# Rumainum-HW2

Lince Rumainum

September 5, 2019

## PROBLEM 1

```
# Load all the libraries for HW-2
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------
------------------------------------ tidyverse 1.2.1 --
```

```
## v ggplot2 3.2.1     v purrr   0.3.2
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   0.8.3     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
```

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

```
library(ggplot2)
library(lme4)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following object is masked from 'package:tidyr':
##
##     expand
```

```
library(reshape2)
```

```
##
## Attaching package: 'reshape2'
```

```
## The following object is masked from 'package:tidyr':
##
##     smiths
```

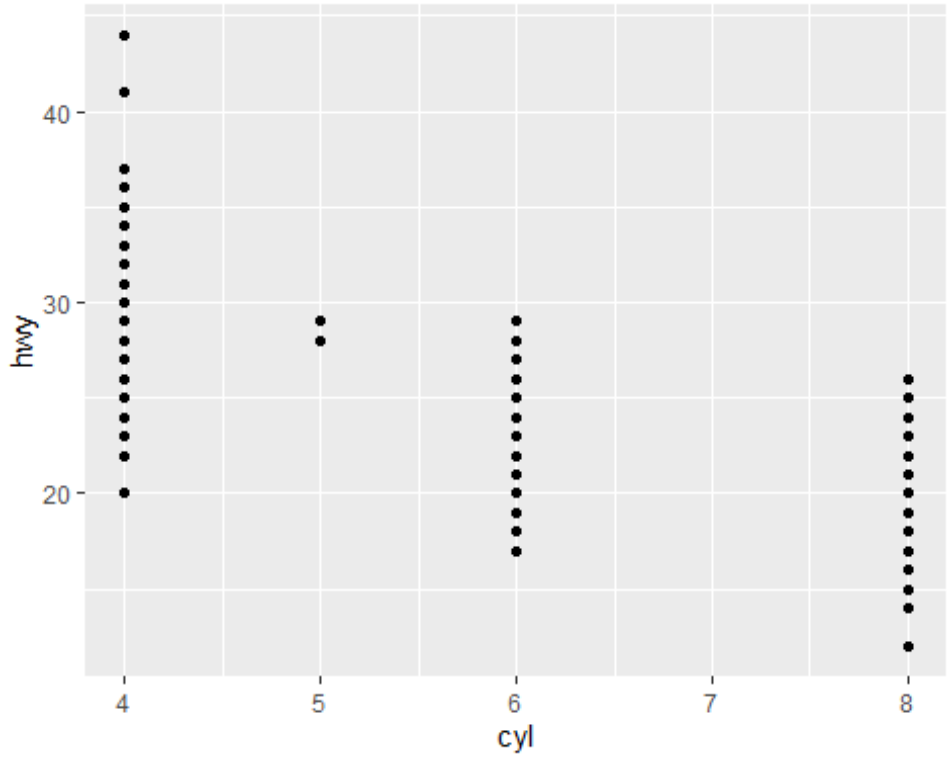
```
library(Amelia)
```

```
## Loading required package: Rcpp
```

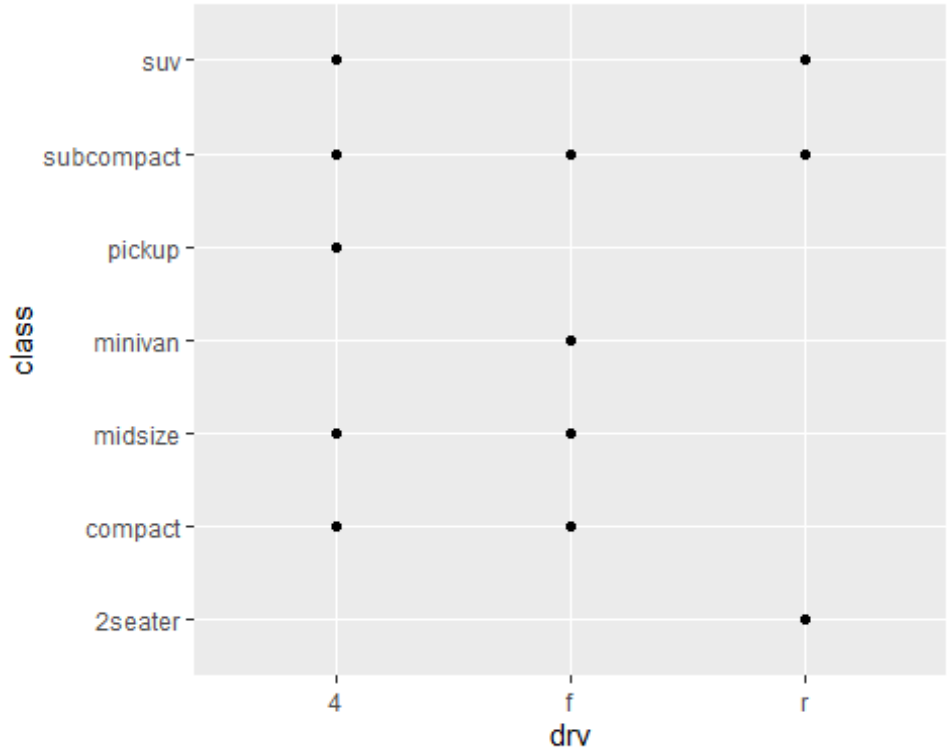
```
## ##
## ## Amelia II: Multiple Imputation
```

```
## ## (Version 1.7.5, built: 2018-05-07)
## ## Copyright (C) 2005-2019 James Honaker, Gary King and Matthew Blackwell
## ## Refer to http://gking.harvard.edu/amelia/ for more information
## ##

library(VIM)

## Loading required package: colorspace

## Loading required package: grid

## Loading required package: data.table

##
## Attaching package: 'data.table'

## The following objects are masked from 'package:reshape2':
##
##     dcast, melt

## The following objects are masked from 'package:dplyr':
##
##     between, first, last

## The following object is masked from 'package:purrr':
##
##     transpose

## Registered S3 methods overwritten by 'car':
##   method                          from
##   influence.merMod                lme4
##   cooks.distance.influence.merMod lme4
##   dfbeta.influence.merMod         lme4
##   dfbetas.influence.merMod        lme4

## VIM is ready to use.
##  Since version 4.0.0 the GUI is in its own package VIMGUI.
##
##           Please use the package to use the new (and old) GUI.

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

##
## Attaching package: 'VIM'

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

# PROBLEM 1
# Problem 1a
# 3.2.4 Exercise #4
```

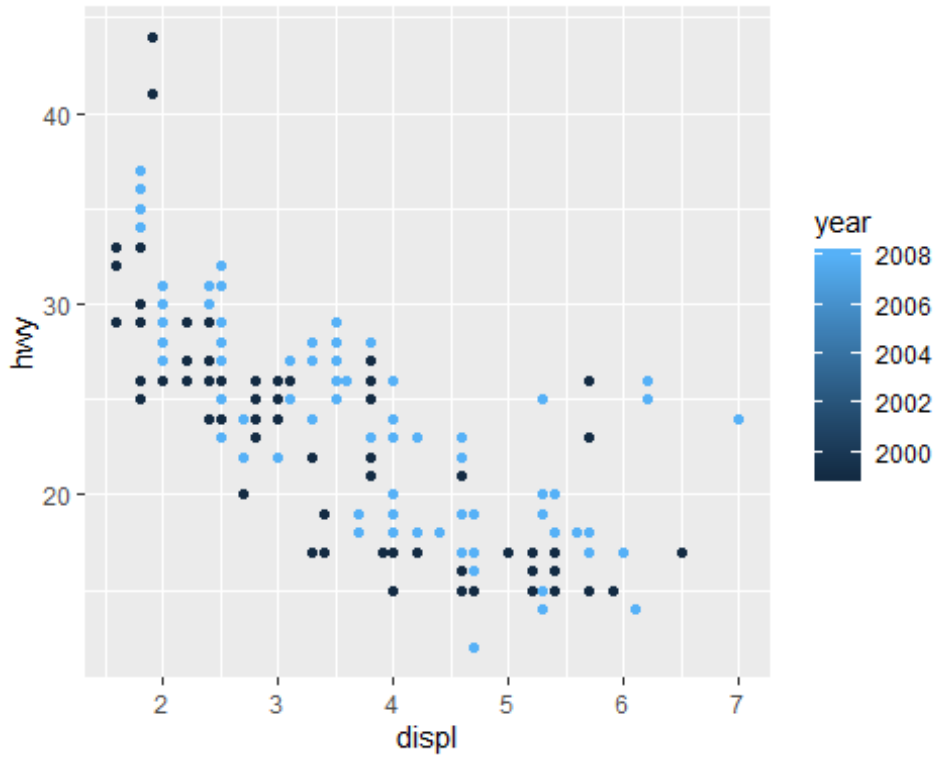
```r
# create a scatter plot from mpg data sets of hwy vs cyl
ggplot(data = mpg) +
  geom_point(mapping = aes(x = cyl, y = hwy))
```




```r
# 3.2.4 Exercise #5
# create a scatter plot from mpg data sets of class vs drv
ggplot(data = mpg) +
  geom_point(mapping = aes(x = drv, y = class))
```

