DS502- HW1

Mahdi Alouane and Rahul Pande

1. 2.4 (question 1)

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

As the sample size is extremely large the certainty of the veracity of the sample mean is high. Therefore the variance will be low, since an unknown sample will not deviate a lot from the sample mean of a large sample. The low number of predictors will prevent overfitting and thus reduce variance. Since we have two factors that reduce variance, in this case, a flexible statistical learning method is expected to be better because it will reduce the bias. The higher variance of more flexible learning method will be countered by the above two factors.

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

The large number of predictors will tend to overfit and perform poor on unseen data, thus increasing variance. Similarly with less number of samples there could be a lot of variablility in the least sqaures which would mean higher variance for unseen data. Therefore in this case an inflexible learning method is expected to perform better since it will counter the high variance.

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

Where the relationship between predictors and response is highly non-linear, an inflexible model will have high bias given that it won't be able to capture the complex relationship between the predictors and response. Therefore a flexible method is expected to perform better in this case since it will reduce the bias.

- (d) The variance of the error terms, i.e. μ^2, is extremely high.

As the variance of error terms is extremely high, the certainty of the veracity of the sample mean is very low. Therefore the model variance will be high, since an unknown sample could deviate a lot from the sample mean. Hence, in this case, an inflexible statistical learning method is expected to be better because it will reduce the variance.

2. 2.4 (question 3)

a-1.bb

- Bias is high for low flexibily as the model is not flexible enough to capture the complexities in the data. As the model becomes more flexible it tries to fit all the data and bias decreases until it reduces to irreducible error.
- Variance is small when the model is simple, with small change in data there is small change in error. In the beginning, Variance does not increase significantly with increase in model complexity for when the model complexity <= true data complexity. When the model becomes too complex, the Variance increases exponentially with increase in flexibility.
- Training error decreases with increase in model flexibily, similarly to the Bias. It is the sum of reducible and irreducible error.
- Test error is lowest when the model complexity is closest to true data complexity. When the model is too simple, test error is high due to high bias, on the other hand when the model is too flexible, test error is high due to high variance.
- Irreducible error is constant and independent of the model flexibily.

3. 2.4 (question 6)

A parametric statistical learning approach assumes a given form for f and reduces the problem to estimate only the parameters of this form in order to predict f, however, a non-parametric approach does not assume any form for f and hence requires a large set of data in order to estimate f.

The advantages of a parametric approach in regression or classification is the simplification of the model which could lead to a good estimation without requiring a large amount of data.

On the other side, this approach could lead to an erroneous model if the initially assumed form is wrong. Another problem could sometimes occur if we try to make the model more flexible until it overfits the training set.

4. 2.4 (question 8)

- (a) Reading `College.csv` into `college` variable

```r
college <- read.csv("College.csv", stringsAsFactors = TRUE)
```

- (b) Set first column as row names and then remove that column from data

```r
# fix(college)
rownames(college) = college[,1]
college = college[,-1]
# fix(college)
```

- (c)

