Linear Regression (LR): The most talked and used algorithm by anyone who starts with data science. Machine Learning basics start with LR. I'll be discussing LR in depth with various techniques.

1st Code go through the analytical process; understanding the Programming Language; pros and cons in using the language.

2nd Understanding each process and adding statistical value to the process.

3rd Adding features (Feature Engineering); running multiple iterations; choosing best fit model.

The analysis is a combinations of results in python, R and STATA, hence the doubt or dilemma in choosing which platform to use will be resolved.

The data used to do the analysis is AUTO data set of USA year 1978, has 74 data points. The data is about the sales of different cars in USA, which is either being made in USA (Domestic) or made elsewhere (Foreign). The data tells about the name of the car (Make and Model) , the price of car (price), miles per gallon (Mileage (mpg)), Repair Record 1978 (rep78), no of headroom (Headroom (in.)), Trunk space (cu. ft.) (Trunk) , weight of the car (Weight (lbs.)), length of car(Length (in.)) , turn for the car(Turn Circle (ft.)), displacement(Displacement (cu. in.)) , gear ratio and foreign (Car type) where 0 means manufactured in domestic i.e. in USA else 1 means is imported from other countries.

Sample of the Auto data set:

make	price	mpg	rep78	headroom	trunk	weight	length	turn	displacement	gear_ratio	foreign
AMC Concord	4099	22	3	2.5	11	2930	186	40	121	3.579999924	0
AMC Pacer	4749	17	3	3	11	3350	173	40	258	2.529999971	0
AMC Spirit	3799	22		3	12	2640	168	35	121	3.079999924	0
Buick Century	4816	20	3	4.5	16	3250	196	40	196	2.930000067	0
Buick Electra	7827	15	4	4	20	4080	222	43	350	2.410000086	0
Buick LeSabre	5788	18	3	4	21	3670	218	43	231	2.730000019	0

Knowing the data is THE most important part for data analysis. Let's start the analysis journey but what is our goal? Linear Regression deals with the response variable i.e. Y or dependent variable with respect to independent variable X; so goal would be to predict the values of MPG (miles per gallon) (this variable is the dependent variable Y)cross-ponding to other independent variable(X) like weight, length, headroom, price , etc.

Let's being our coding and Analysing:

Codes are marked in 

R Codes:

Import Basics libraries, if not present then use

install.packages('Library name') function to download it from the R server.

```
library(foreign)
library(ggplot2)
library(Hmisc)
```

Import the data:

1. auto <- read.dta("http://www.stata-press.com/data/r9/auto.dta")
2. auto <- read.csv("YOUR_LOCATION/auto.csv")
3. auto <- read.csv(file.choose())
4. auto <- read.table("YOUR_LOCATION/auto.csv",sep = ',',header=T)

There are many ways to load the data, few I have stated above

With library (ISLR) Auto data set is their but is somewhat different from I am using.

Understanding the data and Viewing it:

attach(auto) ## use the data set headers (variables) freely else we need to use auto$variablename

View(auto) ## Gives table like view for the data set.

nrow(auto) ## number of rows

dim(auto) ## dimension for the data frame.

str(auto) ## structure of the data frame like variable type **char/int/num/** and best thing in R is that it identifies the **variable** as factor for qualitative/categorical variable.

```
> nrow(auto)
[1] 74
> dim(auto)
[1] 74 12
> View(auto)
> str(auto)
'data.frame':   74 obs. of  12 variables:
 $ make        : chr  "AMC Concord" "AMC Pacer" "AMC Spirit" "Buick Century" ...
 $ price       : int  4099 4749 3799 4816 7827 5788 4453 5189 10372 4082 ...
 $ mpg         : int  22 17 22 20 15 18 26 20 16 19 ...
 $ rep78       : int  3 3 NA 3 4 3 NA 3 3 3 ...
 $ headroom    : num  2.5 3 3 4.5 4 4 3 2 3.5 3.5 ...
 $ trunk       : int  11 11 12 16 20 21 10 16 17 13 ...
 $ weight      : int  2930 3350 2640 3250 4080 3670 2230 3280 3880 3400 ...
 $ length      : int  186 173 168 196 222 218 170 200 207 200 ...
 $ turn        : int  40 40 35 40 43 43 34 42 43 42 ...
 $ displacement: int  121 258 121 196 350 231 304 196 231 231 ...
 $ gear_ratio  : num  3.58 2.53 3.08 2.93 2.41 ...
 $ foreign     : Factor w/ 2 levels "Domestic","Foreign": 1 1 1 1 1 1 1 1 1 1 ...
 - attr(*, "datalabel")= chr "1978 Automobile Data"
 - attr(*, "time.stamp")= chr "13 Apr 2005 17:45"
 - attr(*, "formats")= chr  "%-18s" "%8.0gc" "%8.0g" "%8.0g" ...
 - attr(*, "types")= int  18 252 252 252 254 252 252 252 252 252 ...
 - attr(*, "val.labels")= chr  "" "" "" "" ...
 - attr(*, "var.labels")= chr  "Make and Model" "Price" "Mileage (mpg)" "Repair Record 1978" ...
 - attr(*, "expansion.fields")=List of 2
  ..$ : chr  "_dta" "note0" "1"
  ..$ : chr  "_dta" "note1" "from Consumer Reports with permission"
 - attr(*, "version")= int 8
 - attr(*, "label.table")=List of 1
  ..$ origin: Named int  0 1
  .. ..- attr(*, "names")= chr  "Domestic" "Foreign"
> str(foreign)
 Factor w/ 2 levels "Domestic","Foreign": 1 1 1 1 1 1 1 1 1 1 ...
> str(auto$foreign)
```

summary(auto) ##this is where u get idea how the is spread across the variable.

```
> summary(auto)
     make               price           mpg             rep78          headroom        trunk           weight     
 Length:74          Min.   : 3291   Min.   :12.00   Min.   :1.000   Min.   :1.500   Min.   : 5.00   Min.   :1760  
 Class :character   1st Qu.: 4220   1st Qu.:18.00   1st Qu.:3.000   1st Qu.:2.500   1st Qu.:10.25   1st Qu.:2250  
 Mode  :character   Median : 5006   Median :20.00   Median :3.000   Median :3.000   Median :14.00   Median :3190  
                    Mean   : 6165   Mean   :21.30   Mean   :3.406   Mean   :2.993   Mean   :13.76   Mean   :3019  
                    3rd Qu.: 6332   3rd Qu.:24.75   3rd Qu.:4.000   3rd Qu.:3.500   3rd Qu.:16.75   3rd Qu.:3600  
                    Max.   :15906   Max.   :41.00   Max.   :5.000   Max.   :5.000   Max.   :23.00   Max.   :4840  
                                                    NA's   :5                                                     
     length           turn          displacement     gear_ratio       foreign   
 Min.   :142.0   Min.   :31.00   Min.   : 79.0   Min.   :2.190   Domestic:52  
 1st Qu.:170.0   1st Qu.:36.00   1st Qu.:119.0   1st Qu.:2.730   Foreign :22  
 Median :192.5   Median :40.00   Median :196.0   Median :2.955                
 Mean   :187.9   Mean   :39.65   Mean   :197.3   Mean   :3.015                
 3rd Qu.:203.8   3rd Qu.:43.00   3rd Qu.:245.2   3rd Qu.:3.353                
 Max.   :233.0   Max.   :51.00   Max.   :425.0   Max.   :3.890                
```

The variable which are continuous have 6 point summary (Min, 1st Quartile, Median, Mean, 3rd Quartile, Max) but many says 5 point summary and for categorical or qualitative variable we have frequency summary like as here for foreign.

It identifies the data points which are blank or are NULLs like we have NA's :5 for rep78 variable.

describe(auto) ## Publishes more detailed summary.

```
displacement
       n  missing distinct     Info     Mean      Gmd      .05      .10      .25      .50      .75      .90      .95
      74        0       31    0.993    197.3    103.4    87.95    97.00   119.00   196.00   245.25   340.40   350.00

lowest :  79  85  86  89  90, highest: 304 318 350 400 425
--------------------------------------------------------------------------------------------------------------------------
gear_ratio
       n  missing distinct     Info     Mean      Gmd      .05      .10      .25      .50      .75      .90      .95
      74        0       36    0.996    3.015   0.5226    2.365    2.442    2.730    2.955    3.352    3.714    3.780

lowest : 2.19 2.24 2.26 2.28 2.41, highest: 3.73 3.74 3.78 3.81 3.89
--------------------------------------------------------------------------------------------------------------------------
foreign
       n  missing distinct
      74        0        2

Value      Domestic  Foreign
Frequency        52       22
Proportion    0.703    0.297
```

auto[is.na(rep78),] ## Viewing the null data points.

```
> auto[is.na(rep78), ] #which rows have a missing value
           make price mpg rep78 headroom trunk weight length turn displacement gear_ratio  foreign
3     AMC Spirit  3799  22    NA      3.0    12   2640    168   35          121       3.08 Domestic
7     Buick Opel  4453  26    NA      3.0    10   2230    170   34          304       2.87 Domestic
45 Plym. Sapporo  6486  26    NA      1.5     8   2520    182   38          119       3.54 Domestic
51 Pont. Phoenix  4424  19    NA      3.5    13   3420    203   43          231       3.08 Domestic
64   Peugeot 604 12990  14    NA      3.5    14   3420    192   38          163       3.58  Foreign
```

We will not do any impute

with(auto, summary(price[mpg >= mean(mpg)])) ## When you want to view conditional summary.

