# Meghnath Reddy Challa- OUID (113508098)

# IDA HOMEWORK-2

**Marked in yellow=R CODE  Output=Result when marked R code is executed**

**Required Packages:**

1) Require (tidyverse) #To install and load multiple tidyverse packages
2) Require (plyr) #package to implement split-apply combine pattern
3) Require (mice) #Multivariate Imputation by Chained equation to deal with missing data
4) Require (VIM) #library for Visualization and imputation of Missing values
5) Require (Amelia) #to load "freetrade" data

## Question 1) Learning GGplot

**Question 3.2.4:**

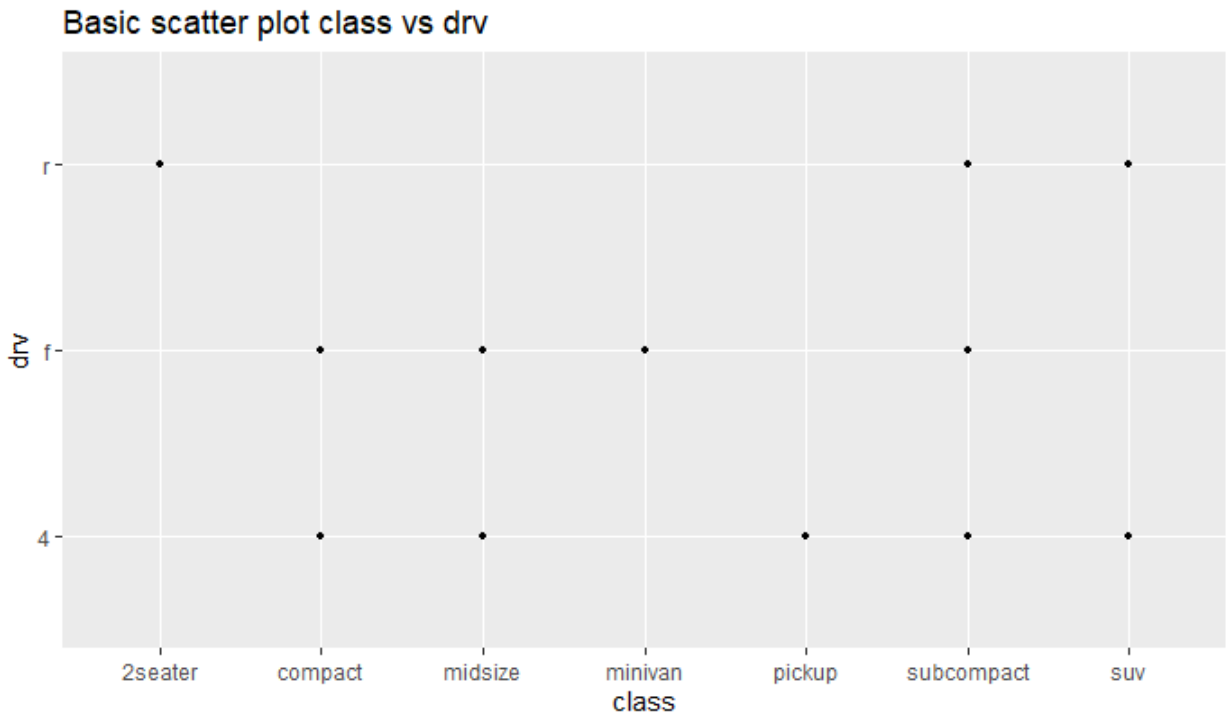
**Exercise 4:** Basic scatter plot hwy vs cyl

```
ggplot(mpg, aes(x=cyl, y=hwy)) + geom_point(size=2, shape=20)
```


**Insight:** With the help of this scatterplot we can easily distinguish range of number of miles the car can travel based on the type of cylinder.

**Exercise 5: Basic scatter plot class vs drv**

```
ggplot(mpg, aes(x=class, y=drv)) + geom_point(size=2, shape=20)
```



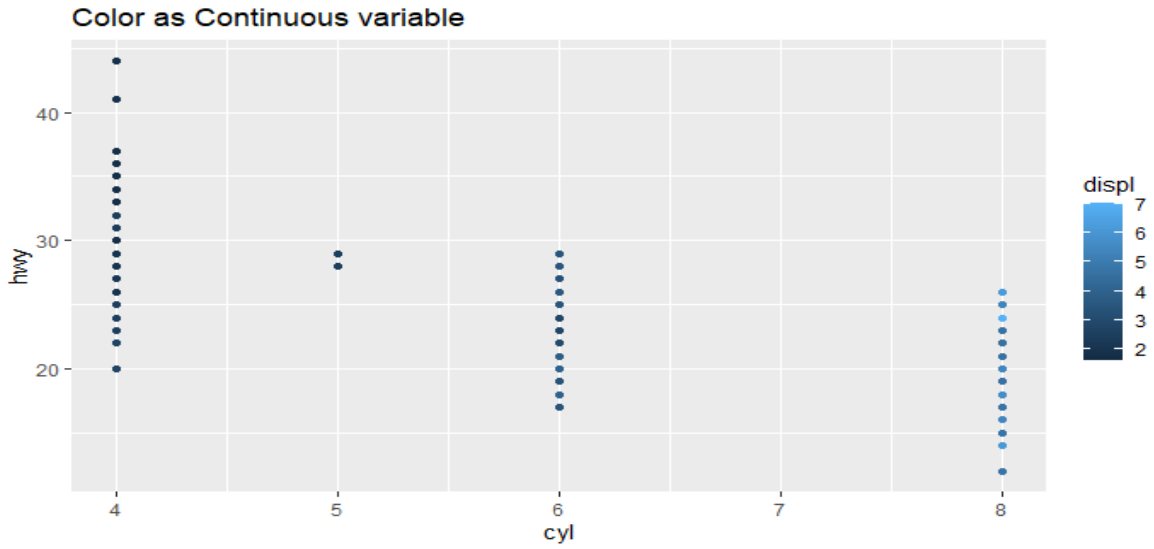

**Insight:** Since both the variables class and drv are categorical values, the number of unique combinations of (x,y) are limited which is 3 for drv * 7 for class. Scatterplot works better when the variables compared are unique and continuous.

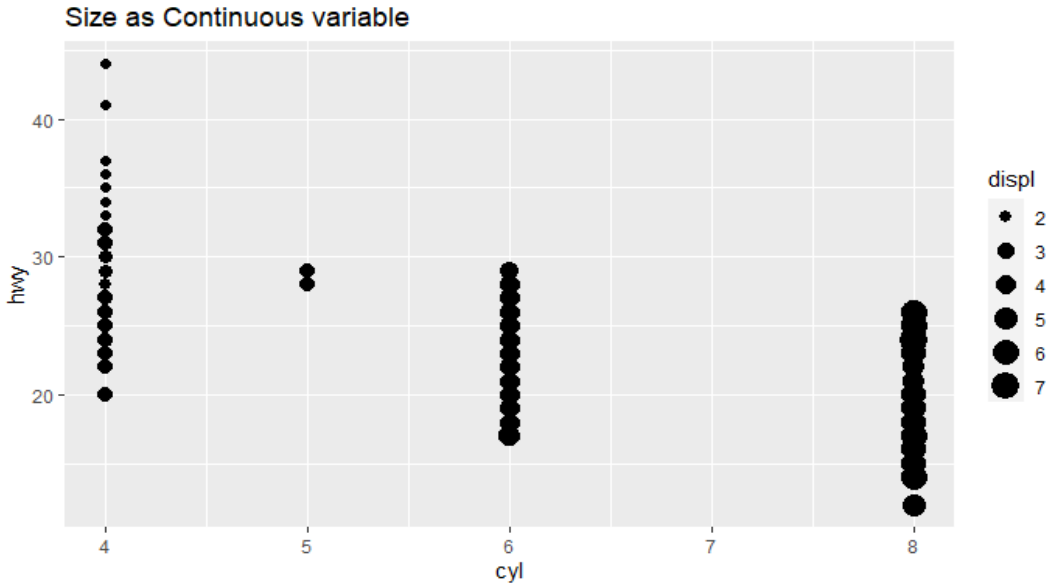
**Question 3.3.1:**

**Exercise 3: Map continuous variable**

a) **ggplot(mpg, aes(x = cyl, y = hwy, color = displ)) + geom_point() #color = displ which is continuous**
b) **ggplot(mpg, aes(x = cyl, y = hwy, size = displ)) + geom_point() #size=displ which is continuous**
c) **ggplot(mpg, aes(x = cyl, y = hwy, shape = displ)) + geom_point() #displ which is continuous**

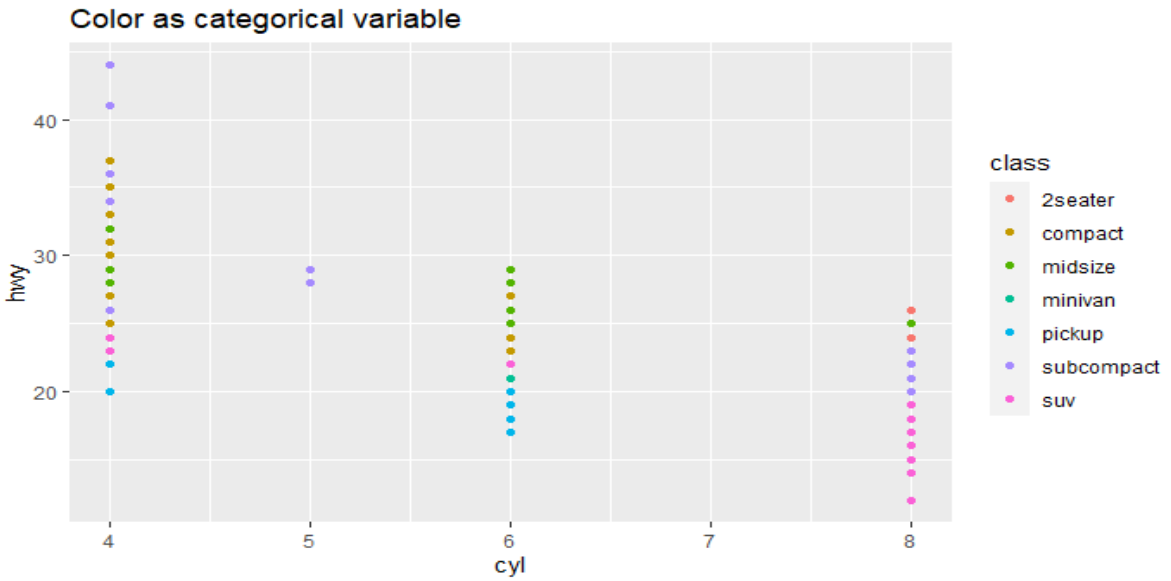
**Insight**: When we use shape as displacement which is a continuous variable, we can get an error since the continuous variables cannot be mapped to shape parameter.