i. Summary of `college` variable

```r
summary(college)
```

```
##  Private        Apps           Accept          Enroll       Top10perc
##  No :212   Min.   :   81   Min.   :   72   Min.   :  35   Min.   : 1.00
##  Yes:565   1st Qu.:  776   1st Qu.:  604   1st Qu.: 242   1st Qu.:15.00
##            Median : 1558   Median : 1110   Median : 434   Median :23.00
##            Mean   : 3002   Mean   : 2019   Mean   : 780   Mean   :27.56
##            3rd Qu.: 3624   3rd Qu.: 2424   3rd Qu.: 902   3rd Qu.:35.00
##            Max.   :48094   Max.   :26330   Max.   :6392   Max.   :96.00
##    Top25perc      F.Undergrad     P.Undergrad         Outstate
##  Min.   :  9.0   Min.   :  139   Min.   :    1.0   Min.   : 2340
##  1st Qu.: 41.0   1st Qu.:  992   1st Qu.:   95.0   1st Qu.: 7320
##  Median : 54.0   Median : 1707   Median :  353.0   Median : 9990
##  Mean   : 55.8   Mean   : 3700   Mean   :  855.3   Mean   :10441
##  3rd Qu.: 69.0   3rd Qu.: 4005   3rd Qu.:  967.0   3rd Qu.:12925
##  Max.   :100.0   Max.   :31643   Max.   :21836.0   Max.   :21700
##    Room.Board       Books           Personal         PhD
##  Min.   :1780   Min.   :  96.0   Min.   : 250   Min.   :  8.00
##  1st Qu.:3597   1st Qu.: 470.0   1st Qu.: 850   1st Qu.: 62.00
##  Median :4200   Median : 500.0   Median :1200   Median : 75.00
##  Mean   :4358   Mean   : 549.4   Mean   :1341   Mean   : 72.66
##  3rd Qu.:5050   3rd Qu.: 600.0   3rd Qu.:1700   3rd Qu.: 85.00
##  Max.   :8124   Max.   :2340.0   Max.   :6800   Max.   :103.00
##     Terminal       S.F.Ratio      perc.alumni        Expend
##  Min.   : 24.0   Min.   : 2.50   Min.   : 0.00   Min.   : 3186
##  1st Qu.: 71.0   1st Qu.:11.50   1st Qu.:13.00   1st Qu.: 6751
##  Median : 82.0   Median :13.60   Median :21.00   Median : 8377
##  Mean   : 79.7   Mean   :14.09   Mean   :22.74   Mean   : 9660
##  3rd Qu.: 92.0   3rd Qu.:16.50   3rd Qu.:31.00   3rd Qu.:10830
##  Max.   :100.0   Max.   :39.80   Max.   :64.00   Max.   :56233
##    Grad.Rate
##  Min.   : 10.00
##  1st Qu.: 53.00
##  Median : 65.00
##  Mean   : 65.46
##  3rd Qu.: 78.00
##  Max.   :118.00
```

ii. Scatterplot matrix of first 10 variables

```r
pairs(college[,1:10])
```

8c ii-1.bb

iii. Boxplot of `Outstate` versus `Private`

```
plot(Outstate~Private, data=college)
```

iv. Elite Universities

```
Elite = rep("No",nrow(college))
Elite[college$Top10perc > 50] = " Yes"
Elite = as.factor(Elite)
```

```r
college = data.frame(college, Elite)
summary(college)
```

```
##  Private        Apps           Accept          Enroll       Top10perc
##  No :212   Min.   :   81   Min.   :   72   Min.   :  35   Min.   : 1.00
##  Yes:565   1st Qu.:  776   1st Qu.:  604   1st Qu.: 242   1st Qu.:15.00
##            Median : 1558   Median : 1110   Median : 434   Median :23.00
##            Mean   : 3002   Mean   : 2019   Mean   : 780   Mean   :27.56
##            3rd Qu.: 3624   3rd Qu.: 2424   3rd Qu.: 902   3rd Qu.:35.00
##            Max.   :48094   Max.   :26330   Max.   :6392   Max.   :96.00
##    Top25perc      F.Undergrad     P.Undergrad         Outstate
##  Min.   :  9.0   Min.   :  139   Min.   :    1.0   Min.   : 2340
##  1st Qu.: 41.0   1st Qu.:  992   1st Qu.:   95.0   1st Qu.: 7320
##  Median : 54.0   Median : 1707   Median :  353.0   Median : 9990
##  Mean   : 55.8   Mean   : 3700   Mean   :  855.3   Mean   :10441
##  3rd Qu.: 69.0   3rd Qu.: 4005   3rd Qu.:  967.0   3rd Qu.:12925
##  Max.   :100.0   Max.   :31643   Max.   :21836.0   Max.   :21700
##    Room.Board       Books           Personal         PhD
##  Min.   :1780   Min.   :  96.0   Min.   : 250   Min.   :  8.00
##  1st Qu.:3597   1st Qu.: 470.0   1st Qu.: 850   1st Qu.: 62.00
##  Median :4200   Median : 500.0   Median :1200   Median : 75.00
##  Mean   :4358   Mean   : 549.4   Mean   :1341   Mean   : 72.66
##  3rd Qu.:5050   3rd Qu.: 600.0   3rd Qu.:1700   3rd Qu.: 85.00
##  Max.   :8124   Max.   :2340.0   Max.   :6800   Max.   :103.00
##     Terminal       S.F.Ratio      perc.alumni        Expend
##  Min.   : 24.0   Min.   : 2.50   Min.   : 0.00   Min.   : 3186
##  1st Qu.: 71.0   1st Qu.:11.50   1st Qu.:13.00   1st Qu.: 6751
##  Median : 82.0   Median :13.60   Median :21.00   Median : 8377
##  Mean   : 79.7   Mean   :14.09   Mean   :22.74   Mean   : 9660
##  3rd Qu.: 92.0   3rd Qu.:16.50   3rd Qu.:31.00   3rd Qu.:10830
##  Max.   :100.0   Max.   :39.80   Max.   :64.00   Max.   :56233
##    Grad.Rate       Elite
##  Min.   : 10.00    Yes: 78
##  1st Qu.: 53.00   No  :699
##  Median : 65.00
##  Mean   : 65.46
##  3rd Qu.: 78.00
##  Max.   :118.00
```

```r
plot(Outstate~Elite, data=college)
```

