Homework 2

Brandon Lucio

8/26/2021

From Textbook

Question 1 Indicate whether we would generally expect the performance of a **flexible** statistical learning method to be better or worse than an **inflexible** method. Justify your answer.

a) The sample size n is extremely large, and the number of predictors p is small.

- Since we have an extremely large sample size we can estimate the small number of predictors. The flexible statistical learning method can be used.

b) The number of predictors p is extremely large, and the number of observations n is small.

- With the large number of predictors using a flexible model would not be better than inflexible because the error would be larger

c) The relationship between the predictors and response is highly non-linear.

- Since our relationship is non-linear, it would be best to use a flexible model, we would get a non-line fit

d) The variance of the error terms, i.e. $\sigma^2 = Var(\epsilon)$, is extremely high.

- With a higher variance in the error term using the flexible model would risk more error from just random noise. Thus, using an inflexible method would be best.

Question 2 Explain whether each scenario is a classification or regression problem, and indicate whether we are most interested in inference or prediction. Finally, provide n and p.

a) We collect a set of data on the top 500 firms in the US. For each firm we record profit, number of employees, industry and the CEO salary. We are interested in understanding which factors affect CEO salary.

- This is more of a regression problem as we have more quantitative predictors. The overall interest is in the inference of what factors affect the CEO salary. n = 500 , p = 3

b) We are considering launching a new product and wish to know whether it will be a success or a failure. We collect data on 20 similar products that were previously launched. For each product we have recorded whether it was a success or failure, price charged for the product, marketing budget, competition price, and ten other variables.

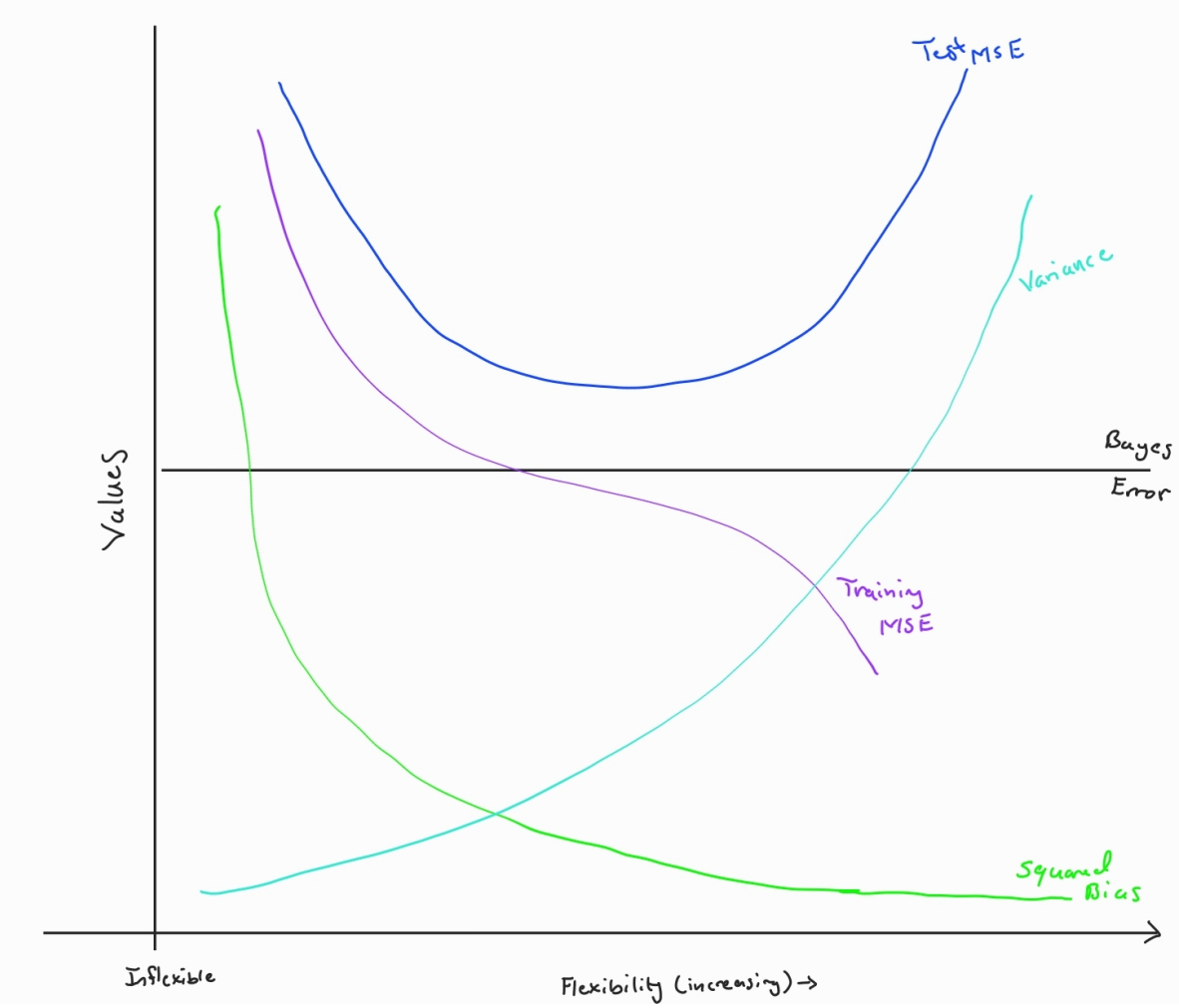
- Here we are solving a classification problem but overall more interested in the prediction of which class we want in the end. n = 20, p = 13

c) We are interested in predicting the % change in the USD/Euro exchange rate in relation to the weekly changes in the world stock markets. Hence we collect weekly data for all of 2012. For each week we record the % change in the British market, and the % change in the German market.

