homework2

Oyindamola Obisesan

9/10/2020

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0

## -- Conflicts ------------------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library ("Amelia")
```

```
## Loading required package: Rcpp

## ##
## ## Amelia II: Multiple Imputation
## ## (Version 1.7.6, built: 2019-11-24)
## ## Copyright (C) 2005-2020 James Honaker, Gary King and Matthew Blackwell
## ## Refer to http://gking.harvard.edu/amelia/ for more information
## ##
```

```
library(mice)
```

```
##
## Attaching package: 'mice'

## The following objects are masked from 'package:base':
##
##     cbind, rbind
```

```
library(VIM)
```

```
## Loading required package: colorspace
```

```
## Loading required package: grid
```

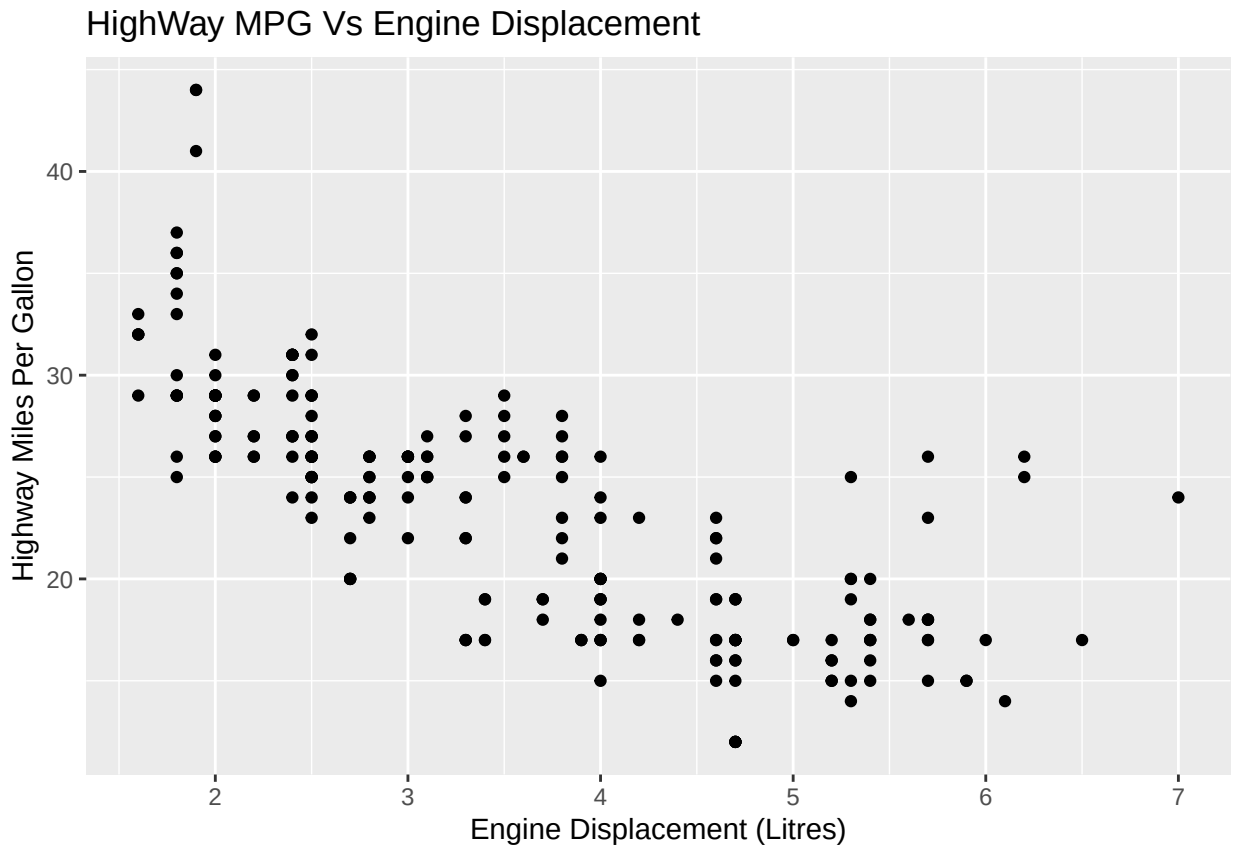
```
## VIM is ready to use.
```

```
## Suggestions and bug-reports can be submitted at: https://github.com/statistikat/VIM/issues
```

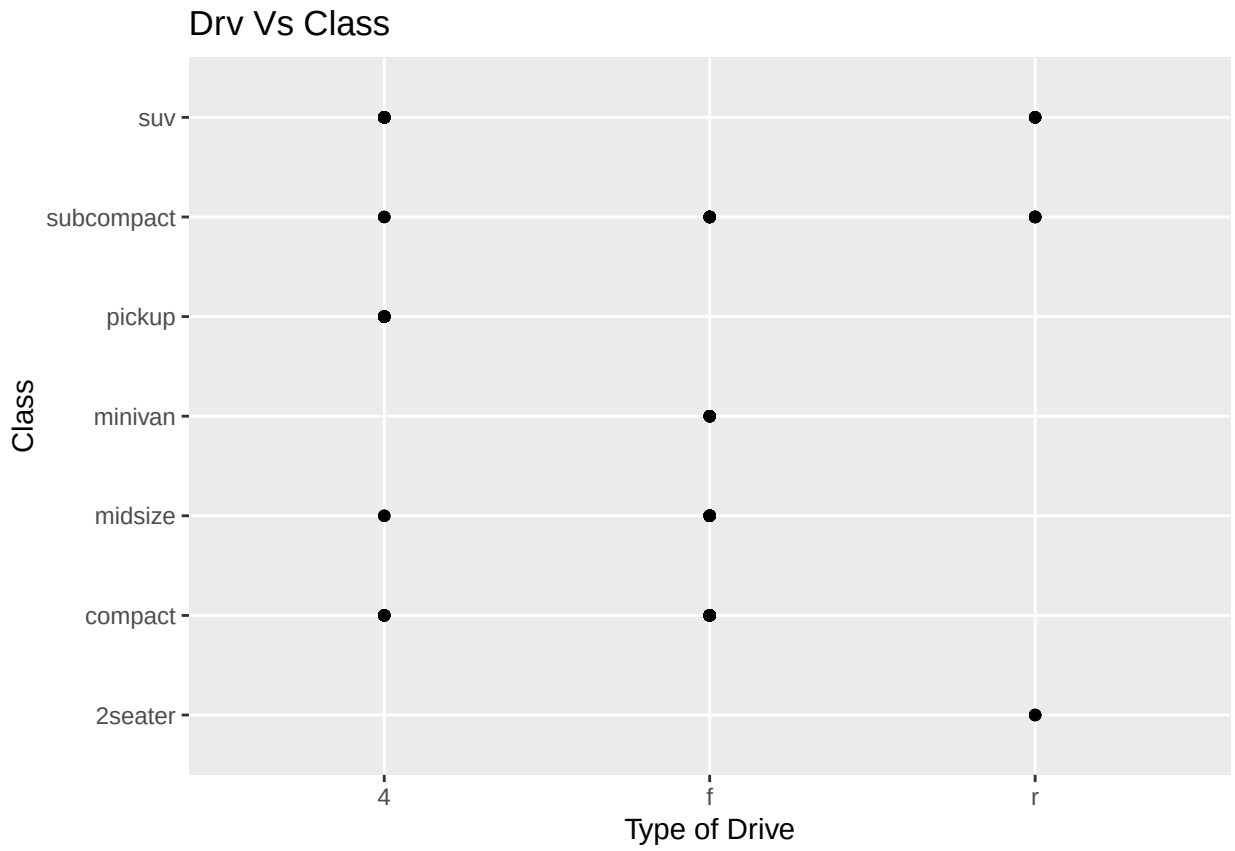
```
##
## Attaching package: 'VIM'
```

```
## The following object is masked from 'package:datasets':
##
##     sleep
```

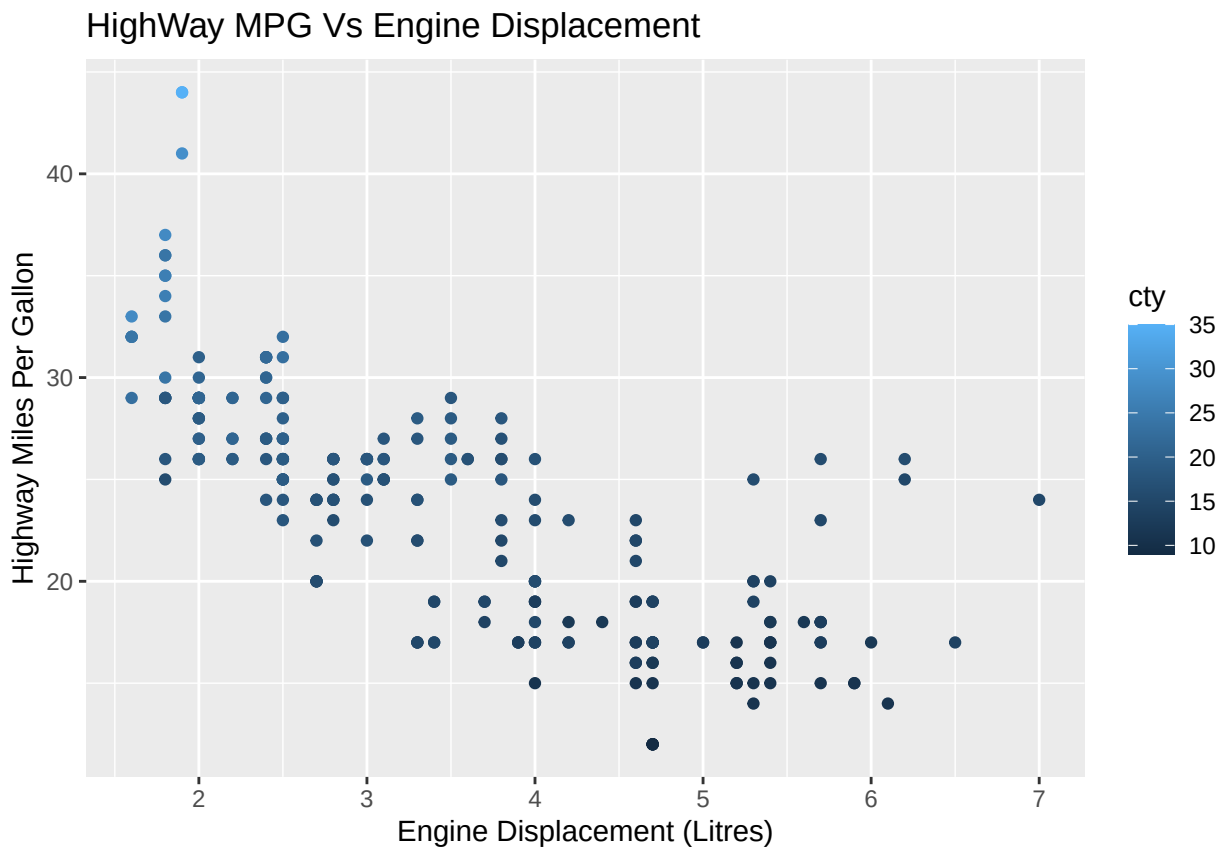
```
#a
ggplot(data= mpg, aes(x=displ, y=hwy)) +geom_point() + labs (x = "Engine Displacement (Litres)", y = "Hi
  ggtitle("HighWay MPG Vs Engine Displacement")
```



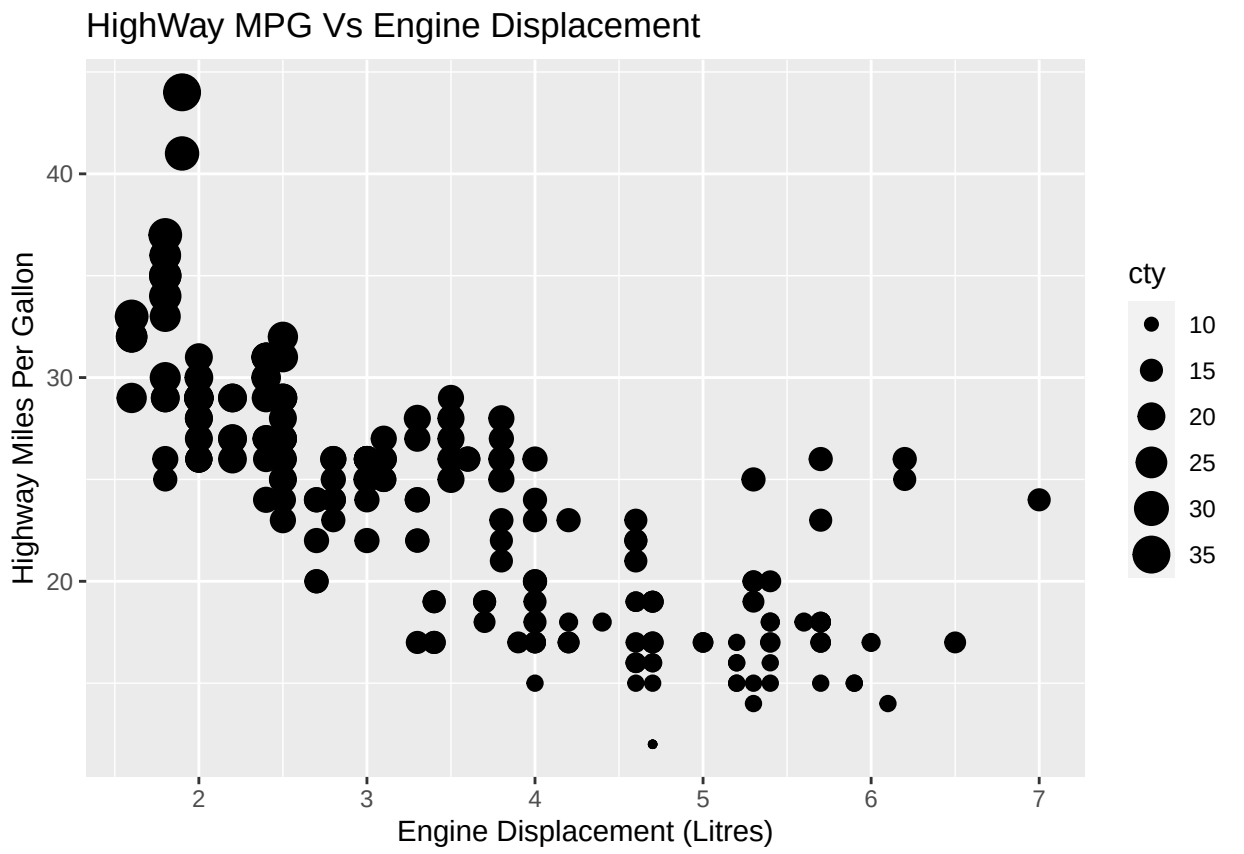
```r
#these shows the relationship between the "hwy" and "displ"