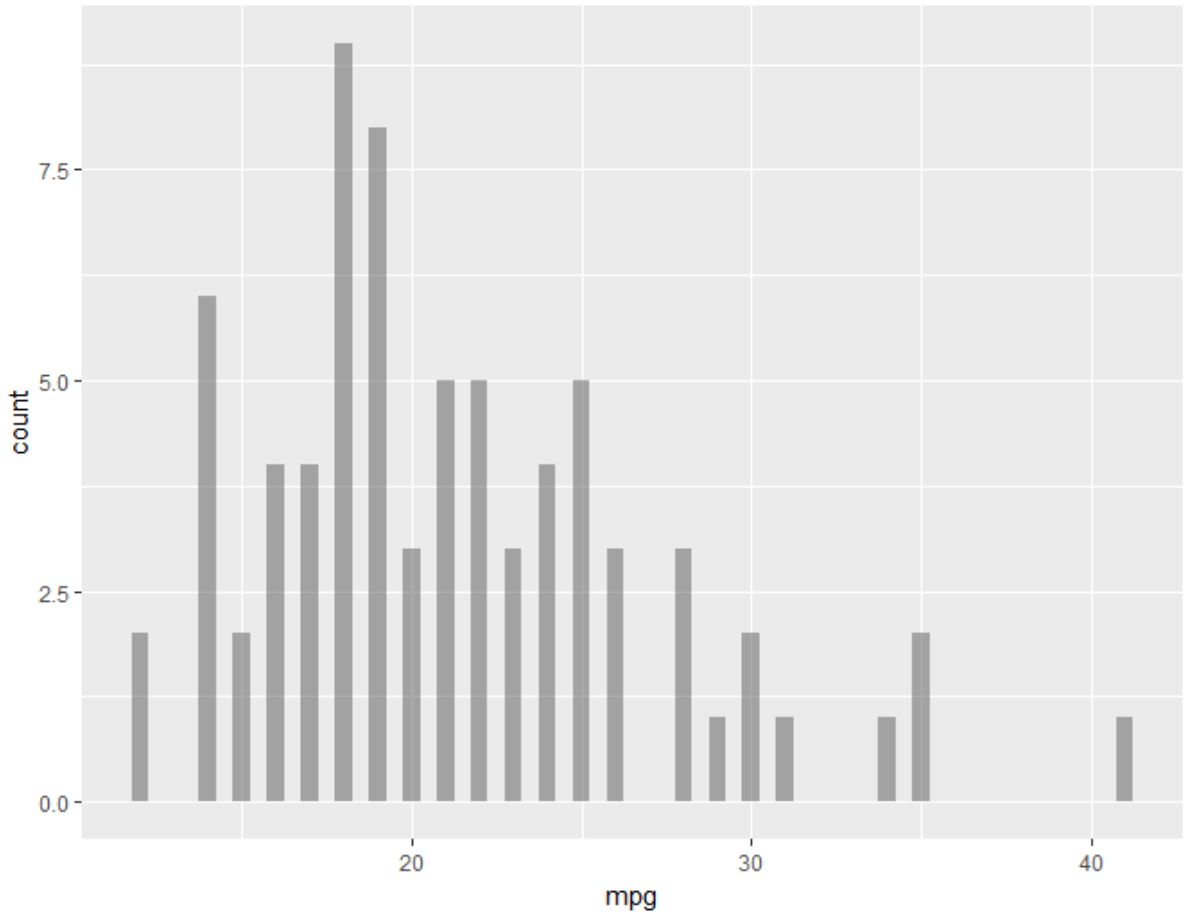
by(auto[, c("price", "mpg")],foreign, summary) ## Split summary wrt to categorical variable.

```
> with(auto, summary(price[mpg >= mean(mpg)])) ##Conditional Summary
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   3299    3990    4482    4880    5250    9735
> by(auto[, c("price", "mpg")],foreign, summary) ##Split summary
foreign: Domestic
     price            mpg
 Min.   : 3291   Min.   :12.00
 1st Qu.: 4186   1st Qu.:16.75
 Median : 4782   Median :19.00
 Mean   : 6072   Mean   :19.83
 3rd Qu.: 6200   3rd Qu.:22.00
 Max.   :15906   Max.   :34.00
--------------------------------------------------------------------------------------------------------------------------
foreign: Foreign
     price            mpg
 Min.   : 3748   Min.   :14.00
 1st Qu.: 4522   1st Qu.:21.00
 Median : 5759   Median :24.50
 Mean   : 6385   Mean   :24.77
 3rd Qu.: 7068   3rd Qu.:27.50
 Max.   :12990   Max.   :41.00
```

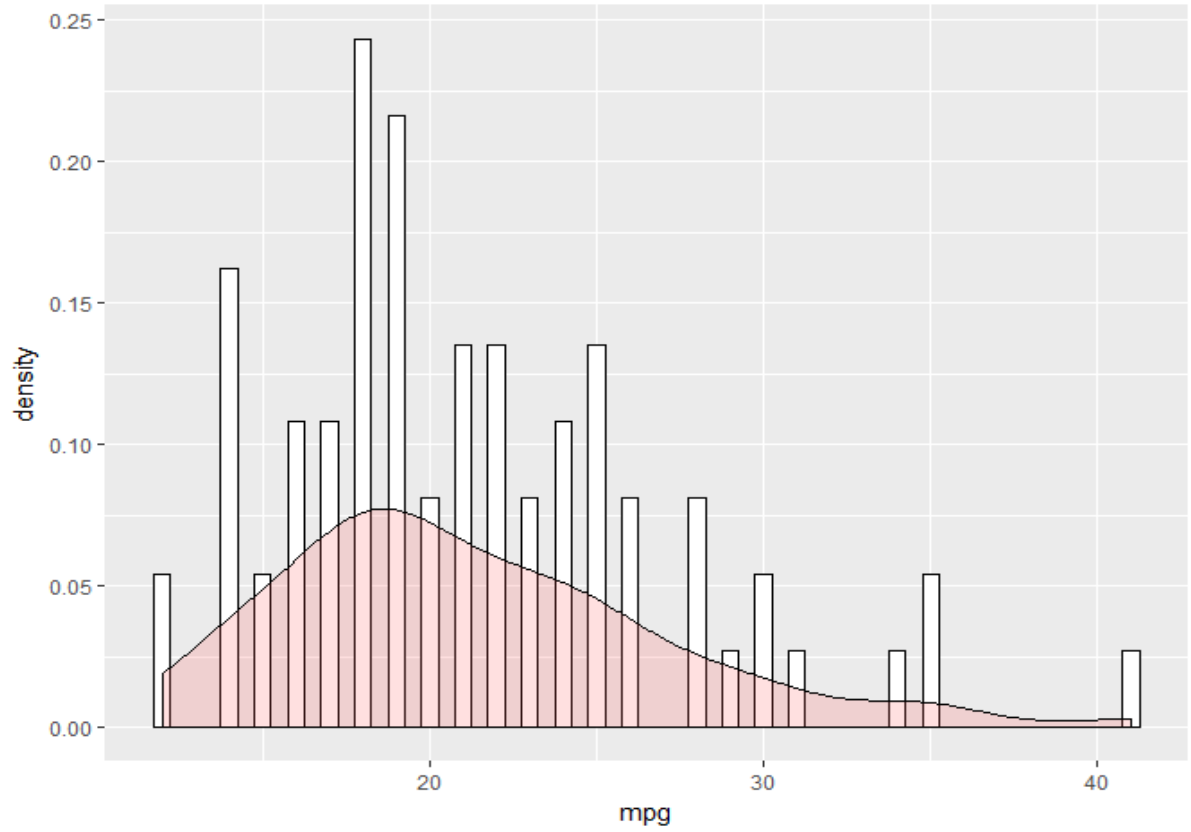
This is basic of knowing data set, now let's have some graphs:

Data Visualisations and interpretations;

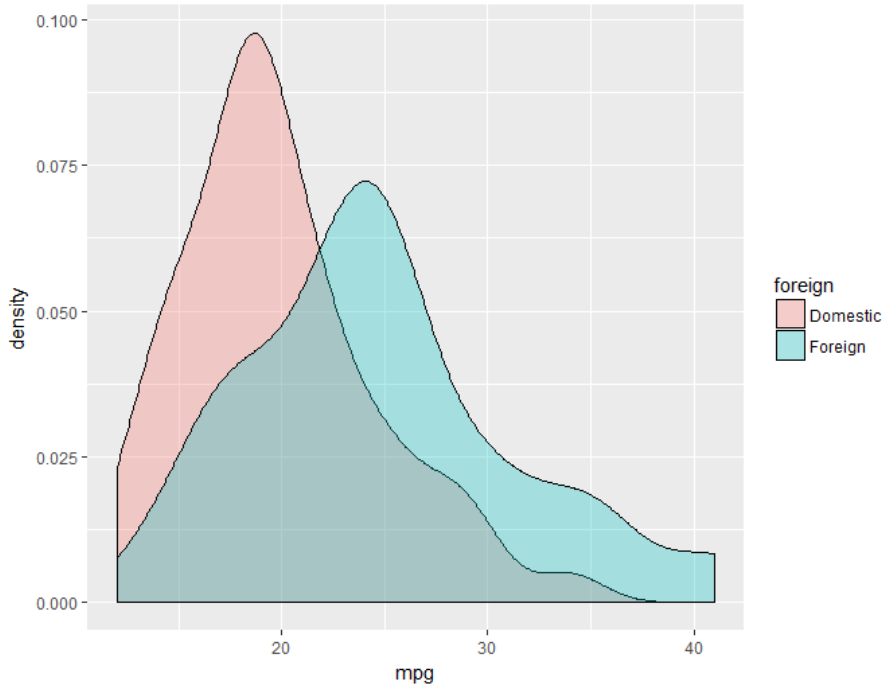
ggplot(auto, aes(x=mpg))+ geom_histogram(binwidth=.5, alpha=.5, position="identity")