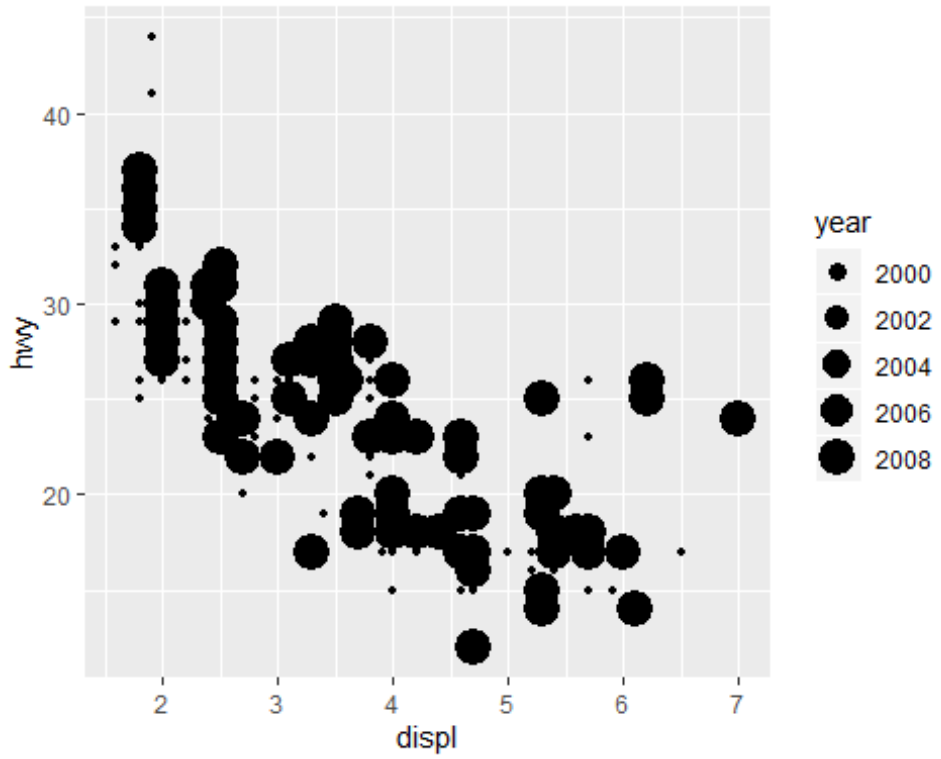



```
# The plot is not useful because it only gives information about the type of
car and whether it uses front-wheel drive, rear-wheel drive, or four-wheel
drive and nothing about its fuel economy.

# 3.3.1 Exercise #3
# create a plot of hwy vs displ and map it with a continuous variable, year
for color aesthetic
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, color = year))
```

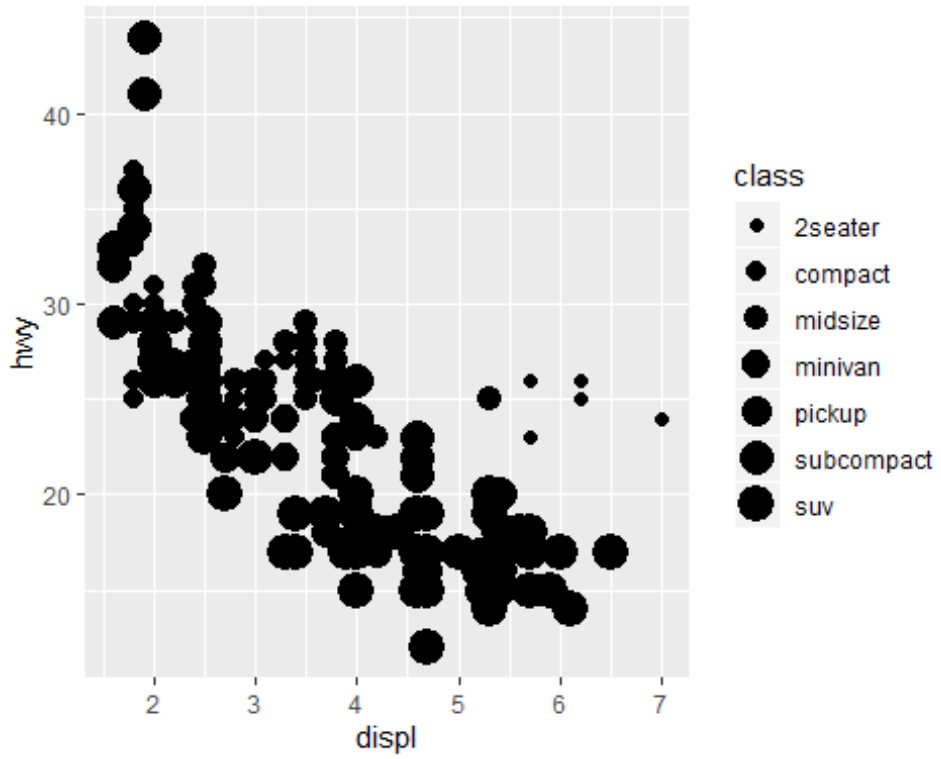



```
# In case of color aesthetic, categorical vs. continous variables would work
either way where categorical variables will create range of distinct colors,
such as, red, blue, green, etc. while the continuous variables will create a
range of continuous shade of a specific color, such as, different shade of
the color blue.

# create a plot of hwy vs displ and map it with a categorical variable, class
for color aesthetic
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, color = class))
```

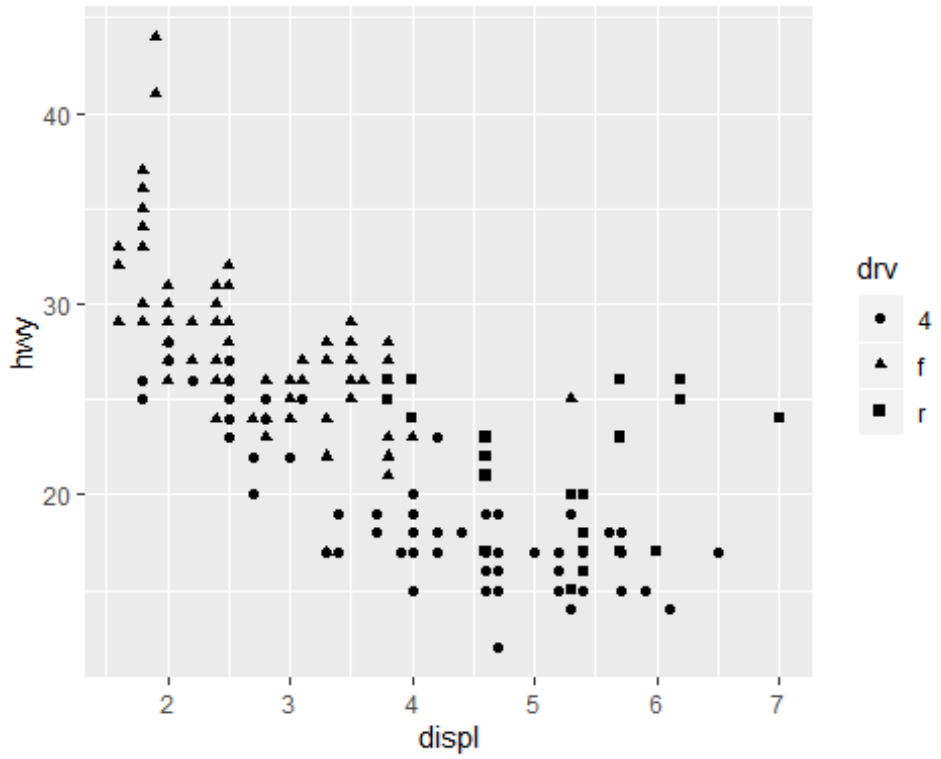

```r
# create a plot of hwy vs displ and map it with a continuous variable, year
for size aesthetic
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, size = year))
```



