HW1_Statistical Machine Learning

Moran Guo

2024-09-08

Question 1 (3.7-5)

We have $\hat{y}_i = x_i\hat{\beta} = x_i\Big(\frac{\sum_{i'=1}^{n} x_{i'}y_{i'}}{\sum_{j=1}^{n} x_j^2}\Big)$ so that $\hat{y}_i = \sum_{i'=1}^{n}\Big(\frac{x_i x_{i'}}{\sum_{j=1}^{n} x_j^2}\Big)y_i'$, therefore

$$a_i' = \frac{x_i x_i'}{\sum_{j=1}^{n} x_j^2}.$$

Question 2 (3.7-6)

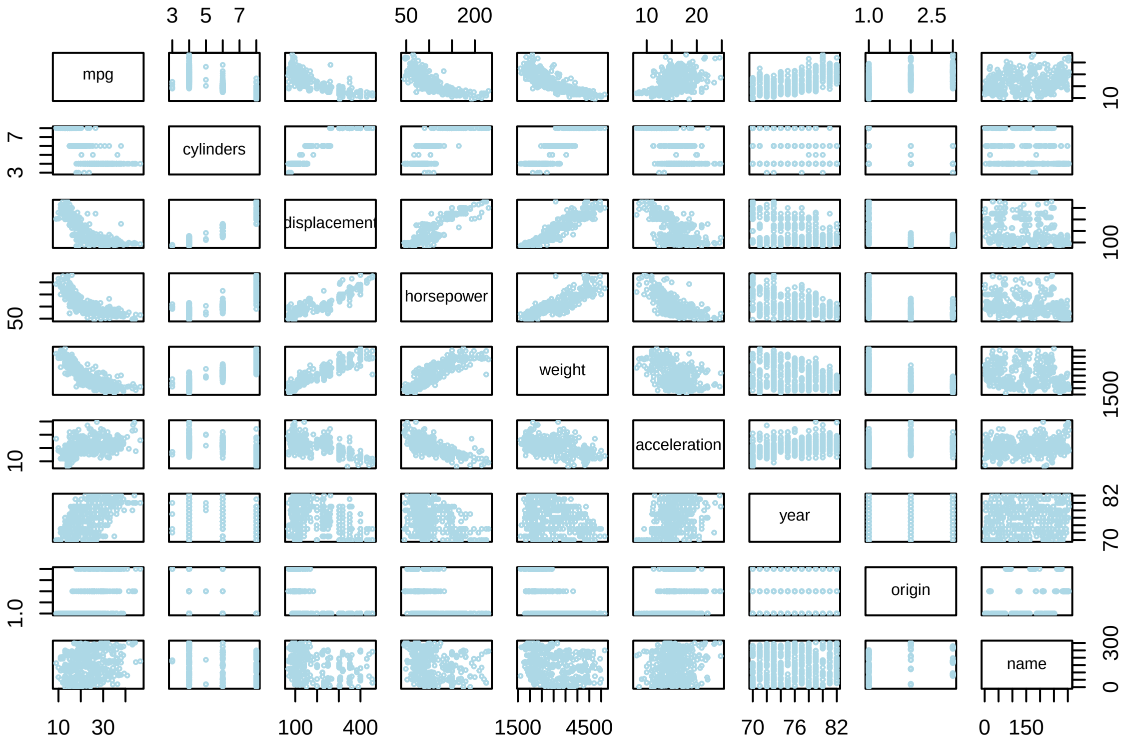
We have the least-square linear regression as $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$, we plug in $x = \bar{x}$ and $\hat{\beta}_0 = \bar{y} - \hat{\beta}_1\bar{x}$ then we have

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

Therefore, the point $(\bar{x}, \bar{y})$ must be on the least-square line.