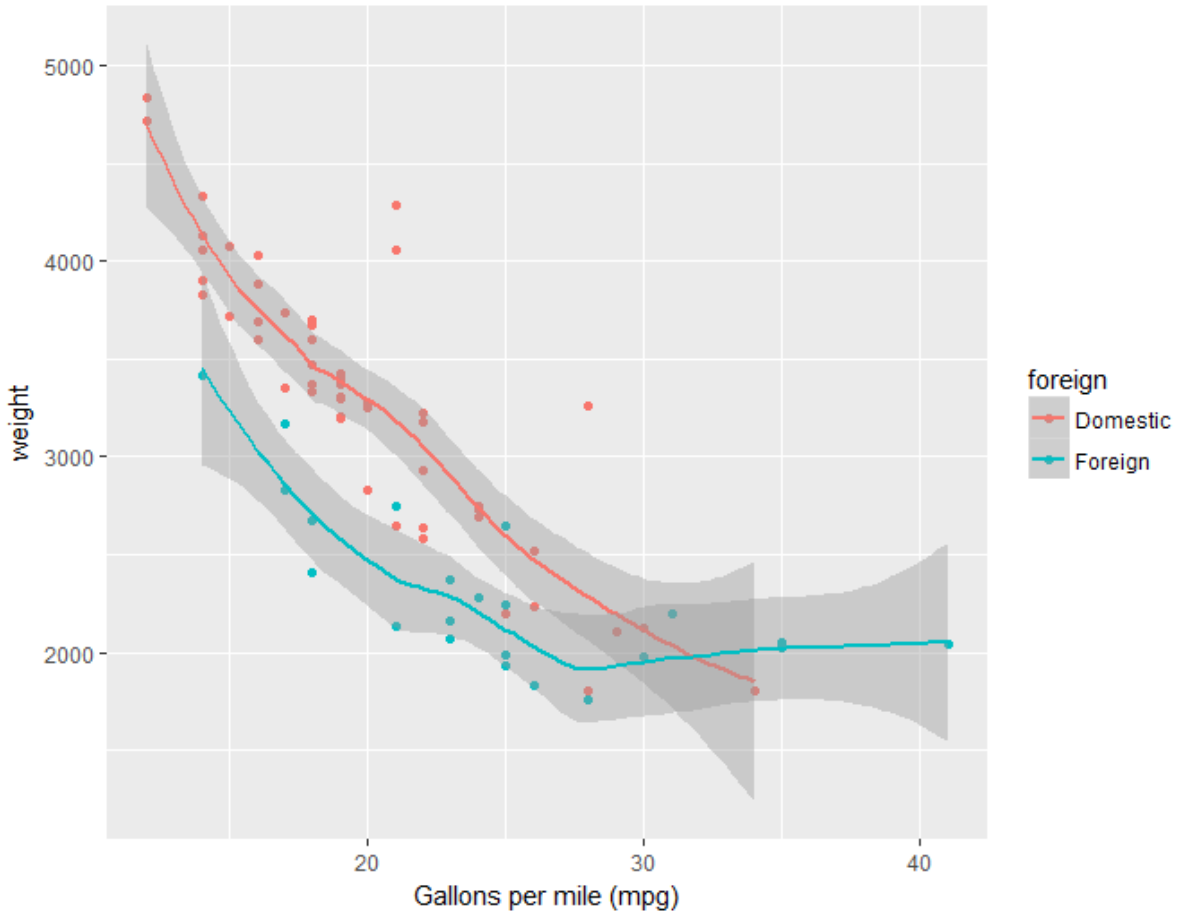

```
ggplot(auto, aes(x=mpg)) +  geom_histogram(aes(y=..density..), binwidth=.5,
colour="black", fill="white") + geom_density(alpha=.2, fill="#FF6666")
```




```
ggplot(auto, aes(x=mpg, fill=foreign)) + geom_density(alpha=.3)
```



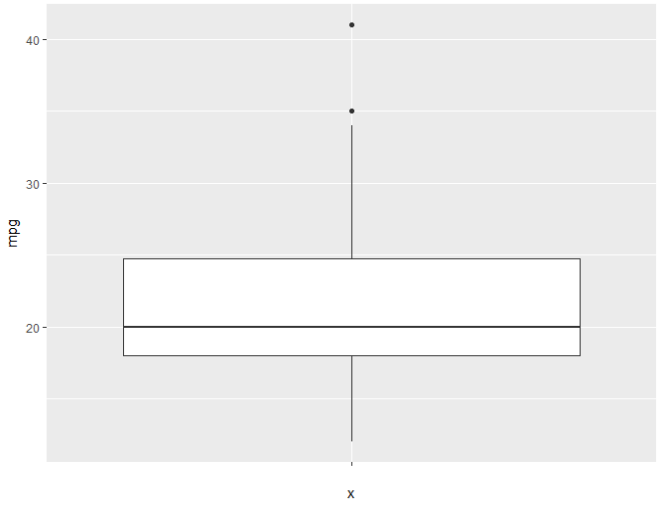

```
qplot(mpg, weight, data = auto, color = foreign, xlab = {"Gallons per mile (mpg)"}, geom = c("point", "smooth"))
```



We can inference that there is some linear relationship between weight and mpg and so is our goal. All though we need to do hypothesis testing for this.

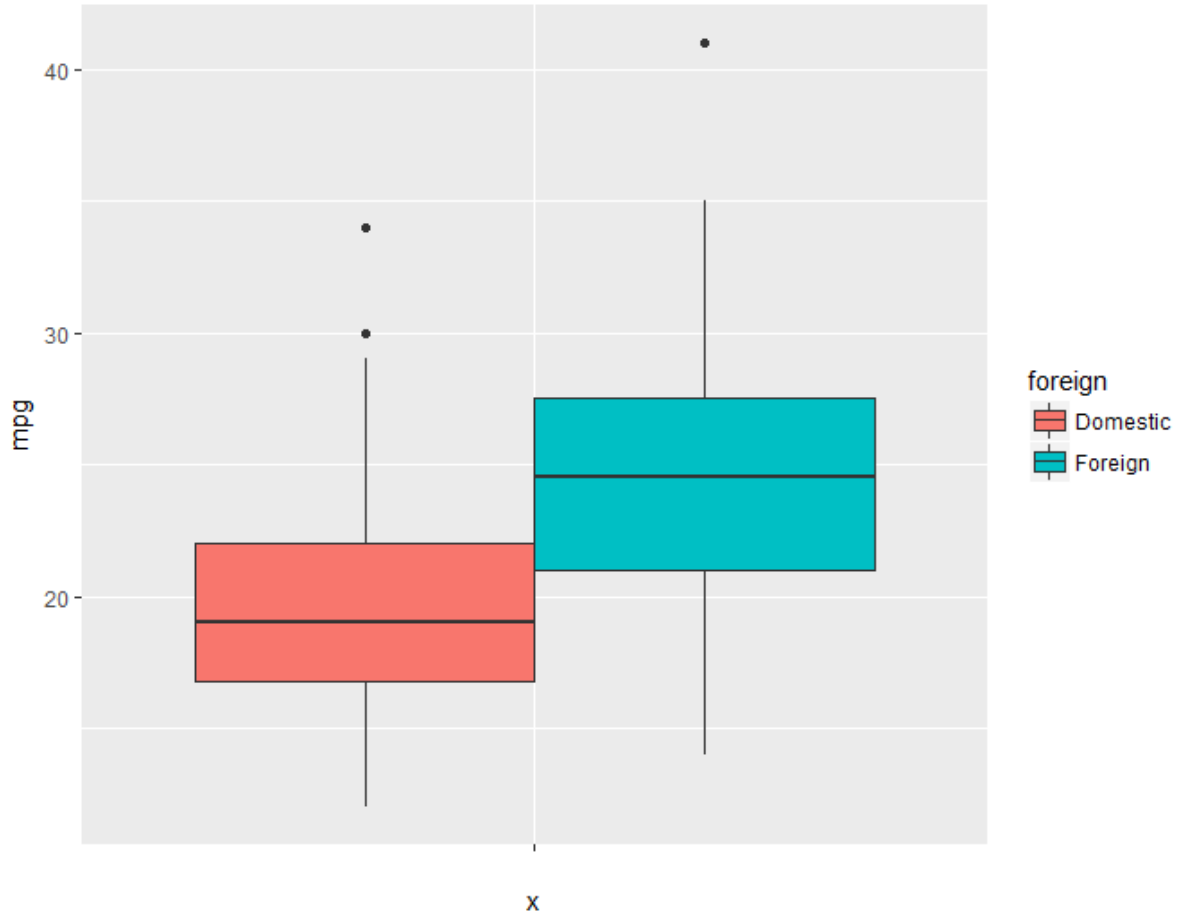
Understanding the summary and outliers:

ggplot(auto, aes(x="", y=mpg)) + geom_boxplot() ## the dots represent outliers , The broad line is Median, this is also used to understand skewness for the variable.

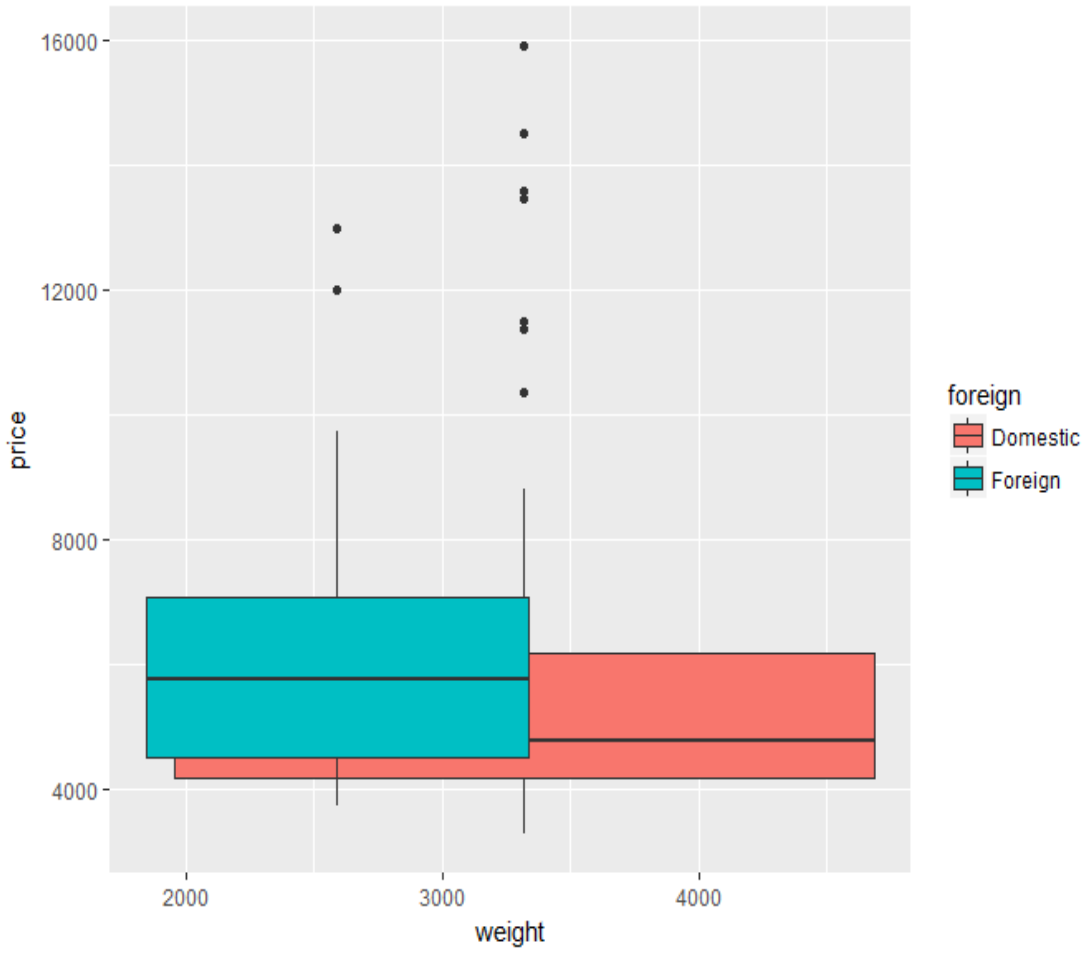


ggplot(auto, aes(x="", y=mpg, fill=foreign)) + geom_boxplot()