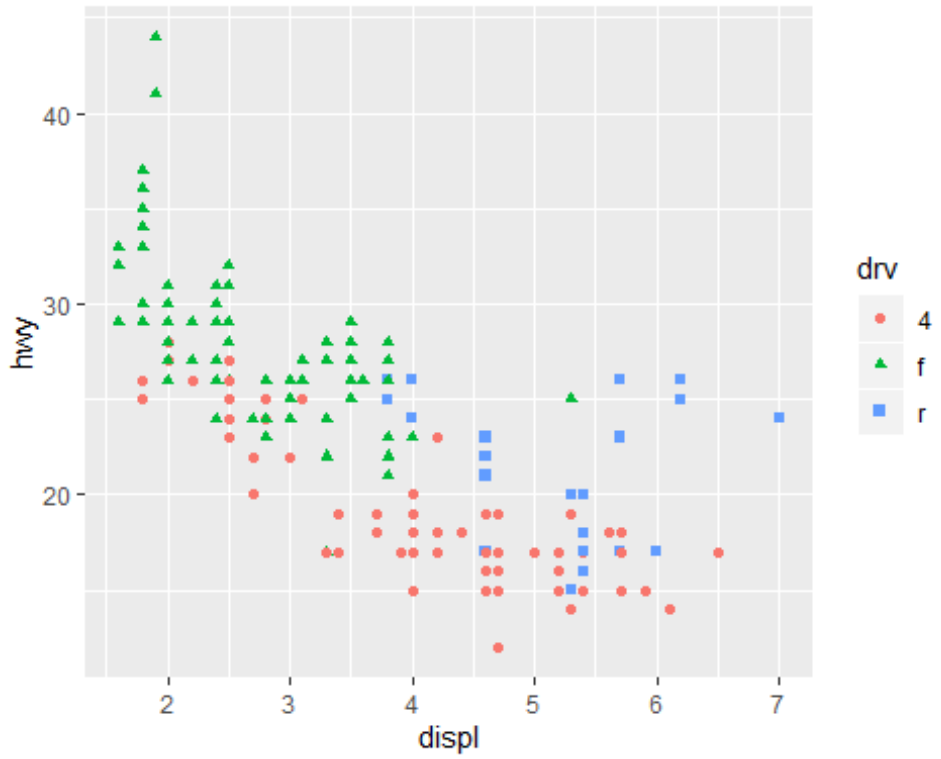

```
# In case of size aesthetic, the use of a discrete (categorical) variables
are not advised because size is an ordered aesthetic and a categorical
variable is an unordered variable.

# create a plot of hwy vs displ and map it with a categorical variable, class
for size aesthetic
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, size = class))

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



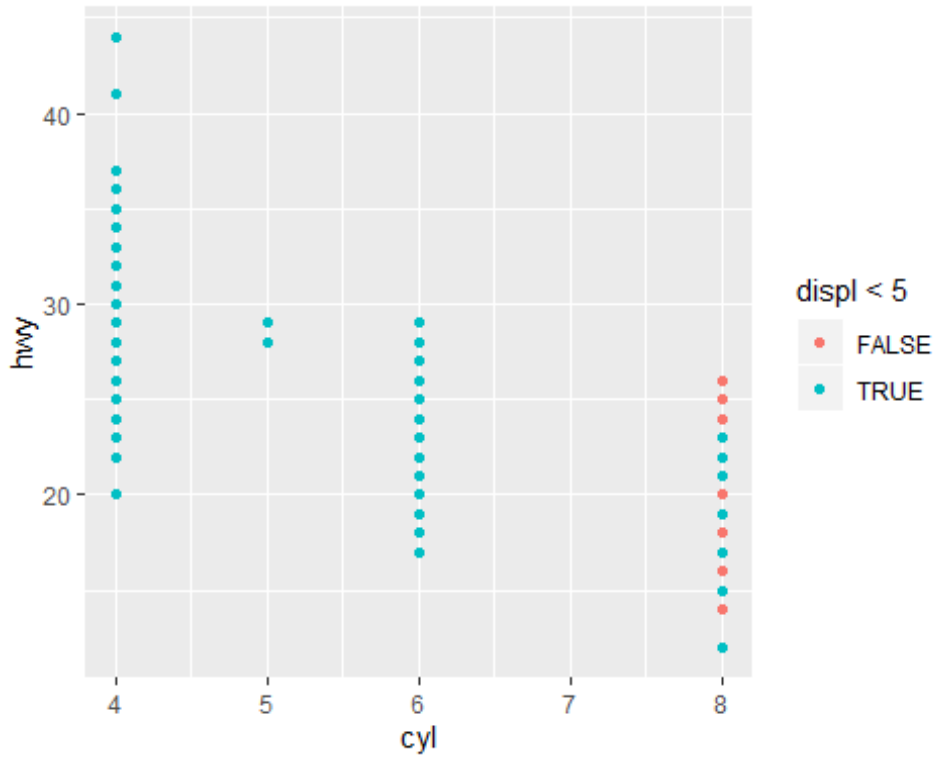

```r
# create a plot of hwy vs displ and map it with a continuous variable, year
for shape aesthetic
# Note: next code is commentted out on purpose because continouse variable
cannot be used for shape aesthetic
#ggplot(data = mpg) +
#  geom_point(mapping = aes(x = displ, y = hwy, shape = year))

# In case of shape aesthetic, continuous variable cannot be used. R will
throw an error message "Error: A continuous variable can not be mapped to
shape". Shape aesthetic can only mapped with categorical variables.

# create a plot of hwy vs displ and map it with a categorical variable, drive
for shape aesthetic
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, shape = drv))
```



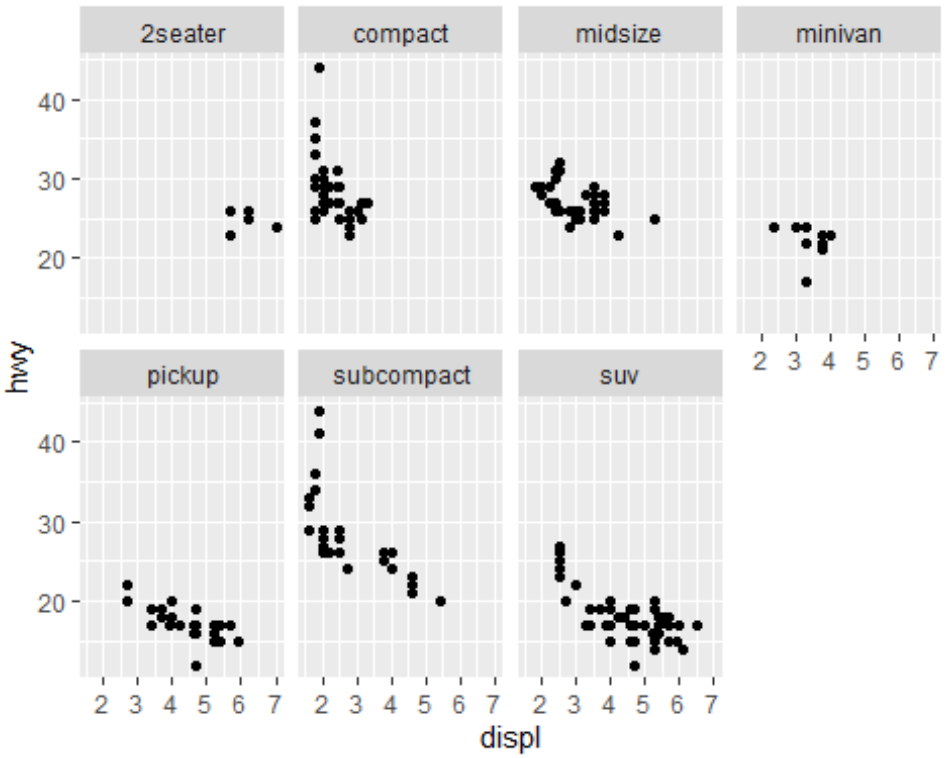

```r
# 3.3.1 Exercise #4
# create a plot of hwy vs displ and map it with a discrete variable, drv for
color and shape aesthetic
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, color = drv, shape = drv))
```




```
# If you map the same variable to multiple aesthetics, ggplot will
automatically chose a scale for the different values of that variable and
constructs a legend for them to explain the mapping of it.