8c iv-1.bb

From the `summary`, we have **78** `Elite` universities.

v. Histograms

```
par(mfrow=c(2,2))

hist_vars   = c("Apps", "S.F.Ratio", "Room.Board", "Grad.Rate")
hist_breaks = c(50, 10, 20, 20)
hist_data = subset(college, select = hist_vars)
make_hist <- function(list.elem, names, breaks) {
  hist(list.elem, main = names, xlab = names, breaks = breaks)
}
mapply(make_hist,
       list.elem = hist_data,
       names = names(hist_data),
       breaks = hist_breaks)
```

8c v-1.bb

vi. Exploration

```
college$Accept.Rate = college$Accept / college$Apps * 100
college$Enroll.Rate = college$Enroll / college$Accept * 100

par(mfrow=c(2,2))
plot(Grad.Rate ~ Outstate, data = college)
plot(Outstate ~ Private, data = college)
```

```r
plot(Outstate ~ S.F.Ratio, data = college)
plot(S.F.Ratio ~ Private, data = college)
```

8c vi-1.bb

```r
par(mfrow=c(2,2))
plot(Apps ~ Private, data = college)
plot(Enroll.Rate ~ Outstate, data = college)
```

8c vi-2.bb

Observations:

- Strong positive correlation between `Grad.Rate` and `Outstate` fees
- Significant difference in `Outstate` fees depending on if the university is `Private` and on the `S.F.Ratio`
- From the second point and above plot, we see that `Private` university colleges have a smaller `S.F.Ratio`
- `Private` university colleges tend to get lot more applications than public
- `Enroll.Rate` is negatively correlated with the college `Outstate` fees

5. 2.4 (question 9)

```r
library(ISLR)
data(Auto)
colSums(sapply(Auto, is.na))
```

```
##          mpg    cylinders displacement   horsepower       weight
##            0            0            0            0            0
## acceleration         year       origin         name
##            0            0            0            0
```

No missing values

- (a) Qualitative and quantitative predictors

```r
summary(Auto)
```

```
##       mpg          cylinders      displacement     horsepower
##  Min.   : 9.00   Min.   :3.000   Min.   : 68.0   Min.   : 46.0
##  1st Qu.:17.00   1st Qu.:4.000   1st Qu.:105.0   1st Qu.: 75.0
##  Median :22.75   Median :4.000   Median :151.0   Median : 93.5
##  Mean   :23.45   Mean   :5.472   Mean   :194.4   Mean   :104.5
##  3rd Qu.:29.00   3rd Qu.:8.000   3rd Qu.:275.8   3rd Qu.:126.0
##  Max.   :46.60   Max.   :8.000   Max.   :455.0   Max.   :230.0
##
##      weight      acceleration        year           origin
##  Min.   :1613   Min.   : 8.00   Min.   :70.00   Min.   :1.000
```

```
##  1st Qu.:2225   1st Qu.:13.78   1st Qu.:73.00   1st Qu.:1.000
##  Median :2804   Median :15.50   Median :76.00   Median :1.000
##  Mean   :2978   Mean   :15.54   Mean   :75.98   Mean   :1.577
##  3rd Qu.:3615   3rd Qu.:17.02   3rd Qu.:79.00   3rd Qu.:2.000
##  Max.   :5140   Max.   :24.80   Max.   :82.00   Max.   :3.000
##
##                  name
##  amc matador       :  5
##  ford pinto        :  5
##  toyota corolla    :  5
##  amc gremlin       :  4
##  amc hornet        :  4
##  chevrolet chevette:  4
##  (Other)           :365
```

```r
hist(Auto$origin, xlab = "origin", main = "Histogram of origin")
```