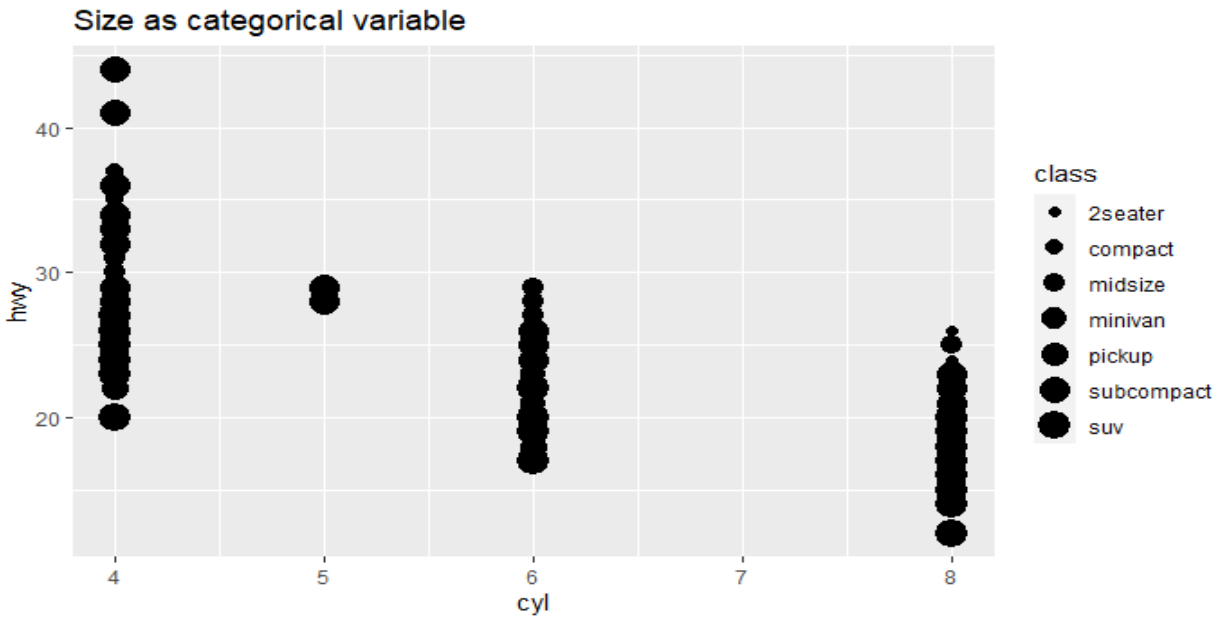


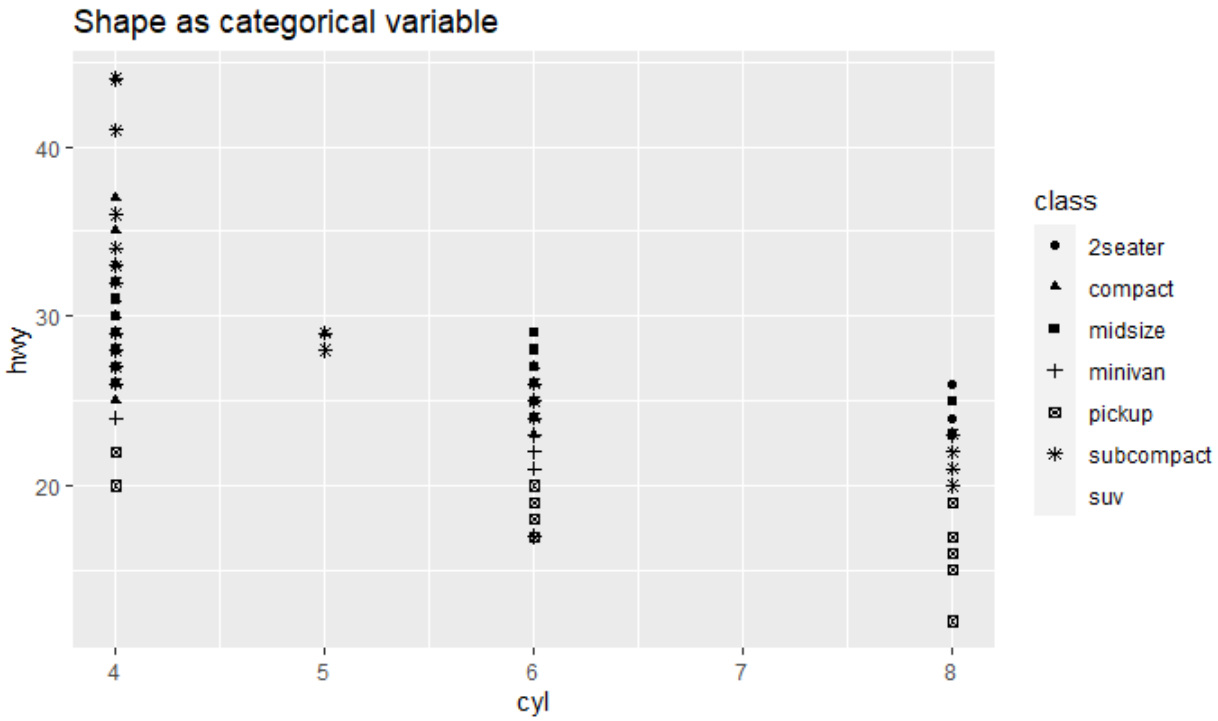



**Exercise 3: Map categorical variables**

**a) ggplot(mpg, aes(x = cyl, y = hwy, color = class)) + geom_point() +**
**labs(title="Color as categorical variable")#color = class which is categorical**
**b) ggplot(mpg, aes(x = cyl, y = hwy, size = class)) + geom_point() +**
**labs(title="Size as categorical variable") #size=class which is categorical**
**c) ggplot(mpg, aes(x = cyl, y = hwy, shape = class)) + geom_point() +**
**labs(title="Shape as categorical variable")#shape=class which is categorical**