- This is a regression problem as we are trying to predict the % change. n = 52, p = 3

Question 3 We now revisit the bias-variance decomposition.

a) Provide a sketch of typical (squared) bias, variance, training, error, test error, and Bayes (or irreducible) error curves, on a single plot, as we go from less flexible statistical learning methods towards more flexible approaches, The x-axis should represent the amount of flexibility in the method , and the y-axis should represent the values for each curve. There should be five curves. Make sure to label each one.



b) Explain why each of the five curves has the shape displayed in part (a)

- As we increase in flexibility the model then we are using more and more parameters which would increase the variance of each error. At the same time the Bias would decrease as the flexibility goes up since we are matching the data closer and closer. Now the test MSE has an interesting U-shape. it starts off high and then goes down as flexibility increases but then at a point it starts to go up again since we begin to over estimate and gain more error. The training MSE is usually always lower than the Test MSE and will consistently decrease as the Flexibility increases.

Question 10 This exercise involves the **Boston** housing data set.

a) To begin, load in the **Boston** data set. The **Boston** data set is part of the **Mass** library in **R**. How many rows are in this data set? How many columns? What do the rows and columns represent?

```
#How many rows?
sprintf('The number of rows: %d', nrow(Boston))
```

```
## [1] "The number of rows: 506"
```

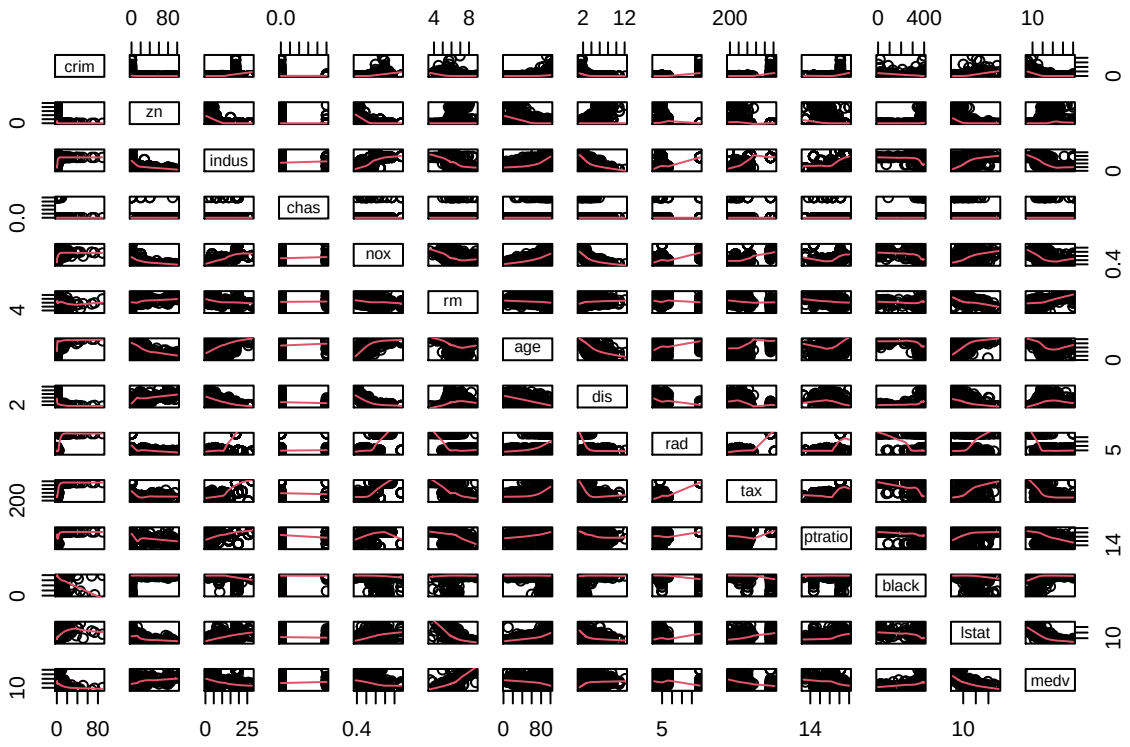
```
#How many columns?
sprintf('The number of columns: %d', ncol(Boston))
```

```
## [1] "The number of columns: 14"
```

```
#What do the rows and columns represent?
#Housing values in suburbs of Boston
```

b) Make some pairwise scatter-plots of the predictors(columns) in this data set. Describe your findings.

```
pairs(Boston, panel = panel.smooth)
```



We can see that there are quite a few trends to be seen here. lstat and medv have some type of relationship. But as age changes rm tends to stay relatively the same. lstat also tends to increase as age goes up.

c) Are any of the predictors associated with per capita crime rate? If so, explain the relationship.

- Yes, we can see that as age increases we can see higher crime rates. Also homes with a lower medv comes higher crime rates.

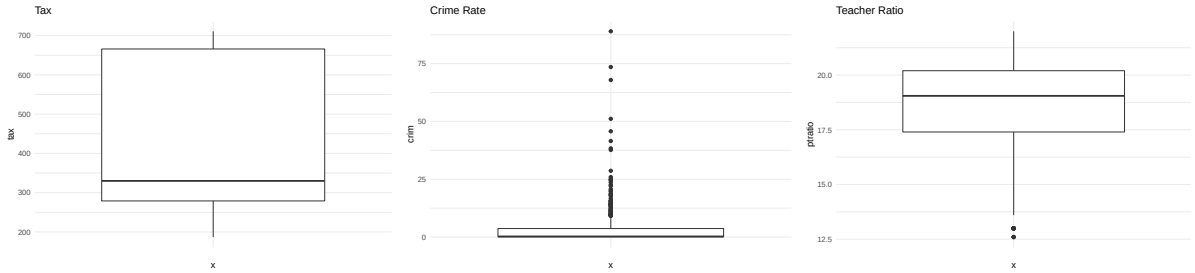
d) Do any of the suburbs of Boston appear to have particularly high crime rates? Tax rates? Pupil-teacher rations? Comment on the range of each predictor.