#ii
ggplot(data=mpg, aes(x= drv, y = class)) +geom_point() + labs (x = "Type of Drive", y = "Class")+ggtitl
```



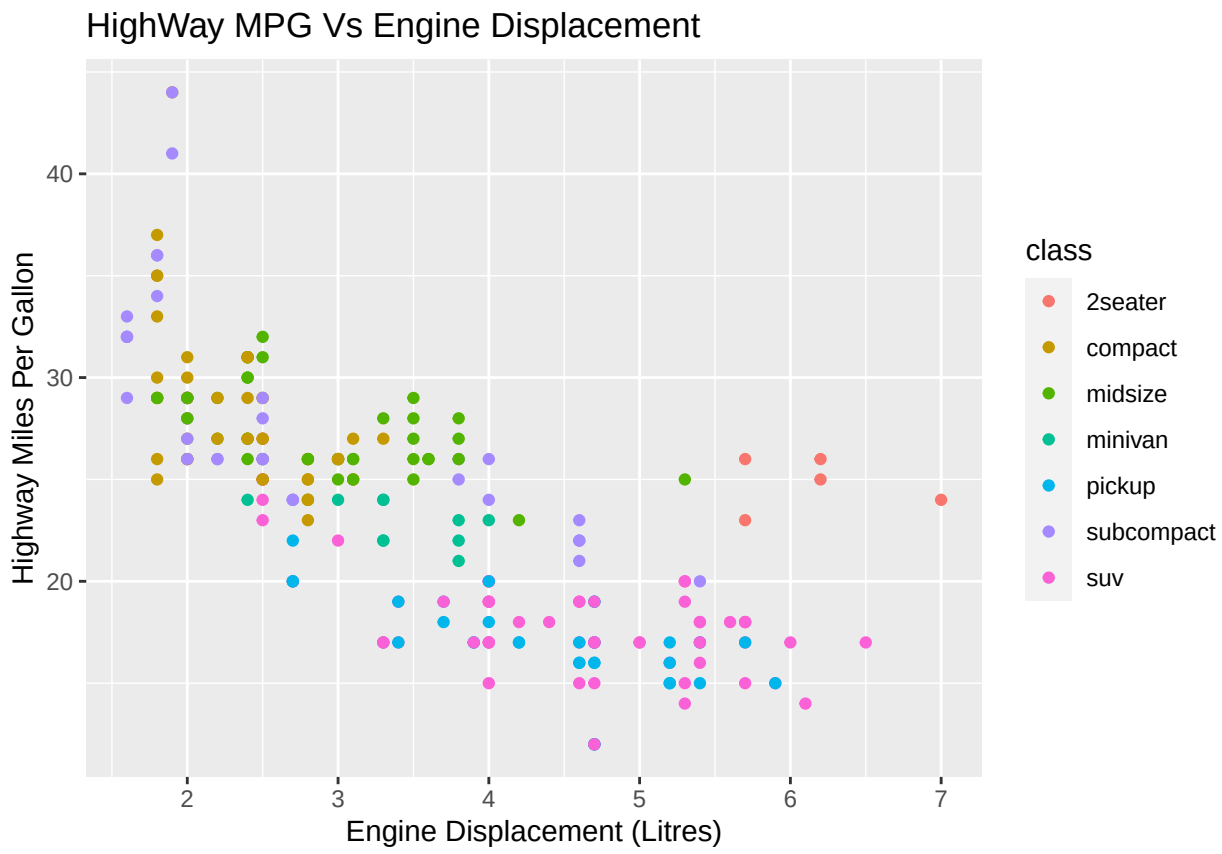

```r
#the two variables are categorical variables and when they are plotted against each other the plot is n

#b
ggplot(data = mpg) + geom_point(mapping = aes(x = displ, y = hwy, color = cty)) +
  labs (x = "Engine Displacement (Litres)", y = "Highway Miles Per Gallon")+ggtitle("HighWay MPG Vs Engi
```




```
ggplot(data = mpg) + geom_point(mapping = aes(x = displ, y = hwy, size = cty)) +
  labs (x = "Engine Displacement (Litres)", y = "Highway Miles Per Gallon")+ggtitle("HighWay MPG Vs Engi
```

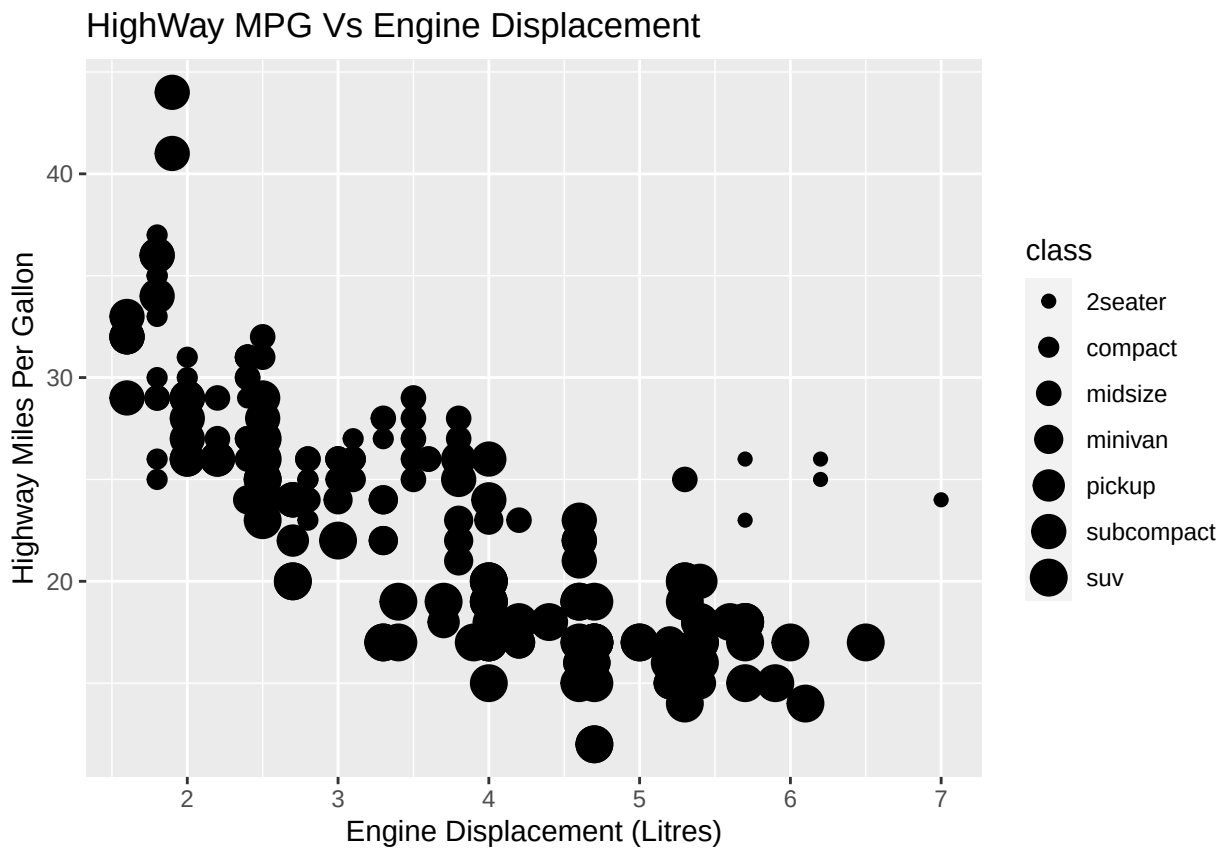



```
#ggplot(data = mpg) + geom_point(mapping = aes(x = displ, y = hwy, shape = cty))+ labs (x = "Engine Dis


#a continuous variable cannot be mapped to shape

ggplot(data = mpg) + geom_point(mapping = aes(x = displ, y = hwy, color = class)) +
  labs (x = "Engine Displacement (Litres)", y = "Highway Miles Per Gallon")+ggtitle("HighWay MPG Vs Engi
```