9a-1.bb

From the summary, we have

- Qualitative predictors: name, origin
- Quantitative predictors: mpg, cylinders, displacement, horsepower, weight, acceleration, year

`origin` is a qualitative predictor and no quantitative. Origin of car (1. American, 2. European, 3. Japanese)

- (b) Range of quantitative predictors

```r
quantitative_preds = c("mpg", "cylinders", "displacement", "horsepower", "weight",
                       "acceleration", "year")
qualitative_preds = c("origin", "name")

ranges = data.frame(lapply(subset(Auto, select = quantitative_preds), range))
knitr::kable(ranges)
```

mpg	cylinders	displacement	horsepower	weight	acceleration	year
9.0	3	68	46	1613	8.0	70
46.6	8	455	230	5140	24.8	82

- (c) Mean and standard deviation of quantitative predictors

```
mean_sd = data.frame(lapply(subset(Auto, select = quantitative_preds), function(x) {
  c(mean = mean(x), std_dev = sd(x))
  }
))
knitr::kable(mean_sd)
```

	mpg	cylinders	displacement	horsepower	weight	acceleration	year
mean	23.445918	5.471939	194.412	104.46939	2977.5842	15.541327	75.979592
std_dev	7.805008	1.705783	104.644	38.49116	849.4026	2.758864	3.683737

- (d) Range, mean and standard deviation of quantitative predictors of **subset data**

```
subset_mean_sd = data.frame(lapply(subset(Auto[- c(10:85),], select = quantitative_preds),
                                   function(x) {
                                     c(range = range(x), mean = mean(x), std_dev = sd(x))
                                     }
                                   )
                            )
knitr::kable(subset_mean_sd)
```

	mpg	cylinders	displacement	horsepower	weight	acceleration	year
range1	11.000000	3.000000	68.00000	46.00000	1649.0000	8.500000	70.000000
range2	46.600000	8.000000	455.00000	230.00000	4997.0000	24.800000	82.000000
mean	24.404430	5.373418	187.24051	100.72152	2935.9715	15.726899	77.145570
std_dev	7.867283	1.654179	99.67837	35.70885	811.3002	2.693721	3.106217

- (e) Predictor Findings

```
factor_vars = c("origin")
newAuto = Auto
newAuto[factor_vars] = lapply(Auto[factor_vars], as.factor)
pairs(newAuto)
```

9e-1.bb

From the scatter plots, we observer that:

- `mpg` decreases non-linearly with increase in displacement
- Over time, `mpg` has shown increase
- `displacement`, `horsepower`, `weight` are all highly correlated with each other
- `origin:2` European and `origin:3` Japanese cars have smaller displacement compared to `origin:2` American cars
- (f) `mpg` prediction

```
par(mfrow = c(1,3))
plot(mpg ~ cylinders + displacement + horsepower + weight + acceleration +
       year + origin, data = newAuto)
```

9f-1.bb

9f-2.bb

```
fit = lm(mpg ~ cylinders + displacement + horsepower + weight + acceleration +
           year + origin, data = newAuto)
summary(fit)
```

```
##
## Call:
## lm(formula = mpg ~ cylinders + displacement + horsepower + weight +
##     acceleration + year + origin, data = newAuto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.0095 -2.0785 -0.0982  1.9856 13.3608
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.795e+01  4.677e+00  -3.839 0.000145 ***
## cylinders    -4.897e-01  3.212e-01  -1.524 0.128215
## displacement  2.398e-02  7.653e-03   3.133 0.001863 **
## horsepower   -1.818e-02  1.371e-02  -1.326 0.185488
## weight       -6.710e-03  6.551e-04 -10.243  < 2e-16 ***
## acceleration  7.910e-02  9.822e-02   0.805 0.421101
## year          7.770e-01  5.178e-02  15.005  < 2e-16 ***
## origin2       2.630e+00  5.664e-01   4.643 4.72e-06 ***
## origin3       2.853e+00  5.527e-01   5.162 3.93e-07 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.307 on 383 degrees of freedom
## Multiple R-squared:  0.8242, Adjusted R-squared:  0.8205
## F-statistic: 224.5 on 8 and 383 DF,  p-value: < 2.2e-16
```

9f-3.bb

From the plots and the fit summary:

- `origin` is a very good predictor of `mpg` since the coefficients are large with small Std. Error (p-stat less than 0.05)
- The next significant predictor is `year` with low p-stat value. We also see in the plot that `mpg` tends to increase year on year
- `weight` is significant predictor but has a small coefficient, i.e. very small effect on the predictions of `mpg`
- `horsepower`, `displacement` and `cylinders` cannot be used as predictor for `mpg` since the their estimated coefficients and Std. Error are close, consequently the coefficients might very well be zero, which is suggested by (p-stat >> 0.05)

6. 3.7 (question 1)

The small p-values in Table 3.4 for TV and radio indicate that, holding the other predictors constant, each of these predictors will, with high likelihood, change the amount of sales. For example, fixing the amount of TV and newspaper advertising, while increasing the amount of radio advertising, will very likely lead to an increase in sales because the p-value in Table 3.4 is very small for radio and radio's coefficient is large and positive.

Similarly, fixing the amount of radio and newspaper advertising, while increasing the amount of TV advertising, will very likely lead to an increase in sales because the p-value in Table 3.4 is very small for TV and radio's coefficient is positive.

One unit increase in TV (while keeping other variables constant) will likely lead to smaller increase in sales than one unit increase in radio (while keeping other variables constant) as the coeff(radio) > coeff(TV).

On the other hand, since the p-value for newspaper is quite large, the data indicates that newspaper advertising in unlikely to have any effect on sales when TV and radio are held fixed. Note, Table 3.3 does show a small p-value for a single linear regression across newspaper and sales, but this is likely because newspaper advertising is predictive of TV and radio advertising, not necessarily because newspaper advertising directly influences sales.

7. 3.7 (question 5)

Given that,

$\hat{y}_i = x_i\hat{\beta}$ and $\hat{\beta} = \frac{\sum_{i'=1}^{n} x_{i'}y_{i'}}{\sum_{j=1}^{n} x_j^2}$

We can write,

$\hat{y}_i = x_i \frac{\sum_{i'=1}^{n} x_{i'}y_{i'}}{\sum_{j=1}^{n} x_j^2} = \sum_{i'=1}^{n} \left(\frac{x_{i'}x_i}{\sum_{j=1}^{n} x_j^2} \right) y_{i'}$

Hence,

$\hat{y}_i = \sum_{i'=1}^{n} a_{i'}y_{i'}$ where $a_{i'} = \frac{x_{i'}x_i}{\sum_{j=1}^{n} x_j^2}$

8. 3.7 (question 6)

On one hand, we know from (3.2) that the least squares line respects this equation

$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$

On the other hand, we know from (3.4) that

$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1\bar{x}$

Hence,

$\hat{y} = (\bar{y} - \hat{\beta}_1\bar{x}) + \hat{\beta}_1 x$

For $x = \bar{x}$,

$\hat{y} = \bar{y} - \hat{\beta}_1\bar{x} + \hat{\beta}_1\bar{x} = \bar{y}$

Consquently, the point $(\bar{x}, \bar{y})$ belongs to the least squares line.

9. 3.7 (question 8)

- (a) The following code performs a simple linear regression on the Auto dataset

```r
# Import the ISLR package
library(ISLR)

# Attach the Auto as the current database
attach(Auto)

# Perform a simple linear regression
lmFit=lm(mpg~horsepower)

#Print the results
summary(lmFit)
```

```
##
## Call:
## lm(formula = mpg ~ horsepower)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.5710  -3.2592  -0.3435   2.7630  16.9240
```

```
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 39.935861   0.717499   55.66   <2e-16 ***
## horsepower  -0.157845   0.006446  -24.49   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.906 on 390 degrees of freedom
## Multiple R-squared:  0.6059, Adjusted R-squared:  0.6049
## F-statistic: 599.7 on 1 and 390 DF,  p-value: < 2.2e-16
```

```r
# Calculate the mean of the response (mpg) (question ii.)
mean(mpg)
```

```
## [1] 23.44592
```

i. Given that the p-value of the F-statistic is close to zero and the F-statistic is larger than 1, we can reject the null-hypothesis stating that all regression coefficients are equal to zero, therefore, we can conclude that there is a significant relationship between horsepower and mpg.

ii. First, the mean value for the response, which is mpg, is 23.446 and since the RSE of lmFit was 4.906, we can observe that the percentage error is about 20.925%. Second, the R^2 of lmFit was about 0.6059 which reflects that 60.59% of the variance in mpg is explained by the predictor horsepower.