Question 3 (3.7-9)

```r
# (a) Scatterplot Matrix
# install.packages("ISLR")
library(ISLR)
data(Auto)
plot(Auto, cex = 0.4, col = "lightblue")
```




```r
# (b) correlation matrix
cor(Auto[, 1:8]) # Excluded the first variable 'name'
```

```
##                     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
```

```r
# (c) multiple linear regression
car_lm_fit <- lm(mpg ~ . - name, data = Auto)
summary(car_lm_fit)
```

```
##
## Call:
## lm(formula = mpg ~ . - name, data = Auto)
##
## Residuals:
```

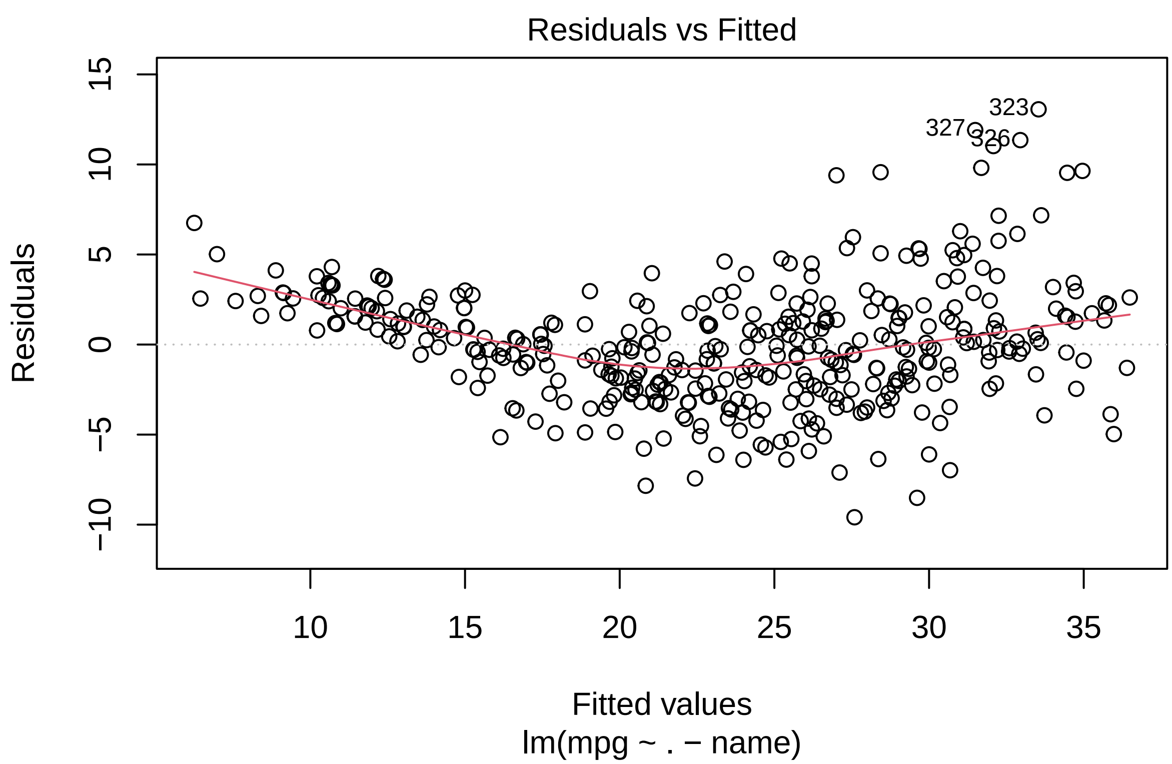
```
##     Min      1Q  Median      3Q     Max
## -9.5903 -2.1565 -0.1169  1.8690 13.0604
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -17.218435   4.644294  -3.707  0.00024 ***
## cylinders     -0.493376   0.323282  -1.526  0.12780
## displacement   0.019896   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
```