```
ggplot(data = mpg) + geom_point(mapping = aes(x = displ, y = hwy, size = class))+
  labs (x = "Engine Displacement (Litres)", y = "Highway Miles Per Gallon")+ggtitle("HighWay MPG Vs Engi
```

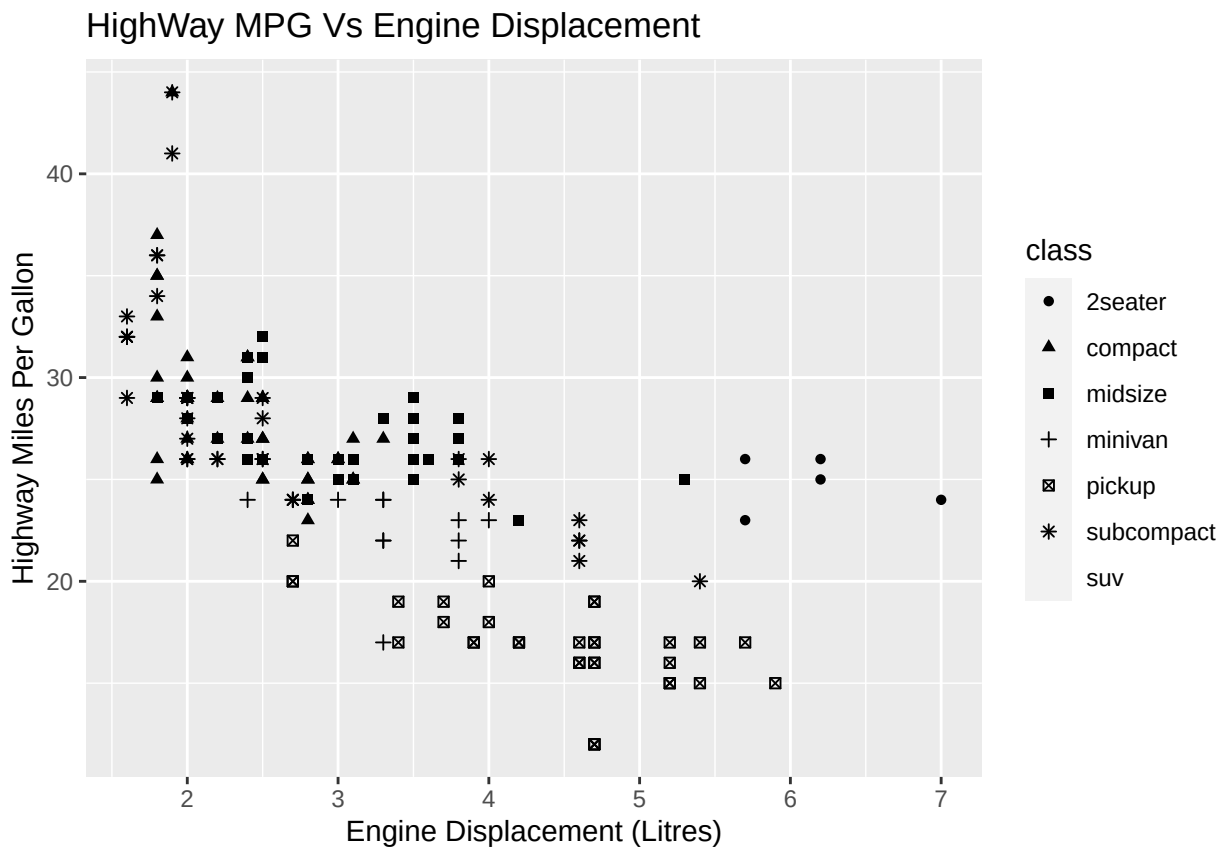
```
## Warning: Using size for a discrete variable is not advised.
```



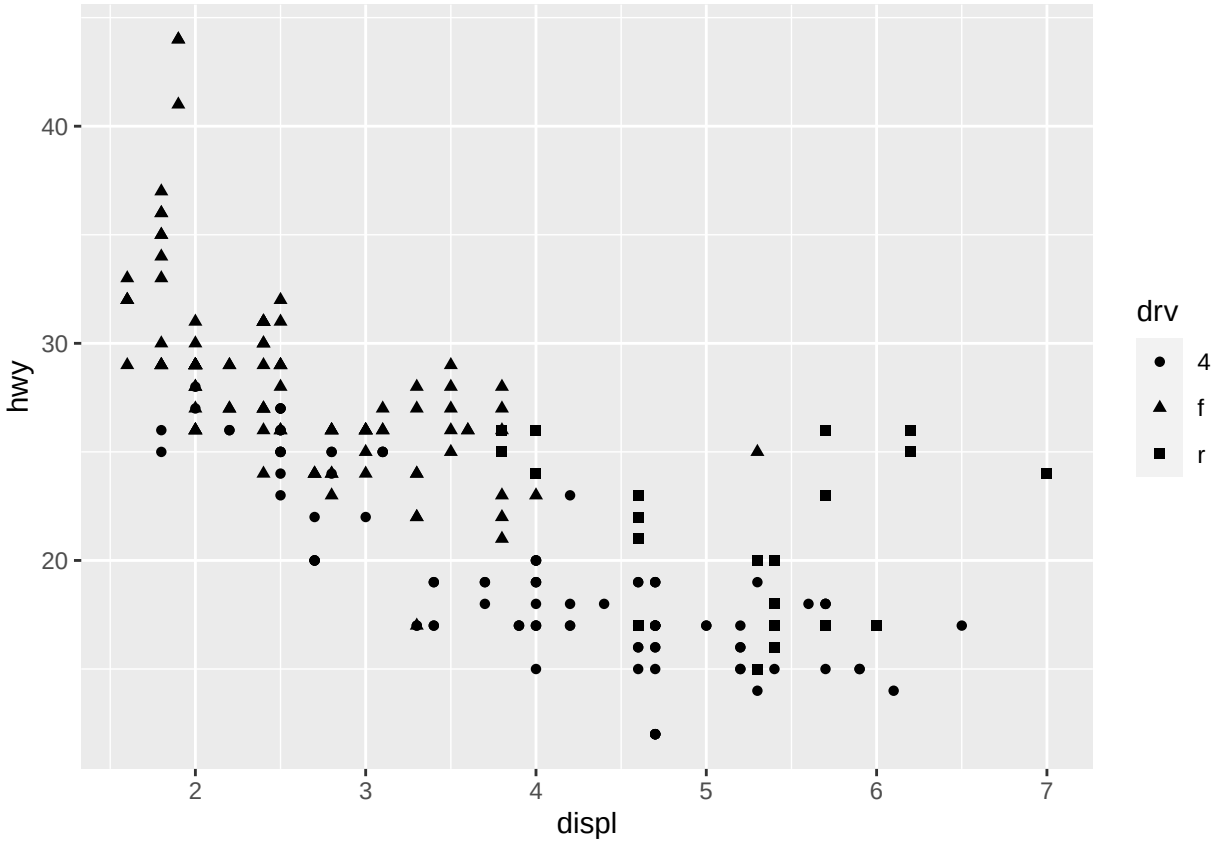

```
ggplot(data = mpg) + geom_point(mapping = aes(x = displ, y = hwy, shape = class))+
  labs (x = "Engine Displacement (Litres)", y = "Highway Miles Per Gallon")+ggtitle("HighWay MPG Vs Engi
```

```
## Warning: The shape palette can deal with a maximum of 6 discrete values because
## more than 6 becomes difficult to discriminate; you have 7. Consider
## specifying shapes manually if you must have them.
```

```
## Warning: Removed 62 rows containing missing values (geom_point).
```

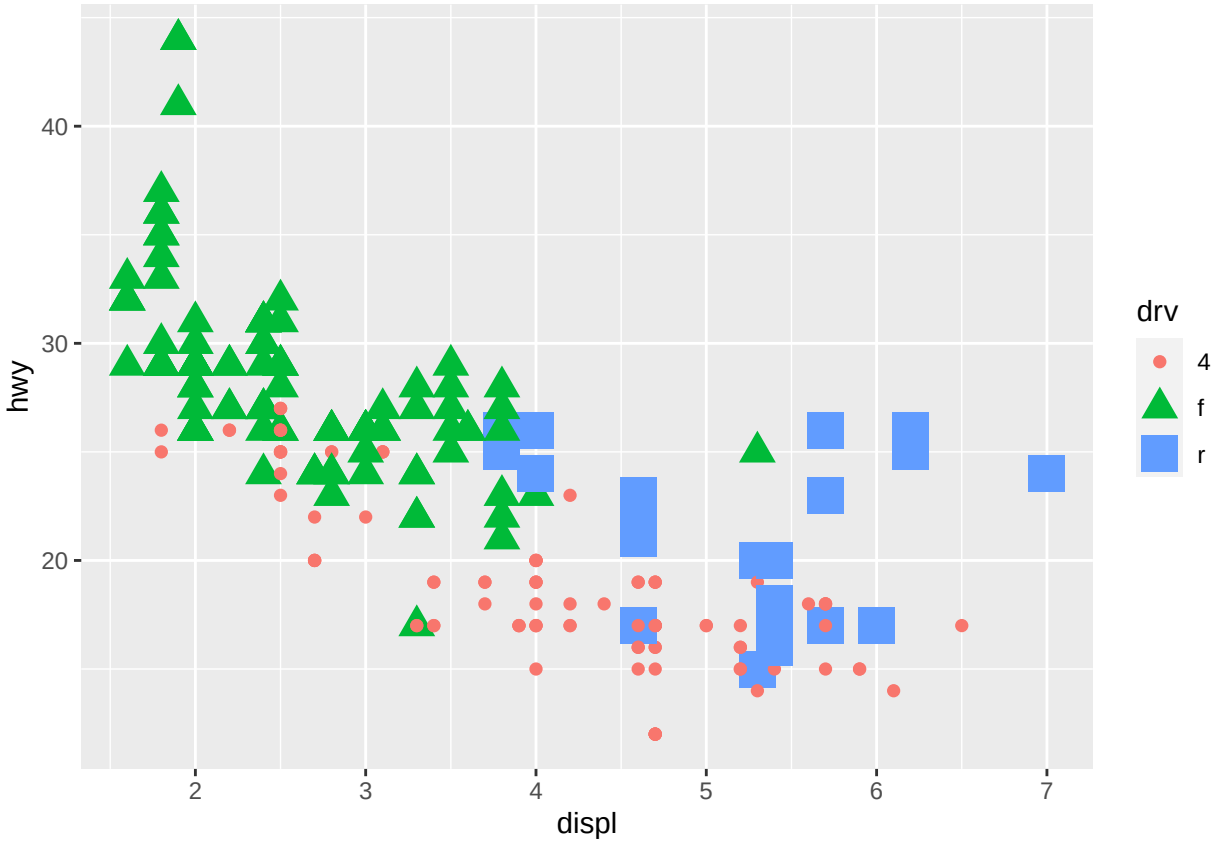



```
ggplot(data = mpg) + geom_point(mapping = aes(x = displ, y = hwy, shape = drv))
```

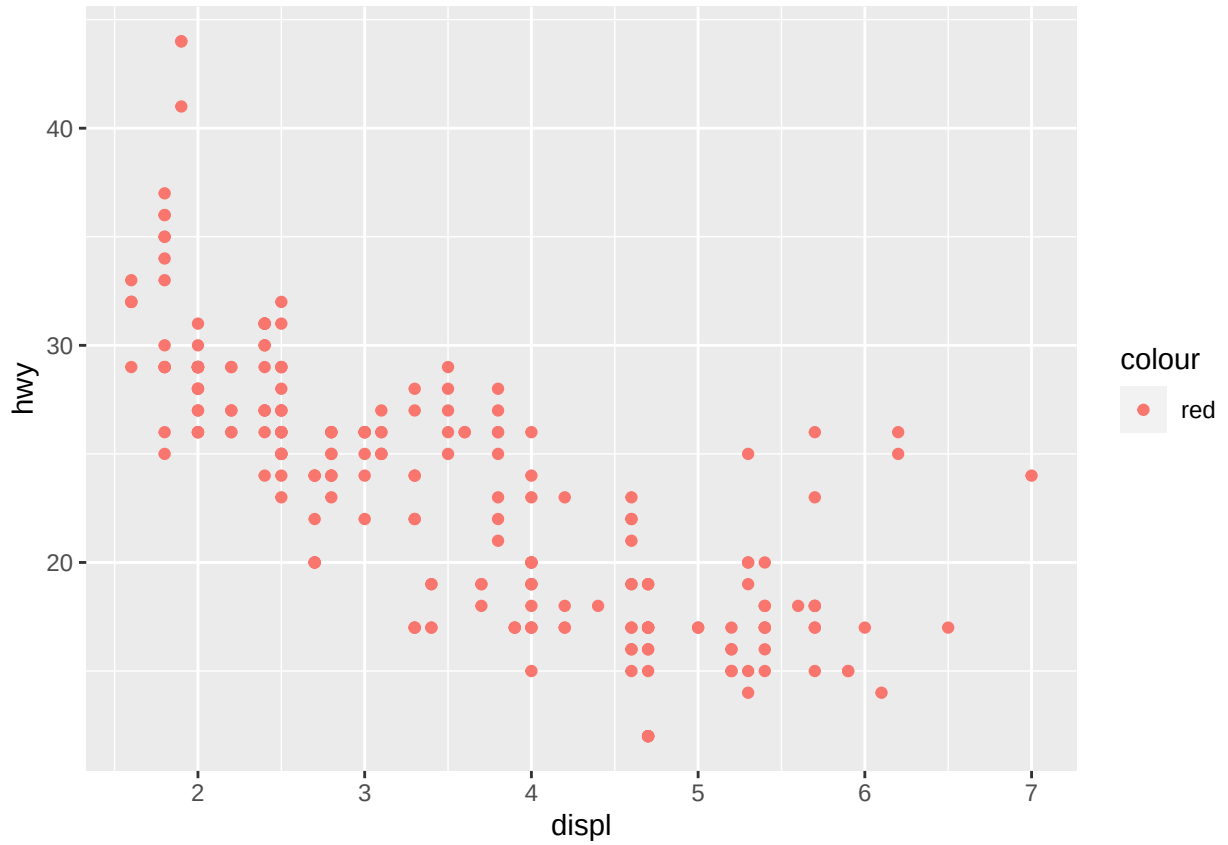