The best way to get a rough idea of high rates (outliers) is with some type of visual

```
# Here we use box plots
ggplot(Boston, mapping = aes(x = '', y = tax)) +
  geom_boxplot() +
  theme_minimal() +
  ggtitle('Tax')

ggplot(Boston, mapping = aes(x = '', y = crim)) +
  geom_boxplot() +
  theme_minimal() +
  ggtitle('Crime Rate')

ggplot(Boston, mapping = aes(x = '', y = ptratio)) +
  geom_boxplot() +
  theme_minimal() +
  ggtitle('Teacher Ratio')
```




```
# You can also pull out potential outliers index with the $out function
# From the images we see that tax does not have any outliers so we omit it
crime_out <- boxplot.stats(Boston$crim)$out
pt_out <- boxplot.stats(Boston$ptratio)$out

# Now using the which() function we can get the exact rows to verify
# We comment out to avoid printing
# Boston[which(Boston$crim %in% c(crime_out)),]
# Boston[which(Boston$ptratio %in% c(pt_out)),]
```

e) How many of the suburbs in this data set bound the Charles river?

```
# Start by filtering data for the indicator variable
# Then getting the nrows in that new dataframe
sprintf('There are %d suburbs bound by the Charles river',nrow(filter(Boston, chas == 1)))
```

```
## [1] "There are 35 suburbs bound by the Charles river"
```

f) What is the median pupil-teacher ratio among the towns in this data set?

```
median(Boston$ptratio)
```

```
## [1] 19.05
```

g) Which suburb of Boston has lowest median value of owner-occupied homes? What are the values of the other predictors for that suburb, and how do those values compare to the overall ranges for those predictors? Comment on your findings.

```
Boston[which(Boston$medv == min(Boston$medv)),]
```

```
##        crim zn indus chas   nox    rm age    dis rad tax ptratio  black lstat
## 399 38.3518  0  18.1    0 0.693 5.453 100 1.4896  24 666    20.2 396.90 30.59
## 406 67.9208  0  18.1    0 0.693 5.683 100 1.4254  24 666    20.2 384.97 22.98
##     medv
## 399    5
## 406    5
```

- These suburbs also have higher crime rates, tax, ptratio, and black being at the 3rd quartile. They also have the max age of the suburbs. This shows that the homes with the lowest values are the oldest, suggesting that age of the home has a big effect on its value.

h) In this data set, how many of the suburbs average more than seven rooms per dwelling? More than eight rooms per dwelling? Comment on the suburbs that average more than eight rooms per dwelling.

```
sprintf('There are %d subrbs with more than 7 rooms.', nrow(filter(Boston, rm > 7)))
```

```
## [1] "There are 64 subrbs with more than 7 rooms."
```

```
sprintf('There are %d subrbs with more than 8 rooms.', nrow(filter(Boston, rm > 8)))
```

```
## [1] "There are 13 subrbs with more than 8 rooms."
```

```
# Boston[which(Boston$rm > 8),]
```

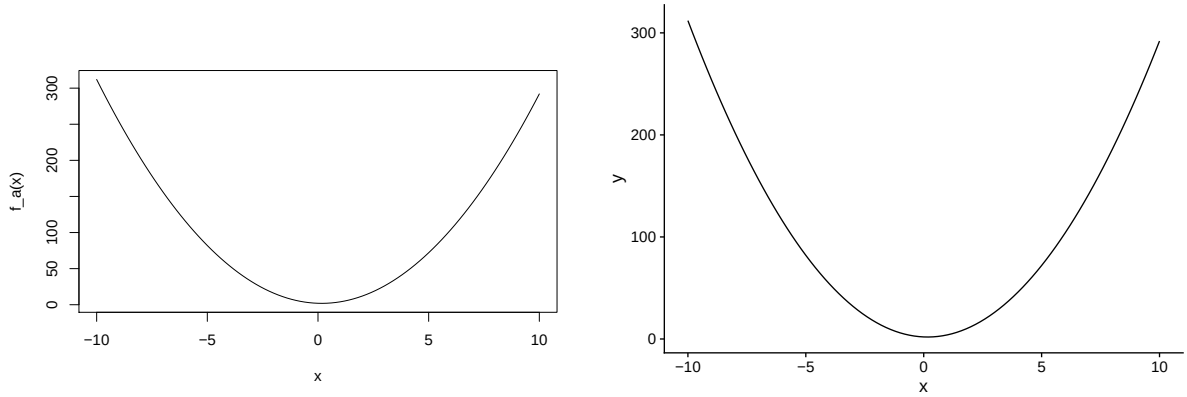
- The crime rate for these dwellings is less then the mean crime rate of all suburbs. We can also see that tax and ptratio are also lower than the mean value per suburb.