# 3.3.1 Exercise #6
# create a plot of hwy vs cyl and map it with a continuous variable, disp for
color aesthetic
ggplot(data = mpg) +
  geom_point(mapping = aes(x = cyl, y = hwy, color = displ < 5))
```

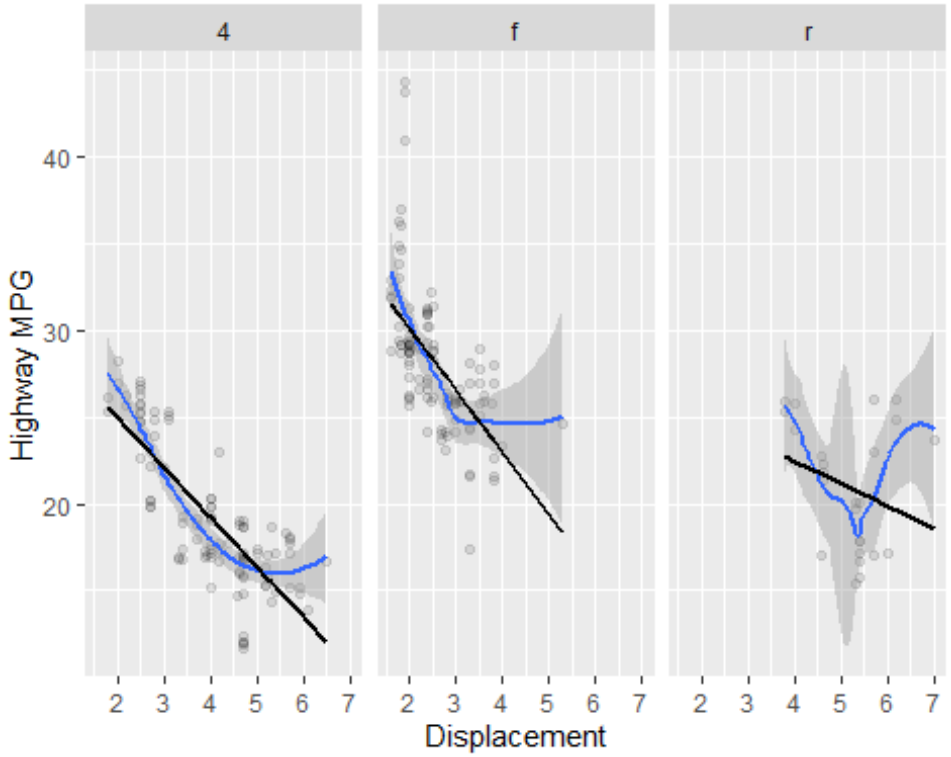



```
# As plotted, ggplot create a legend that shows two different values (TRUE
and FALSE) for the color scale which indicate if the engine displacement is
less than five litres (blue) or if it is five litres and above (red).


# 3.5.1 Exercise #4
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
  facet_wrap(~ class, nrow = 2)
```




```
# The advantages to using faceting instead of the color aesthetic are that
with faceting you will be able to see more clearly the data for each
different subset since clustered data can be divided into several subsets
data, and you can analysis an even closer relationship between variables
using facet_grid() if needed to be.
# The disadvateges to using faceting instead of the color aesthetic are you
will need an extra step to see how close are the values between each
different class and whether they do overlap as a whole.
# The balance might change with a larger dataset because having only two rows
of plots will probably less effective than having ten rows x ten columns
subsets plot or the more symmetric and/or easy to read subsets plot
(depending on the data that is presented).


# Problem 1b
# create linear model for hwy vs displ to use to create the fitted line
linModel <- lm(data=mpg, hwy~displ)
#We estimate a mixed model using lmer() from the lme4 package, with a fixed
effect of
#displ, and random intercepts and coefficients for displ by drv
mixed <- lmer(hwy ~ displ + (1+displ|drv), data=mpg)

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl =
## control$checkConv, : Model failed to converge with max|grad| = 0.00509545
## (tol = 0.002, component 1)
```

```r
#fitted values from the mixed model
mpg$fit_mix <- predict(mixed)

# plot according to figure 1
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) + # use data from mpg
dataset with mapping from displ and hwy variables
  geom_point(aes(jitter(mpg$displ,0.5), jitter(mpg$hwy,2)), alpha = 0.1) +
#add jitters to x and y values and alpa aesthetic
  geom_smooth(method ="loess") + # create the smoothed plot using loess
method
  geom_line(aes(y = fit_mix), size = 1) + # create the fitted line with size
of 1
  xlab("Displacement") + # x-label
  ylab("Highway MPG") + # y-label
  facet_wrap(~drv) # facet wrap by drv variable
```



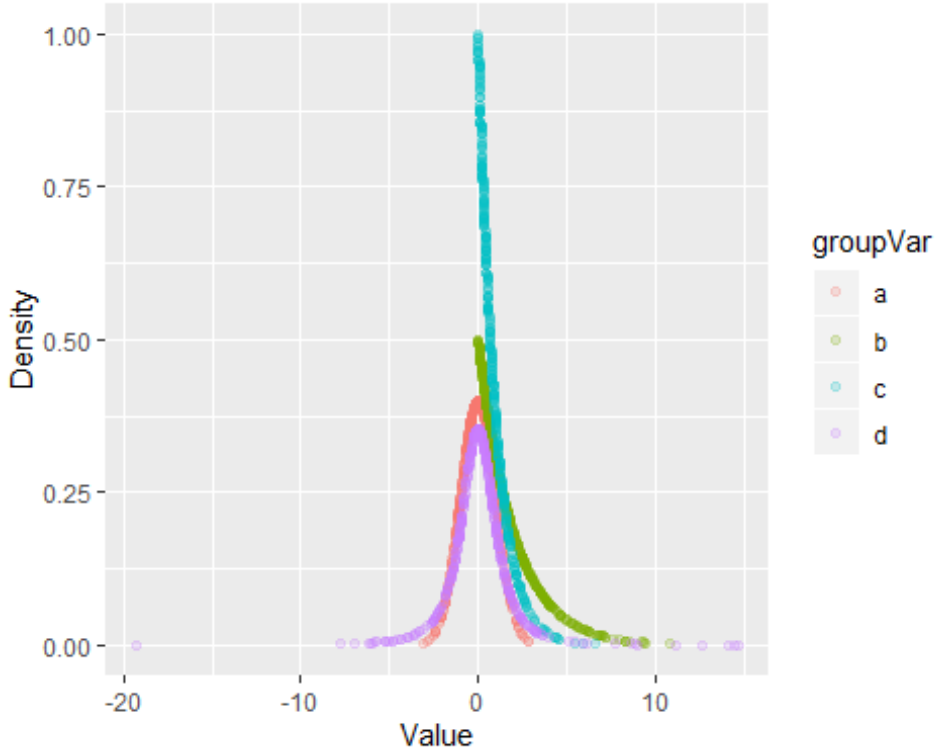

```r
#####################
# END OF PROBLEM 1 #
#####################
```

## PROBLEM 2

```r


# Note and sources for distributions in Problem 2 :
# a - normal distribution https://stat.ethz.ch/R-manual/R-
devel/library/stats/html/Normal.html
```

```r
# b - chi-square distribution https://stat.ethz.ch/R-manual/R-devel/library/stats/html/Chisquare.html
# c - exponential distribution https://stat.ethz.ch/R-manual/R-devel/library/stats/html/Exponential.html
# d - the student t distribution https://stat.ethz.ch/R-manual/R-devel/library/stats/html/TDist.html

# Problem 2
# Problem 2a
# Create data frame of n-value from 1 to 1000
N = 500
df <- data.frame(n = seq(1, N, by = 1))

# the generated random distributions for variables a, b, c, and d are used as follow:
# rnorm(n, mean = 0, sd = 1)
# rchisq(n, df, ncp = 0)
# rexp(n, rate = 1)
# rt(n, df, ncp)
# where n is # of observations, sd is vector of standard deviations, rate is vector of rates,
# df is degree of freedom, ncp is non-centrality parameter (non-negative)

df$a <- rnorm(df$n, mean = 0, sd = 1) # normal distribution
df$b <- rchisq(N, 2,ncp = 0) # chi-square distribution
df$c <- rexp(N, 1) # exponential distribution
df$d <- rt (N, 2) #the student t distribution

df <- df[,-1] # deleting the first column n so only a, b, c, and d left

# using melt from reshape2 to reshape the data frame with groupVar and value columns
df2 <- melt(df, variable.name = "groupVar", value.names = "value")

## No id variables; using all as measure variables

# Problem 2b
# calculation for the density distributions for variables a, b, c, and d are used as follow:
# dnorm(x, mean = 0, sd = 1, log = FALSE)
# dchisq(x, df, ncp = 0, log = FALSE)
# dexp(x, rate = 1, log = FALSE)
# dt(x, df, ncp, log = FALSE)
# where x is vector of quantiles, sd is vector of standard deviations, rate is vector of rates,
# df is degree of freedom, ncp is non-centrality parameter (non-negative)

df2$density[1:500] <- dnorm(df2$value[1:500], mean = 0, sd = 1)
df2$density[501:1000] <- dchisq(df2$value[501:1000], 2, ncp = 0, log = FALSE)
df2$density[1001:1500] <- dexp(df2$value[1001:1500], rate = 1, log = FALSE)
```