```
ggplot(data = mpg) + geom_point(mapping = aes(x = displ, y = hwy, color = drv, shape = drv, size = drv))
```

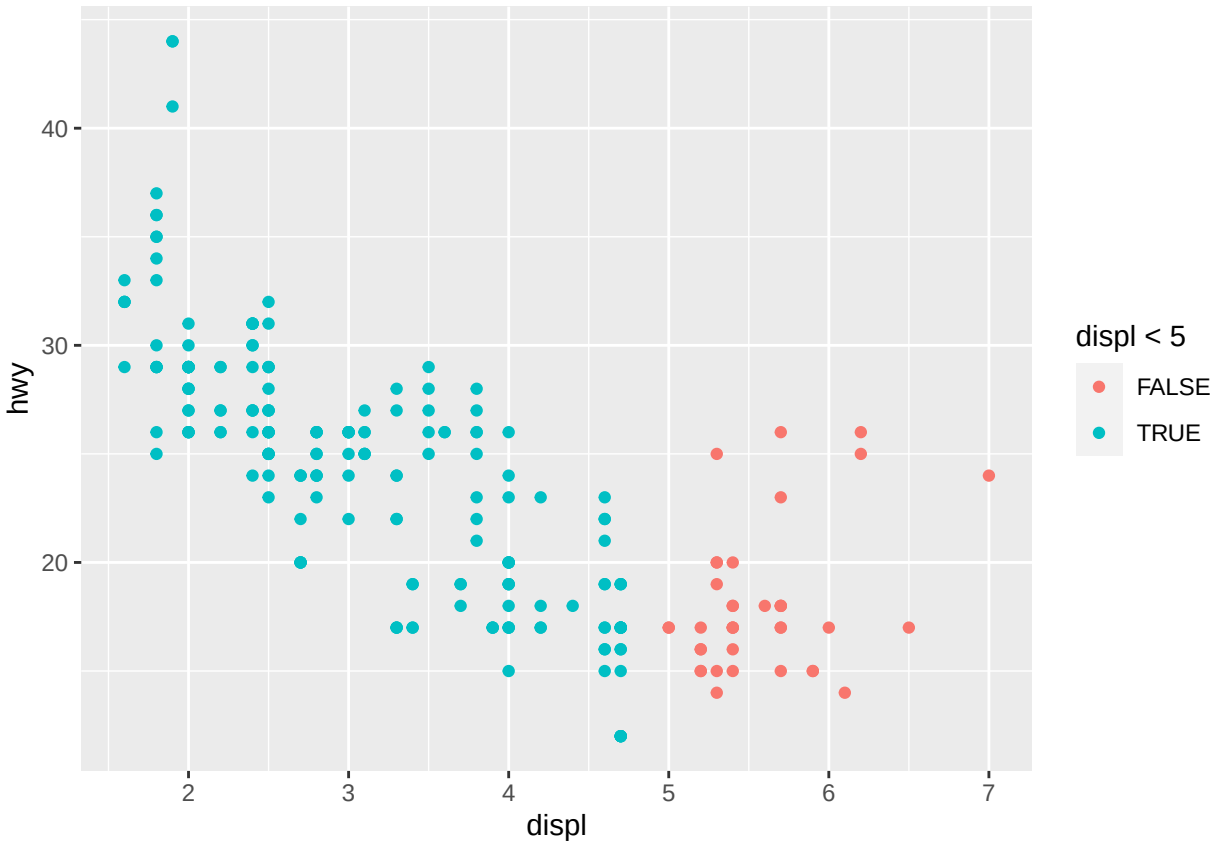
```
## Warning: Using size for a discrete variable is not advised.
```




```
#ggplot(data = mpg) + geom_point(mapping = aes(x = displ, y = hwy, stroke = class))
#ggplot(data = mpg, aes(x=displ, y = hwy)) + geom_point(aes(stroke = 2))

#aiii
ggplot(data = mpg, aes(x=displ, y = hwy)) + geom_point(aes(color = "red"))
```




```
#changing the aes color to red changes the color of all the data points to red.
ggplot(data = mpg, aes(x=displ, y = hwy)) + geom_point(aes(color = displ<5))
```



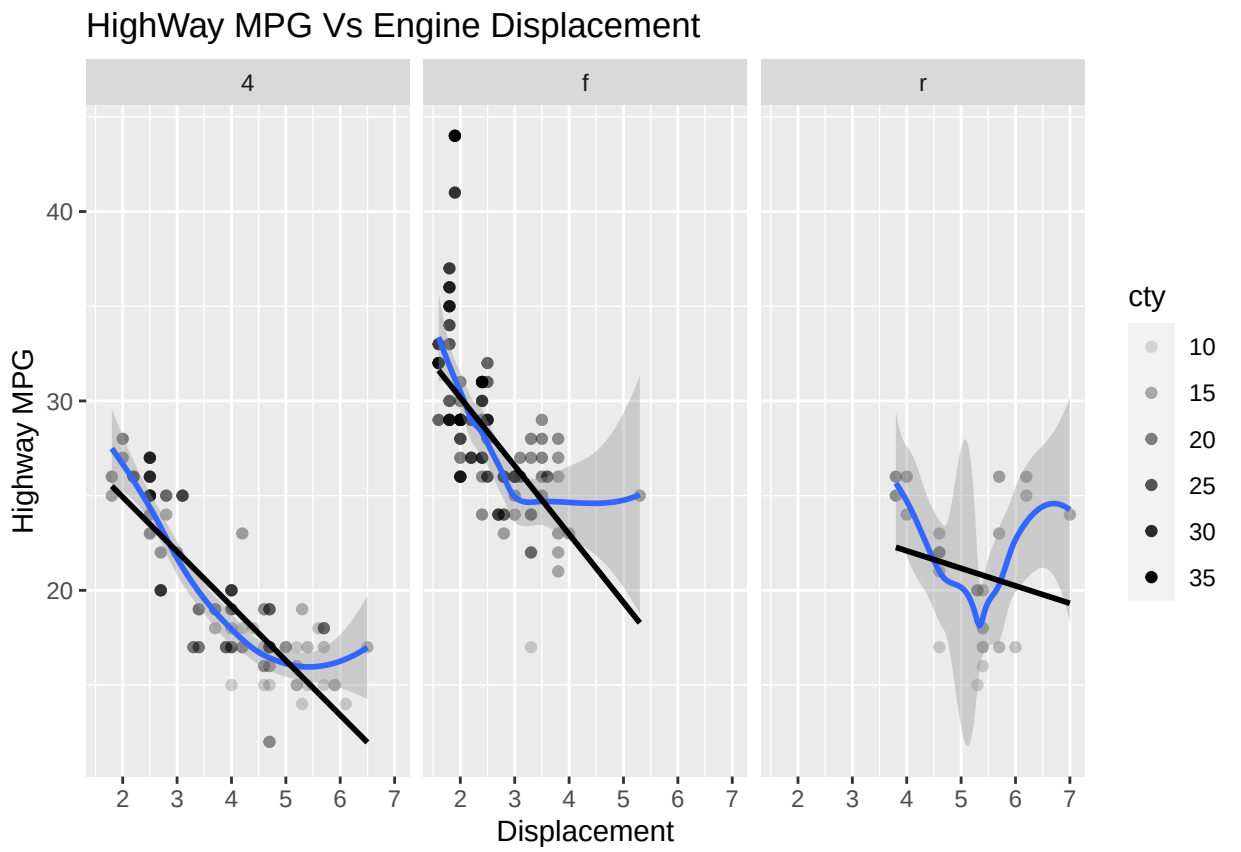

```
#the aesthetic is specified to color the data points based on the value of teh displ. The mark point fo
#The data is colored into above 5 and below 5.


#ggplot(data = mpg, aes(x=displ, y = hwy)) + geom_point(aes(alpha = cty)) +geom_smooth(aes(linetype = d

linmodel = lm(data = mpg, hwy~displ)
#install.packages("plm")
#library(plm)
linmodel
```

```
##
## Call:
## lm(formula = hwy ~ displ, data = mpg)
##
## Coefficients:
## (Intercept)        displ
##      35.698       -3.531
```

```
ggplot(data = mpg, aes(x=displ, y = hwy)) + geom_point(aes(alpha = cty)) +geom_smooth() + facet_wrap(~d
  geom_smooth (method = lm, se = FALSE, colour = "black") +
  labs (x = "Displacement", y = "Highway MPG") + ggtitle("HighWay MPG Vs Engine Displacement")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

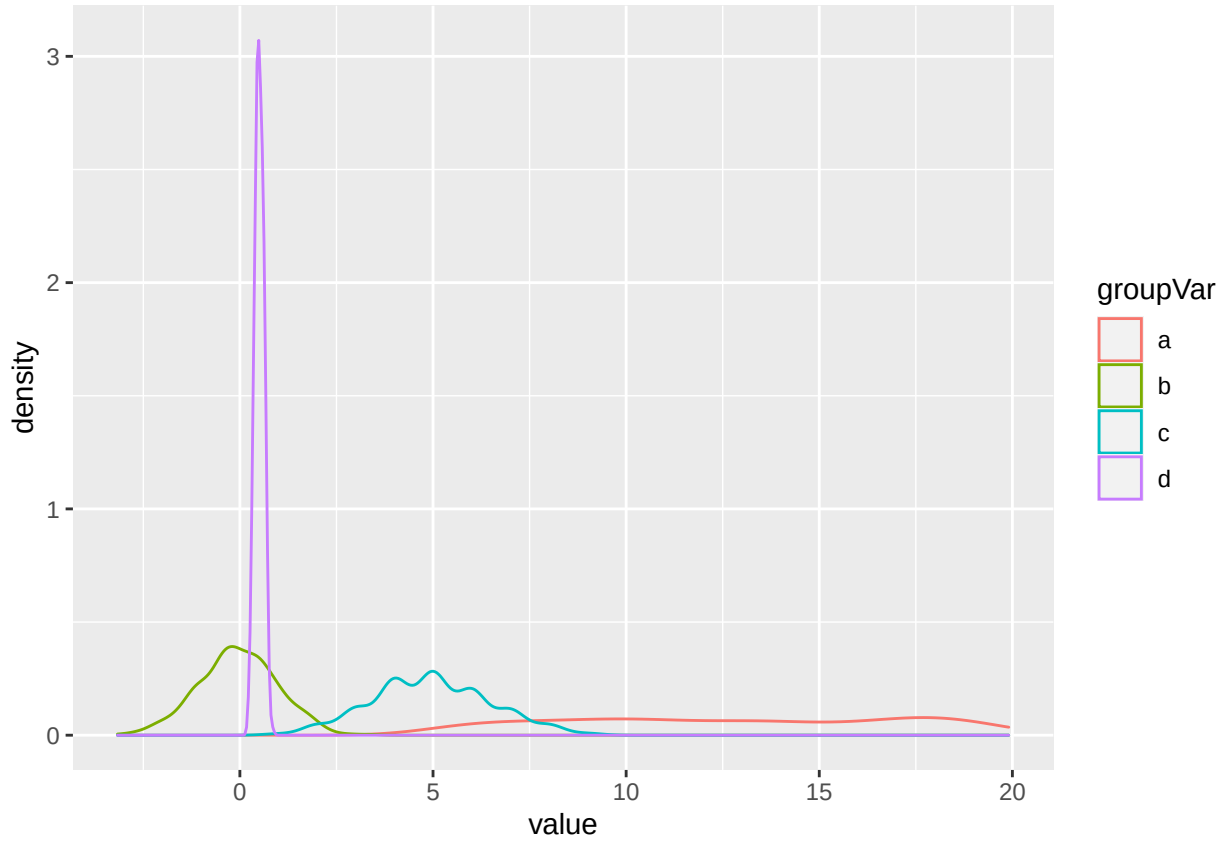
```
## `geom_smooth()` using formula 'y ~ x'
```