Not from text book

1. Suppose you have the following functions. Write an R function to each of them and then make a plot for each one

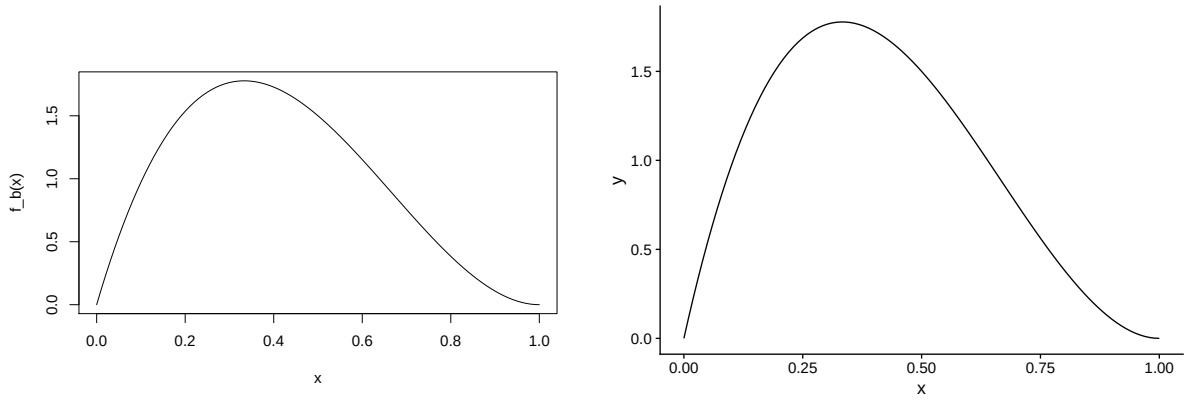
   a) $f(x) = 2 + 3x^2 - x$, in the range of $(-10, 10)$.

```
# we can do this a couple of different ways
# First, using the curve method
f_a <- function(x){ return(2 + 3*x^2 - x)}
curve(expr = f_a, from = -10, to = 10)
# Second, using ggplot
ggplot(data.frame(x = c(-10,10)), aes(x = x )) +
  stat_function(fun = f_a) +
  theme_cowplot()
```



b) $f(x) = \frac{1}{B(2,3)} x(1-x)^2$, for $0 < x < 1$, where $B(\alpha, \beta)$ is a beta function, in the range of $(0, 1)$

```
#Similarly as part a
f_b <- function(x){ return((1/ beta(2,3))* x * (1-x)^2)}
curve(expr = f_b, from = 0, to = 1)
ggplot(data.frame(x = c(0,1) ), aes(x = x )) +
  stat_function(fun = f_b) +
  theme_cowplot()
```



2. Create a data frame with the following command
   >set.seed(123)
   >df = data.frame(x1 = rnorm(10), x2 = rpois(10,3), x3 = runif(10,-1,1), x4= rgamma(10,2,3))

a) Obtain the means of all columns using apply

```
set.seed(123)
df = data.frame(x1 = rnorm(10), x2 = rpois(10,3), x3 = runif(10,-1,1), x4= rgamma(10,2,3))
apply(df, 2, mean)
```

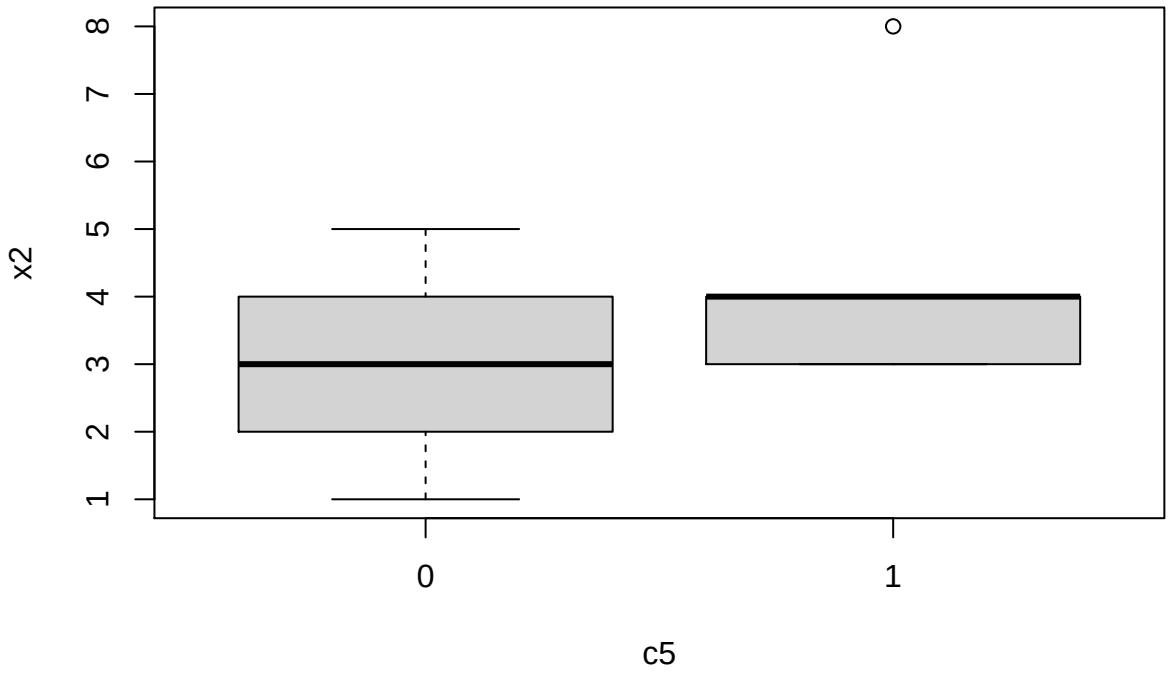
```
##         x1         x2         x3         x4
## 0.07462564 3.70000000 0.07571598 0.41745145
```

b) Add another column, named c5, which is 1 for all $x1 \geq 0$ and 0 otherwise.

```
# Creating new column with mutate
df = mutate(df,c5 = ifelse(x1 >= 0 , 1 , 0 ))
```

c) Draw boxplots of $x2$ for different c5 values

```
boxplot(x2 ~ c5, data = df)
```



3. In this problem, you search the internet using "auto_mpg dataset" to find an automobile mpg data. Most likely you would be able to find it in either at "kaggle", or "UCI Machine Learning Repository" Note that you do not use the original data.