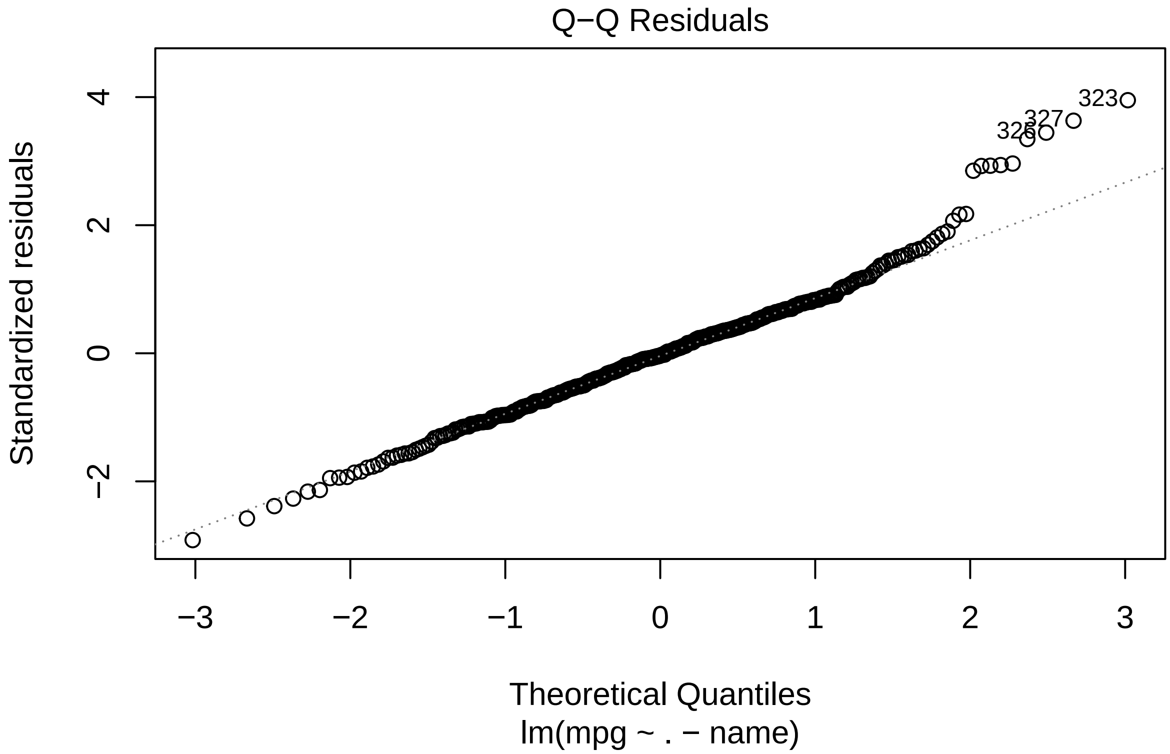
(i) From the result, we see a significant p-value and a sufficiently large F statistics for the overall model fit, which suggests that there exist a linear relationship between mpg and other predictors.

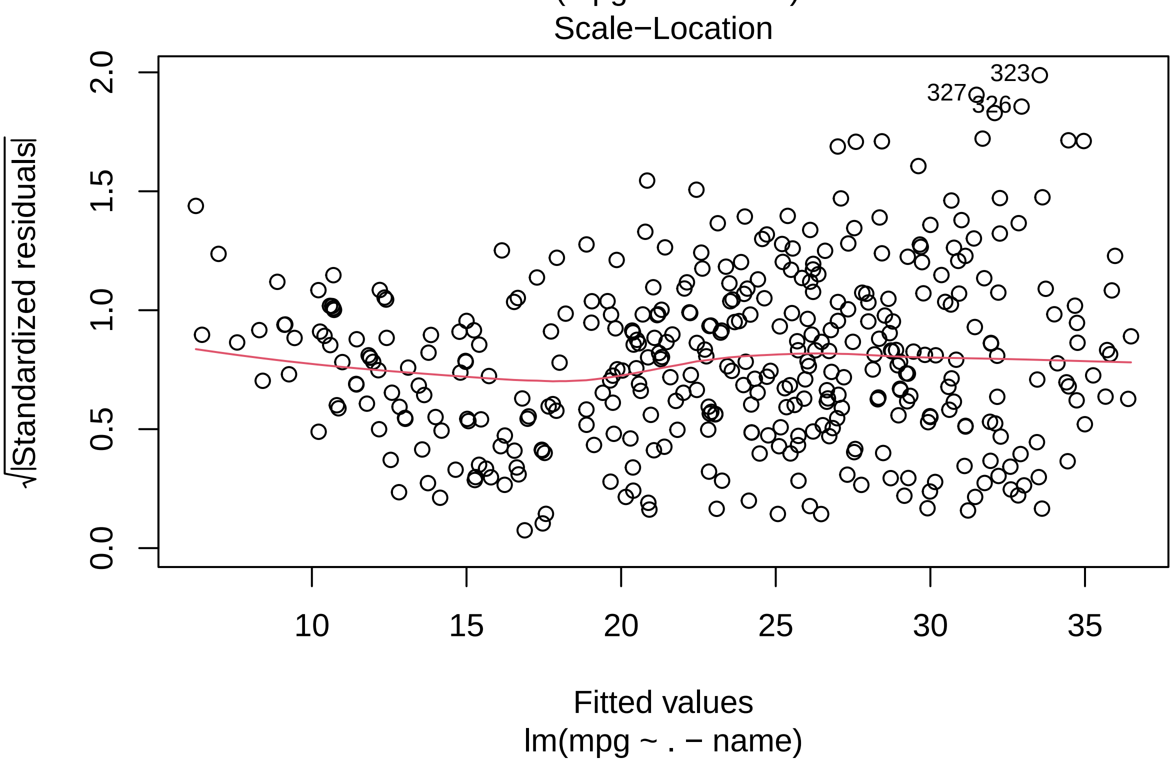
(ii) From the above linear model fit summary, we see variables 'weight', 'year', and 'origin' are the most significant predictors.