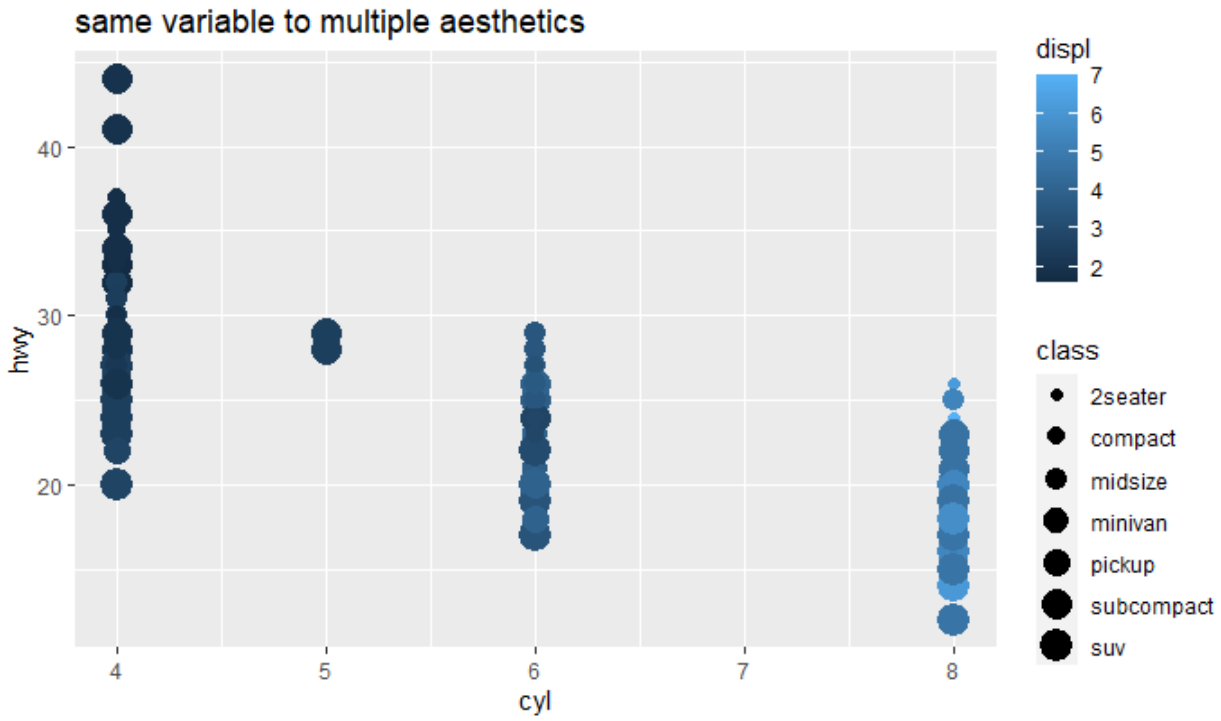






**Insight:** Categorical variable can we work as shape parameter as well.

**Exercise 4: Same variable to multiple aesthetics**

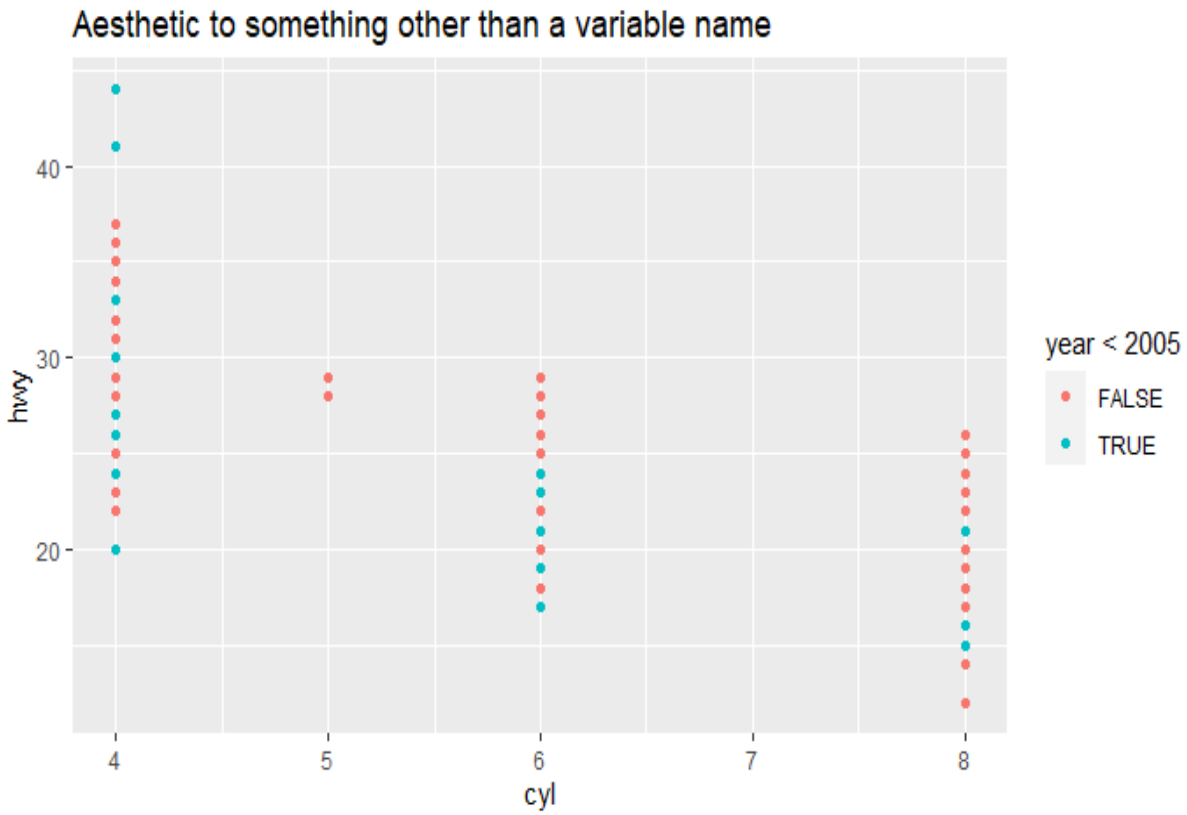
**ggplot(mpg, aes(x = cyl, y = hwy, color = displ,size=class)) + geom_point() #color = displ which is continuous**



**Insight:** Code runs without any error, but we can avoid allocating same variable to multiple aesthetics to prevent redundancy.

**Exercise 6: Aesthetic to something other than a variable name**

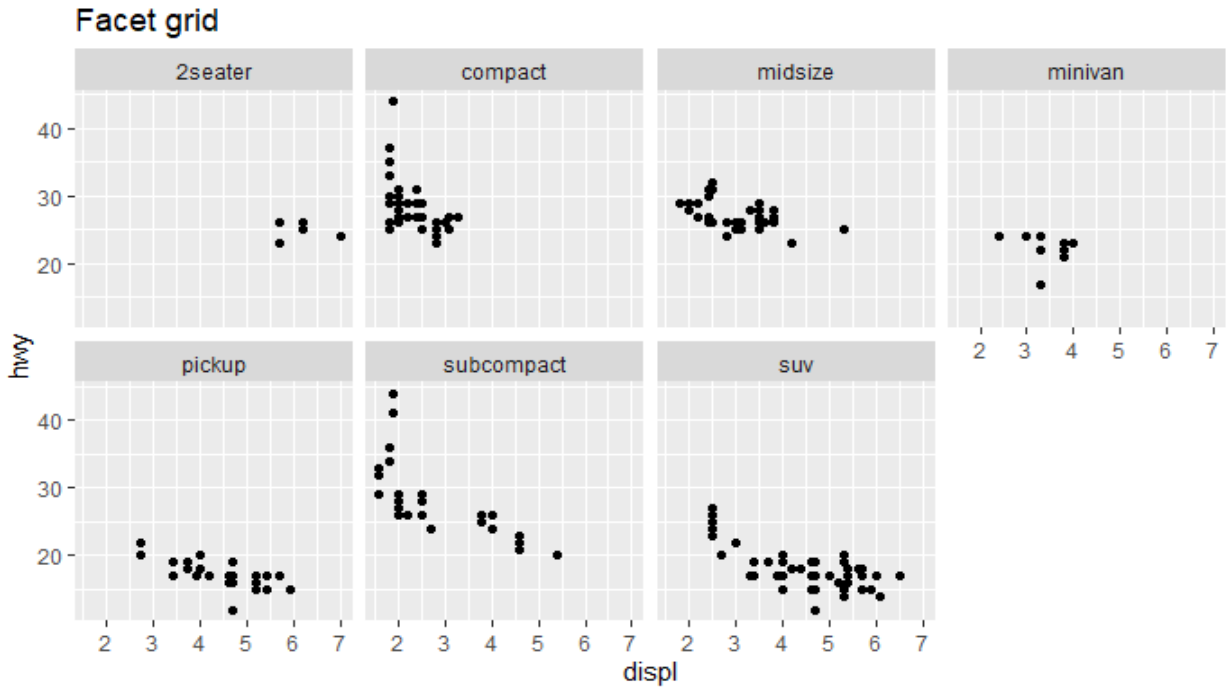
**ggplot(mpg, aes(x = cyl, y = hwy,color = year<2005)) + geom_point()**




**Insight:** The expression year<2005 acts as a logical variable which generates set of True and False features which are mapped onto the plot with 2 different colors to differentiate them easily.

**a) iii) Problem 3.5.1- Exercise 4**

```
ggplot(data = mpg) + geom_point(mapping = aes(x = displ, y = hwy)) +
  facet_wrap(~ class, nrow = 2)
```



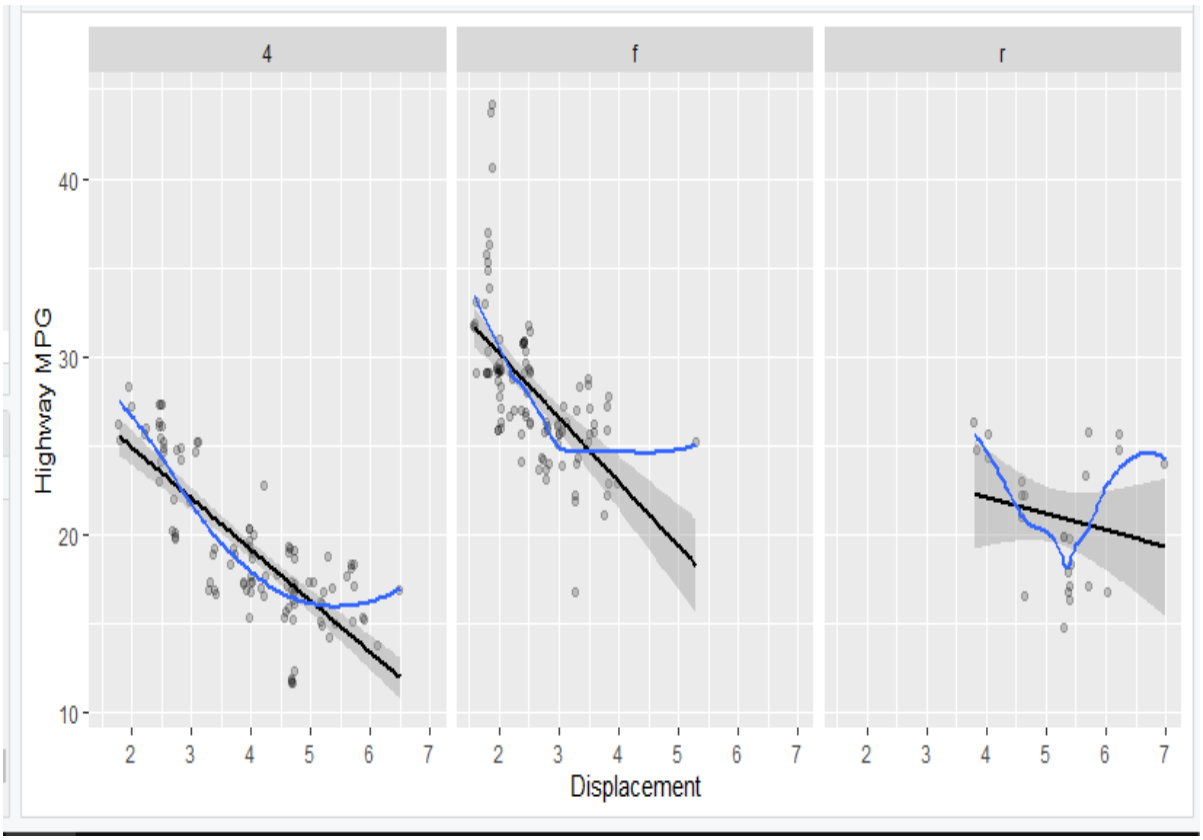

**Advantages vs Disadvantages of plotting class as "facet" instead of color are:**

1) We can easily plot the difference between all the features within the class between the given "displ" and "hwy" where as if we do as color= "class" is difficult since all the plotting is done in one graph there are instances where there is an overlap and differentiating between classes which are having similarities in color.
   For example: Differentiating a class with "Blue" and "Light blue" if in case they are overlapped would be hard.
2) Advantage of having everything in one graph between X & Y axis makes it easy in drawing conclusion in differences of features between the class varieties whereas in facet grid since they are on different grids it causes problem in drawing an insight on the overall problem.
3) If we have a large dataset it might be advisable to use a facet grid because it might get messy if the number of "classes" increases say like 15 there might be a lot of overlap though we might use jitter and alpha to make life easy.