a) Create a new project and import the data into your RStudio. Show the first 3 rows of the data by using *head* command.

```
auto_mpg <- read.csv('c:/users/lucio/OneDrive/School/STA-6923-Data_Mining/Homework/Homework_2/auto-
head(auto_mpg, 3)
```

```
##   mpg cylinders displacement horsepower weight acceleration model.year origin
## 1  18         8          307        130   3504         12.0         70      1
## 2  15         8          350        165   3693         11.5         70      1
## 3  18         8          318        150   3436         11.0         70      1
##                   car.name
## 1 chevrolet chevelle malibu
## 2         buick skylark 320
## 3        plymouth satellite
```

b) Check the classes of your variable by using the *sapply* command. What are the classes of **horsepower**, **model_year** and **name**?

```
sapply(auto_mpg, class)
```

```
##          mpg    cylinders displacement   horsepower       weight acceleration
##    "numeric"    "integer"    "numeric"  "character"    "integer"    "numeric"
##   model.year       origin     car.name
##    "integer"    "integer"  "character"
```

- We can see that Horsepower: character, model_year: integer, name: character

c) From the original data, **horsepower** is supposed to be numeric. Do you see any problem? in R, any missing value is labeled as "NA". Try to clean the data(actually the **horsepower** column) and replace any character to "NA".

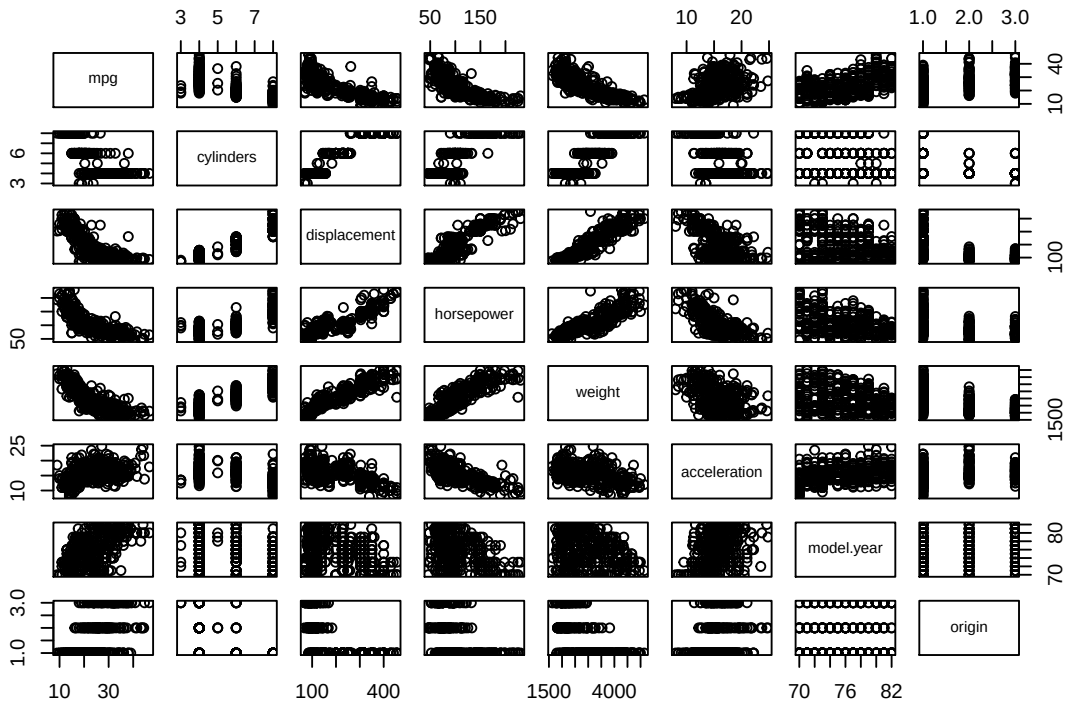
```
auto_mpg$horsepower <-ifelse(auto_mpg$horsepower == '?', NA,auto_mpg$horsepower)
auto_mpg$horsepower <- as.numeric(auto_mpg$horsepower)
```

d) Do a summary analysis of the data (numeric variables) by checking each variable's range, extreme values, mean, median, standard deviation, etc. Check the correlations among the variables and plot pairwise graph between each two variables by using command *pairs*.

```
summary(auto_mpg[which(sapply(auto_mpg, is.numeric))])
```