iii. The relationship between the predictor and the response is negative meaning that the more horsepower a given car would have the less mpg efficiency it would have.

iv.

```r
predict(lmFit, data.frame(horsepower=98), interval="confidence")
```

```
##        fit      lwr      upr
## 1 24.46708 23.97308 24.96108
```

```r
predict(lmFit, data.frame(horsepower=98), interval="prediction")
```

```
##        fit     lwr      upr
## 1 24.46708 14.8094 34.12476
```

The predicted mpg associated with a horsepower of 98 is 24.46708 with a confidence interval of [23.97308, 24.96108] and a prediction interval of [14.8094, 34.12476]

- (b)

```r
# Plot the response and the predictor
plot(horsepower, mpg)

# Add the least squares regression line
abline(lmFit)
```

8b-1.bb

- (c)

```
par(mfrow=c(2,2))
plot(lmFit)
```

8c-1.bb

According to the residuals plots, we can observe that they are highly biased at lower and higher values and respectively less biased in the middle. We can note then some signs of non-linearity. In addition, the plot of residuals vs leverage reveals some high leverage points and some outliers outside the range [-2,2]

10. 3.7 (question 9)

- (a)

```r
# Produce a scatterplot
pairs(Auto)
```

9a-1.bb

- (b)

```r
# Remove the names variable
names(Auto)
```

```
## [1] "mpg"          "cylinders"    "displacement" "horsepower"
## [5] "weight"       "acceleration" "year"         "origin"
## [9] "name"
```

```r
# Compute the correlation matrix
cor(Auto[1:8])
```

```
##                     mpg  cylinders displacement horsepower     weight
## mpg           1.0000000 -0.7776175   -0.8051269 -0.7784268 -0.8322442
## cylinders    -0.7776175  1.0000000    0.9508233  0.8429834  0.8975273
## displacement -0.8051269  0.9508233    1.0000000  0.8972570  0.9329944
## horsepower   -0.7784268  0.8429834    0.8972570  1.0000000  0.8645377
## weight       -0.8322442  0.8975273    0.9329944  0.8645377  1.0000000
## acceleration  0.4233285 -0.5046834   -0.5438005 -0.6891955 -0.4168392
## year          0.5805410 -0.3456474   -0.3698552 -0.4163615 -0.3091199
## origin        0.5652088 -0.5689316   -0.6145351 -0.4551715 -0.5850054
##               acceleration       year     origin
## mpg              0.4233285  0.5805410  0.5652088
## cylinders       -0.5046834 -0.3456474 -0.5689316
## displacement    -0.5438005 -0.3698552 -0.6145351
## horsepower      -0.6891955 -0.4163615 -0.4551715
```

```
## weight         -0.4168392 -0.3091199 -0.5850054
## acceleration    1.0000000  0.2903161  0.2127458
## year            0.2903161  1.0000000  0.1815277
## origin          0.2127458  0.1815277  1.0000000
```

- (c)

```r
lmFit2 = lm(mpg ~ . - name, data = Auto)
summary(lmFit2)
```

```
##
## Call:
## lm(formula = mpg ~ . - name, data = Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.5903 -2.1565 -0.1169  1.8690 13.0604
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -17.218435   4.644294  -3.707  0.00024 ***
## cylinders     -0.493376   0.323282  -1.526  0.12780
## displacement   0.019896   0.007515   2.647  0.00844 **
## horsepower    -0.016951   0.013787  -1.230  0.21963
## weight        -0.006474   0.000652  -9.929  < 2e-16 ***
## acceleration   0.080576   0.098845   0.815  0.41548
## year           0.750773   0.050973  14.729  < 2e-16 ***
## origin         1.426141   0.278136   5.127 4.67e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.328 on 384 degrees of freedom
## Multiple R-squared:  0.8215, Adjusted R-squared:  0.8182
## F-statistic: 252.4 on 7 and 384 DF,  p-value: < 2.2e-16
```

i. We can observe that the p-value of the F-statistic is close to zero and the F-statistic is greater than 1, hence, we can reject the null-hypothesis and confirm the relationship between `mpg` and the other predictors.

ii. By referring to the p-values associated with each predictor's t-statistic, we can observe that the variables `year`, `weight`, `origin`, `displacement` are statistically significant to the response.

iii. The coefficient of the `year` variable suggests that an increase of 1 year (the car is a year newer) involves a 0.75 mile per gallon increase (car is more fuel efficient) while the other predictors remain constant.