**Problem 1)**

**b) Reproduce the plot in Figure 1**

```
ggplot(data=mpg) + geom_point(mapping = aes(x=displ,y=hwy),
 position = "jitter",alpha=1/5) +facet_grid(. ~ drv)+
 geom_smooth(mapping = aes(x=displ,y=hwy),method=lm,color="black")+
 geom_smooth(mapping = aes(x=displ,y=hwy),se=FALSE)+
 xlab("Displacement") +
 ylab("Highway MPG")
```



**Problem 2: Generating data and advanced density plots**

a) **I) Generating data**

```
a<-rnorm(500)
b<-rnorm(500)
c<-rnorm(500)
d<-rnorm(500)
df <- data.frame(a,b,c,d)
df <- setNames(df, c("a","b","c","d")) #dataframe is created with 4 columns
```

Few rows in the df are as follows:

```
           a          b          c          d
1 -0.4764211  0.3209298  0.8790321  0.9920121
2 -0.3214650  0.9851651 -0.2252255  1.1114469
3 -0.9854158  0.5559759  0.4551216 -1.7682325
4  0.2483124  0.5743480  1.1951763 -1.1577704
5  1.7470084 -1.6769331  1.8124707 -0.6681815
6  1.2654985  1.7487617 -0.3273073 -1.7986183
```

**II) Gather function**

```
df2<-df %>% gather(groupVar,value,a:d)
```

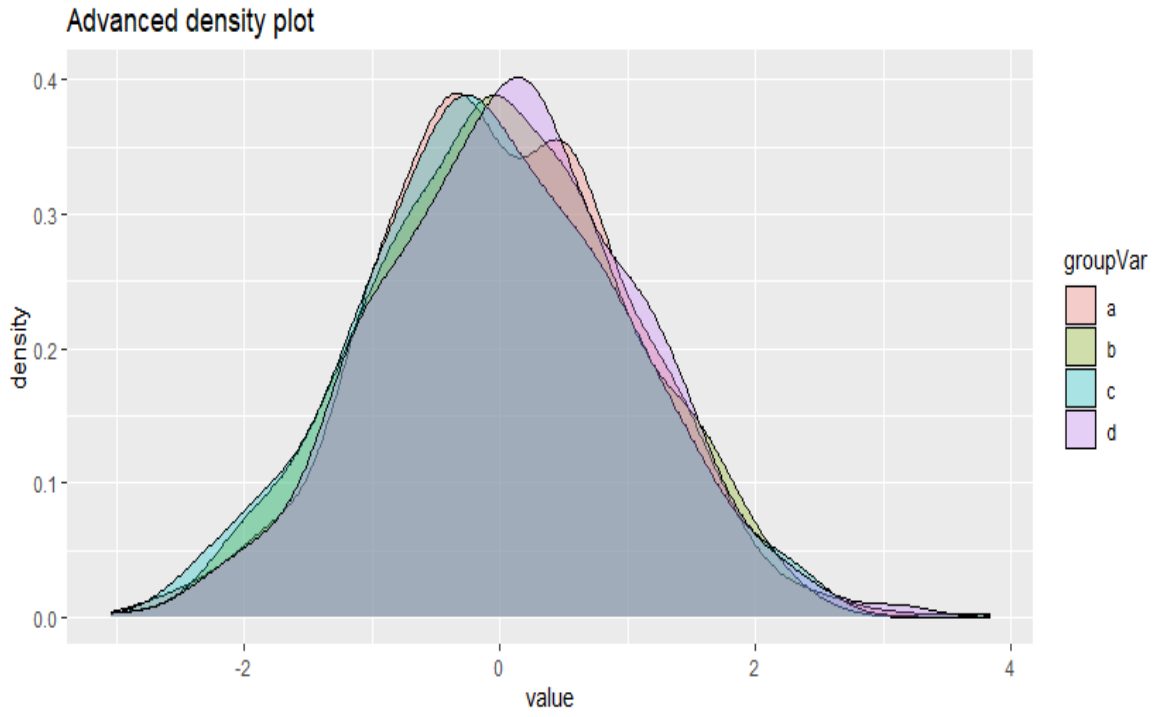
```
  groupVar      value
1        a -0.4764211
2        a -0.3214650
3        a -0.9854158
4        a  0.2483124
5        a  1.7470084
6        a  1.2654985
```

```
> nrow(df2)
[1] 2000
```

**Insight:** By using the gather function the original dataframe "df" with 500 rows is transformed to 2000 rows in "df2".

b) **Density plots**

```
density_plot <- ggplot(df2, aes(x=value,fill=groupVar)) +
  geom_density(alpha=0.3)+labs(title="Advanced density plot")
```




**Problem 3: House prices data**

```
housing_data<-read.csv('housingData.csv') #read the data
summary(housing_data) #to give overall idea of the features
```

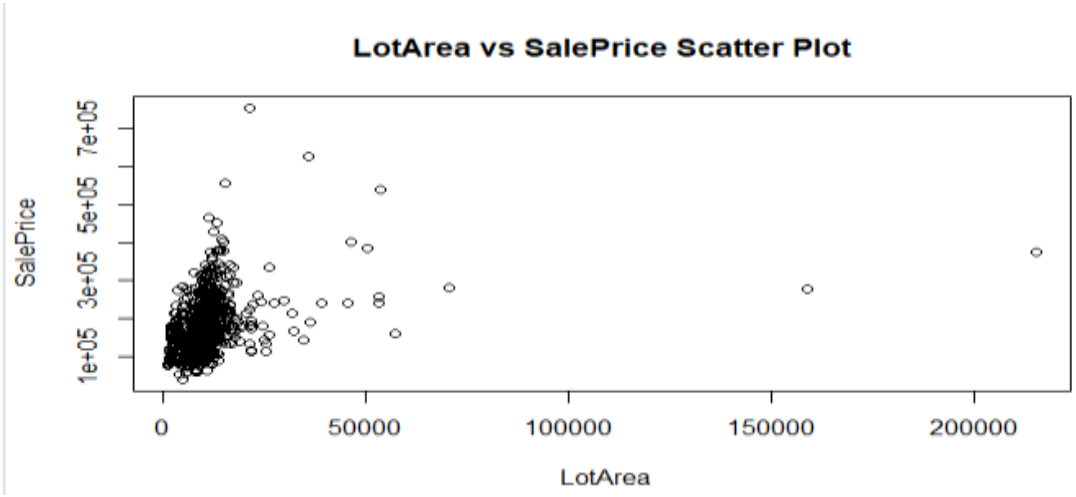
```
   EncPorchSF        PoolArea          PoolQc             Fence
 Min.   :  0.00   Min.   :  0.000   Length:1000        Length:1000
 1st Qu.:  0.00   1st Qu.:  0.000   Class :character   Class :character
 Median :  0.00   Median :  0.000   Mode  :character   Mode  :character
 Mean   : 40.64   Mean   :  1.224
 3rd Qu.:  0.00   3rd Qu.:  0.000
 Max.   :508.00   Max.   :648.000

 MiscFeature          MiscVal            MoSold           YrSold
 Length:1000        Min.   :    0.00   Min.   : 1.000   Min.   :2006
 Class :character   1st Qu.:    0.00   1st Qu.: 4.000   1st Qu.:2007
 Mode  :character   Median :    0.00   Median : 6.000   Median :2008
                    Mean   :   27.21   Mean   : 6.207   Mean   :2008
                    3rd Qu.:    0.00   3rd Qu.: 8.000   3rd Qu.:2009
                    Max.   :3500.00   Max.   :12.000   Max.   :2010

   SaleType           SalePrice
 Length:1000        Min.   : 39300
 Class :character   1st Qu.:130000
 Mode  :character   Median :160000
                    Mean   :174561
                    3rd Qu.:205000
                    Max.   :755000
```

**Insight:** We can easily draw conclusion such as Max and min of salesprice. Features like PoolQc, Fence, MiscFeature, Saletype can be seperated under categorical**.**