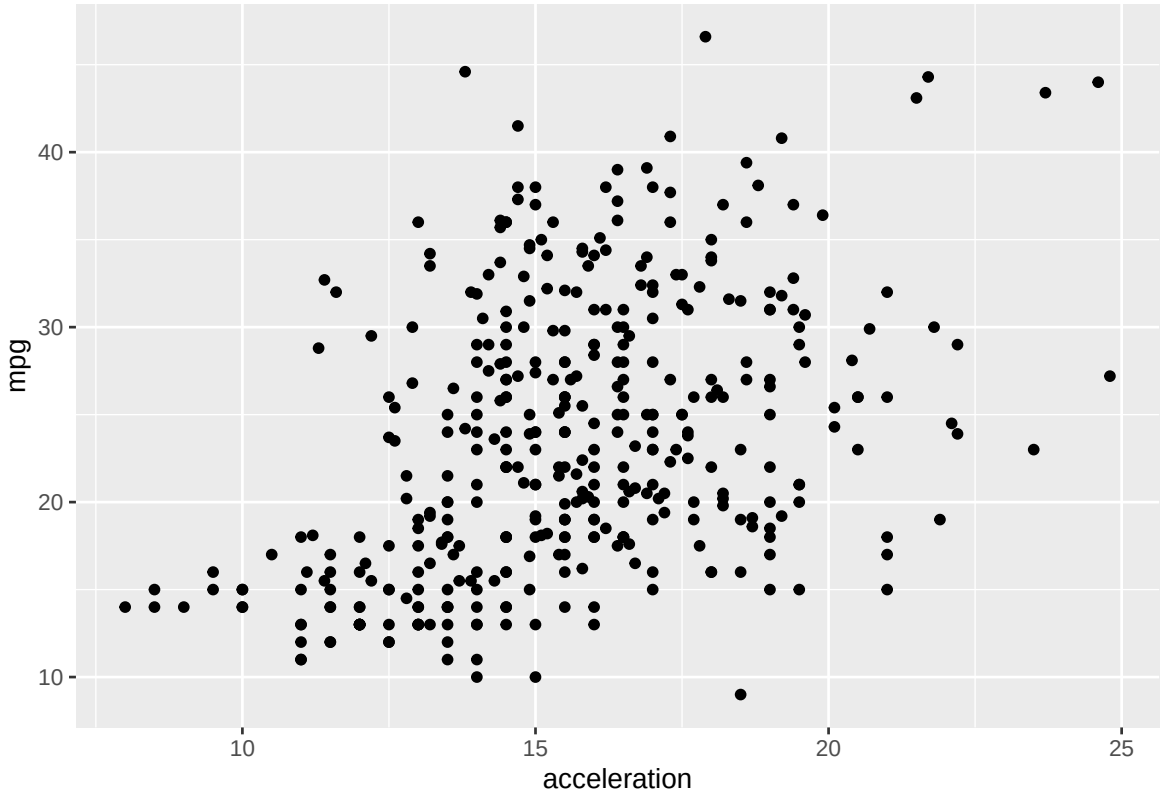
```
##       mpg          cylinders      displacement     horsepower        weight
##  Min.   : 9.00   Min.   :3.000   Min.   : 68.0   Min.   : 46.0   Min.   :1613
##  1st Qu.:17.50   1st Qu.:4.000   1st Qu.:104.2   1st Qu.: 75.0   1st Qu.:2224
##  Median :23.00   Median :4.000   Median :148.5   Median : 93.5   Median :2804
##  Mean   :23.51   Mean   :5.455   Mean   :193.4   Mean   :104.5   Mean   :2970
##  3rd Qu.:29.00   3rd Qu.:8.000   3rd Qu.:262.0   3rd Qu.:126.0   3rd Qu.:3608
##  Max.   :46.60   Max.   :8.000   Max.   :455.0   Max.   :230.0   Max.   :5140
##                                                  NA's   :6
##   acceleration     model.year        origin
##  Min.   : 8.00   Min.   :70.00   Min.   :1.000
##  1st Qu.:13.82   1st Qu.:73.00   1st Qu.:1.000
##  Median :15.50   Median :76.00   Median :1.000
##  Mean   :15.57   Mean   :76.01   Mean   :1.573
##  3rd Qu.:17.18   3rd Qu.:79.00   3rd Qu.:2.000
##  Max.   :24.80   Max.   :82.00   Max.   :3.000
##
```

```
pairs(auto_mpg[which(sapply(auto_mpg, is.numeric))])
```



e) Create a two-variable data, with only *acceleration* and *mpg* in it. Make a scatter plot between them by using *mpg* as y-axis variable. Do you see strong correlation between the two variables? In addition, what is a correlation?

```
a_mpg <- dplyr::select(auto_mpg, mpg, acceleration)
a_mpg %>%
  ggplot(mapping = aes(x = acceleration, y = mpg)) +
  geom_point()
```



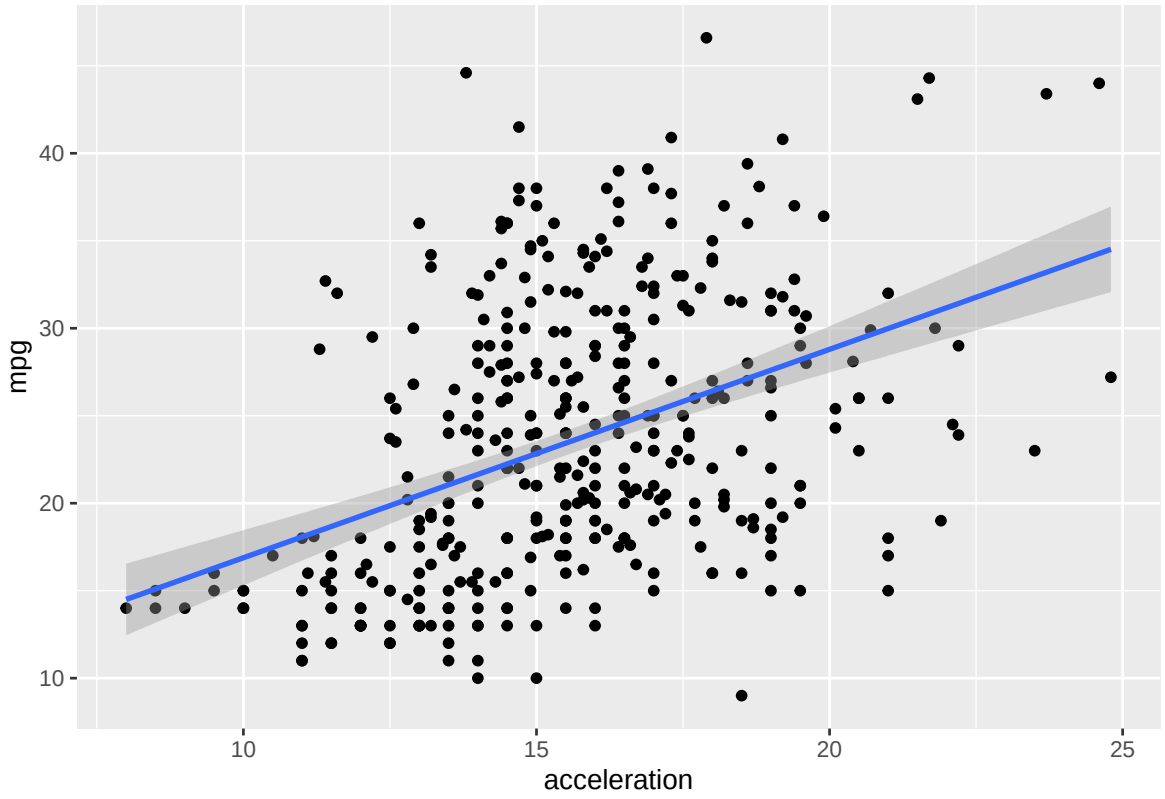
- Correlation refers to the measure of the linear relationship between X and Y. From the graph I do not see a strong correlation between acceleration and mpg.