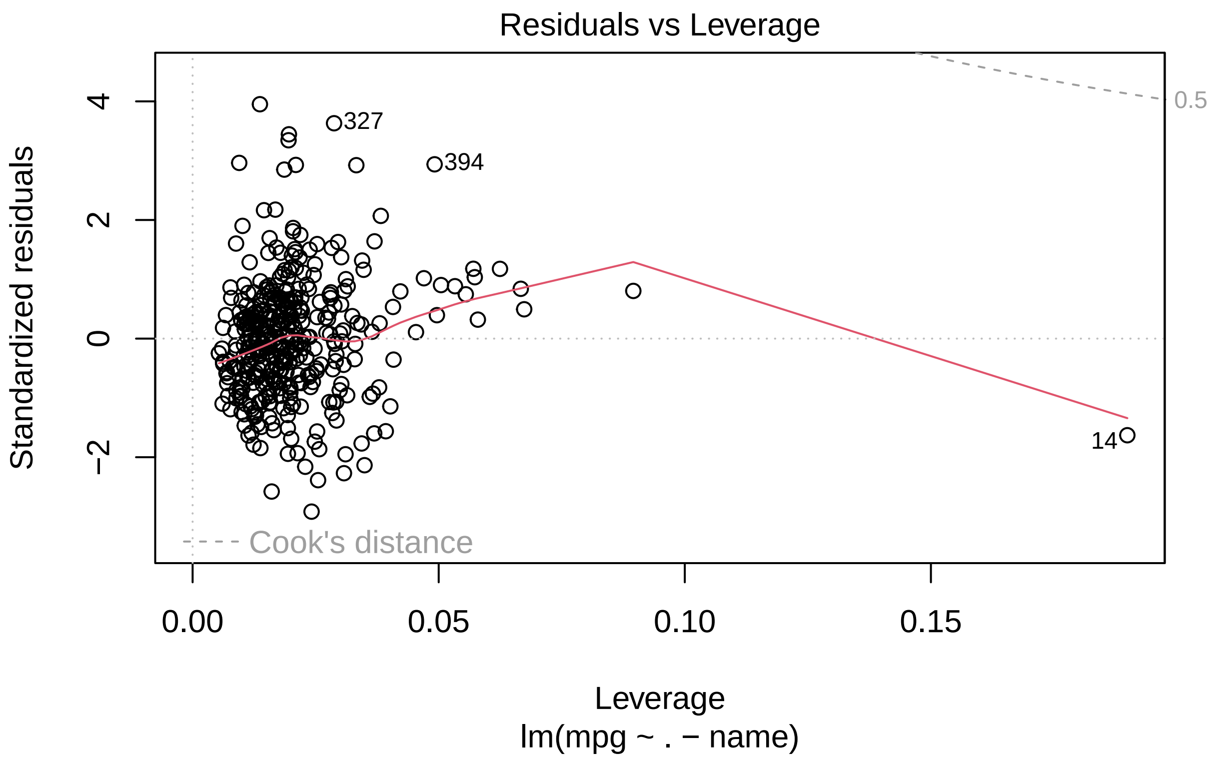
(iii) The coefficient of 'year' is statistical significant and positive, which suggests that year is positively associated with the outcome variable mpg. This implies that as time goes by, the car manufacturers improved cars' mpg, possibly by innovating new technology.

```
# (d) diagnostic plots
plot(car_lm_fit)
```