1) **Scatterplot**

   i) **Price vs lot area**

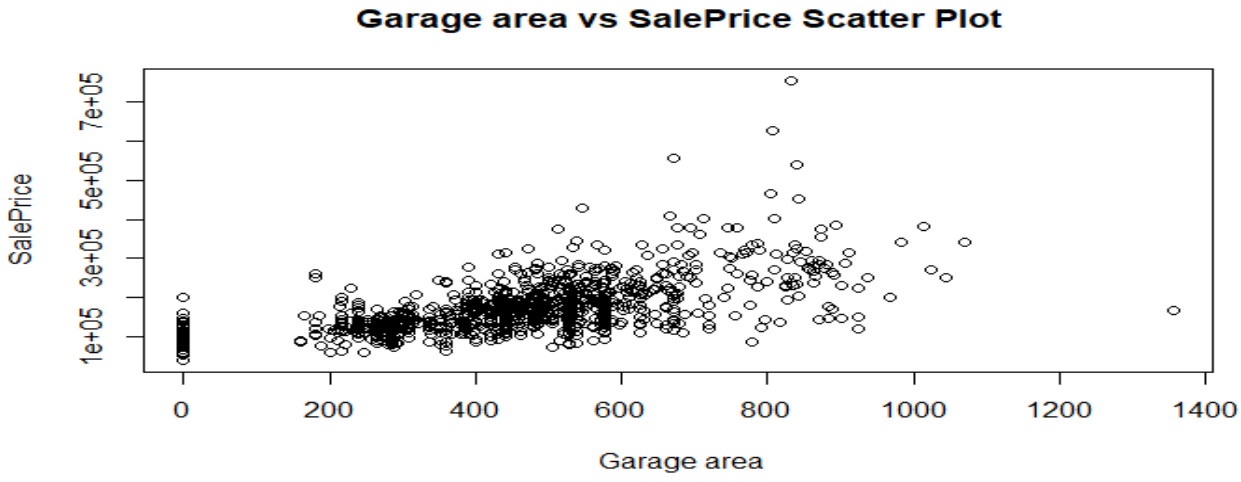
**plot(SalePrice ~ LotArea , data=housing_data,xlab=" LotArea", ylab="SalePrice",main="LotArea vs SalePrice Scatter Plot")**




**Insight:** Most of houses lie between below 50000 lot area and only 2 properties are above 150000.

   ii) **Price vs garage area**

**plot(SalePrice ~ GarageArea , data=housing_data,xlab=" Garage area", ylab="SalePrice", main="Garage area vs SalePrice Scatter Plot")**

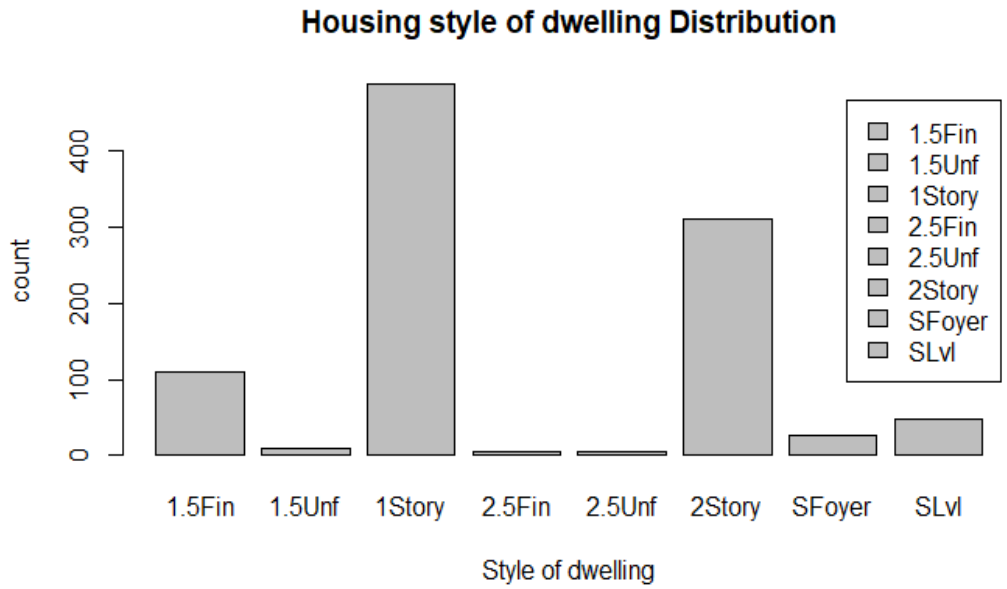



**Insight:** Even though the size of the garage area increases the sales price seems to be not affected by it.

2) **Bar plot:**

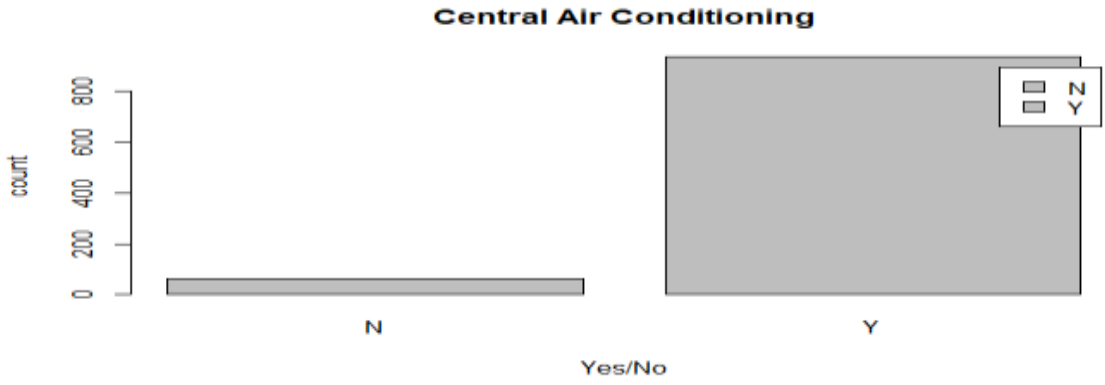
i) **Bar Plot with count of different dwelling styles-bar plot**

```
counts <- table(housing_data$HouseStyle)

barplot(counts, main="Housing style of dwelling Distribution",

      legend=rownames(counts),xlab = "Style of dwelling",ylab="count")
```



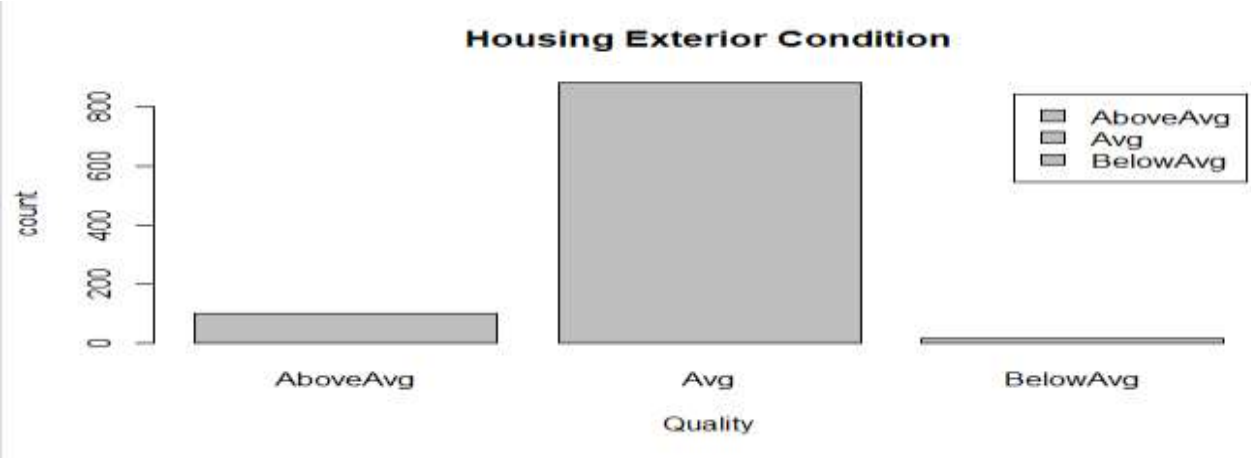

ii) **Bar Plot with count of Central A/C**

```
counts1 <- table(housing_data$CentralAir)

barplot(counts1, main="Central Air Conditioning",

      legend=rownames(counts1),xlab = "Yes/No",ylab="count")
```



iii) **Bar Plot for Exterior condition**

```
barplot(counts2, main="Housing Exterior Condition",
legend=rownames(counts2),xlab = "Quality",ylab="count")
```



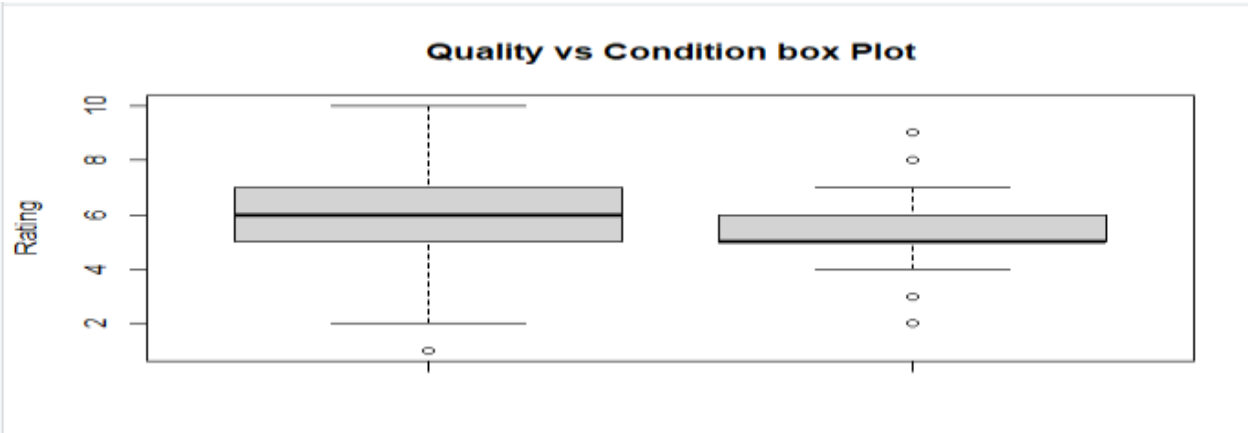

3) **Table** gives the count of number of properties that are built in that specific year vs Overall quality on the scale of 1 to 10.

```
table(housing_data$YearBuilt,housing_data$OverallQual)
```

