# Min 1Q Median 3Q Max   
# -3.9472 -0.9820 0.1045 1.0878 4.7756   
#   
# Coefficients:  
# Estimate Std. Error t value Pr(>|t|)   
# (Intercept) 43.3905247 2.9055518 14.934 <2e-16 \*\*\*  
# EngineSize -0.0372003 0.8193240 -0.045 0.9639   
# Horsepower -0.0003816 0.0099024 -0.039 0.9694   
# Weight -0.0039702 0.0016147 -2.459 0.0169 \*   
# Originnon-USA 0.4004637 0.5945816 0.674 0.5032   
# Cylinders4 -4.0187443 2.1792342 -1.844 0.0702 .   
# Cylinders5 -5.3011756 3.1105624 -1.704 0.0936 .   
# Cylinders6 -5.5512410 2.4671908 -2.250 0.0282 \*   
# Cylinders8 -4.8511770 2.9430051 -1.648 0.1046   
# Fuel.tank.capacity -0.2825404 0.2038712 -1.386 0.1710   
# ---  
# Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
#   
# Residual standard error: 1.967 on 59 degrees of freedom  
# Multiple R-squared: 0.7913, Adjusted R-squared: 0.7595   
# F-statistic: 24.86 on 9 and 59 DF, p-value: < 2.2e-16

The summary command gives us concise report on given model. At this point, we will not go in detail on individual terms but (R2, Adj R2) indicate good amount of variation explained by our model (~80%). Lets see if we can further improve this model using **backward selection approach**.

**Applying backward selection :** The backward selection method removes non-significant terms based on AIC score. Though the selected model doesn't improve existing model significant enough, it is good practice to apply one of the model selection techniques.

step(Model, direction = "backward")

# Start: AIC=102.55  
# MPG.city ~ EngineSize + Horsepower + Weight + Origin + Cylinders +   
# Fuel.tank.capacity  
#   
# Df Sum of Sq RSS AIC  
# - Horsepower 1 0.0057 228.25 100.55  
# - EngineSize 1 0.0080 228.25 100.55  
# - Origin 1 1.7549 230.00 101.07  
# - Cylinders 4 27.3737 255.62 102.36  
# <none> 228.24 102.55  
# - Fuel.tank.capacity 1 7.4301 235.67 102.76  
# - Weight 1 23.3871 251.63 107.28  
#   
# Step: AIC=100.55  
# MPG.city ~ EngineSize + Weight + Origin + Cylinders + Fuel.tank.capacity  
#   
# Df Sum of Sq RSS AIC  
# - EngineSize 1 0.0055 228.25 98.549  
# - Origin 1 1.8146 230.06 99.093  
# - Cylinders 4 27.8551 256.11 100.492  
# <none> 228.25 100.547  
# - Fuel.tank.capacity 1 7.5095 235.76 100.781  
# - Weight 1 30.0856 258.34 107.091  
#   
# Step: AIC=98.55  
# MPG.city ~ Weight + Origin + Cylinders + Fuel.tank.capacity  
#   
# Df Sum of Sq RSS AIC  
# - Origin 1 2.340 230.59 97.252  
# - Cylinders 4 27.850 256.11 98.492  
# <none> 228.25 98.549  
# - Fuel.tank.capacity 1 8.026 236.28 98.933  
# - Weight 1 39.202 267.46 107.485  
#   
# Step: AIC=97.25  
# MPG.city ~ Weight + Cylinders + Fuel.tank.capacity  
#   
# Df Sum of Sq RSS AIC  
# - Fuel.tank.capacity 1 6.173 236.77 97.075  
# <none> 230.59 97.252  
# - Cylinders 4 29.141 259.74 97.464  
# - Weight 1 47.695 278.29 108.224  
#   
# Step: AIC=97.08  
# MPG.city ~ Weight + Cylinders  
#   
# Df Sum of Sq RSS AIC  
# <none> 236.77 97.075  
# - Cylinders 4 33.544 270.31 98.217  
# - Weight 1 154.532 391.30 129.740