f) Run a linear regression between the variables in part e) and use *mpg* as the response variable, *acceleration* as the predictor. What's your conclusion for this analysis? In addition, add the regression line to the plot in part e).

```
lm.model = lm(mpg~ acceleration, data = a_mpg)
summary(lm.model)
```

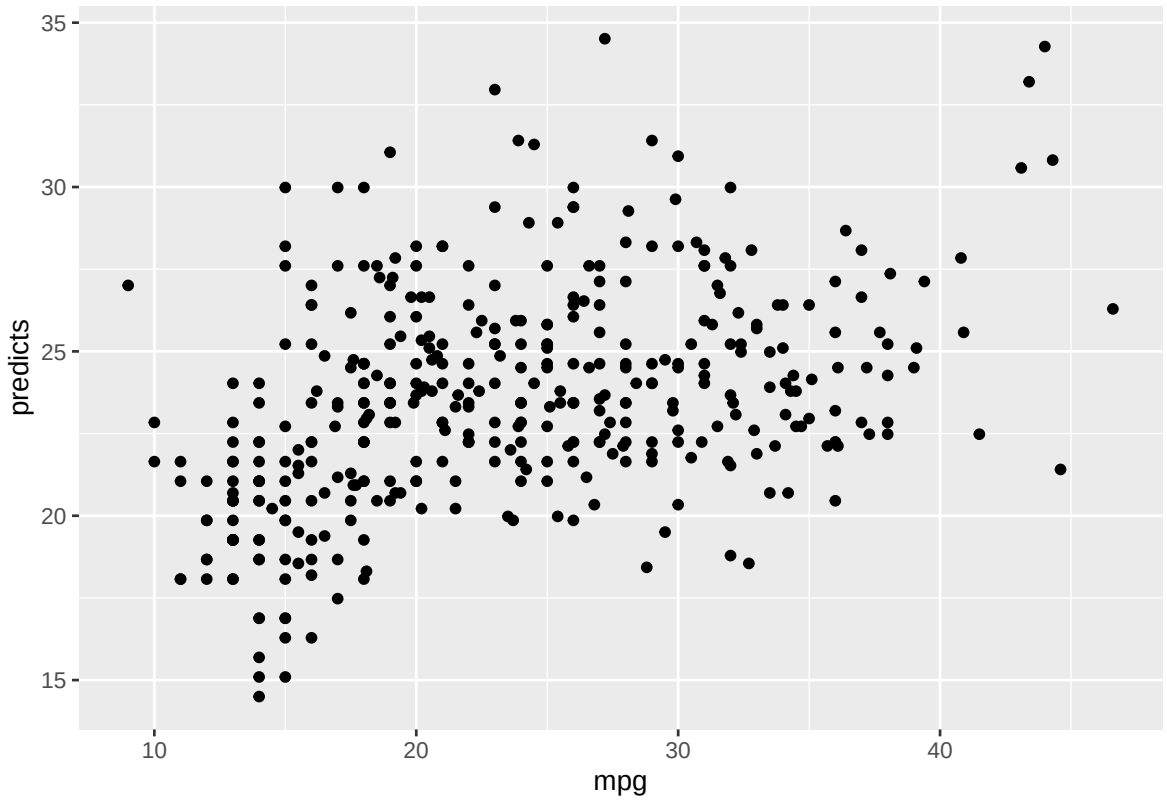
```
##
## Call:
## lm(formula = mpg ~ acceleration, data = a_mpg)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -18.007  -5.636  -1.242   4.758  23.192
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    4.9698     2.0432   2.432   0.0154 *
## acceleration   1.1912     0.1292   9.217   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.101 on 396 degrees of freedom
## Multiple R-squared:  0.1766, Adjusted R-squared:  0.1746
## F-statistic: 84.96 on 1 and 396 DF,  p-value: < 2.2e-16
```

```
a_mpg %>%
  ggplot(aes(x = acceleration, y = mpg)) +
  geom_point() +
  geom_smooth( formula = y ~ x ,method = 'lm')
```



g) Using the regression in part f), make a prediction of *mpg* for each *acceleration* values in the data. Draw a scatter plot between the original *mpg* and the predicted *mpg*. Comment.

```
a_mpg$predicts <- predict(lm.model,  , newdata =data.frame(a_mpg['acceleration']))
a_mpg %>%
  ggplot(aes(x = mpg , y =predicts)) +
  geom_point()
```



h) **MSE** is an abbreviation for **Mean Squared Error**, which is the average of the squared differences between the estimated and the truth value (or observed value). For the results in part g), treat the original *mpg* as true values, and predicted *mpg* as estimates. Find the **MSE** of this prediction.

```
# We have two ways to calculate the MSE
print( mean((a_mpg$mpg-a_mpg$predicts)^2) )
```

```
## [1] 50.17219
```

```
print(mean(lm.model$residuals^2))
```