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1875 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1880 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 0 | 0 | 0 |
| 1882 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1885 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1890 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 1892 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |

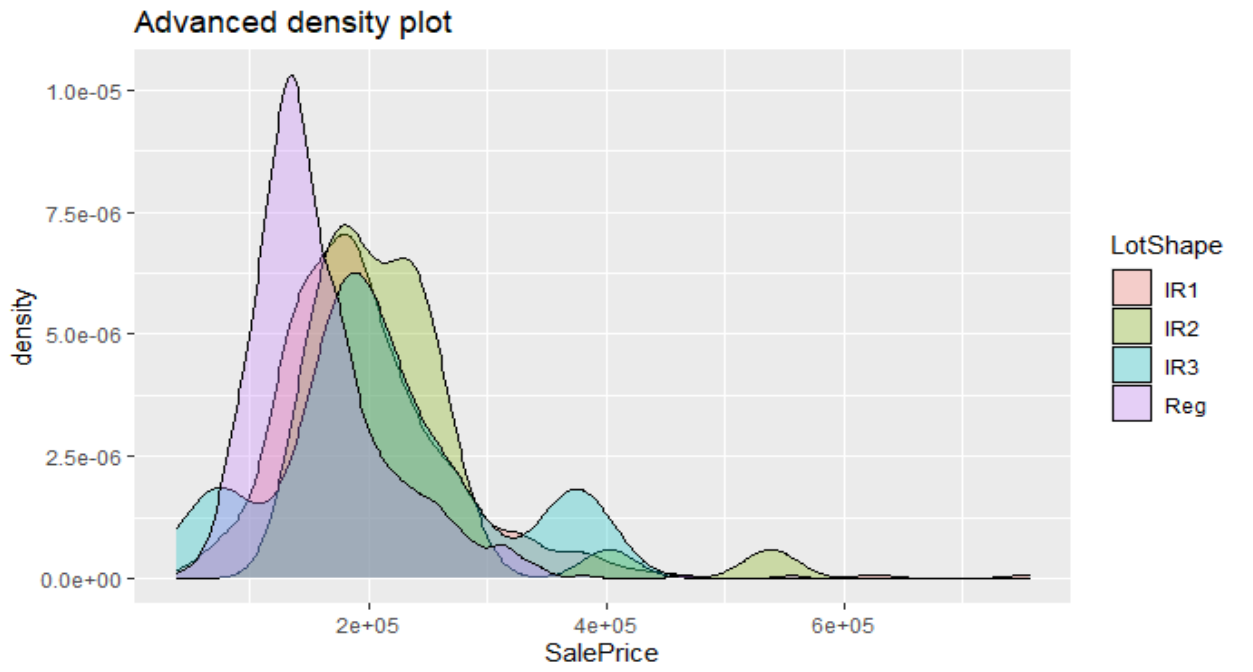
4) **Boxplot: To find mean, outliers, min and max at one go.**

```
boxplot (housing_data$OverallQual,housing_data$OverallCond, ylab="Rating",
        main="Quality vs Condition box Plot")
```



**Density plot:**

```
density_plot1 <- ggplot(housing_data, aes(x=SalePrice,fill=LotShape)) +
 geom_density(alpha=0.3)+labs(title="Advanced density plot")
```



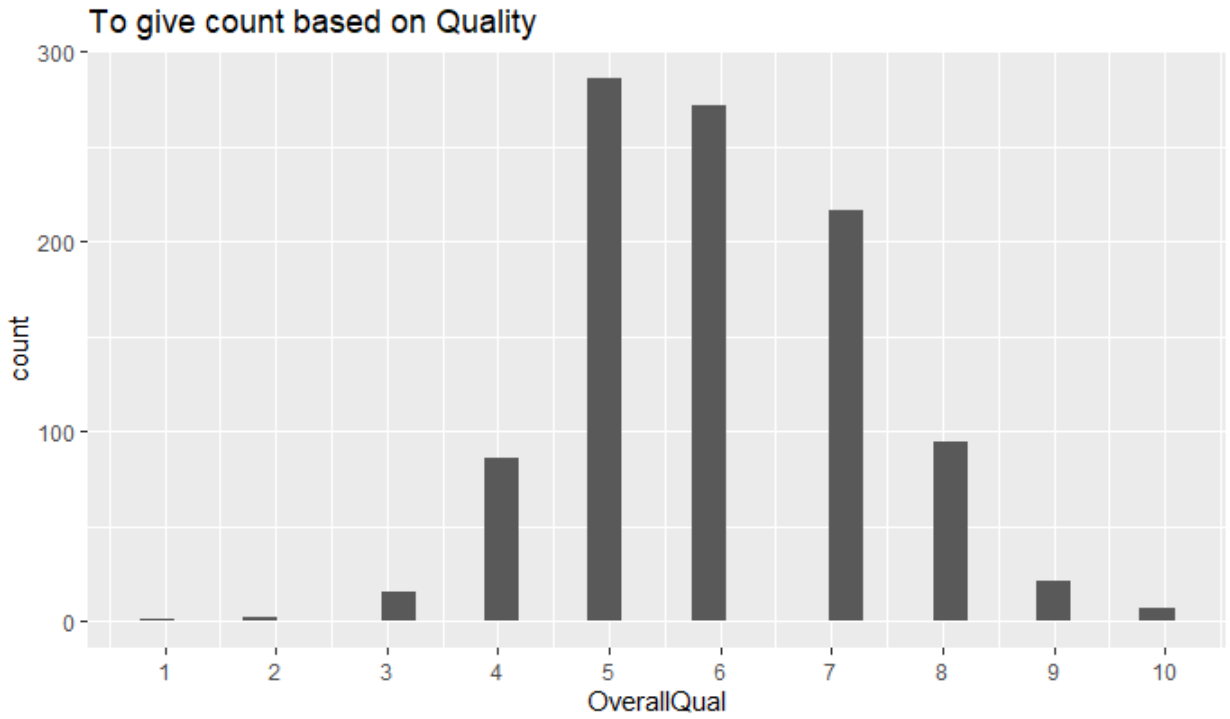

**Insight:** ***Regular lotshape seems have highest sales price***

**Stat_bins:**

```
ggplot(housing_data,aes(x=OverallQual))+stat_bin()+
scale_x_continuous(breaks=c(1,2,3,4,5,6,7,8,9,10)) + labs(title="To give count based on Quality")
```

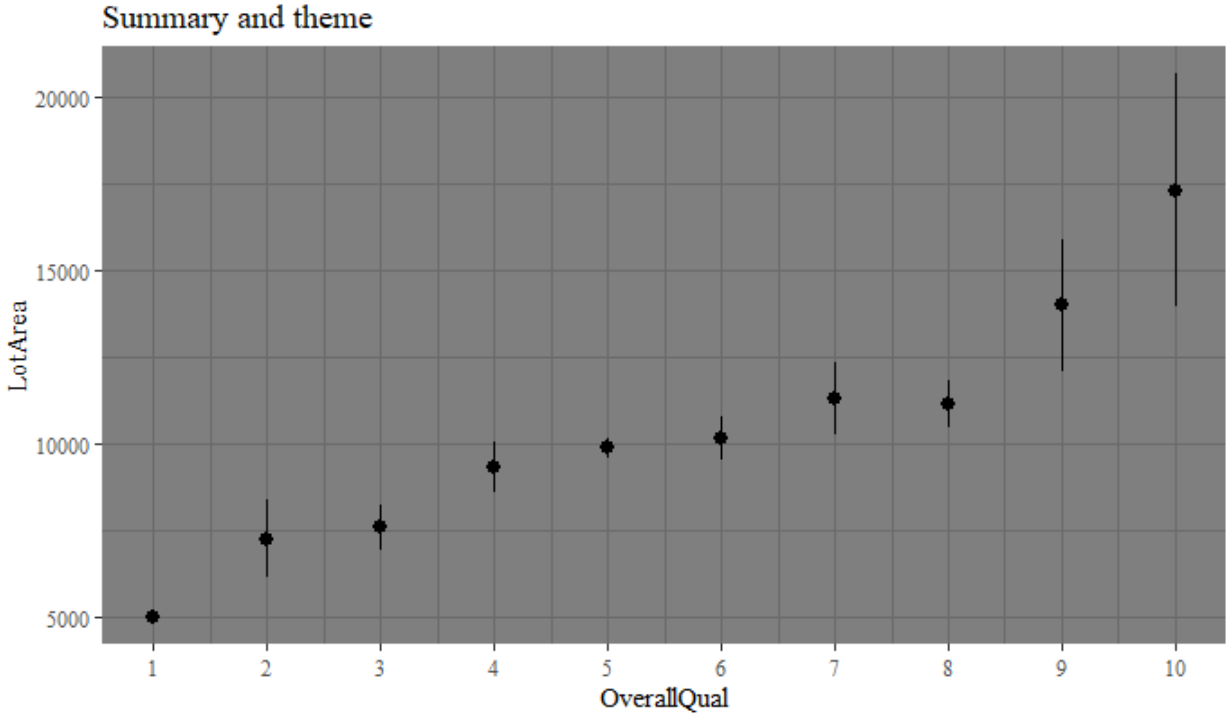
**Insight:** The houses with overall quality ratedas 5 seems to be high in number, followed by 6 whereas properties receiving 1and 2 are negligible.