Q−Q Residuals
Standardized residuals
Theoretical Quantiles
lm(mpg ~ . − name)
323
327
326
4
2
0
−2
−3
−2
−1
0
1
2
3


Scale−Location
√|Standardized residuals|
Fitted values
lm(mpg ~ . − name)
323
327
326
2.0
1.5
1.0
0.5
0.0
10
15
20
25
30
35



From the diagnostic plots (especially the Residuals vs. Leverage plot), we can see that observation 327, 394 seems to have a large residuals compared to the rest, which suggests that they are potentially outliers. Observation 14 has a large leverage value so that it is a point of large impact.

```
# (e) linear regression w/ interaction effects
summary(update(car_lm_fit, . ~ . + horsepower:acceleration))
```

```
##
## Call:
## lm(formula = mpg ~ cylinders + displacement + horsepower + weight +
##     acceleration + year + origin + horsepower:acceleration, data = Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.0329 -1.8177 -0.1183  1.7247 12.4870
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)            -32.499820   4.923380  -6.601 1.36e-10 ***
## cylinders                0.083489   0.316913   0.263 0.792350
## displacement            -0.007649   0.008161  -0.937 0.349244
## horsepower               0.127188   0.024746   5.140 4.40e-07 ***
## weight                  -0.003976   0.000716  -5.552 5.27e-08 ***
## acceleration             0.983282   0.161513   6.088 2.78e-09 ***
## year                     0.755919   0.048179  15.690  < 2e-16 ***
## origin                   1.035733   0.268962   3.851 0.000138 ***
## horsepower:acceleration -0.012139   0.001772  -6.851 2.93e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 3.145 on 383 degrees of freedom
## Multiple R-squared:  0.841,  Adjusted R-squared:  0.8376
## F-statistic: 253.2 on 8 and 383 DF,  p-value: < 2.2e-16
```

```r
summary(update(car_lm_fit, . ~ . + horsepower:weight))
```

```
##
## Call:
## lm(formula = mpg ~ cylinders + displacement + horsepower + weight +
##     acceleration + year + origin + 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
```

Intuitively, we might think that the interactions between horsepower & acceleration **and** horsepower & car weight may have significant influence on car's mpg.

From the result of linear model fit with added interaction terms, we can see that as acceleration increases, the effect of horsepower on car's mpg becomes negative with statistical significance. Cars with higher acceleration and higher horsepower will tend to have even lower mpg than would be expected from each of these two variables individually.

For the interactions between horsepower and car weight, we see a significantly positive effect, whereas car weight & horsepower on their own have negative effects on car mpg. This suggests that as weight increases, the negative effect of horsepower on car's mpg is moderated. Or in other words, the drop of fuel efficiency of high horsepower car is less severe for high weight cars compared to lighter cars.

```r
# (f) different transformation of variables

# log transform of horsepower

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

```
##
## Call:
## lm(formula = mpg ~ . + log(horsepower) - name, 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
# quadratic transform of horsepower

car_lm_fit_sq_hp <- lm (mpg ~ . + I((horsepower)^2) - name, data = Auto)
summary(car_lm_fit_sq_hp)
```

```
##
## Call:
## lm(formula = mpg ~ . + I((horsepower)^2) - name, data = Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.5497 -1.7311 -0.2236  1.5877 11.9955
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        1.3236564  4.6247696   0.286 0.774872
## cylinders          0.3489063  0.3048310   1.145 0.253094
## displacement      -0.0075649  0.0073733  -1.026 0.305550
## horsepower        -0.3194633  0.0343447  -9.302  < 2e-16 ***
## weight            -0.0032712  0.0006787  -4.820 2.07e-06 ***
## acceleration      -0.3305981  0.0991849  -3.333 0.000942 ***
## year               0.7353414  0.0459918  15.989  < 2e-16 ***
## origin             1.0144130  0.2545545   3.985 8.08e-05 ***
## I((horsepower)^2)  0.0010060  0.0001065   9.449  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.001 on 383 degrees of freedom
## Multiple R-squared:  0.8552, Adjusted R-squared:  0.8522
## F-statistic: 282.8 on 8 and 383 DF,  p-value: < 2.2e-16
```

From the model fit results, we can see that both the natural log and the quadratic transform of variable 'horsepower' showed significant effects. We can also observe from the F-statistics that the model improved with the addition of transformed variables. So we would expect that horsepower has a non-linear relationship with mpg, which is the reason that adding transformed variables improves the overall model.

Question 4 (3.7-15)