```
df2$density[1501:2000] <- dt(df2$value[1501:2000],2, log = FALSE)

# use data from df2 data frame to create a plot of density vs. value
ggplot(data = df2, mapping = aes(x = value, y = density)) +
  geom_point(aes(color = groupVar), alpha = 0.2) +
  xlab("Value") +
  ylab("Density")
```

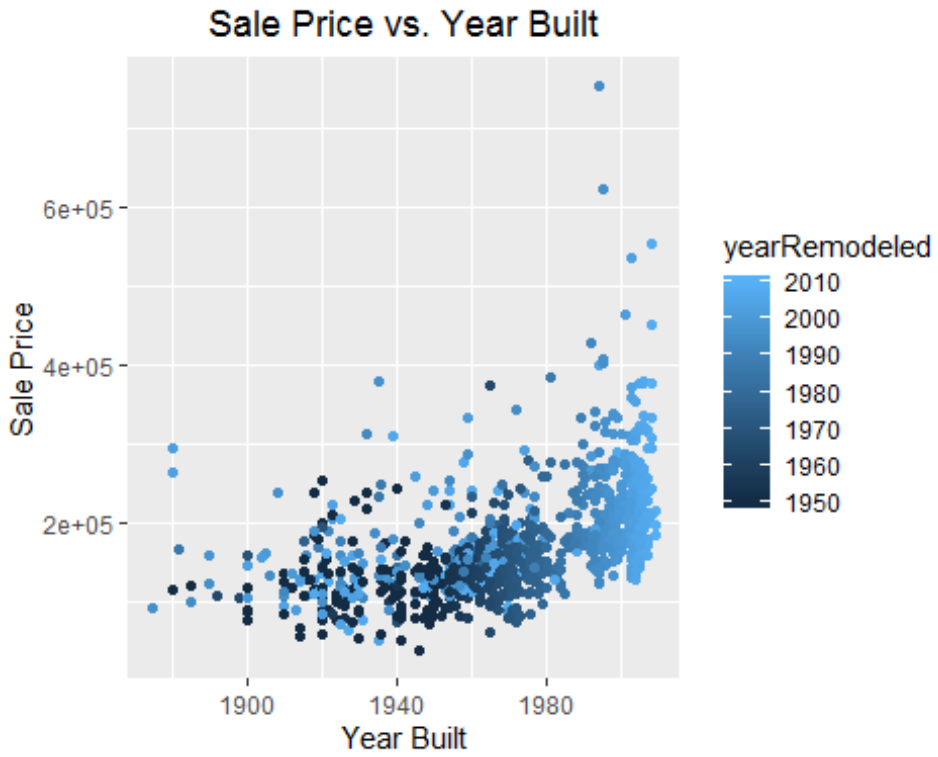



```
####################
# END OF PROBLEM 2 #
####################
```

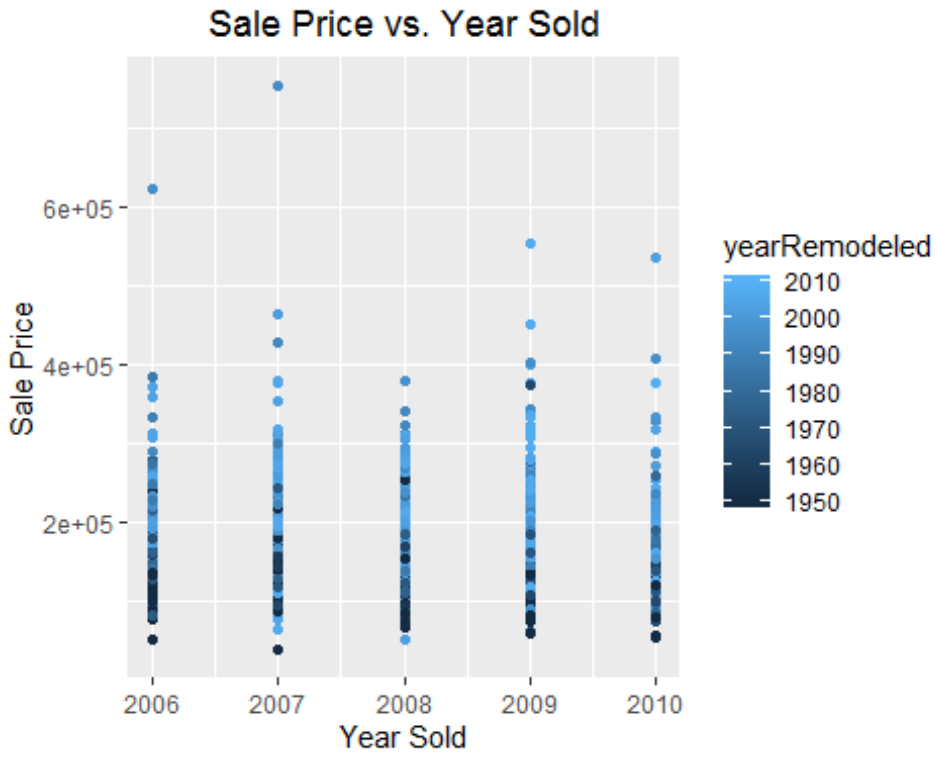
## PROBLEM 3

```
# Problem 3
# read the excel file
housingData <- read.csv(file="housingData.csv", header=TRUE, sep=",")
#View (housingData)
#summary(housingData)

yearRemodeled <- housingData$YearRemodAdd

# create plot for Sale Price vs Year Built with year remodeled color
aesthetic
```

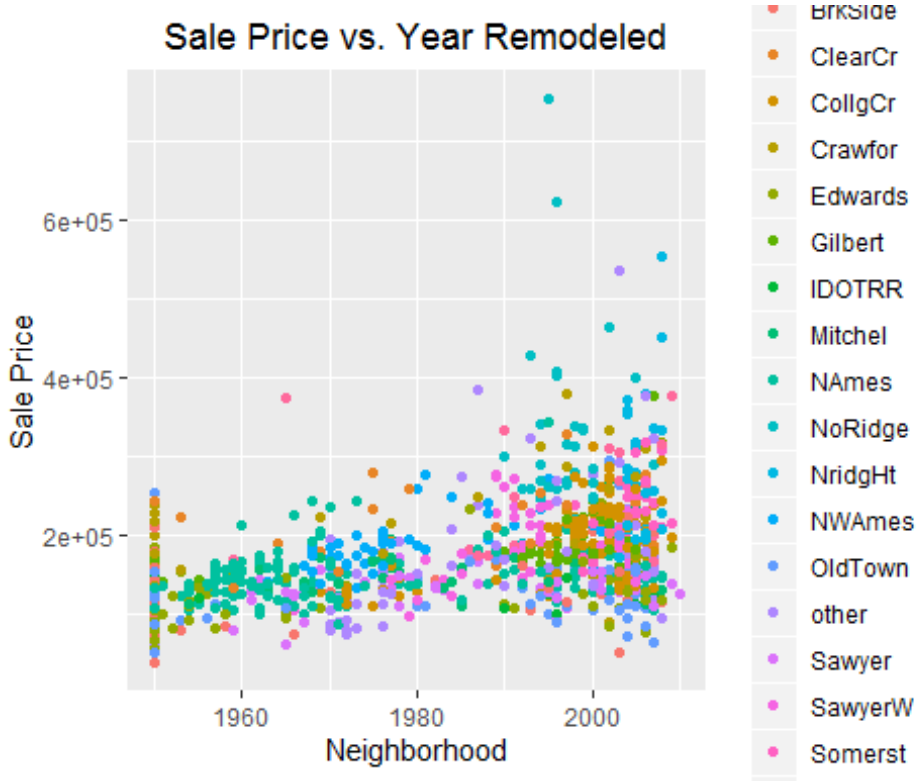
```r
ggplot(data = housingData, mapping = aes(x = YearBuilt, y = SalePrice)) +
  geom_point(mapping = aes(color = yearRemodeled)) +
  theme(plot.title = element_text(hjust = 0.5))+
  ggtitle ("Sale Price vs. Year Built") +
  xlab("Year Built") +
  ylab("Sale Price")
```



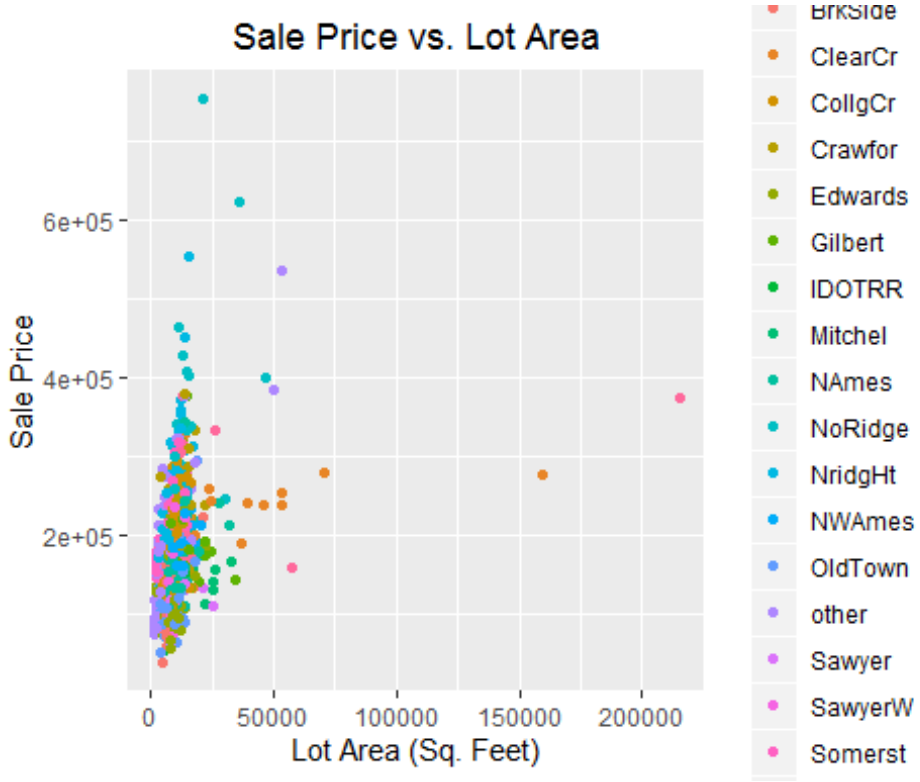