- (d)

```r
# 2 x 2 pictures on one plot
par(mfrow = c(2, 2))
#
plot(lmFit2)
```

9d-1.bb

The residuals vs fitted graph shows a slight non-linearity in the relationship between the response and the predictors. We can also observe that the residuals seem to be heteroscedastic (as the fitted values increase, the range of the residuals increases). We notice that the residuals vs leverage plot suggests the presence of some outliers (outside the interval [-2,2]) and the presence of some observations with unusually high leverage (such as point 14)

- (e)

```
lmFit2 = lm(mpg ~ year + weight + origin + year:origin, data = Auto)
summary(lmFit2)
```

```
##
## Call:
## lm(formula = mpg ~ year + weight + origin + year:origin, data = Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.7639 -1.8651 -0.1118  1.6828 12.4639
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.702e+00  8.525e+00   1.021 0.307993
## year         4.195e-01  1.066e-01   3.934 9.89e-05 ***
## weight      -6.238e-03  2.598e-04 -24.016  < 2e-16 ***
## origin      -1.489e+01  4.539e+00  -3.280 0.001133 **
## year:origin  2.071e-01  5.851e-02   3.539 0.000451 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.299 on 387 degrees of freedom
```

```
## Multiple R-squared:  0.8232, Adjusted R-squared:  0.8213
## F-statistic: 450.4 on 4 and 387 DF,  p-value: < 2.2e-16
```

In this model, we tried to combine the most significant predictors and add an interaction between `year` and `origin`. The results show an improvement in the adjusted R-squared (we use the adjusted R-squared since the two models don't have the same number of predictors) which reflects that the new model explains better the variablity in `mpg`. We notice also that the value of the p-value for the interaction term is close to zero which explains that the term is significant in the model.

```r
lmFit2 = lm(mpg ~.-name+horsepower:weight, data = Auto)
summary(lmFit2)
```

```
##
## Call:
## lm(formula = mpg ~ . - name + horsepower:weight, data = Auto)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.589 -1.617 -0.184  1.541 12.001
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        2.876e+00  4.511e+00   0.638 0.524147
## cylinders         -2.955e-02  2.881e-01  -0.103 0.918363
## displacement       5.950e-03  6.750e-03   0.881 0.378610
## horsepower        -2.313e-01  2.363e-02  -9.791  < 2e-16 ***
## weight            -1.121e-02  7.285e-04 -15.393  < 2e-16 ***
## acceleration      -9.019e-02  8.855e-02  -1.019 0.309081
## year               7.695e-01  4.494e-02  17.124  < 2e-16 ***
## origin             8.344e-01  2.513e-01   3.320 0.000986 ***
## horsepower:weight  5.529e-05  5.227e-06  10.577  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.931 on 383 degrees of freedom
## Multiple R-squared:  0.8618, Adjusted R-squared:  0.859
## F-statistic: 298.6 on 8 and 383 DF,  p-value: < 2.2e-16
```

In this model, we take the initial model and add an interaction between `horsepower` and `weight`. The results show that the variability in `mpg` is even better explained than the previous model (85.9% for adjusted R-squared compared to 82.13% in the previous model).. The interaction term `horsepower:weight` has a low p-value which means that it is statistically significant to the response.