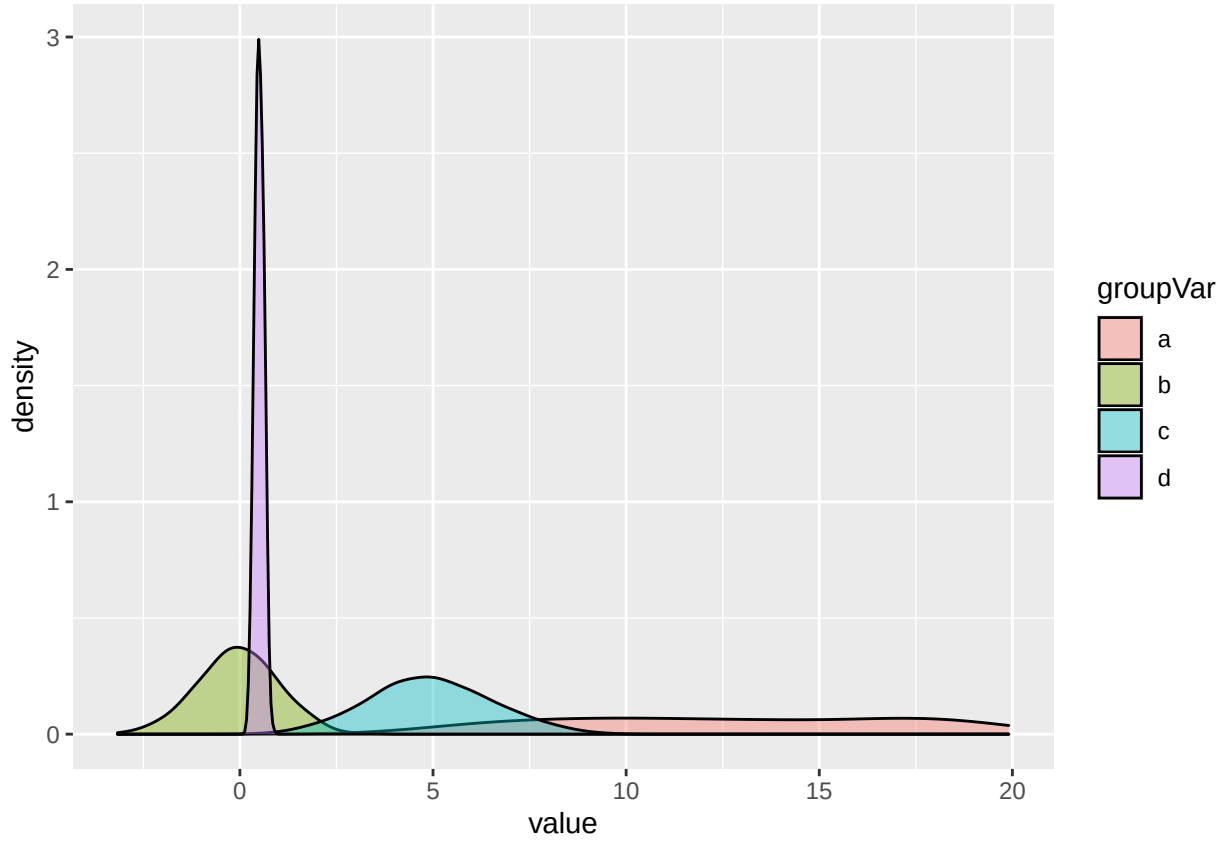

```
summary(linmodel)
```

```
##
## Call:
## lm(formula = hwy ~ displ, data = mpg)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.1039 -2.1646 -0.2242  2.0589 15.0105
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  35.6977     0.7204   49.55   <2e-16 ***
## displ        -3.5306     0.1945  -18.15   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.836 on 232 degrees of freedom
## Multiple R-squared:  0.5868, Adjusted R-squared:  0.585
## F-statistic: 329.5 on 1 and 232 DF,  p-value: < 2.2e-16
```

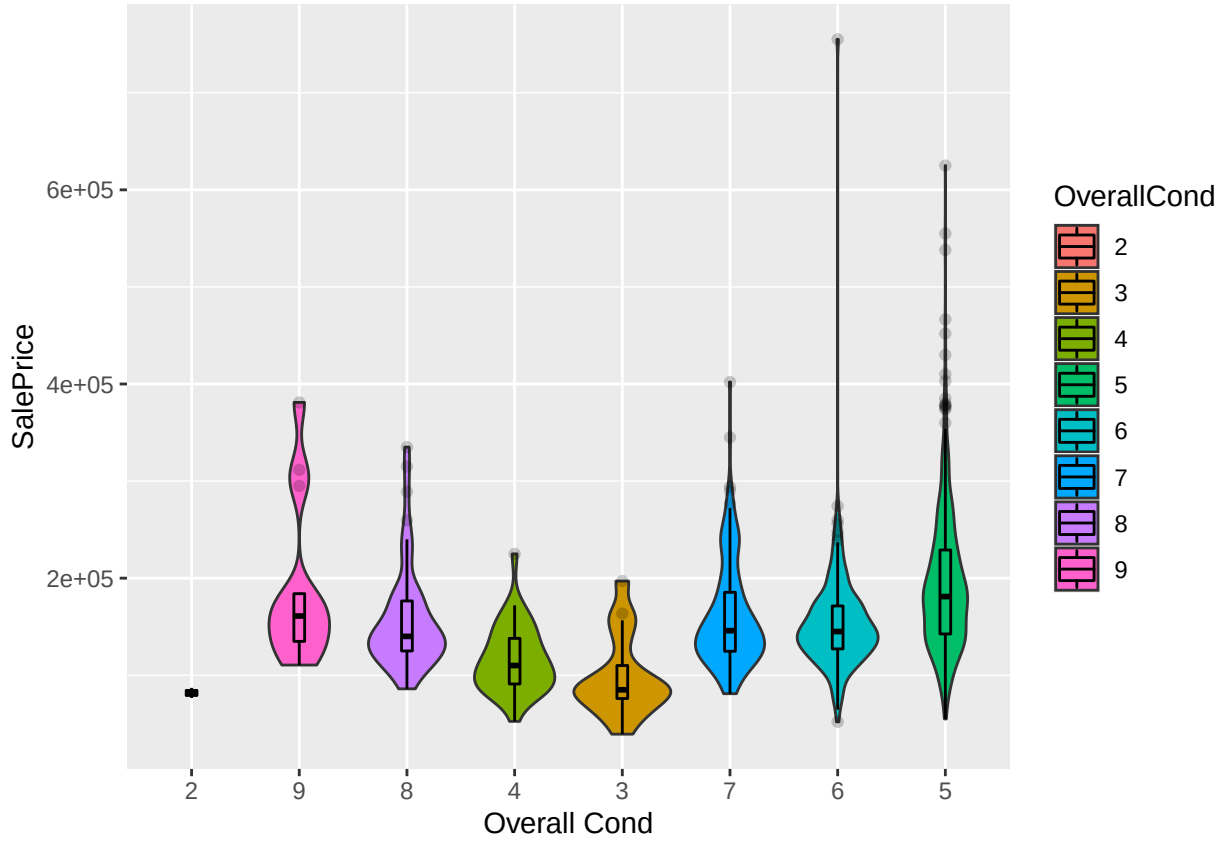
```r
#2a
a = c(runif(500, min = 5, max = 20))
b = c(rnorm(500, mean  =0, sd=1))
c = c(rbinom(p=0.5, size =10, n=500))
d = c(rbeta(n=500, 10,10))

df = data.frame (a, b, c, d)
df = gather(df, key = "groupVar", value="value")


ggplot(data = df)+aes(x=value)+geom_density((aes(color = groupVar)))
```



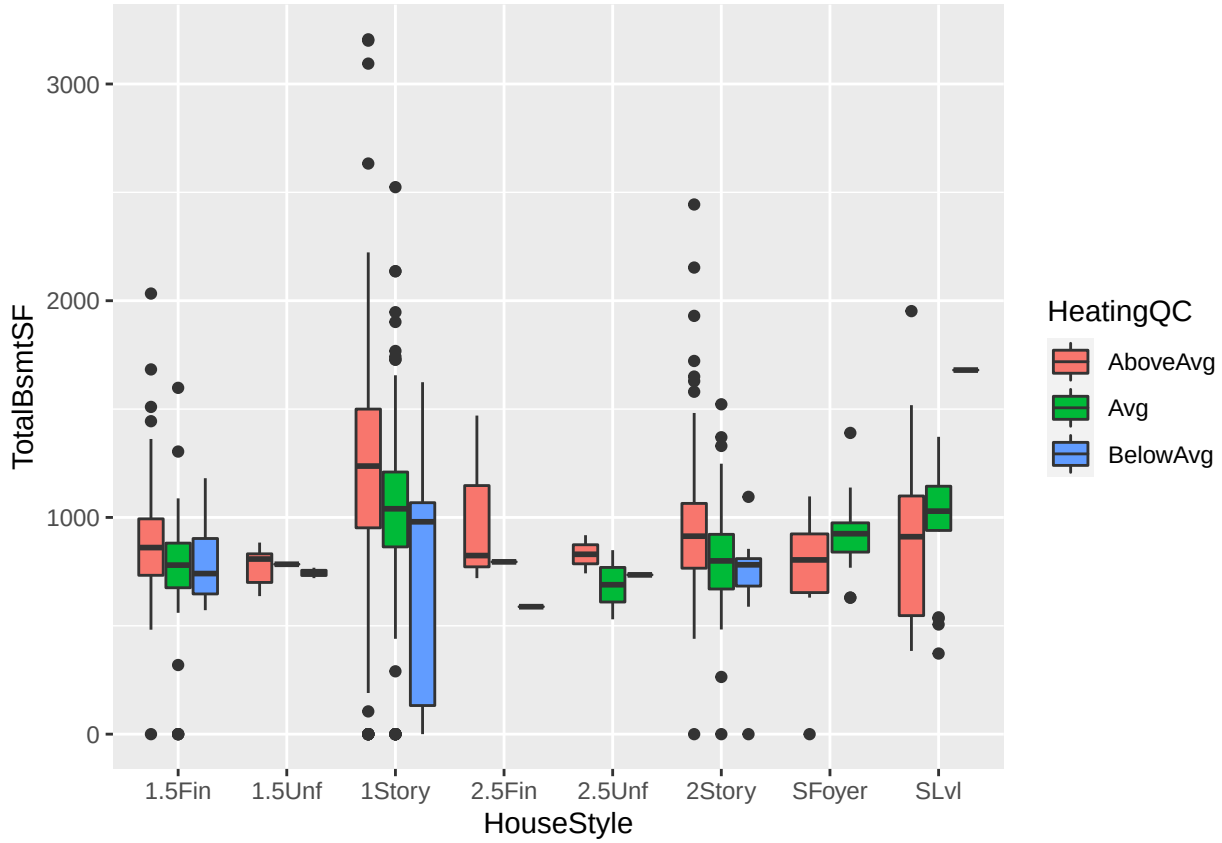

```r
ggplot(data = df, aes(x=value, group = groupVar, fill=groupVar))+geom_density(adjust = 1.5, alpha = 0.4
```



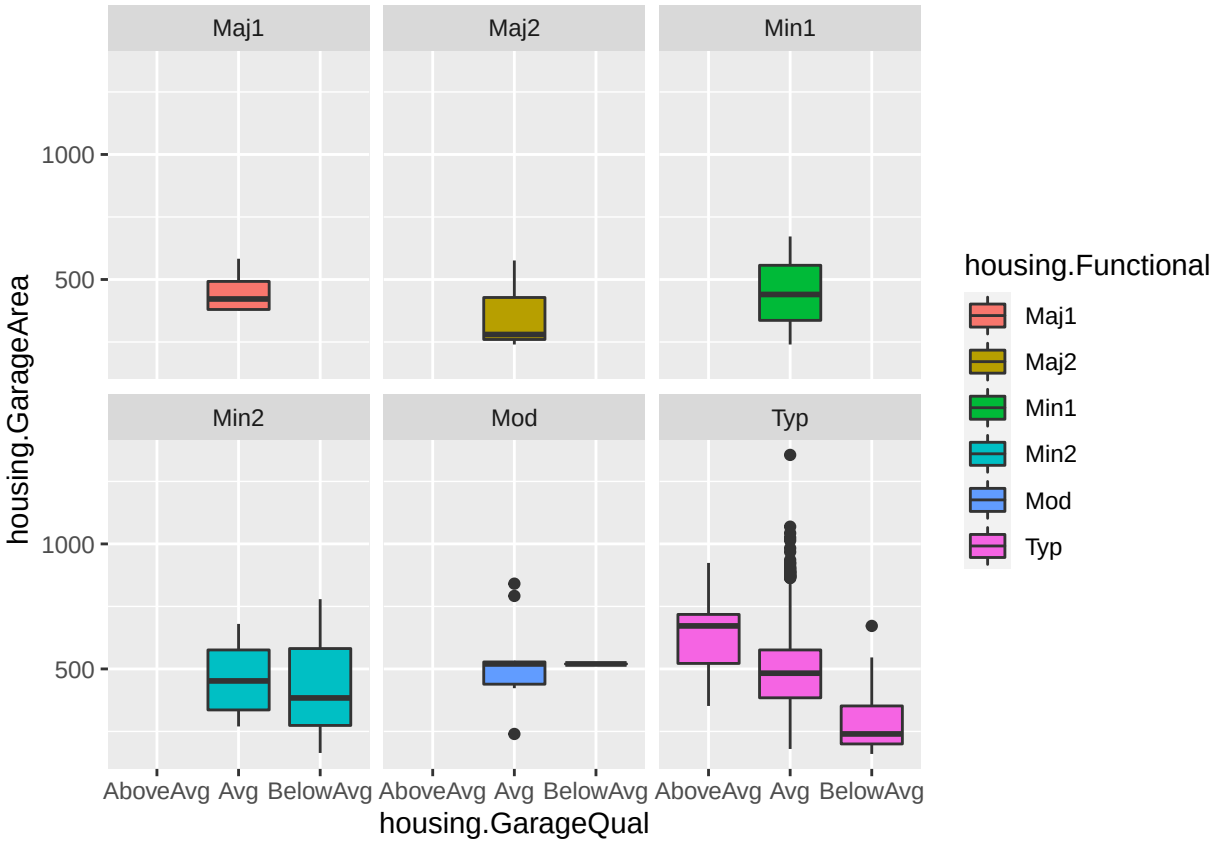

```
#3
housing = read.csv(file = 'housingData.csv',header = TRUE)
view(housing)

view(data)

housing$OverallCond = with(housing, reorder(OverallCond,YearBuilt))

ggplot(data= housing, aes(x=OverallCond, y = SalePrice, fill = OverallCond))+geom_violin()+
  xlab("Overall Cond")+geom_boxplot(width=0.1, color="black", alpha=0.2)
```