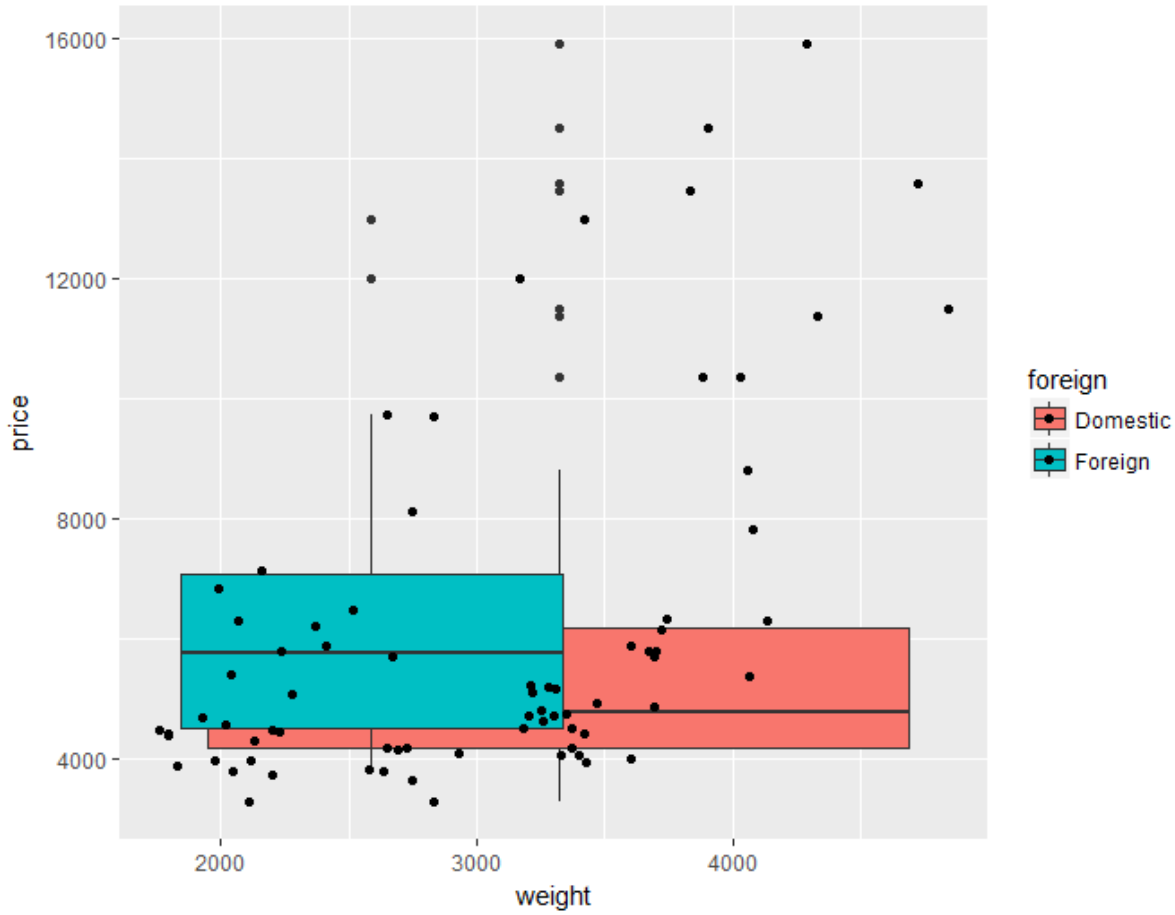
Note: fill is command used in graphs has foreign variable i.e. categorical variable which is used to give a comprehensive analysis:




```
ggplot(auto, aes(x=weight, y=price,  fill=foreign)) +  geom_boxplot()
```




```
ggplot(auto, aes(x=weight, y=price,  fill=foreign)) +  geom_boxplot()+
geom_jitter(width = 0.2)
```



There are lot parameters with boxplot or say with any plot, more you practice more features can be added. R has very rich developed library GGPLOT2.

Lets see how we do it in python:

Python Codes:

Import Basics library

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns ##for statstical anaylis
%matplotlib inline
```

Data import:

```
auto =  pd.read_csv("YOUR_LOCATION/auto.csv")
```

DataView and summary:

```
auto.size
```

```
auto.columns
```

```
auto.head(10)
```

```
auto.dtypes ## displays the data type of variable
```

```
make            object
price            int64
mpg              int64
rep78          float64
headroom       float64
trunk            int64
weight           int64
length           int64
turn             int64
displacement     int64
gear_ratio     float64
foreign          int64
dtype: object
```

Note: Like in R categorical variable were identified as Factor we miss this here.

auto.describe() ## 8 point summary(Count,mean,std,min,25%,50%,75%,max) but difference in results with repect to R.

	price	mpg	rep78	headroom	trunk	weight	length	turn	displacement	gear_ratio	foreign
count	74.000000	74.000000	69.000000	74.000000	74.000000	74.000000	74.000000	74.000000	74.000000	74.000000	74.000000
mean	6165.256757	21.297297	3.405797	2.993243	13.756757	3019.459459	187.932432	39.648649	197.297297	3.014865	0.297297
std	2949.495885	5.785503	0.989932	0.845995	4.277404	777.193567	22.266340	4.399354	91.837219	0.456287	0.460188
min	3291.000000	12.000000	1.000000	1.500000	5.000000	1760.000000	142.000000	31.000000	79.000000	2.190000	0.000000
25%	4220.250000	18.000000	3.000000	2.500000	10.250000	2250.000000	170.000000	36.000000	119.000000	2.730000	0.000000
50%	5006.500000	20.000000	3.000000	3.000000	14.000000	3190.000000	192.500000	40.000000	196.000000	2.955000	0.000000
75%	6332.250000	24.750000	4.000000	3.500000	16.750000	3600.000000	203.750000	43.000000	245.250000	3.352500	1.000000
max	15906.000000	41.000000	5.000000	5.000000	23.000000	4840.000000	233.000000	51.000000	425.000000	3.890000	1.000000

Let's see the comparison between the variables:

Python > auto['mpg'].describe() vs R > summary(auto$mpg)

Summary	Python	R
count	74	??
mean	21.2973	21
std	5.785503	??
min	12	12
25%	18	18
50%/ Median	20	20
75%	24.75	25
max	41	41

Almost both covers relevant summary like mean, inter-quartile region, max, min; with python giving more insight like std and count of data points but as shown earlier more detail summary in R is given by describe(auto$mpg).

```
> describe(auto$mpg)
auto$mpg
       n  missing distinct     Info     Mean      Gmd      .05      .10      .25      .50      .75      .90      .95
      74        0       21    0.995     21.3    6.355    14.00    14.30    18.00    20.00    24.75    28.70    32.05

lowest : 12 14 15 16 17, highest: 30 31 34 35 41
```

auto[['mpg','price']].describe() ## Selectively summary close look at the syntax [[‘’ , ‘’]] for more than one variable.

	mpg	price
count	74.000000	74.000000
mean	21.297297	6165.256757
std	5.785503	2949.495885
min	12.000000	3291.000000
25%	18.000000	4220.250000
50%	20.000000	5006.500000
75%	24.750000	6332.250000
max	41.000000	15906.000000

auto.groupby('foreign')[['mpg','price']].describe() ## Conditional summary but hey look at the complexity ☹.

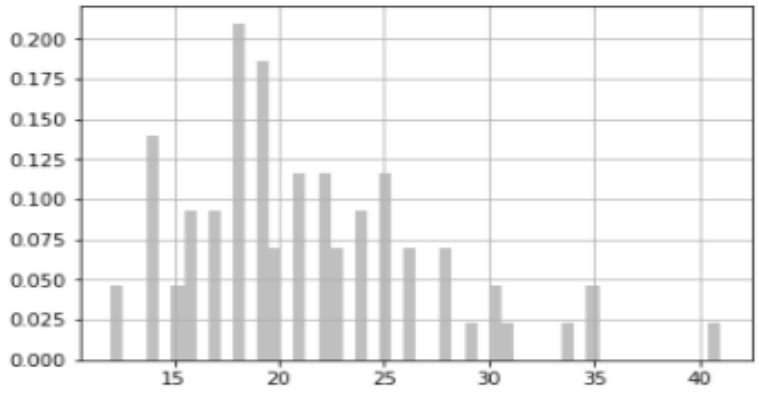
	mpg								price							
	count	mean	std	min	25%	50%	75%	max	count	mean	std	min	25%	50%	75%	max
foreign																
0	52.0	19.826923	4.743297	12.0	16.75	19.0	22.0	34.0	52.0	6072.423077	3097.104279	3291.0	4185.5	4782.5	6199.5	15906.0
1	22.0	24.772727	6.611187	14.0	21.00	24.5	27.5	41.0	22.0	6384.681818	2621.915083	3748.0	4521.5	5759.0	7067.5	12990.0

auto.groupby('foreign').agg({ 'price' : np.mean ,'headroom': np.mean}) ## One more way to get to the summary, lot more are their, more you practice more will you know about the language.

Visualisation:

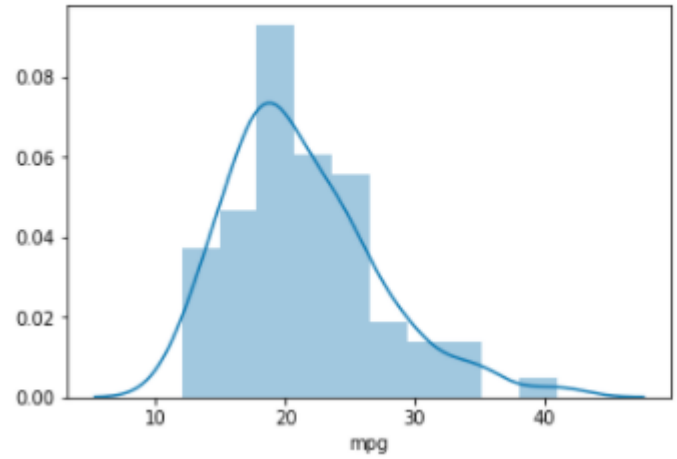
```
plt.hist(auto['mpg'], 50, normed=1, facecolor='grey', alpha=0.5)
plt.grid(True)
plt.show()
```