```r
# (a) simple regression model fit
#install.packages("MASS")
library(MASS)
predictors_bos <- names(Boston)[names(Boston) != "crim"] # Exclude response variable 'crim'

#Empty lists to store statistics
coef_l <- c()
pval_l <- c()
r2_l <- c()

#For loop, looping through all variables
for (i in predictors_bos) {
  formula <- as.formula(paste("crim ~", i))
  model <- lm(formula, data=Boston)
  model_summary <- summary(model)

  #Extract estimates
  coef <- model_summary$coefficients[2, 1]
  pval <- model_summary$coefficients[2, 4]
  r2 <- model_summary$r.squared

  #Store the results in lists
  coef_l <- c(coef_l, coef)
  pval_l <- c(pval_l, pval)
  r2_l <- c(r2_l, r2)
}

#Create dataframe
summary_tab <- data.frame(
  Predictor = predictors_bos,
  Coefficient = coef_l,
  P_value = pval_l,
  R_squared = r2_l
)

# Display the table
summary_tab
```

```
##    Predictor Coefficient      P_value   R_squared
## 1         zn -0.07393498 5.506472e-06 0.040187908
## 2      indus  0.50977633 1.450349e-21 0.165310070
## 3       chas -1.89277655 2.094345e-01 0.003123869
## 4        nox 31.24853120 3.751739e-23 0.177217182
## 5         rm -2.68405122 6.346703e-07 0.048069117
## 6        age  0.10778623 2.854869e-16 0.124421452
## 7        dis -1.55090168 8.519949e-19 0.144149375
## 8        rad  0.61791093 2.693844e-56 0.391256687
## 9        tax  0.02974225 2.357127e-47 0.339614243
## 10   ptratio  1.15198279 2.942922e-11 0.084068439
## 11     black -0.03627964 2.487274e-19 0.148274239
```

```
## 12     lstat  0.54880478 2.654277e-27 0.207590933
## 13      medv -0.36315992 1.173987e-19 0.150780469
```

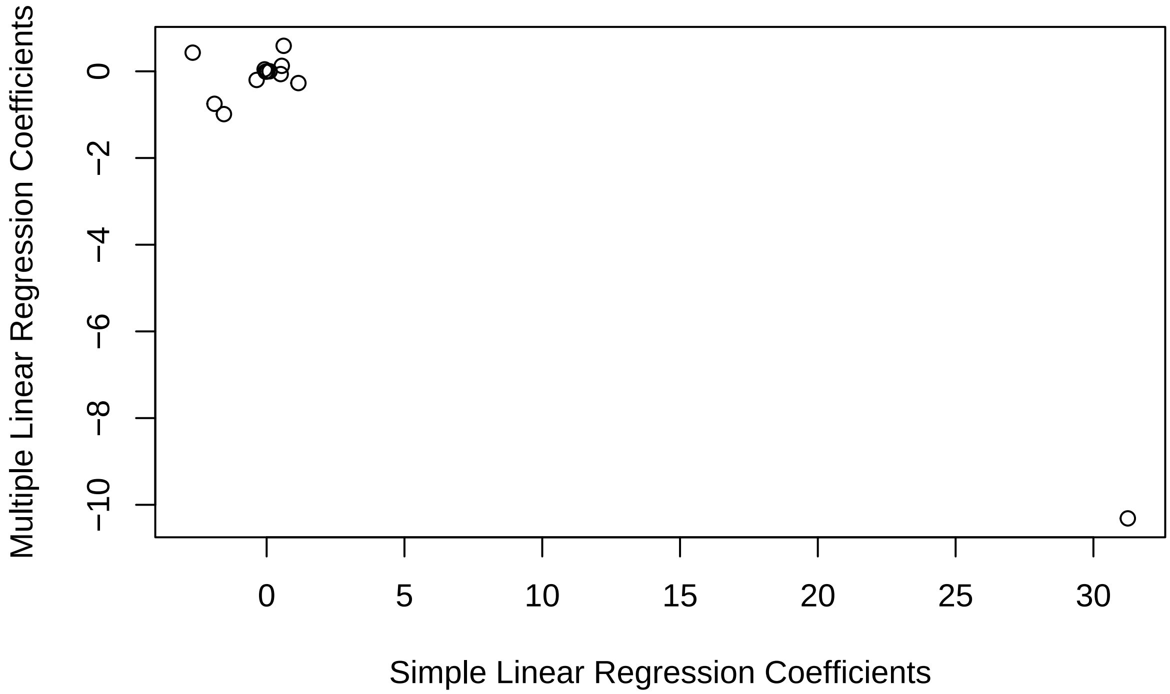
From the integrated result summary table, we can observe that variables ‘rad’ and ‘tax’ have the most significant effect on the response variable, criminal rate per Capita.