```r
# create plot for Sale Price vs Year Sold with year remodeled color aesthetic
ggplot(data = housingData, mapping = aes(x = YrSold, y = SalePrice)) +
  geom_point(mapping = aes(color = yearRemodeled)) +
  theme(plot.title = element_text(hjust = 0.5))+
  ggtitle ("Sale Price vs. Year Sold") +
  xlab("Year Sold") +
  ylab("Sale Price")
```



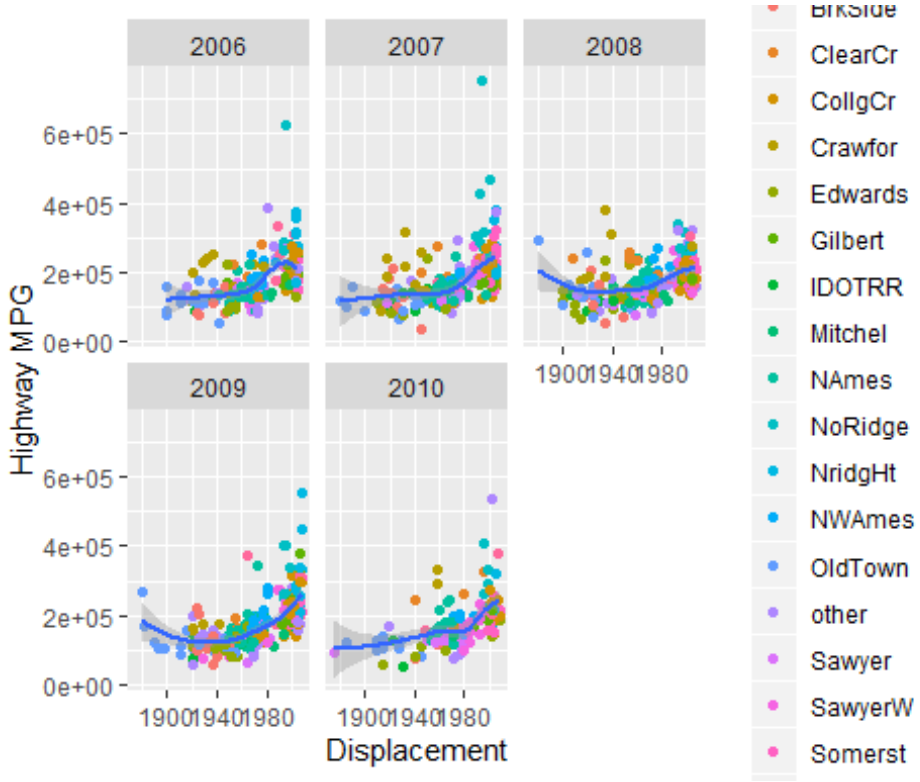

```r
# create plot for Sale Price vs year remodeled with neighborhood color aesthetic
ggplot(data = housingData, mapping = aes(x = yearRemodeled, y = SalePrice)) +
  geom_point(mapping = aes(color = Neighborhood)) +
  theme(plot.title = element_text(hjust = 0.5))+
  ggtitle ("Sale Price vs. Year Remodeled") +
  xlab("Neighborhood") +
  ylab("Sale Price")
```




```r
# create plot for Sale Price vs Lot Area with neighborhood color aesthetic
ggplot(data = housingData, mapping = aes(x = LotArea, y = SalePrice)) +
  geom_point(mapping = aes(color = Neighborhood)) +
  theme(plot.title = element_text(hjust = 0.5))+
  ggtitle ("Sale Price vs. Lot Area") +
  xlab("Lot Area (Sq. Feet)") +
  ylab("Sale Price")
```



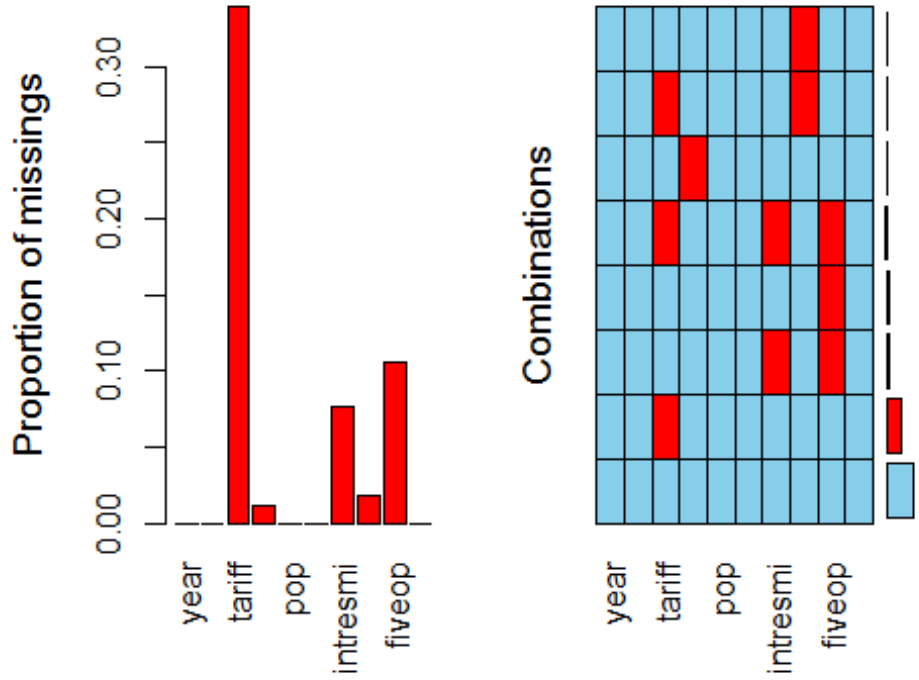

```r
ggplot(data = housingData, mapping = aes(x = YearBuilt, y = SalePrice)) +
  geom_point(mapping = aes(color = Neighborhood)) +
  geom_smooth(method ="loess") + # create the smoothed plot using loess
method
  xlab("Displacement") + # x-label
  ylab("Highway MPG") + # y-label
  facet_wrap(~YrSold) # facet wrap by drv variable
```