```
plt.hist(auto['mpg'], 50, normed=1, facecolor='grey', alpha=0.5)
plt.grid(True)
plt.show()
```



sns.distplot(auto['mpg']) ; ## plot with normal distribution curve default is histogram

```
sns.distplot(auto['mpg']);
```



auto.hist(bins=50,figsize=(15,15))

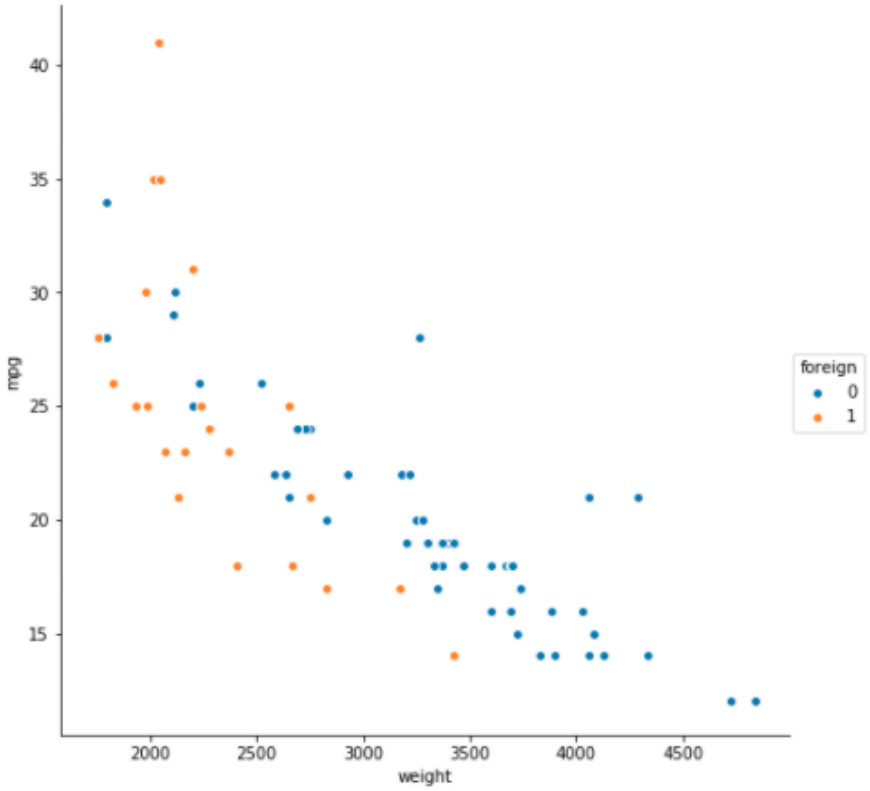
plt.show() ## display all the variable histogram

?sns.pairplot ## use this command to know more about the function.

sns.pairplot(auto, x_vars=['weight'], y_vars='mpg',size= 7 ,hue="foreign" ,aspect= 1) ## as above in R same plot.

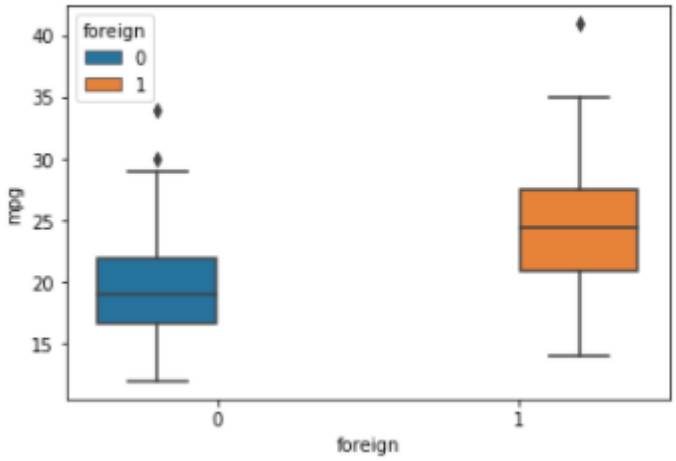
```
sns.pairplot(auto, x_vars=['weight'], y_vars='mpg',size= 7 ,hue="foreign" ,aspect= 1)
```

```
<seaborn.axisgrid.PairGrid at 0x2707807c8d0>
```



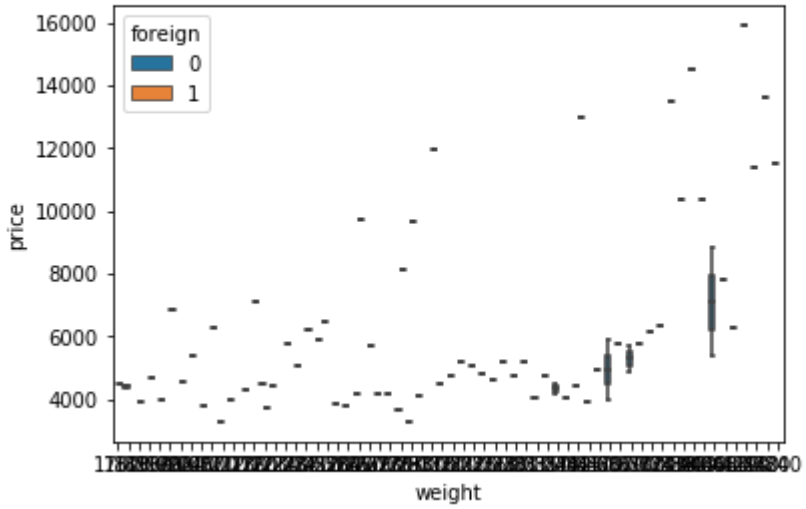
sns.boxplot(x="foreign", y="mpg", hue="foreign", data=auto);

```
sns.boxplot(x="foreign", y="mpg", hue="foreign", data=auto);
```



sns.boxplot(x="price", y="weight", hue="foreign", data=auto); ## while replicating the same from R, the graphs are not that great.

```
sns.boxplot(x="weight", y="price", hue="foreign", data=auto);
```



So my take, R still leads in terms of EDA from Python.

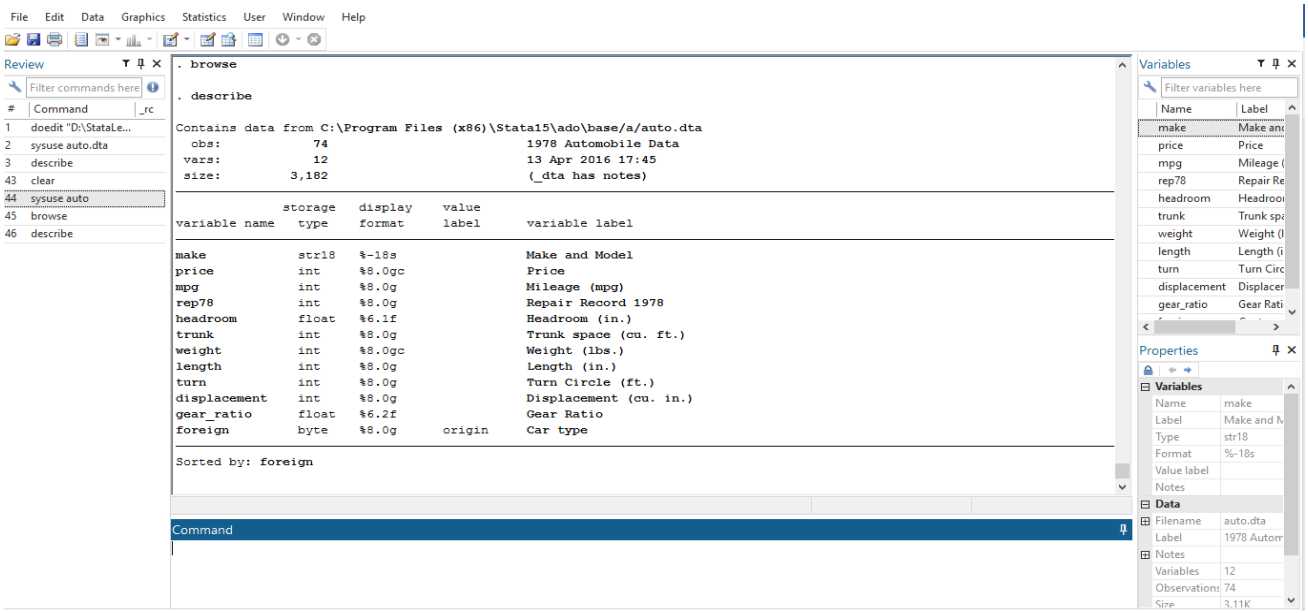
Let's see how STATA treats the auto data set;

STATA Workings:

sysuse auto ##has default data set in their library

browse ## help you view the data just like View in R, an excel look ☺

It takes one data set at a time; the stata Look as below, has a command tab, recent command history tab on left, variables of data set on right Top followed by properties options on right bottom, similar to Rstudio ☺.



describe ## Gives the details of the data set, vairable type , number of records, even the sizeof data; a much more details compared to R and python.