- (f)

```r
lmFit2 = lm(mpg ~.-name+ log(horsepower), data = Auto)
summary(lmFit2)
```

```
##
## Call:
## lm(formula = mpg ~ . - name + log(horsepower), data = Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.5777 -1.6623 -0.1213  1.4913 12.0230
##
## Coefficients:
```

```
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      8.674e+01  1.106e+01   7.839 4.54e-14 ***
## cylinders       -5.530e-02  2.907e-01  -0.190 0.849230
## displacement    -4.607e-03  7.108e-03  -0.648 0.517291
## horsepower       1.764e-01  2.269e-02   7.775 7.05e-14 ***
## weight          -3.366e-03  6.561e-04  -5.130 4.62e-07 ***
## acceleration    -3.277e-01  9.670e-02  -3.388 0.000776 ***
## year             7.421e-01  4.534e-02  16.368  < 2e-16 ***
## origin           8.976e-01  2.528e-01   3.551 0.000432 ***
## log(horsepower) -2.685e+01  2.652e+00 -10.127  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.959 on 383 degrees of freedom
## Multiple R-squared:  0.8592, Adjusted R-squared:  0.8562
## F-statistic: 292.1 on 8 and 383 DF,  p-value: < 2.2e-16
```

```r
lmFit2 = lm(mpg ~.-name+ I((weight)^2), data = Auto)
summary(lmFit2)
```

```
##
## Call:
## lm(formula = mpg ~ . - name + I((weight)^2), data = Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.4706 -1.6701 -0.1488  1.6383 12.5429
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.479e+00  4.614e+00   0.321  0.74867
## cylinders     -2.840e-01  2.917e-01  -0.974  0.33083
## displacement   1.371e-02  6.793e-03   2.019  0.04418 *
## horsepower    -2.435e-02  1.243e-02  -1.959  0.05083 .
## weight        -2.049e-02  1.580e-03 -12.970  < 2e-16 ***
## acceleration   6.571e-02  8.895e-02   0.739  0.46055
## year           7.999e-01  4.615e-02  17.331  < 2e-16 ***
## origin         7.418e-01  2.603e-01   2.850  0.00461 **
## I((weight)^2)  2.237e-06  2.341e-07   9.556  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.994 on 383 degrees of freedom
## Multiple R-squared:  0.8558, Adjusted R-squared:  0.8528
## F-statistic: 284.2 on 8 and 383 DF,  p-value: < 2.2e-16
```

```r
lmFit2 = lm(mpg ~.-name+ log(horsepower) + I((horsepower)^2), data = Auto)
summary(lmFit2)
```

```
##
## Call:
## lm(formula = mpg ~ . - name + log(horsepower) + I((horsepower)^2),
##     data = Auto)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -8.5905 -1.6648 -0.1141  1.5048 12.0359
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        9.434e+01  2.863e+01   3.295 0.001076 **
## cylinders         -1.004e-01  3.305e-01  -0.304 0.761422
## displacement      -4.068e-03  7.358e-03  -0.553 0.580679
## horsepower         2.246e-01  1.688e-01   1.331 0.184017
## weight            -3.406e-03  6.714e-04  -5.073 6.14e-07 ***
## acceleration      -3.234e-01  9.796e-02  -3.301 0.001054 **
## year               7.428e-01  4.547e-02  16.337  < 2e-16 ***
## origin             8.910e-01  2.541e-01   3.506 0.000509 ***
## log(horsepower)   -2.930e+01  8.904e+00  -3.291 0.001092 **
## I((horsepower)^2) -1.016e-04  3.526e-04  -0.288 0.773389
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.963 on 382 degrees of freedom
## Multiple R-squared:  0.8592, Adjusted R-squared:  0.8559
## F-statistic:   259 on 9 and 382 DF,  p-value: < 2.2e-16
```

```r
lmFit2 = lm(mpg ~.-name+ log(displacement) + I((cylinders)^2), data = Auto)
summary(lmFit2)
```

```
##
## Call:
## lm(formula = mpg ~ . - name + log(displacement) + I((cylinders)^2),
##     data = Auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.9306  -1.8122   0.0016   1.7027  11.7299
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        4.666e+01  8.643e+00   5.399 1.18e-07 ***
## cylinders          1.281e+00  1.544e+00   0.830  0.40708
## displacement       8.198e-02  1.106e-02   7.414 7.97e-13 ***
## horsepower        -4.258e-02  1.313e-02  -3.243  0.00129 **
## weight            -4.426e-03  6.536e-04  -6.772 4.82e-11 ***
## acceleration      -2.656e-02  9.275e-02  -0.286  0.77472
## year               7.864e-01  4.718e-02  16.670  < 2e-16 ***
## origin             4.179e-01  2.824e-01   1.480  0.13977
## log(displacement) -1.657e+01  2.302e+00  -7.199 3.24e-12 ***
## I((cylinders)^2)  -1.081e-01  1.280e-01  -0.844  0.39912
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.056 on 382 degrees of freedom
## Multiple R-squared:  0.8502, Adjusted R-squared:  0.8467
## F-statistic: 240.9 on 9 and 382 DF,  p-value: < 2.2e-16
```

We notice in the previous models, that the value of adjusted R-squared has slightly increased and that all the new added terms are statistically significant. The RSS value is almost stable.

However, this absence of big improvement could be explained by the fact that the relationship between `mpg` and the predictors could be closer to a linear model, which explains the ineffectiveness of the application of non-linear models.