**Screenshot of the results are in the next page. Continued,**



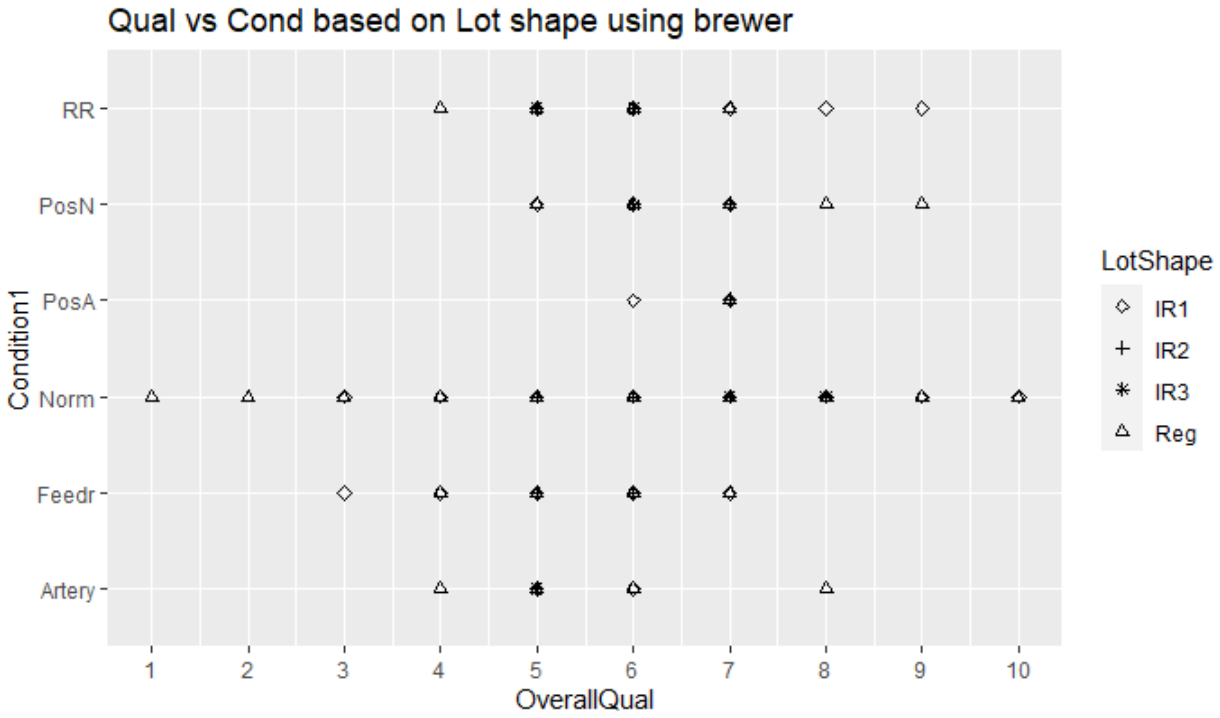

## Stat_summary

```
ggplot(housing_data,aes(x=OverallQual,y=LotArea))+stat_summary()+scale_x_continuous(breaks=c(1,2,3,4,5,6,7,8,9,10))+theme_dark(base_family = "serif")+labs(title="Summary and theme")
```

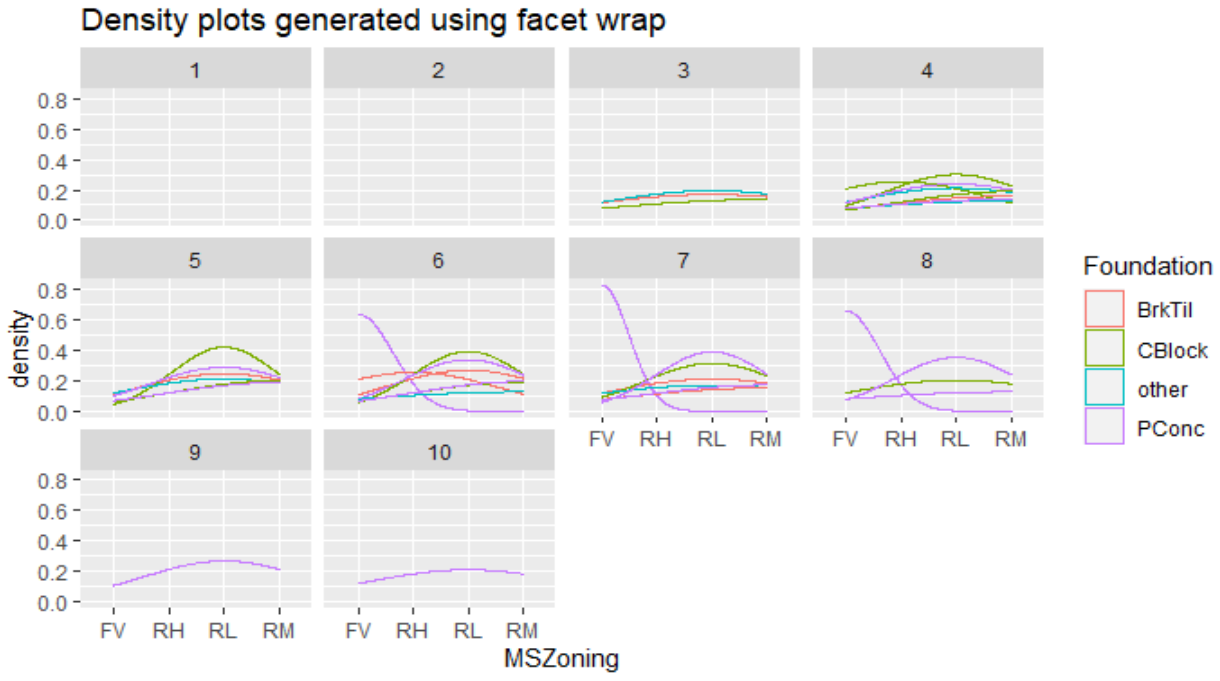


## Scale_shape_manual:

**ggplot(housing_data,aes(x=OverallQual,y=Condition1,shape=LotShape))+geom_point()+scale_shape_manual(values=c(5,3,8,2))+scale_color_brewer(type="qual")+scale_x_continuous(breaks=c(1,2,3,4,5,6,7,8,9,10))+labs(title="Qual vs Cond based on Lot shape using brewer")**




## Facet_wrap()



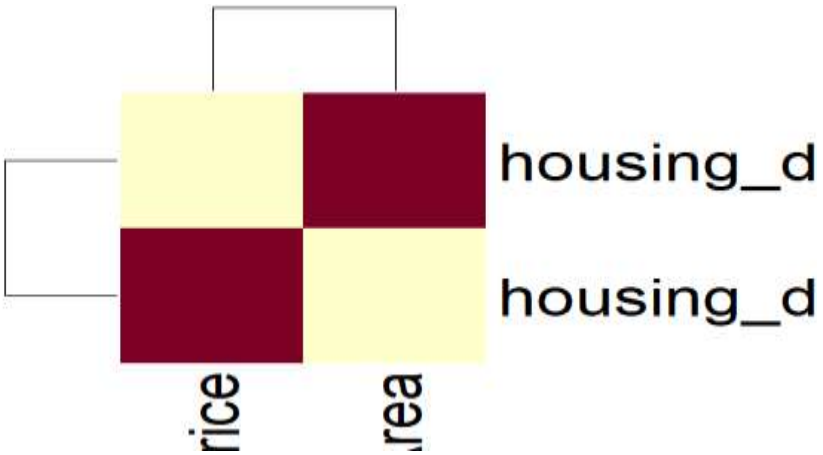
## Heatmap:

**x<-data.frame(housing_data$SalePrice,housing_data$LotArea)**

**cor(x) #value close to "1" shows high correlation**

**heatmap(cor(x)) #visual representation of correlation**

```
                      housing_data.SalePrice housing_data.LotArea
housing_data.SalePrice               1.000000             0.314726
housing_data.LotArea                 0.314726             1.000000
```




## Linear Model:

**linModel<-lm(data=housing_data,OverallQual~OverallCond)**

**summary(linModel)**

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  6.67798    0.21281  31.380  < 2e-16 ***
OverallCond -0.12398    0.03703  -3.348 0.000844 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.303 on 998 degrees of freedom
Multiple R-squared:  0.01111,   Adjusted R-squared:  0.01012
F-statistic: 11.21 on 1 and 998 DF,  p-value: 0.0008444
```

**Insight:** Because of negligable P value is less than any alpha value which we ideally choose therefore reject null.

**Problem 4: Missing Data**

**data("freetrade") #load data**

**summary(freetrade) #to check the summary**