```
. describe

Contains data from C:\Program Files (x86)\Stata15\ado\base/a/auto.dta
  obs:            74                          1978 Automobile Data
 vars:            12                          13 Apr 2016 17:45
 size:         3,182                          (_dta has notes)
-------------------------------------------------------------------------------
              storage   display    value
variable name   type    format     label      variable label
-------------------------------------------------------------------------------
make            str18   %-18s                 Make and Model
price           int     %8.0gc                Price
mpg             int     %8.0g                 Mileage (mpg)
rep78           int     %8.0g                 Repair Record 1978
headroom        float   %6.1f                 Headroom (in.)
trunk           int     %8.0g                 Trunk space (cu. ft.)
weight          int     %8.0gc                Weight (lbs.)
length          int     %8.0g                 Length (in.)
turn            int     %8.0g                 Turn Circle (ft.)
displacement    int     %8.0g                 Displacement (cu. in.)
gear_ratio      float   %6.2f                 Gear Ratio
foreign         byte    %8.0g       origin    Car type
-------------------------------------------------------------------------------
Sorted by: foreign
```

Am saving the data and re-import the data, lets see the change in describe output.

export delimited using "YOUR_Locations\autofulldataset.csv", replace

import delimited D:\StataLearning\StataCourse\autofulldataset.csv, clear

##Clear is always use to clean the memory for the new data set, one data set can be used for operations.

Describe

```
. describe

Contains data
  obs:            74
 vars:            12
 size:         3,330
-------------------------------------------------------------------------------
              storage   display    value
variable name   type    format     label      variable label
-------------------------------------------------------------------------------
make            str17   %17s
price           int     %8.0g
mpg             byte    %8.0g
rep78           byte    %8.0g
headroom        float   %9.0g
trunk           byte    %8.0g
weight          int     %8.0g
length          int     %8.0g
turn            byte    %8.0g
displacement    int     %8.0g
gear_ratio      float   %9.0g
foreign         str8    %9s
-------------------------------------------------------------------------------
Sorted by:
     Note: Dataset has changed since last saved.
```

The result set has less details. Statistical software like R and STATA have detail descriptions of their library data set compared to others.

summarize ## STATA is 5 point summary where relevant information is given. How do we see the NA's, for the variable rep78 glance at the output, Obs is 69 but total Obs is 74, so what goes missing are the 5 data points.

There is no summary for categorical variable (make and foreign) that's something eye catching.

```
. summarize
```

Variable	Obs	Mean	Std. Dev.	Min	Max
make	0				
price	74	6165.257	2949.496	3291	15906
mpg	74	21.2973	5.785503	12	41
rep78	69	3.405797	.9899323	1	5
headroom	74	2.993243	.8459948	1.5	5
trunk	74	13.75676	4.277404	5	23
weight	74	3019.459	777.1936	1760	4840
length	74	187.9324	22.26634	142	233
turn	74	39.64865	4.399354	31	51
displacement	74	197.2973	91.83722	79	425
gear_ratio	74	3.014865	.4562871	2.19	3.89
foreign	0				

summarize mpg

```
. summarize mpg
```

Variable	Obs	Mean	Std. Dev.	Min	Max
mpg	74	21.2973	5.785503	12	41

A comprehensive comparisons between Python Vs R Vs STATA

Summary	Python	R	STATA
count	74	??	74
mean	21.3	21.3	21.3
std	5.786	??	5.786
min	12	12	12
25%	18	18	??
50%/Median	20	20	??
75%	24.75	24.75	??
max	41	41	41

For sure python publishes much more fine-tuned insightful results, R is good.

by foreign: summarize price mpg ##Conditional summary