```
#####################
# END OF PROBLEM 3 #
#####################
```

## PROBLEM 4

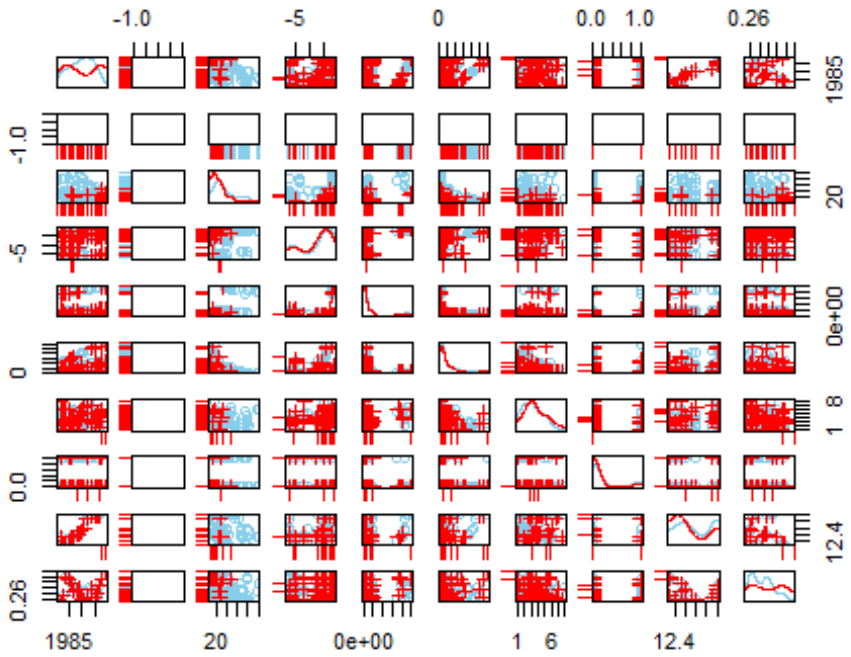
```
# Problem 4
# load freetrade data from Amelia package
data ("freetrade",package = "Amelia")

# get the overall information that is missing on freetrade data frame using
VIM's aggr function
missingInfo <- aggr(freetrade)
```



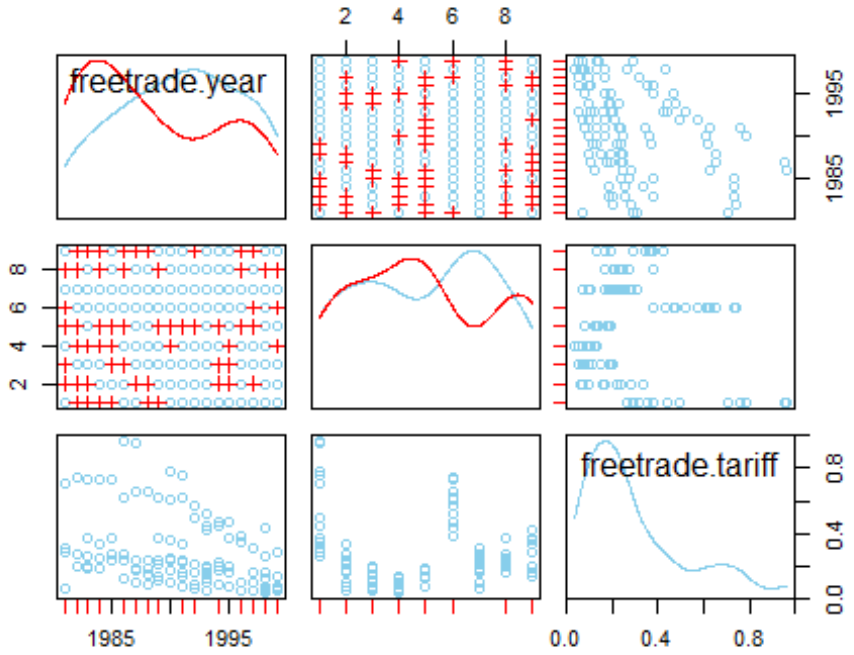

```
summary(missingInfo) # summary of missing information for each variables
```

```
##
##  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
```

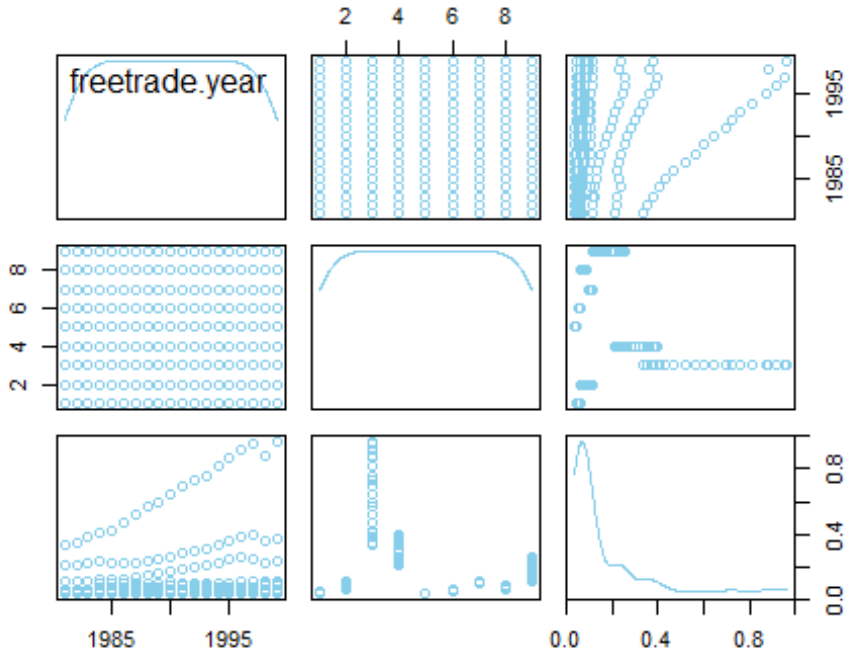
```
# scattmatrixMiss that shows the correlations of all variables in freetrade
data frame
scattmatrixMiss(freetrade)

## Warning in data.matrix(z): NAs introduced by coercion
```



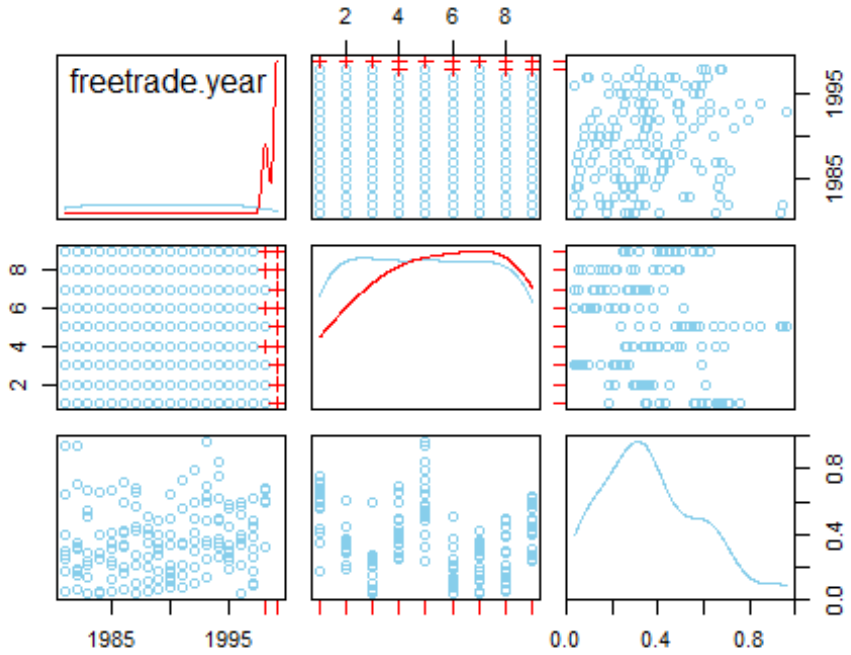

```
# from summary, we know that there are missing data from tariff, gdp.pc,
intresmi, signed, and gdp.pc variables
# so, now, we look at scattmatrixMiss plot for the year and country against
each of those variable
scattmatrixMiss(data.frame(freetrade$year, freetrade$country,
freetrade$tariff))
```



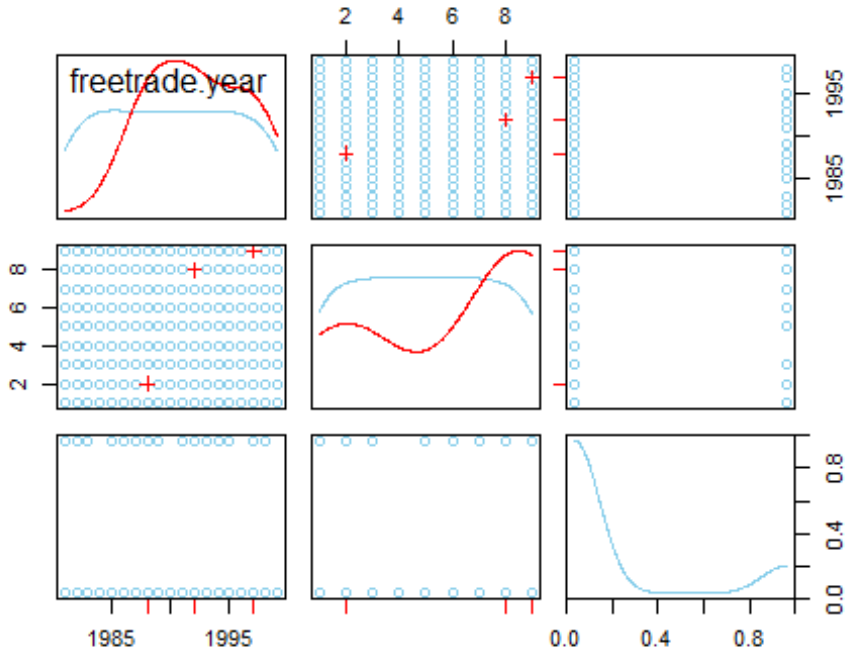