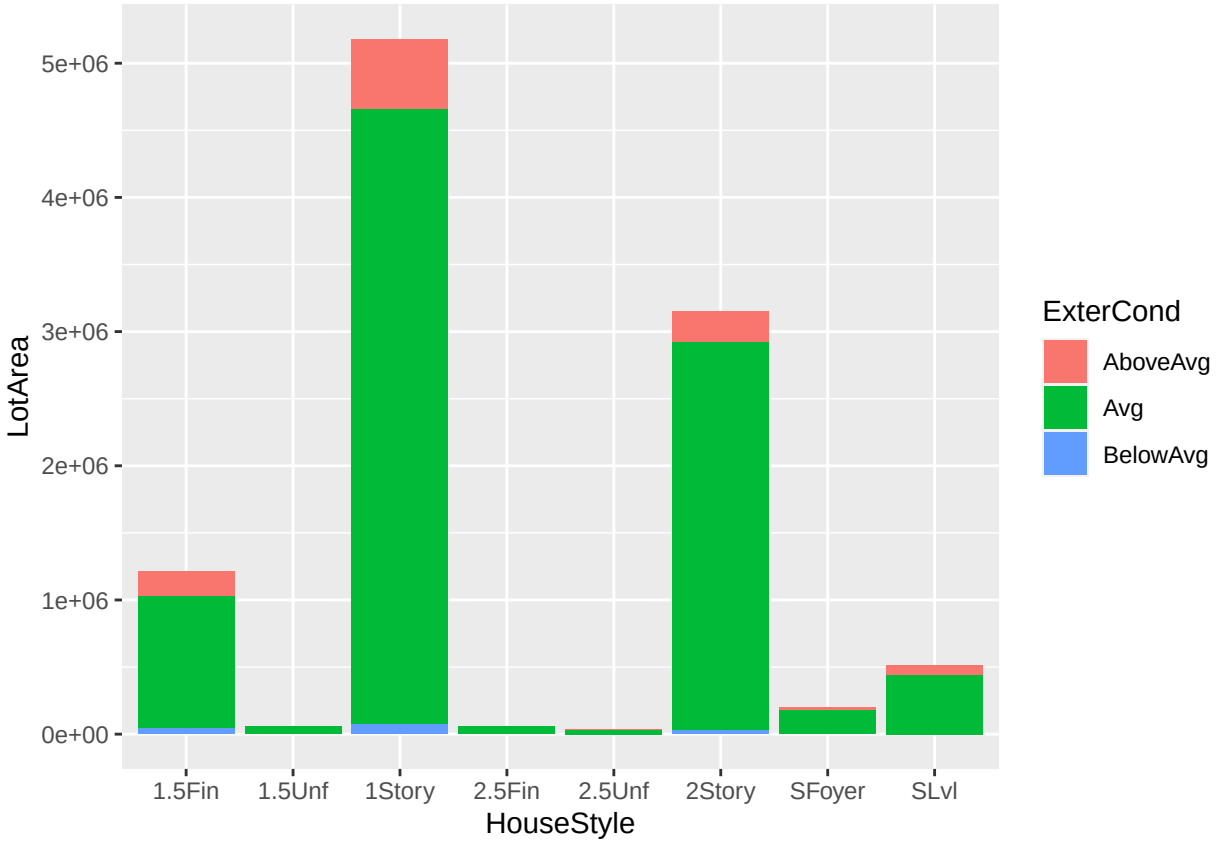

```
#this plot represent the Year the house was bulit vs the Overall conditions of the house.
#The plots allows us to view the ranking of the "Overall Cond" with the Sale price
#and also the distribution of the Overall Cond. The boxplots show that the higher the overall increases
#overall cond - 9 has the highest median value and the Overall cond- 3 has the lowest median.
#This shows that the relationship between the Overall Cond and the Sale price is positive.

ggplot(housing, aes(x = HouseStyle, y = TotalBsmtSF, fill=HeatingQC))+geom_boxplot()
```



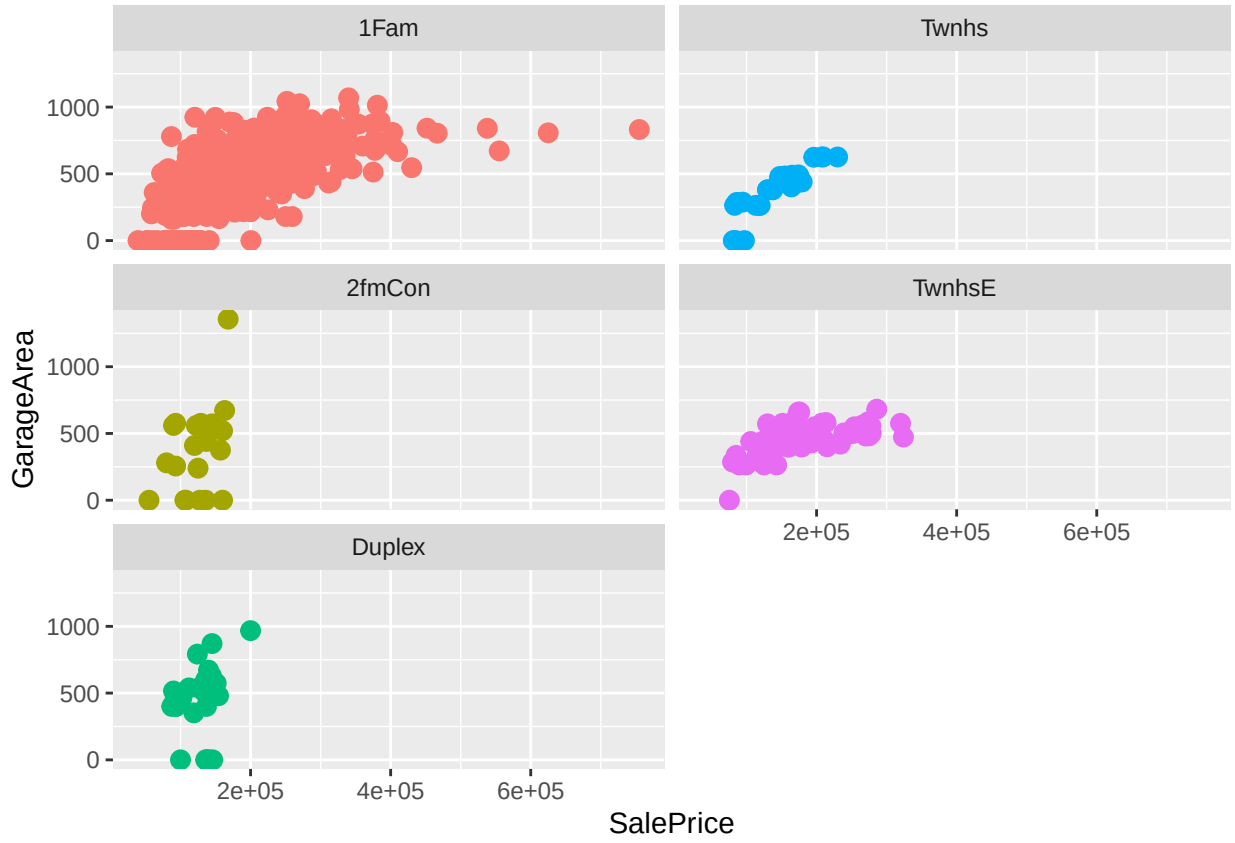

```
#this is a plot of Total basement SF vs the the style of the house.
#The box plot show the heating Quality. The plot shows the boxplot for the three categorical variables
#Heating values "above average" , "Avg" and "Below Avg".
#The total Basemnt Square foot was ploted on the y axis. For 1.5FIn, 1 story, 2 story, a trend of above
#average heating value was seen with higher Total Basement SF.
#The mean of the box plot for each of the box plots show that higher basement area have better heating

housing_new = data.frame(housing$Functional, housing$GarageQual, housing$GarageArea)
housing_new = na.exclude(housing_new)
ggplot(housing_new, aes(x = housing.GarageQual, y = housing.GarageArea, fill=housing.Functional))+geom_
  facet_wrap(~housing.Functional)
```




```
ggplot(housing, aes(fill=ExterCond, y = LotArea, x = HouseStyle)) +
  geom_bar(position="stack", stat="identity")
```




```
#this shows the Lot Area vs the style of the house. The 1 story building has the biggest lot area and
#the stacked bar chart also shows that the quality of the material of the exterior is highest in 1 stor
#which is the case when the LOT area is the highest.
#This shows that the Higher the lot area, the higher the proportion of the above average condition.
#This trend is als observed in 1.5uNF which as a very small lot area and
#therefore a very low proportion of "above average" exterior condition.
#The highest response is "Avg".

ggplot( housing , aes(x=SalePrice, y= GarageArea, color=as.factor(BldgType)  )) +
  geom_point(size=3) +
  facet_wrap(~BldgType , dir="v")  +
  theme(legend.position="none")
```




```
#the plot shows a positive relationship between the sales price of the house and the
#Garage area for various types of dwellings.
#The plot shows that a positive relationship occurs between the Garage area and sales price.
#The means that more expensive houses have larger garages

#4
#install.packages("Amelia")


data(freetrade)
aggregate(freetrade, by = list(freetrade$country), function(x) mean(is.na(x)))
```

```
##       Group.1 year country    tariff     polity pop gdp.pc   intresmi
## 1       India    0       0 0.3157895 0.00000000   0      0 0.05263158
## 2   Indonesia    0       0 0.4210526 0.00000000   0      0 0.05263158
## 3       Korea    0       0 0.2631579 0.05263158   0      0 0.05263158
## 4    Malaysia    0       0 0.3684211 0.00000000   0      0 0.10526316
## 5       Nepal    0       0 0.6315789 0.00000000   0      0 0.05263158
## 6    Pakistan    0       0 0.1578947 0.00000000   0      0 0.10526316
## 7 Philippines    0       0 0.0000000 0.05263158   0      0 0.05263158
## 8    SriLanka    0       0 0.4210526 0.00000000   0      0 0.10526316
## 9    Thailand    0       0 0.4736842 0.00000000   0      0 0.10526316
##       signed    fiveop usheg
## 1 0.00000000 0.1052632     0
## 2 0.05263158 0.1052632     0
## 3 0.00000000 0.1052632     0
```