```
. by foreign: summarize price mpg
```

-> foreign = Domestic

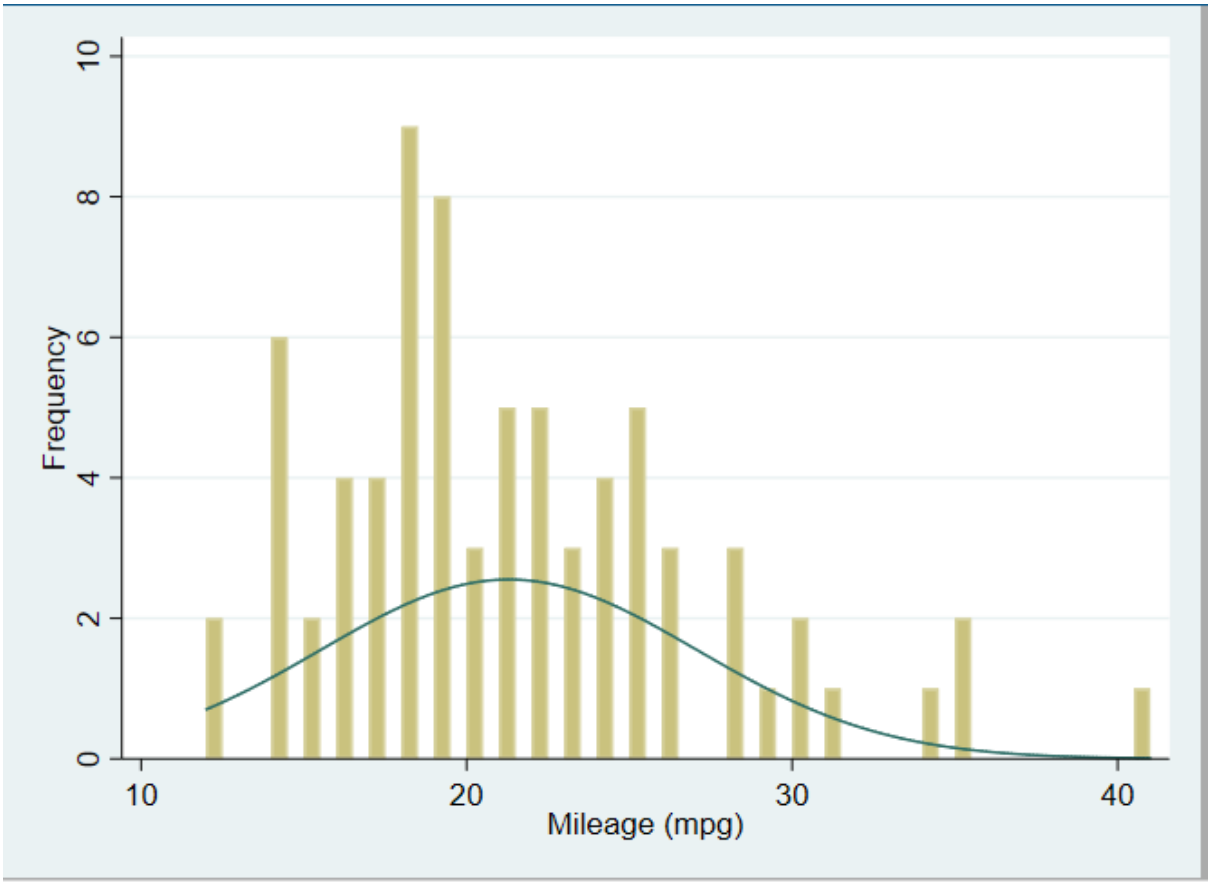
Variable	Obs	Mean	Std. Dev.	Min	Max
price	52	6072.423	3097.104	3291	15906
mpg	52	19.82692	4.743297	12	34

-> foreign = Foreign

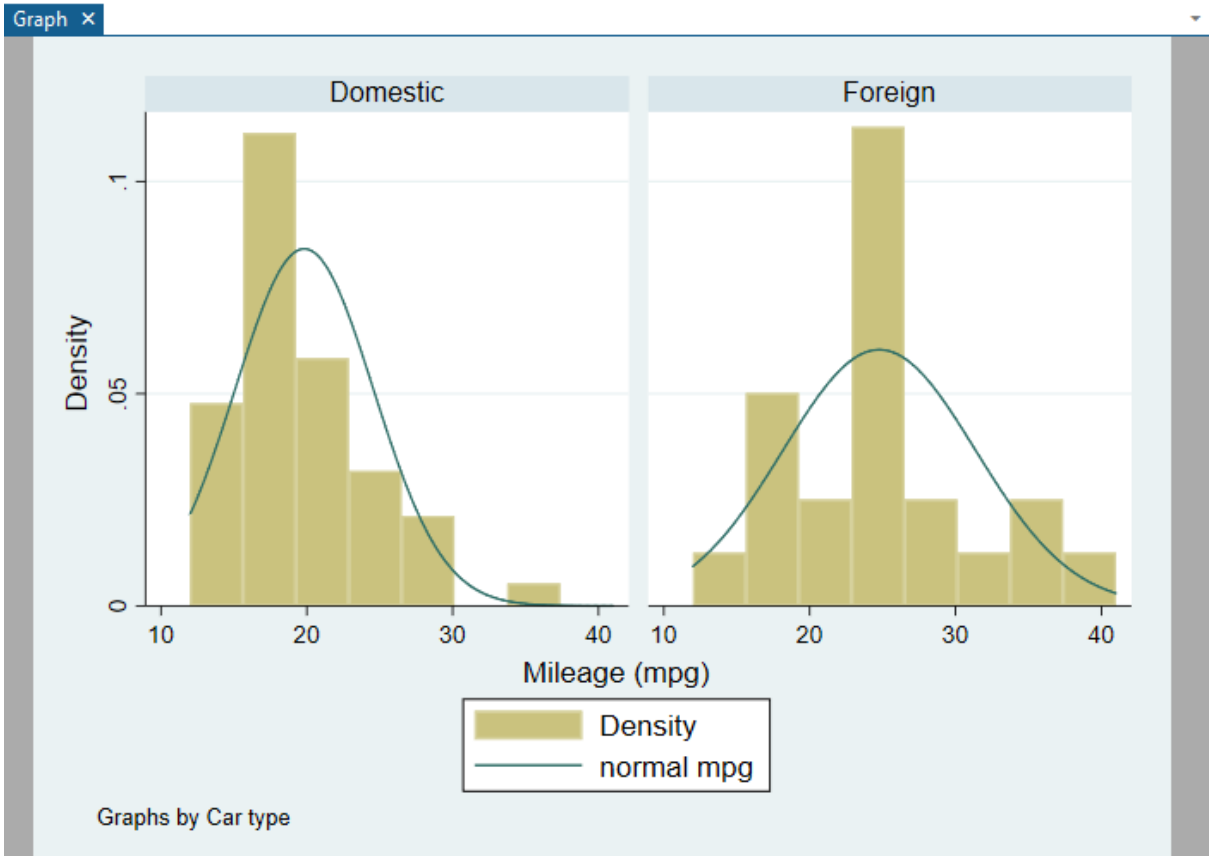
Variable	Obs	Mean	Std. Dev.	Min	Max
price	22	6384.682	2621.915	3748	12990
mpg	22	24.77273	6.611187	14	41

Visualisations:

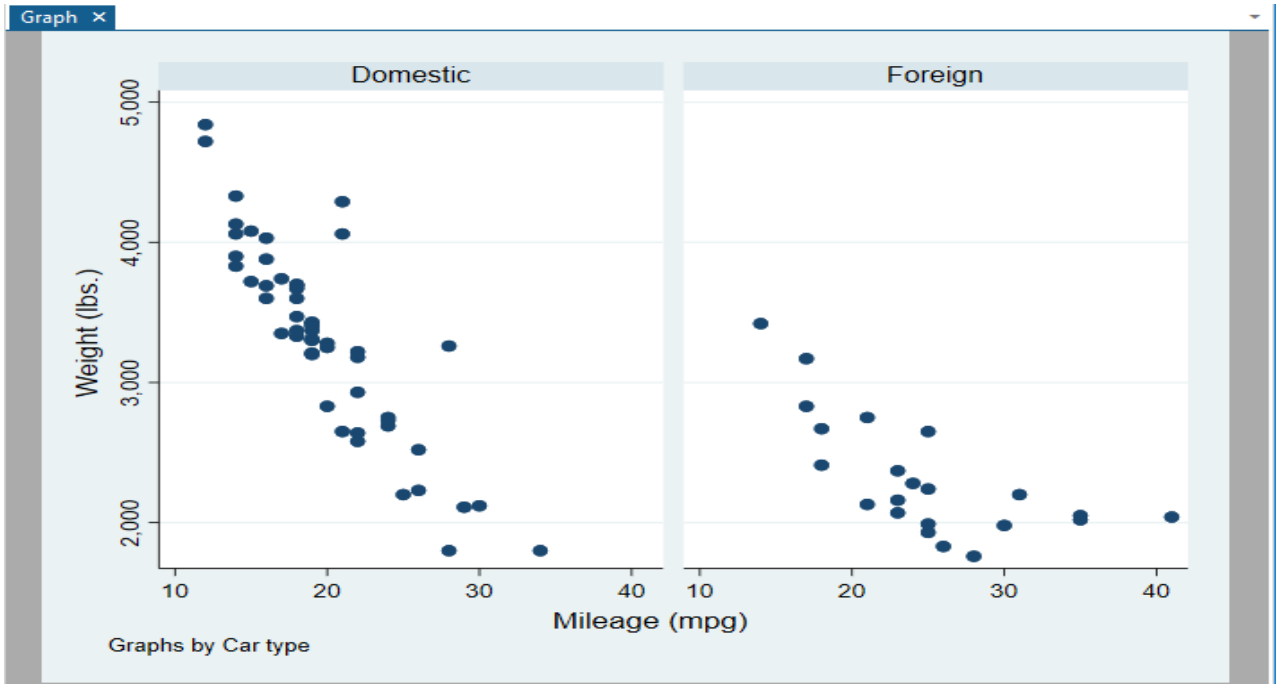
histogram mpg, width(.5) frequency normal ## the graph is good with normal distribution line is coming by default .. that's super cool.



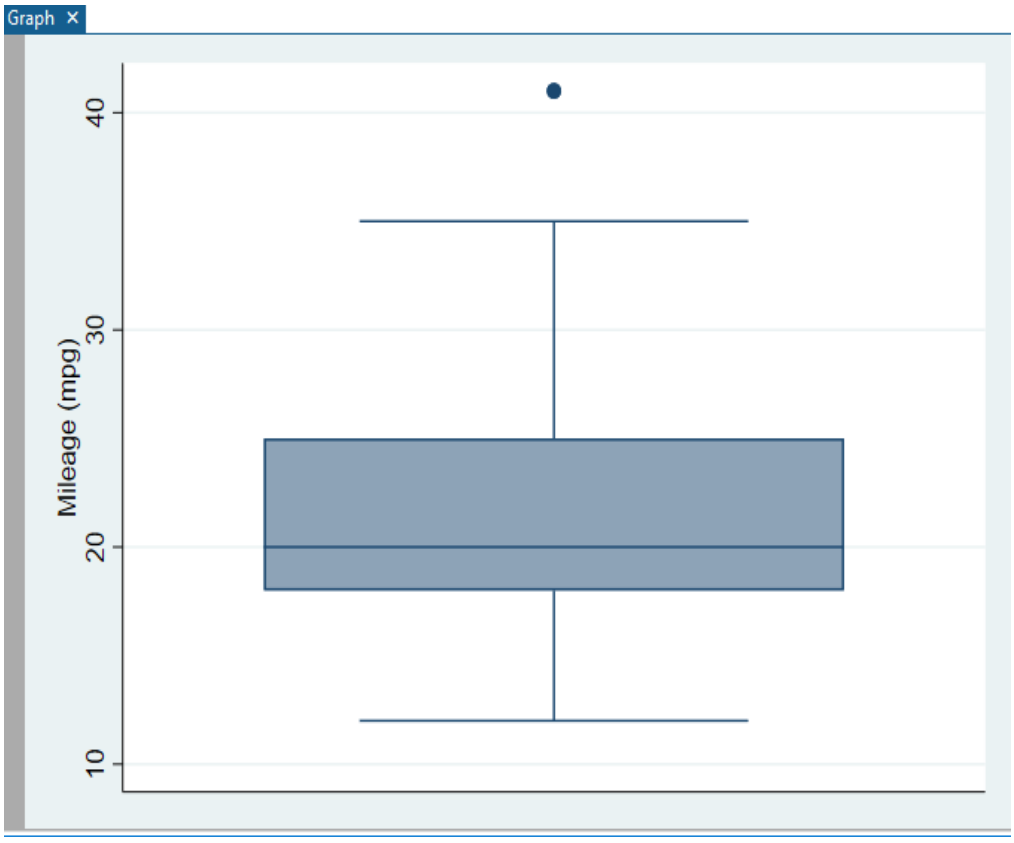
histogram mpg, normal by(foreign)



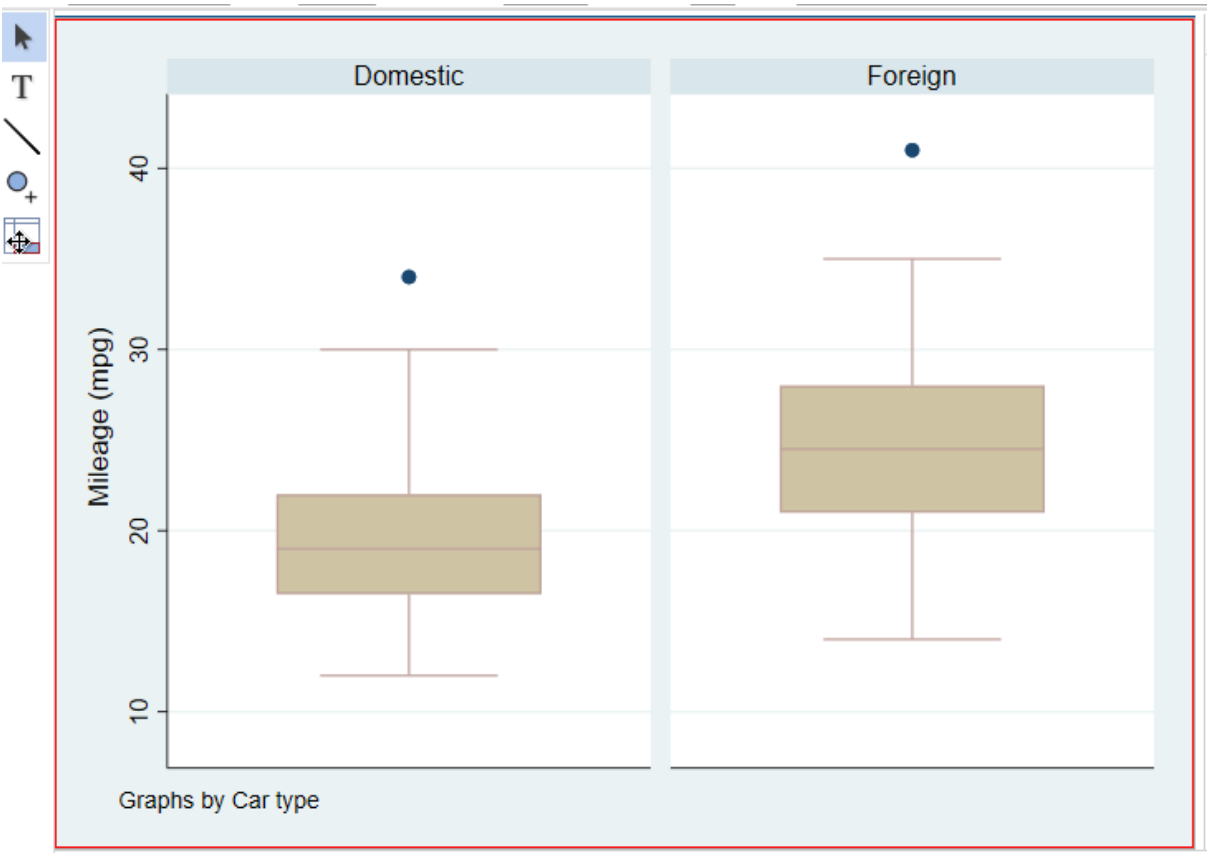
twoway (scatter weight mpg, sort), by(foreign) ##terrific graph, compared to R which has overlapping visuals but this is also fine in intimidating what R and Pyhton did. apart from others parameter graph looks good.



graph box mpg ## the graph has only one outlaier while others gave us two ☺.



graph box mpg, by(foreign) ## other graphs we have seen with difference in colour wrt Foreign variable.



graph box weight price, by(foreign) ## good representation overall, ideal for analysis.

So we have completed basics for all the 3 RPS, python has good summary statistics but not good for EDA but R and STATA have powerful EDA.

THE BASICS IS DONE,

LINEAR REGRESSION

We will start with two sample T-test for the variables miles per gallon and foreign; lets follow the process and why can explained later when statics for the problem is taken cared. Till now what ever we did is to understand the programming languages:

R codes:

```
with(auto, t.test(mpg ~ foreign, var.equal = TRUE))
```

```
> with(auto, t.test(mpg ~ foreign, var.equal = TRUE))

        Two Sample t-test

data:  mpg by foreign
t = -3.6308, df = 72, p-value = 0.0005254
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -7.661225 -2.230384
sample estimates:
mean in group Domestic  mean in group Foreign
              19.82692                24.77273

>
```

Ponder over WHY this test ??

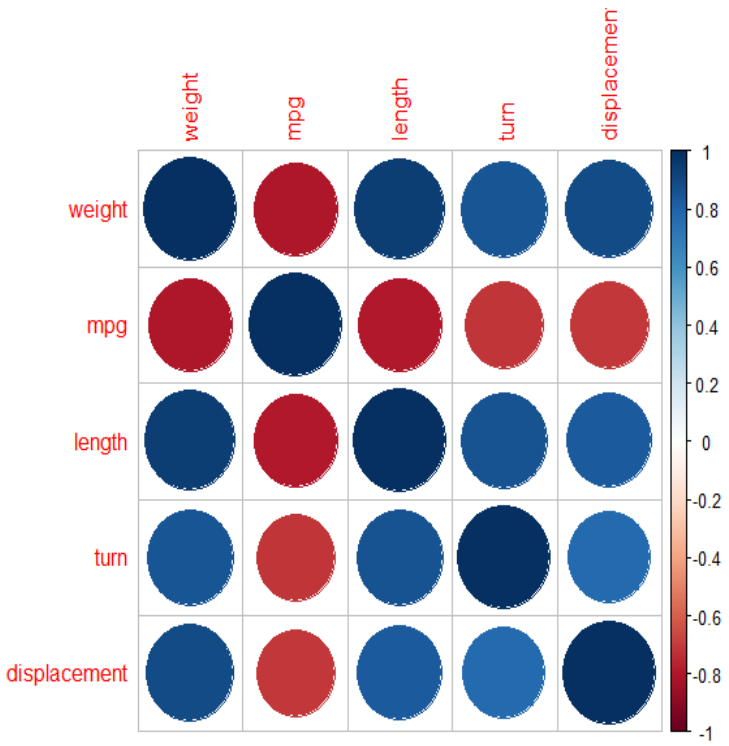
Correlations:

```
library(corrplot)

cor(auto[, c("weight", "mpg", "length", "turn", "displacement")])
```

```
> cor(auto[, c("weight", "mpg", "length", "turn", "displacement")])
                weight        mpg     length       turn displacement
weight       1.0000000 -0.8071749  0.9460086  0.8574429    0.8948958
mpg         -0.8071749  1.0000000 -0.7957794 -0.7191863   -0.7056426
length       0.9460086 -0.7957794  1.0000000  0.8642612    0.8351400
turn         0.8574429 -0.7191863  0.8642612  1.0000000    0.7767647
displacement 0.8948958 -0.7056426  0.8351400  0.7767647    1.0000000
```

But Hey graphs are better to visualise;

```
M <- cor(auto[, c("weight", "mpg", "length", "turn", "displacement")])

corrplot(M, method = "circle")
```



Now less do regression:

qplot(mpg, weight, data = auto) ##Check previous graph for relationship

##A plot of mileage by weight shows the expected, decreasing, and relationship.

Modelling:

We want to model the relationship between MPG and weight,

and we will estimate a linear regression model using the lm() command.

mod1 <- lm(mpg ~ weight + foreign, auto)

summary(mod1)

```
> mod1 <- lm(mpg ~ weight + foreign, auto)
> summary(mod1)

Call:
lm(formula = mpg ~ weight + foreign, data = auto)

Residuals:
    Min      1Q  Median      3Q     Max
-6.1529 -1.9712 -0.4534  0.8083 14.4096

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)    41.6797023  2.1655472  19.247  < 2e-16 ***
weight         -0.0065879  0.0006371 -10.340 8.28e-16 ***
foreignForeign -1.6500291  1.0759941  -1.533     0.13
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.407 on 71 degrees of freedom
Multiple R-squared:  0.6627,    Adjusted R-squared:  0.6532
F-statistic: 69.75 on 2 and 71 DF,  p-value: < 2.2e-16
```

Not discussing anything now lets execute the same in python; this section is Programming language understandings and its implications, pros and cons;

Python Codes:

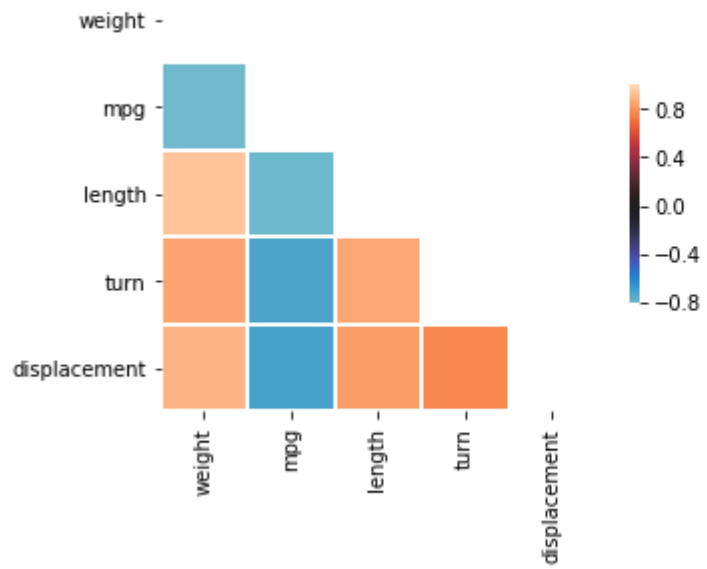
```
from scipy import stats

stats.ttest_ind(auto['foreign'],auto['mpg'], equal_var = True)
```

```
Ttest_indResult(statistic=-31.126083662971933, pvalue=2.514773373571505
9e-66)
```

Correlations:

```
# Generate a mask for the upper triangle
mask = np.zeros_like(corr_matrix, dtype=np.bool)
mask[np.triu_indices_from(mask)] = True

# Draw the heatmap with the mask and correct aspect ratio
sns.heatmap(corr_matrix, vmax=1,mask=mask, center=0,square=True,
linewidths=1, cbar_kws={"shrink": .5})
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x27078f9fba8>
```



Linear Regression In Python;

```
from sklearn.linear_model import LinearRegression
lr = LinearRegression()
lr.fit(x_train,y_train)
```

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=1, normalize=F
alse)
```

```
print (lr.intercept_)
print (lr.coef_)
```

```
[ 41.67970233]
[[-0.00658789 -1.65002911]]
```

Python output are same but it is not in detailed as in R, on knowing the output in details we can appreciate the outputs; latter we gona discuss this.

Let’s look how STATA response it:

STATA code:

T-Test

```
by foreign, sort : ttest mpg == 0
```

```
-> foreign = Domestic

One-sample t test
```

Variable	Obs	Mean	Std. Err.	Std. Dev.	[95% Conf.	Interval]
mpg	52	19.82692	.657777	4.743297	18.50638	21.14747

```
    mean = mean(mpg)                                          t =  30.1423
Ho: mean = 0                                  degrees of freedom =       51

    Ha: mean < 0                 Ha: mean != 0                 Ha: mean > 0
 Pr(T < t) = 1.0000         Pr(|T| > |t|) = 0.0000          Pr(T > t) = 0.0000
```

```
-> foreign = Foreign

One-sample t test
```

Variable	Obs	Mean	Std. Err.	Std. Dev.	[95% Conf.	Interval]
mpg	22	24.77273	1.40951	6.611187	21.84149	27.70396

```
    mean = mean(mpg)                                          t =  17.5754
Ho: mean = 0                                  degrees of freedom =       21

    Ha: mean < 0                 Ha: mean != 0                 Ha: mean > 0
 Pr(T < t) = 1.0000         Pr(|T| > |t|) = 0.0000          Pr(T > t) = 0.0000
```

OMG so detailed, this is how it should be unlike python, so funcatinlaity wise for stastics stata gives more comprehensive results, than R.

correlate weight mpg length turn displacement

```
. correlate weight mpg length turn displacement
(obs=74)
```

	weight	mpg	length	turn	displa~t
weight	1.0000				
mpg	-0.8072	1.0000			
length	0.9460	-0.7958	1.0000		
turn	0.8574	-0.7192	0.8643	1.0000	
displacement	0.8949	-0.7056	0.8351	0.7768	1.0000

Have not found way to do the correlation matrix graph;

Linear Regresoion:

regress mpg weight foreign

```
. regress mpg weight  foreign
```

Source	SS	df	MS	Number of obs	=	74
				F(2, 71)	=	69.75
Model	1619.2877	2	809.643849	Prob > F	=	0.0000
Residual	824.171761	71	11.608053	R-squared	=	0.6627
				Adj R-squared	=	0.6532
Total	2443.45946	73	33.4720474	Root MSE	=	3.4071

mpg	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
weight	-.0065879	.0006371	-10.34	0.000	-.0078583	-.0053175
foreign	-1.650029	1.075994	-1.53	0.130	-3.7955	.4954422
_cons	41.6797	2.165547	19.25	0.000	37.36172	45.99768

So to conclude, we have witnessed how the Programming languages/Tool are aligned with statistical learning, so my call to any beginner can start with R and then take python; STATA or any paid tool can handy if we know the process flow and interpretation of the results. We are done with running code, in my next set let's analyse the results with Statistical tarkka.

Hope you have liked the bit and pieces of learning I can provide, hit the like button if u have not liked it; please comment down your suggestions and feedbacks.

See you on reading my next post ☺

Happy Learning ☺

Learn and Let Learn ☺