```
scattmatrixMiss(data.frame(freetrade$year, freetrade$country,
freetrade$gdp.pc))
```



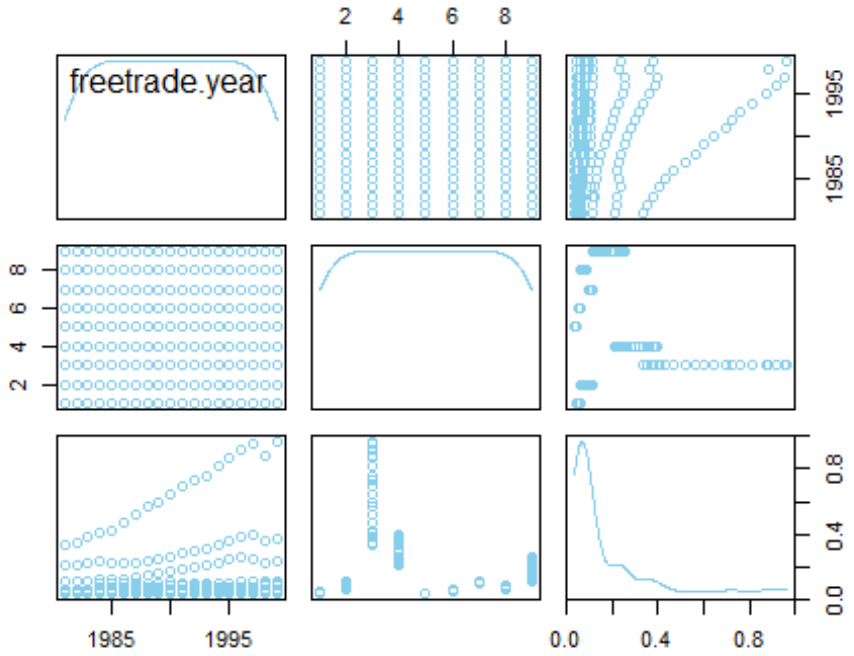
```
scattmatrixMiss(data.frame(freetrade$year, freetrade$country,
freetrade$intresmi))
```



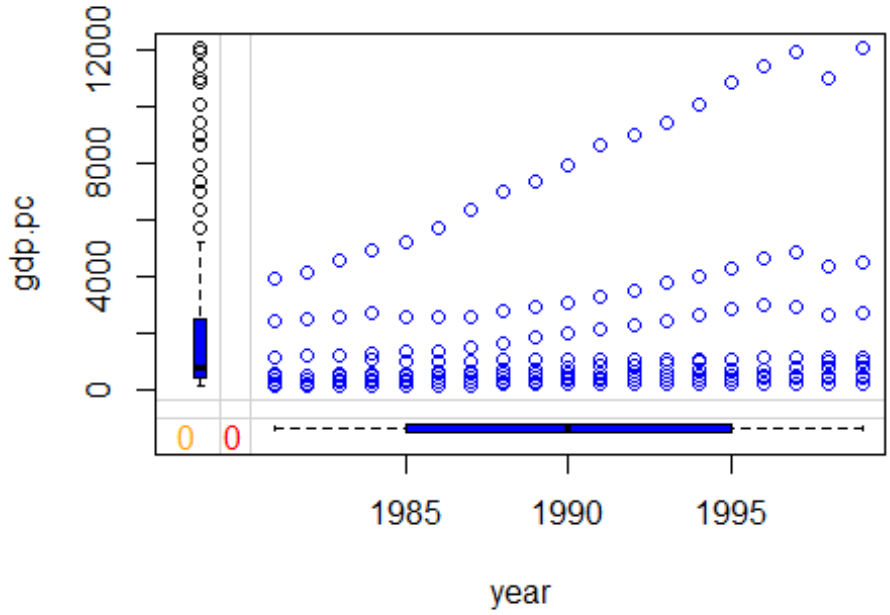

```
scattmatrixMiss(data.frame(freetrade$year, freetrade$country,
freetrade$signed))
```



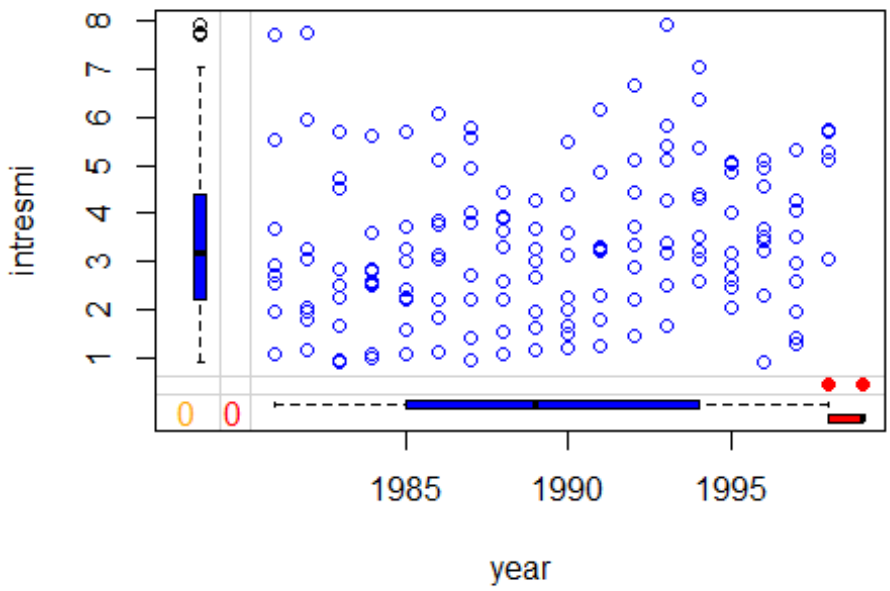

```
scattmatrixMiss(data.frame(freetrade$year, freetrade$country,
freetrade$gdp.pc))
```



```r
# we can also use the marginplot to see the relationship between two
variables from freetrade data frame, such as, below:
marginplot(freetrade[c("year","gdp.pc")], col = c("blue", "red", "orange")) #
relation between year and gdp.pc values
```




```r
marginplot(freetrade[c("year","intresmi")], col = c("blue", "red", "orange"))
# relation between year and intresmi values
```




```
# Problem 4b
# create country and tariff variables
country <- freetrade$country
tariff <- freetrade$tariff

chisq.test(country, tariff) # do chi-square test
```

```
## Warning in chisq.test(country, tariff): Chi-squared approximation may be
## incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  country and tariff
## X-squared = 831.96, df = 736, p-value = 0.007819
```

```
# Exclude Nepal from the freetrade data
freetrade.woNepal <- freetrade[!(freetrade$country == "Nepal"),]
# update country and tariff variables
country <- freetrade.woNepal$country
tariff <- freetrade.woNepal$tariff

chisq.test(country, tariff) # do chi-square test
```

```
## Warning in chisq.test(country, tariff): Chi-squared approximation may be
## incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  country and tariff
## X-squared = 684.79, df = 602, p-value = 0.01063
```

```
# Exclude Philippines from the freetrade data
freetrade.woPhilippines <- freetrade[!(freetrade$country == "Philippines"),]
# update country, and tariff variables
country <- freetrade.woPhilippines$country
tariff <- freetrade.woPhilippines$tariff

chisq.test(country, tariff) # do chi-square test
```

```
## Warning in chisq.test(country, tariff): Chi-squared approximation may be
## incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  country and tariff
## X-squared = 639.33, df = 574, p-value = 0.03012
```

```
# From the test, you can see that tariff is independent with the country
variable. If you obsevered each chi-square test, they are not related and it
also shows from the p-value that is less than the significance level alpha of
0.05. Excluding Nepal or Philippines changes the p-value of the chi-square
test because Nepal have significant numbers of missing tariff while
Philippines have all their tariff values.

####################
# END OF PROBLEM 4 #
####################
```