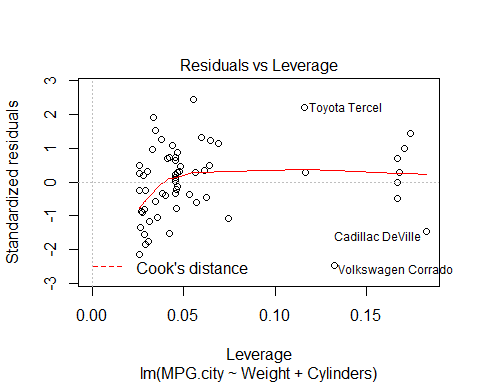
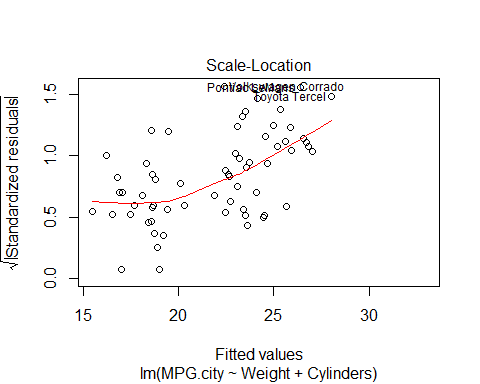
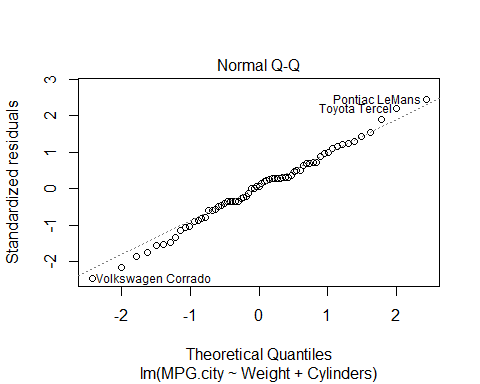
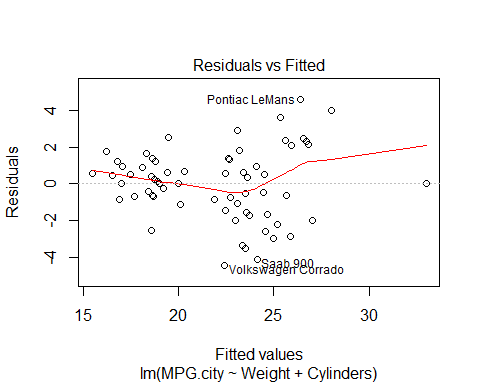
#   
# Call:  
# lm(formula = MPG.city ~ Weight + Cylinders, data = TrainSet)  
#   
# Coefficients:  
# (Intercept) Weight Cylinders4 Cylinders5 Cylinders6   
# 44.012917 -0.005385 -4.946270 -6.537653 -6.444306   
# Cylinders8   
# -5.947460

**The summary of selected model..**

Model <- lm(formula = MPG.city ~ Weight + Cylinders, data = TrainSet)  
summary(Model)

#   
# Call:  
# lm(formula = MPG.city ~ Weight + Cylinders, data = TrainSet)  
#   
# Residuals:  
# Min 1Q Median 3Q Max   
# -4.4359 -1.0723 0.1183 1.2064 4.5888   
#   
# Coefficients:  
# Estimate Std. Error t value Pr(>|t|)   
# (Intercept) 44.0129167 2.5899543 16.994 < 2e-16 \*\*\*  
# Weight -0.0053853 0.0008398 -6.412 2.09e-08 \*\*\*  
# Cylinders4 -4.9462703 2.0506758 -2.412 0.0188 \*   
# Cylinders5 -6.5376528 2.9209747 -2.238 0.0288 \*   
# Cylinders6 -6.4443057 2.3242689 -2.773 0.0073 \*\*   
# Cylinders8 -5.9474599 2.6157637 -2.274 0.0264 \*   
# ---  
# Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
#   
# Residual standard error: 1.939 on 63 degrees of freedom  
# Multiple R-squared: 0.7835, Adjusted R-squared: 0.7663   
# F-statistic: 45.61 on 5 and 63 DF, p-value: < 2.2e-16

### Diagnostic plots



**Residual vs. Fitted Plot:** Ideally, we would expect the points to be equally spreaded around residual mean = 0. The plot indicates some sort of pattern in our data as there are many under fitted values in middle. We should go back and further investigate the linearity among variables.

**Q-Q Plot:** The data should approximately lie along the normal reference line. In our case, there is slight deviation from the line, indicating some sort of pattern in the data. Also, the graph shows outliers with labels (row names).

#### Residual normality test

# Shapiro-Wilk Normality Test  
shapiro.test(Model$residuals)

#   
# Shapiro-Wilk normality test  
#   
# data: Model$residuals  
# W = 0.99159, p-value = 0.9273

### Predictions

Now, we go one step further and run the model to make predictions. The predictions on test data can be used to evaluate model fitting.

TestSet$MPG.PRED <- predict(Model, newdata = TestSet)  
TestSet[, c("MPG.city", "MPG.PRED")]

# MPG.city MPG.PRED  
# Eagle Summit 29 26.70741  
# Ford Festiva 31 29.13079  
# Ford Escort 23 25.44186  
# Ford Taurus 21 19.66252  
# Geo Storm 30 25.73806  
# Hyundai Excel 29 26.43814  
# Hyundai Scoupe 26 26.76126  
# Plymouth Laser 23 24.84948  
# Pontiac Bonneville 19 18.74702  
# Volkswagen Passat 21 22.99156

TrainSet$MPG.PRED <- predict(Model, newdata = TrainSet)  
TrainSet[1:5, c("MPG.city", "MPG.PRED")]

# MPG.city MPG.PRED  
# Acura Integra 25 24.49944  
# Acura Legend 18 18.39698  
# Audi 90 20 19.39326  
# Audi 100 19 19.23170  
# BMW 535i 22 19.46419

### Final note on building model and evaluation

As I mentioned earlier, building model is an iterative process in which we learn at every step. At times, we would have to go back and gather more appropriate data, choose another model etc. The diagnostic plots are very handy in assessing the overall model and decide if we can improve the same using another technique.

We have covered till step 4 at this point. The final two stages (Model validation and deployment) would make more sense in real scenario. The model validation is primarily important to evaluate overall soundness of the developed model before it is deployed into production. We will skip these two steps but I hope you get the idea.

## Final note on the exercise

The main focus of this exercise was to familiarize one with typical set of steps taken in process of building model. As I mentioned earlier, the whole process is iterative in itself and involves various stack holders such as project sponsor, client, data engineers etc. We discussed linear regression for this exercise and made predictions too.

In next paper, we repeat the same process but with another technique - **logistic regression**. I would love to have feedback/suggestions, if any. Thank you for taking time to read this introductory paper.