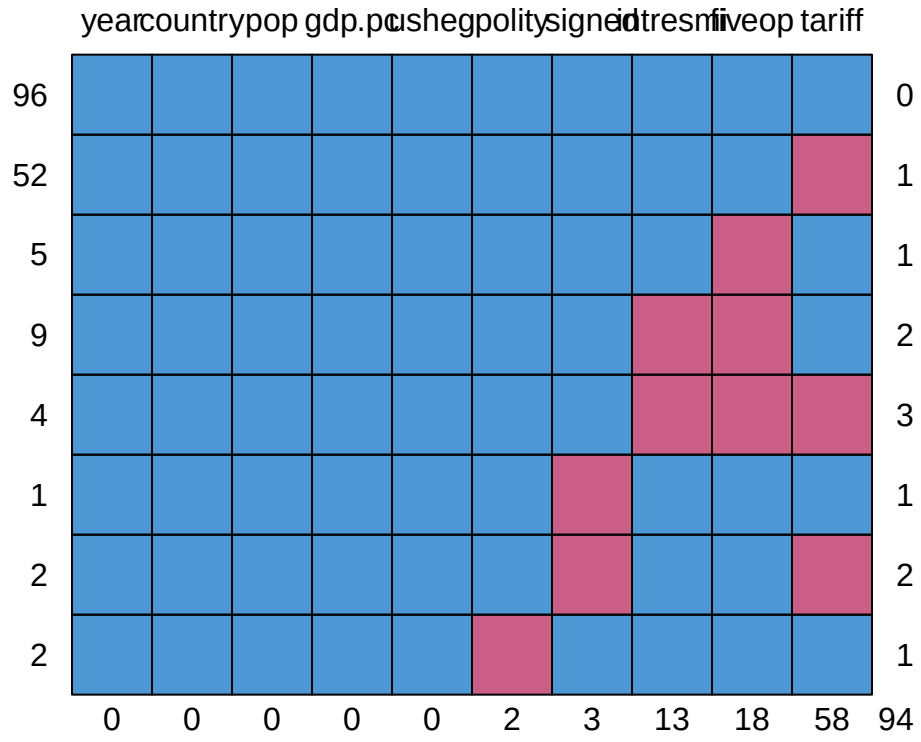
```
## 4 0.00000000 0.1052632     0
## 5 0.00000000 0.1052632     0
## 6 0.00000000 0.1052632     0
## 7 0.00000000 0.1052632     0
## 8 0.05263158 0.1052632     0
## 9 0.05263158 0.1052632     0
```

```
#install.packages("mice")


#?md.pairs
md.pairs(freetrade)     #exploring the missingness of the data using the mice package. The rr shows
```

```
## $rr
##          year country tariff polity pop gdp.pc intresmi signed fiveop usheg
## year      171     171    113    169 171    171      158    168    153   171
## country   171     171    113    169 171    171      158    168    153   171
## tariff    113     113    113    111 113    113      104    112     99   113
## polity    169     169    111    169 169    169      156    166    151   169
## pop       171     171    113    169 171    171      158    168    153   171
## gdp.pc    171     171    113    169 171    171      158    168    153   171
## intresmi  158     158    104    156 158    158      158    155    153   158
## signed    168     168    112    166 168    168      155    168    150   168
## fiveop    153     153     99    151 153    153      153    150    153   153
## usheg     171     171    113    169 171    171      158    168    153   171
##
## $rm
##          year country tariff polity pop gdp.pc intresmi signed fiveop usheg
## year        0       0     58      2   0      0       13      3     18     0
## country     0       0     58      2   0      0       13      3     18     0
## tariff      0       0      0      2   0      0        9      1     14     0
## polity      0       0     58      0   0      0       13      3     18     0
## pop         0       0     58      2   0      0       13      3     18     0
## gdp.pc      0       0     58      2   0      0       13      3     18     0
## intresmi    0       0     54      2   0      0        0      3      5     0
## signed      0       0     56      2   0      0       13      0     18     0
## fiveop      0       0     54      2   0      0        0      3      0     0
## usheg       0       0     58      2   0      0       13      3     18     0
##
## $mr
##          year country tariff polity pop gdp.pc intresmi signed fiveop usheg
## year        0       0      0      0   0      0        0      0      0     0
## country     0       0      0      0   0      0        0      0      0     0
## tariff     58      58      0     58  58     58       54     56     54    58
## polity      2       2      2      0   2      2        2      2      2     2
## pop         0       0      0      0   0      0        0      0      0     0
## gdp.pc      0       0      0      0   0      0        0      0      0     0
## intresmi   13      13      9     13  13     13        0     13      0    13
## signed      3       3      1      3   3      3        3      0      3     3
## fiveop     18      18     14     18  18     18        5     18      0    18
## usheg       0       0      0      0   0      0        0      0      0     0
##
## $mm
##          year country tariff polity pop gdp.pc intresmi signed fiveop usheg
```

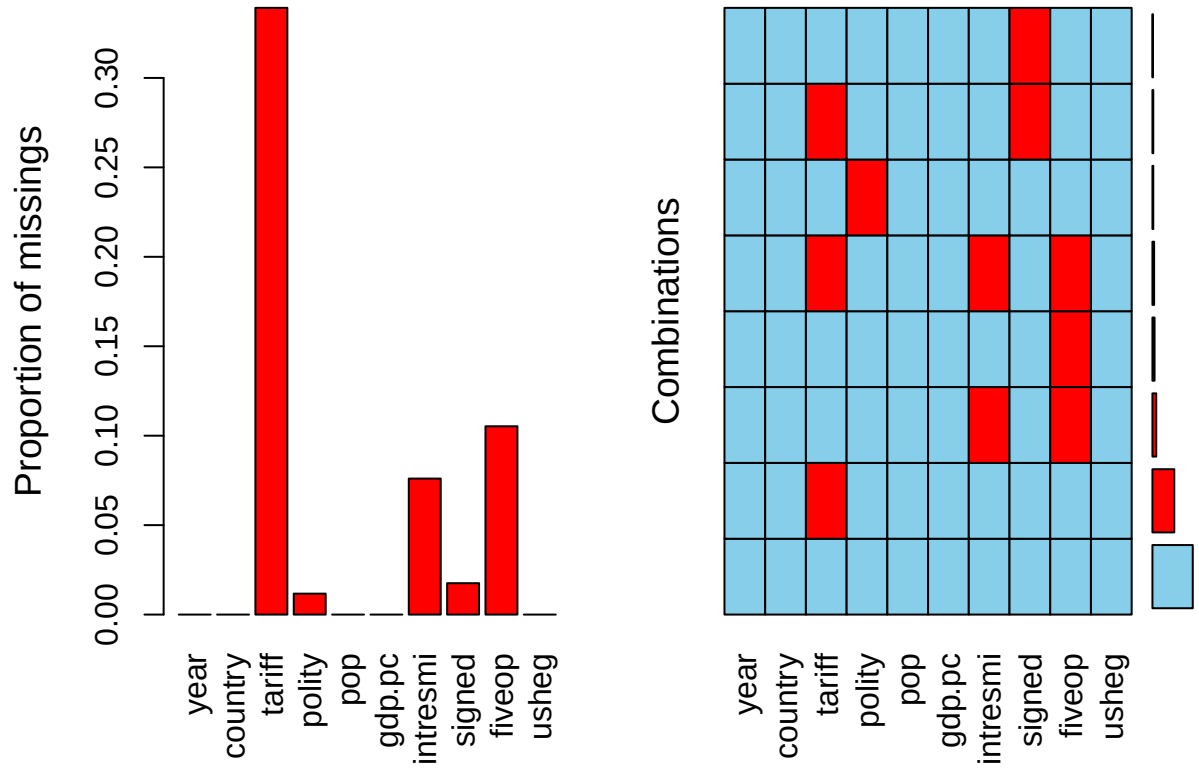
```
## year        0       0      0       0   0       0        0       0       0      0
## country     0       0      0       0   0       0        0       0       0      0
## tariff      0       0     58       0   0       0        4       2       4      0
## polity      0       0      0       2   0       0        0       0       0      0
## pop         0       0      0       0   0       0        0       0       0      0
## gdp.pc      0       0      0       0   0       0        0       0       0      0
## intresmi    0       0      4       0   0       0       13       0      13      0
## signed      0       0      2       0   0       0        0       3       0      0
## fiveop      0       0      4       0   0       0       13       0      18      0
## usheg       0       0      0       0   0       0        0       0       0      0
```

```
#the all the available data with no missing data. rm- the rows are observed but
#the column is missing for each data point.
#mr - row is missing but the column is available. mm- both variables are missing
md.pattern(freetrade)   #shows the pattern of the missing data.
```




```
##    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
```

```
#gives the total number of missing data. there are 96 values with no missing values and there
#52 times with tariff data missing.
#There are 5 times with fiveop missing and
#9 data with intresemi and fiveop missing.4 times when intresmi,
#fiveop and tariff is missing. 1 time when polity data is missing, 2 times when signed
#and tarfiff data is missing and 2 times when polity data is missing.
#This shows the number of times each data is missing and the pattern of the data.

#install.packages("VIM")

a = aggr(freetrade)
```

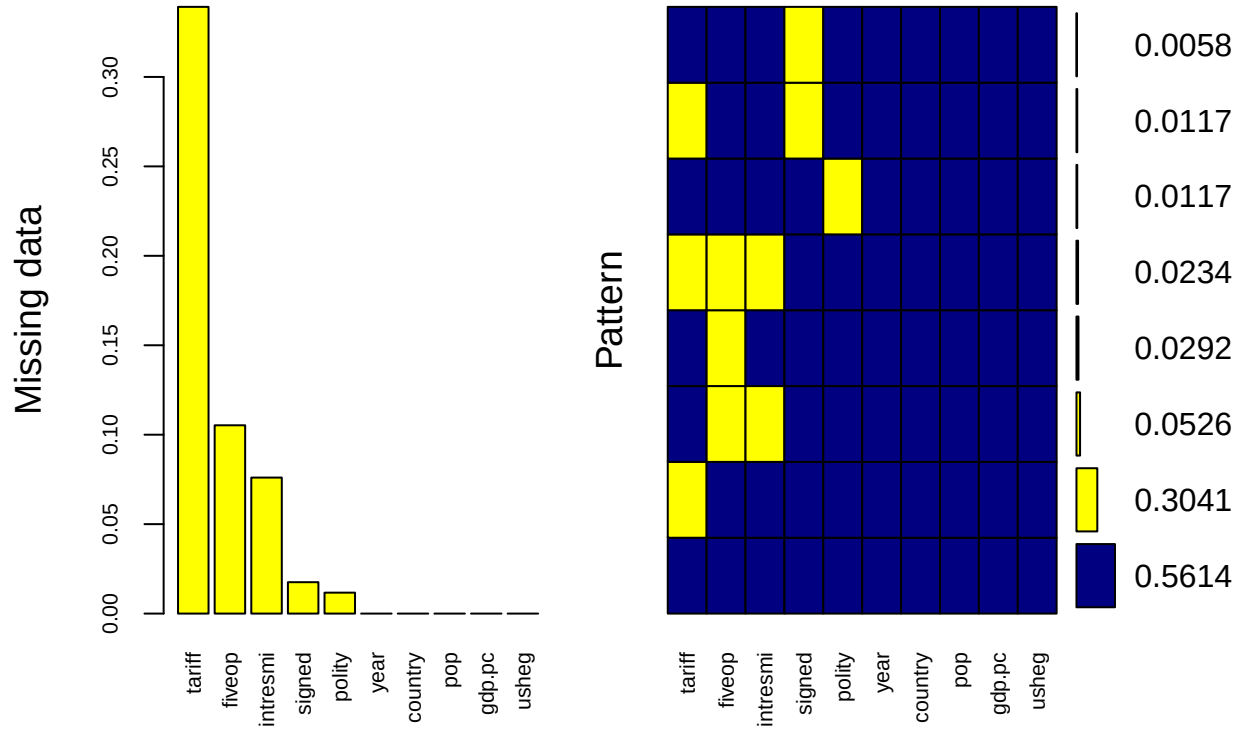



```
summary(a)
```

```
##
##  Missings per variable:
##  Variable Count
##      year     0
##   country     0
##    tariff    58
##    polity     2
##       pop     0
##    gdp.pc     0
##  intresmi    13
##    signed     3
```

```
##    fiveop    18
##     usheg     0
##
##  Missings in combinations of variables:
##         Combinations Count    Percent
##  0:0:0:0:0:0:0:0:0:0    96 56.1403509
##  0:0:0:0:0:0:0:0:1:0     5  2.9239766
##  0:0:0:0:0:0:0:1:0:0     1  0.5847953
##  0:0:0:0:0:0:1:0:1:0     9  5.2631579
##  0:0:0:1:0:0:0:0:0:0     2  1.1695906
##  0:0:1:0:0:0:0:0:0:0    52 30.4093567
##  0:0:1:0:0:0:0:1:0:0     2  1.1695906
##  0:0:1:0:0:0:1:0:1:0     4  2.3391813
```