```
      year        country             tariff          polity       
 Min.   :1981   Length:171         Min.   :  7.10   Min.   :-8.000  
 1st Qu.:1985   Class :character   1st Qu.: 16.30   1st Qu.:-2.000  
 Median :1990   Mode  :character   Median : 25.20   Median : 5.000  
 Mean   :1990                      Mean   : 31.65   Mean   : 2.905  
 3rd Qu.:1995                      3rd Qu.: 40.80   3rd Qu.: 8.000  
 Max.   :1999                      Max.   :100.00   Max.   : 9.000  
                                   NA's   :58       NA's   :2       
      pop                 gdp.pc           intresmi         signed       
 Min.   : 14105080   Min.   :  149.5   Min.   :0.9036   Min.   :0.0000  
 1st Qu.: 19676715   1st Qu.:  420.1   1st Qu.:2.2231   1st Qu.:0.0000  
 Median : 52799040   Median :  814.3   Median :3.1815   Median :0.0000  
 Mean   :149904501   Mean   : 1867.3   Mean   :3.3752   Mean   :0.1548  
 3rd Qu.:120888400   3rd Qu.: 2462.9   3rd Qu.:4.4063   3rd Qu.:0.0000  
 Max.   :997515200   Max.   :12086.2   Max.   :7.9346   Max.   :1.0000  
                                       NA's   :13       NA's   :3       
     fiveop           usheg        
 Min.   :12.30   Min.   :0.2558  
 1st Qu.:12.50   1st Qu.:0.2623  
 Median :12.60   Median :0.2756  
 Mean   :12.74   Mean   :0.2764  
 3rd Qu.:13.20   3rd Qu.:0.2887  
 Max.   :13.20   Max.   :0.3083  
 NA's   :18      
> 
```

**Insight:** From summary it is clear that tariff has most NA with “58” instances. If the missing values in column is more than 5% of observations it is ideal to drop the feature.

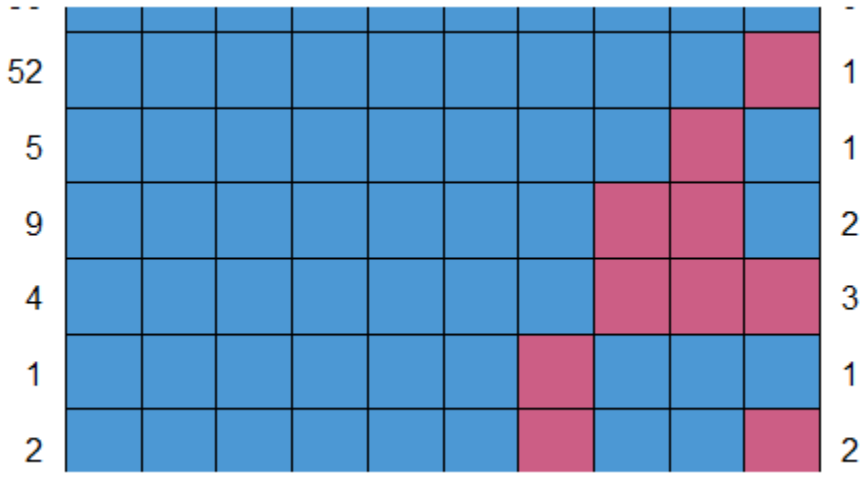
### i) Using MICE library

```
pMiss <- function(x){sum(is.na(x))/length(x)*100} #calculate missing variable math function
apply(freetrade,1,pMiss)
apply(freetrade,2,pMiss) #tariff,intresmi, fiveop features have more than 5% of missing values
md.pattern(freetrade) #pattern represntation
```

**Insight:** 96 rows have no missing features,similarly 52 rows have only tariff as missing

```
    year   country    tariff    polity       pop    gdp.pc  intresmi
0.000000  0.000000 33.918129  1.169591  0.000000  0.000000  7.602339
  signed    fiveop     usheg
1.754386 10.526316  0.000000
```

**Percentage of missing**



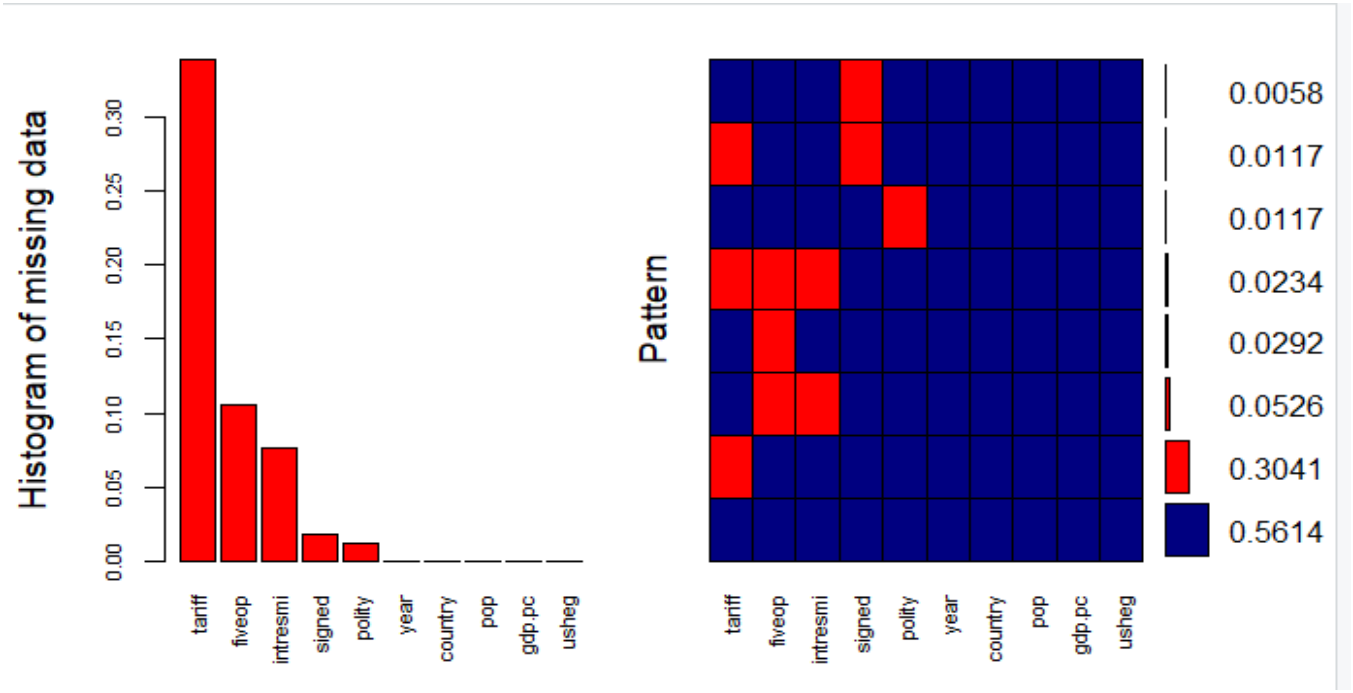

**Graphical representation**

```
    year country pop gdp.pc usheg polity signed intresmi fiveop tariff
96     1       1   1      1     1      1      1        1      1      1  0
52     1       1   1      1     1      1      1        1      1      0  1
5      1       1   1      1     1      1      1        1      0      1  1
9      1       1   1      1     1      1      1        0      0      1  2
4      1       1   1      1     1      1      1        0      0      0  3
1      1       1   1      1     1      1      0        1      1      1  1
2      1       1   1      1     1      1      0        1      1      0  2
2      1       1   1      1     1      0      1        1      1      1  1
       0       0   0      0     0      2      3       13     18     58 94
```

**Tabular :** Gives the same insight i.e, 96 instances have no missing values. Similarly 2 instances have polity as a missing value.

**ii) Using VIM library**

```
aggr_plot <- aggr(freetrade, col=c('navyblue','red'), numbers=TRUE, sortVars=TRUE,
labels=names(data),  cex.axis=.7, gap=3, ylab=c("Histogram of missing
data","Pattern"))
```




```
Variables sorted by number of missings:
Variable      Count
  tariff 0.33918129
  fiveop 0.10526316
intresmi 0.07602339
  signed 0.01754386
  polity 0.01169591
    year 0.00000000
 country 0.00000000
     pop 0.00000000
  gdp.pc 0.00000000
   usheg 0.00000000
>
```

**Insight:** Almost 56% of data is not missing any value whereas somewhere around 30% miss only tariff feature and so on we can draw details from the pattern and histogram above.

## Problem 5: Extra Credit

## Using ANOVA:

**real_data <- aov(is.na(freetrade$tariff) ~ freetrade$country,**

**freetrade) #is.na() returns boolean value**

```
Terms:
                freetrade$country Residuals
Sum of Squares            5.16959  33.15789
Deg. of Freedom                 8       162

Residual standard error: 0.4524139
Estimated effects may be unbalanced
> |
```

**summary(real_data) # Summary of the analysis**

```
                   Df Sum Sq Mean Sq F value  Pr(>F)
freetrade$country   8   5.17  0.6462   3.157 0.00238 **
Residuals         162  33.16  0.2047
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> |
```

**Insight**: With the orginal data when ANOVA test is run we get a P-value 0.00238 which way less than any alpha we choose. Thereore we reject null and conclude there is tariff and country are dependent.

## Remove Nepal

**nepalr<-freetrade[!freetrade$country=="Nepal",]**

**without_Nepal<-aov(is.na(nepalr$tariff) ~ nepalr$country, nepalr)**

**summary(without_Nepal) # Summary of the analysis**

```
                 Df Sum Sq Mean Sq F value Pr(>F)
nepalr$country    7  3.342  0.4774   2.392 0.0241 *
Residuals       144 28.737  0.1996
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> |
```

**Insight**: When we remove Nepal from the data, still we observe that the P value is less than alpha.Hence, we can reject null and conclude there is dependency between tariff and country.

**Remove Philippines**

```
Philr<-freetrade[!freetrade$country=="Philippines",]

without_Philippines<-aov(is.na(Philr$tariff) ~ Philr$country, Philr)

summary(without_Philippines) # Summary of the analysis
```

```
> summary(without_Philippines)
               Df Sum Sq Mean Sq F value Pr(>F)
Philr$country   7   2.71  0.3872   1.682  0.118
Residuals     144  33.16  0.2303
>
```

**Insight:** When we remove “Phillippines” from the country fetaure we see a significant increase in P value which is greater than alpha of “0.05” which we ideally choose. So, we don’t reject null and conclude tariff and country are independent.