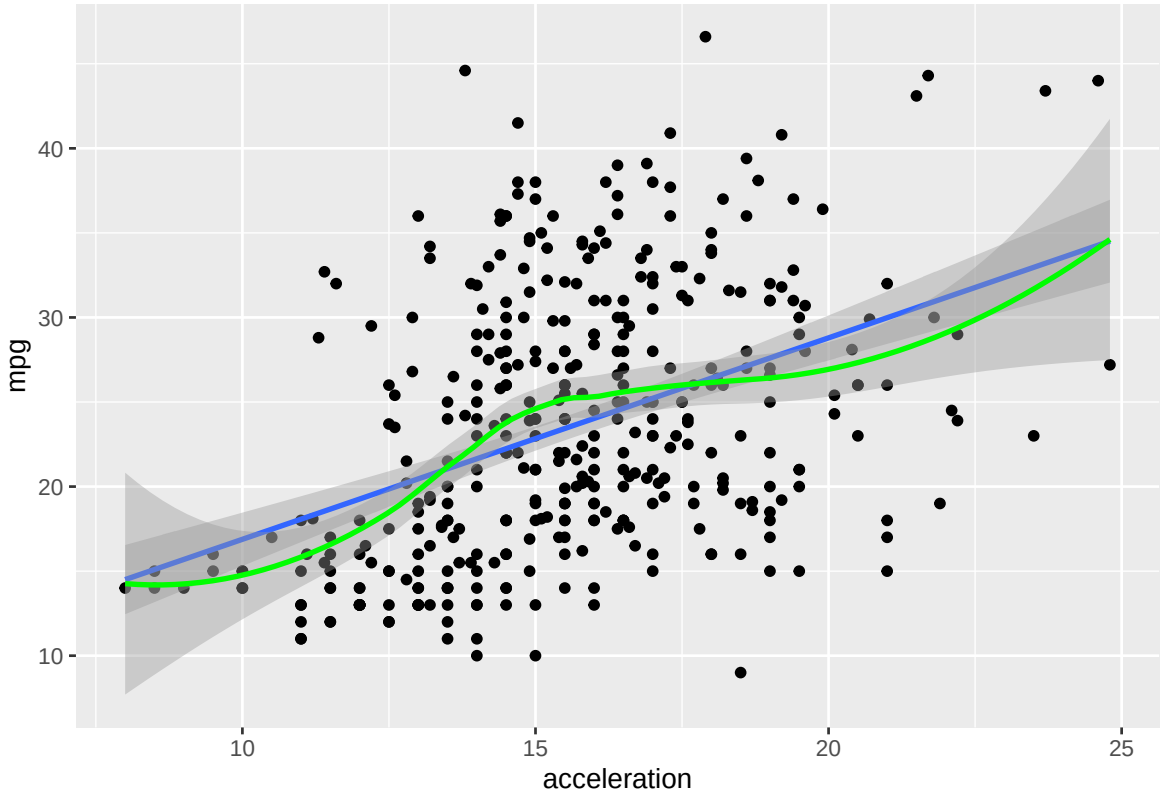
```
## [1] 50.17219
```

i) The Locally Estimated Scatterplot Smoothing, or LOESS, is a moving regression to fit data more smoothly. Use the **loess** function in R to make a LOESS regression between *acceleration* and *mpg*. What is the MSE of the prediction in this case? Comment, including the results in part h). Add the LOESS regression line into the graph you drew for part h)

```
loess.model <- loess(formula = mpg~ acceleration, data = a_mpg )
print(mean(loess.model$residuals^2))
```

```
## [1] 47.67229
```

```
a_mpg %>%
  ggplot(aes(x = acceleration, y = mpg)) +
  geom_point() +
  geom_smooth( formula = y ~ x ,method = 'lm') +
  geom_smooth( formula = y ~ x ,method = 'loess', color = 'green' )
```



j) Using **summary** to check the result of your LOESS regression in part i). The **span**, in the *Control settings*, is a smoothing parameter. Now try to run another (or more if you like) LOESS regression by adding **span** option in your loess command. Comment on the results

```
for (i in seq(.1,1,.05)){
    model <- loess(mpg ~ acceleration, data = a_mpg, span = i)
    print(paste("Span:", sprintf("%.2f",i) , "MSE:",
          round(mean(model$residuals^2),4)), quote = FALSE)
}
```

```
## [1] Span: 0.10 MSE: 42.0865
## [1] Span: 0.15 MSE: 44.5906
## [1] Span: 0.20 MSE: 46.3622
## [1] Span: 0.25 MSE: 46.5026
## [1] Span: 0.30 MSE: 46.5074
## [1] Span: 0.35 MSE: 46.6461
## [1] Span: 0.40 MSE: 46.7513
## [1] Span: 0.45 MSE: 46.8195
## [1] Span: 0.50 MSE: 46.9521
## [1] Span: 0.55 MSE: 47.0661
## [1] Span: 0.60 MSE: 47.2813
## [1] Span: 0.65 MSE: 47.4192
## [1] Span: 0.70 MSE: 47.5966
## [1] Span: 0.75 MSE: 47.6723
## [1] Span: 0.80 MSE: 47.7715
## [1] Span: 0.85 MSE: 47.9451
## [1] Span: 0.90 MSE: 48.0724
```

```
## [1] Span: 0.95 MSE: 48.2951
## [1] Span: 1.00 MSE: 48.9563
```

- Running through multiple values of span using anything lower than span = .1 gives a small error but at .1 we get a smaller MSE of 42.0865 gradually getting higher as the span gets bigger.