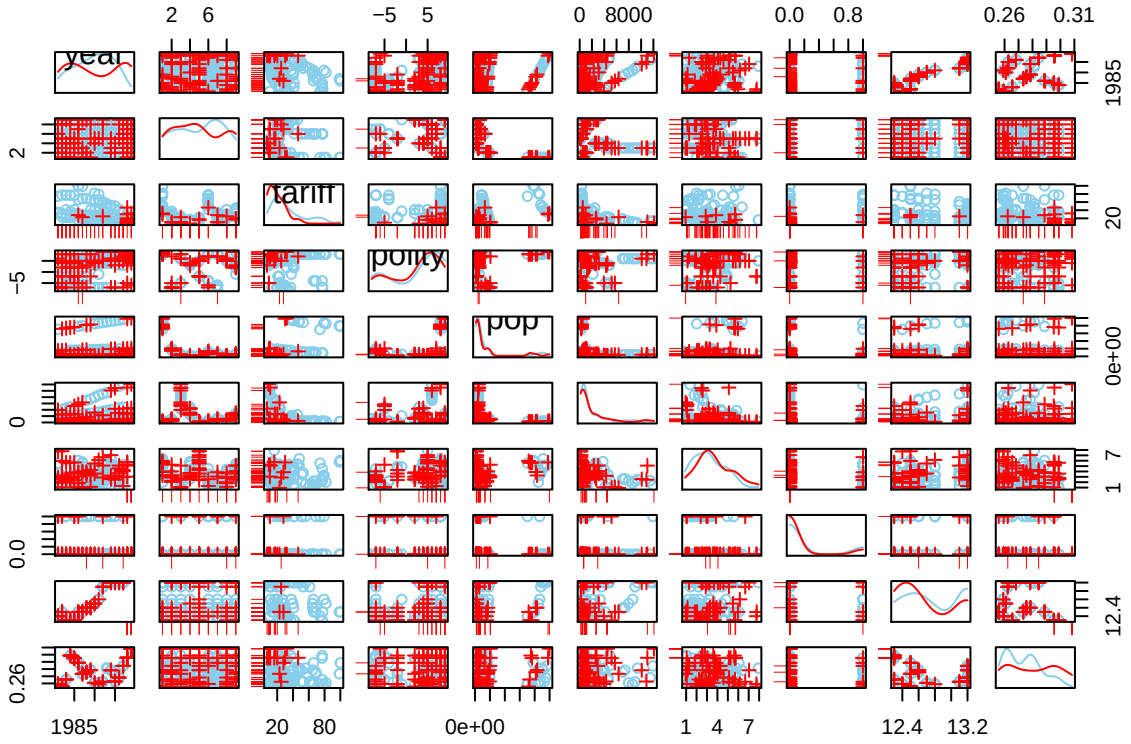
```
aggr(freetrade, col=c('navyblue','yellow'),
                numbers=TRUE, sortVars=TRUE,
                labels=names(freetrade), cex.axis=.7,
                gap=3, ylab=c("Missing data","Pattern"))
```




```
##
##  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
```

```
#tariff is the variable with the most missingness as shown in the plot.

scattmatrixMiss(freetrade)
```




```
#5
new  = data.frame(freetrade$country, freetrade$tariff)    #dataframe for just freetrade country and Tar

#new = na.exclude(new)
view(new)
mean(is.na(freetrade$tariff)) #shows the percentage of the missing data in the tariff variable
```

```
## [1] 0.3391813
```

```
md.pairs(new)
```

```
## $rr
```

```
##                   freetrade.country freetrade.tariff
## freetrade.country               171              113
## freetrade.tariff                113              113
##
## $rm
##                   freetrade.country freetrade.tariff
## freetrade.country                 0               58
## freetrade.tariff                  0                0
##
## $mr
##                   freetrade.country freetrade.tariff
## freetrade.country                 0                0
## freetrade.tariff                 58                0
##
## $mm
##                   freetrade.country freetrade.tariff
## freetrade.country                 0                0
## freetrade.tariff                  0               58
```

```
aggregate(new, by = list(new$freetrade.country), function(x) mean(is.na(x)))
```

```
##       Group.1 freetrade.country freetrade.tariff
## 1       India                 0        0.3157895
## 2   Indonesia                 0        0.4210526
## 3       Korea                 0        0.2631579
## 4    Malaysia                 0        0.3684211
## 5       Nepal                 0        0.6315789
## 6    Pakistan                 0        0.1578947
## 7 Philippines                 0        0.0000000
## 8    SriLanka                 0        0.4210526
## 9    Thailand                 0        0.4736842
```

```
#shows the individual percentage of the missing data of each varibale
freetradeagg = aggregate(new, by = list(new$freetrade.country), function(x) sum(is.na(x)))
#aggregate the data frame to get the total number of missing data for tariff varibale
freetradeagg_na = freetradeagg$freetrade.tariff       #the column that has the tariff
x = freetradeagg_na
#create new varibale x to store the total number of missing tariff for each country
freetradeagg1 = aggregate(new, by = list(new$freetrade.country), function(x) sum(!(is.na(x))))
#aggregate the data frame to get the total number of obsevered data for tariff varibale
freetradeagg1_ = freetradeagg1$freetrade.tariff
y = freetradeagg1_
#create new varibale x to store the total number of observed tariff for each country

ddf = data.frame(x,y)   #create new dataframe to store the total number of missing and observed value p


ddf$rowtotal = with(ddf, (x+y))       #new column to add rows x and y i.e. row total
rbind(ddf, colSums(ddf[,1:2]))
```

```
##     x   y rowtotal
## 1   6  13       19
## 2   8  11       19
```

```
## 3   5  14       19
## 4   7  12       19
## 5  12   7       19
## 6   3  16       19
## 7   0  19       19
## 8   8  11       19
## 9   9  10       19
## 10 58 113       58
```

```
ddf$chi_x = with(ddf, (rowtotal* 58)/171) #new column to get the nee values
ddf$chi_y = with(ddf, (rowtotal* 113)/171) #new column to get new values for the

ddf$test_stat_na = with (ddf, ((x-chi_x)^2)/chi_x)     #test statistics for the mising tariff varibale
ddf$test_stat = with (ddf, ((y-chi_y)^2)/chi_y)      #test statistics for the observed tariff varibale
ddf
```

```
##    x  y rowtotal    chi_x    chi_y test_stat_na  test_stat
## 1  6 13       19 6.444444 12.55556   0.03065134 0.01573255
## 2  8 11       19 6.444444 12.55556   0.37547893 0.19272370
## 3  5 14       19 6.444444 12.55556   0.32375479 0.16617502
## 4  7 12       19 6.444444 12.55556   0.04789272 0.02458210
## 5 12  7       19 6.444444 12.55556   4.78927203 2.45821042
## 6  3 16       19 6.444444 12.55556   1.84099617 0.94493609
## 7  0 19       19 6.444444 12.55556   6.44444444 3.30776794
## 8  8 11       19 6.444444 12.55556   0.37547893 0.19272370
## 9  9 10       19 6.444444 12.55556   1.01340996 0.52015733
```

```
chi_square = sum(ddf$test_stat[1:9]) + sum(ddf$test_stat_na[1:9])

df_ = (9-1) * (2-1)   #degree of freedom
prob = 1-pchisq(chi_square,df_)    #calculate the p value for the test stastics
prob
```

```
## [1] 0.003282555
```

```
#since p value is very small (close to zero), we do  reject the null hypothesis that they are independe
#The evidence show that the country and the tariff data missingness are related. Independence is reject
rbind(ddf, colSums(ddf[,1:2]))
```

```
##     x   y rowtotal      chi_x    chi_y test_stat_na   test_stat
## 1   6  13       19   6.444444 12.55556   0.03065134  0.01573255
## 2   8  11       19   6.444444 12.55556   0.37547893  0.19272370
## 3   5  14       19   6.444444 12.55556   0.32375479  0.16617502
## 4   7  12       19   6.444444 12.55556   0.04789272  0.02458210
## 5  12   7       19   6.444444 12.55556   4.78927203  2.45821042
## 6   3  16       19   6.444444 12.55556   1.84099617  0.94493609
## 7   0  19       19   6.444444 12.55556   6.44444444  3.30776794
## 8   8  11       19   6.444444 12.55556   0.37547893  0.19272370
## 9   9  10       19   6.444444 12.55556   1.01340996  0.52015733
## 10 58 113       58 113.000000 58.00000 113.00000000 58.00000000
```

```
ddf
```

```
##    x  y rowtotal    chi_x    chi_y test_stat_na  test_stat
## 1  6 13       19 6.444444 12.55556   0.03065134 0.01573255
## 2  8 11       19 6.444444 12.55556   0.37547893 0.19272370
## 3  5 14       19 6.444444 12.55556   0.32375479 0.16617502
## 4  7 12       19 6.444444 12.55556   0.04789272 0.02458210
## 5 12  7       19 6.444444 12.55556   4.78927203 2.45821042
## 6  3 16       19 6.444444 12.55556   1.84099617 0.94493609
## 7  0 19       19 6.444444 12.55556   6.44444444 3.30776794
## 8  8 11       19 6.444444 12.55556   0.37547893 0.19272370
## 9  9 10       19 6.444444 12.55556   1.01340996 0.52015733
```

```
#after removing Nepal
new2 = new[!(new$freetrade.country == "Nepal"), ]

aggregate(new2, by = list(new2$freetrade.country), function(x) mean(is.na(x)))
```

```
##       Group.1 freetrade.country freetrade.tariff
## 1       India                 0        0.3157895
## 2   Indonesia                 0        0.4210526
## 3       Korea                 0        0.2631579
## 4    Malaysia                 0        0.3684211
## 5    Pakistan                 0        0.1578947
## 6 Philippines                 0        0.0000000
## 7    SriLanka                 0        0.4210526
## 8    Thailand                 0        0.4736842
```

```
#shows the individual percentage of the missing data of each varibale
freetradeagg = aggregate(new2, by = list(new2$freetrade.country), function(x) sum(is.na(x)))
#aggregate the data frame to get the total number of missing data for tariff varibale

freetradeagg_na = freetradeagg$freetrade.tariff       #the column that has the tariff
x = freetradeagg_na       #create new varibale x to store the total number of missing tariff for each c
freetradeagg1 = aggregate(new2, by = list(new2$freetrade.country), function(x) sum(!(is.na(x))))
#aggregate the data frame to get the total number of obsevered data for tariff varibale
freetradeagg1_ = freetradeagg1$freetrade.tariff
y = freetradeagg1_           #create new varibale x to store the total number of observed tariff for ea

ddf = data.frame(x,y)   #create new dataframe to store the total number of missing and observed value p


ddf$rowtotal = with(ddf, (x+y))       #new column to add rows x and y i.e. row total
rbind(ddf, colSums(ddf[,1:2]))
```

```
##    x   y rowtotal
## 1  6  13       19
## 2  8  11       19
## 3  5  14       19
## 4  7  12       19
## 5  3  16       19
## 6  0  19       19
```

```
## 7  8  11       19
## 8  9  10       19
## 9 46 106       46
```

```
ddf$chi_x = with(ddf, (rowtotal* 46)/152) #new column to get the nee values
ddf$chi_y = with(ddf, (rowtotal* 106)/152) #new column to get new values for the

ddf$test_stat_na = with (ddf, ((x-chi_x)^2)/chi_x)     #test statistics for the mising tariff varibale
ddf$test_stat = with (ddf, ((y-chi_y)^2)/chi_y)      #test statistics for the observed tariff varibale
ddf
```

```
##   x  y rowtotal chi_x chi_y test_stat_na   test_stat
## 1 6 13       19  5.75 13.25   0.01086957 0.004716981
## 2 8 11       19  5.75 13.25   0.88043478 0.382075472
## 3 5 14       19  5.75 13.25   0.09782609 0.042452830
## 4 7 12       19  5.75 13.25   0.27173913 0.117924528
## 5 3 16       19  5.75 13.25   1.31521739 0.570754717
## 6 0 19       19  5.75 13.25   5.75000000 2.495283019
## 7 8 11       19  5.75 13.25   0.88043478 0.382075472
## 8 9 10       19  5.75 13.25   1.83695652 0.797169811
```

```
chi_square = sum(ddf$test_stat[1:8]) + sum(ddf$test_stat_na[1:8])
df_ = (8-1) * (2-1)   #degree of freedom
prob = 1-pchisq(chi_square,df_)    #calculate the p value for the test stastics
prob
```

```
## [1] 0.02665832
```

```
#the probability value changes slightly, but the value is still very small (close to zero)
#and this means the two varibales are  not independent
```

Including Plots

You can also embed plots, for example:

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.