```r
# (b) multiple linear regression model fit

model_fit_multi_bos <- lm(crim ~ . , data = Boston)
summary(model_fit_multi_bos)
```

```
##
## Call:
## lm(formula = crim ~ ., data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.924 -2.120 -0.353  1.019 75.051
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  17.033228   7.234903   2.354 0.018949 *
## zn            0.044855   0.018734   2.394 0.017025 *
## indus        -0.063855   0.083407  -0.766 0.444294
## chas         -0.749134   1.180147  -0.635 0.525867
## nox         -10.313535   5.275536  -1.955 0.051152 .
## rm            0.430131   0.612830   0.702 0.483089
## age           0.001452   0.017925   0.081 0.935488
## dis          -0.987176   0.281817  -3.503 0.000502 ***
## rad           0.588209   0.088049   6.680 6.46e-11 ***
## tax          -0.003780   0.005156  -0.733 0.463793
## ptratio      -0.271081   0.186450  -1.454 0.146611
## black        -0.007538   0.003673  -2.052 0.040702 *
## lstat         0.126211   0.075725   1.667 0.096208 .
## medv         -0.198887   0.060516  -3.287 0.001087 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.439 on 492 degrees of freedom
## Multiple R-squared:  0.454,  Adjusted R-squared:  0.4396
## F-statistic: 31.47 on 13 and 492 DF,  p-value: < 2.2e-16
```

From the multiple linear regression result, it can be observed that variables ‘dis’ and ‘rad’ are the two most significant variables compared to the rest, with some of them have no significant effects on the response.

```r
# (c) Compare coefficients of two models

plot(summary_tab$Coefficient, model_fit_multi_bos$coefficients[-1], xlab = "Simple Linear Regression Co
```



From the result we can see that the coefficient of variable 'nox' changed from around 31 in the simple linear regression model to around -10 in the multiple linear regression model. The coefficients of other variables also changed, but not as much as 'nox'.

```r
# (d) test for non-linear association

predictors_compare_bos <- names(Boston)[names(Boston) != "crim"]

predictor_names <- c()
f_stats <- c()
p_value <- c()

for (i in predictors_compare_bos) {
  #Fit a simple linear model
  linear_model <- lm(as.formula(paste("crim ~", i)), data = Boston)
  #Fit a model with quadratic & cubic terms
  nonlinear_model <- lm(as.formula(paste("crim ~", i, "+ I(", i, "^2)", "+ I(", i, "^3)")),
  data = Boston)
  #Use ANOVA to compare the two models
  anova_result <- anova(linear_model, nonlinear_model)
  #Extract statistics
  f_stats_temp <- anova_result$F[2]
  p_value_temp <- anova_result$`Pr(>F)`[2]

  predictor_names <- c(predictor_names, i)
  f_stats <- c(f_stats, f_stats_temp)
  p_value <- c(p_value, p_value_temp)

}

#Create a data frame to store the results
comparison_tab <- data.frame(
```

```r
  Predictors = predictor_names,
  F_statistics = f_stats,
  P_value = p_value
)

comparison_tab[order(comparison_tab$P_value), ]
```

```
##    Predictors F_statistics      P_value
## 13       medv  116.6340058 2.504778e-42
## 7         dis   46.4603654 3.071837e-19
## 4         nox   42.7581707 7.122383e-18
## 2       indus   31.9869602 8.408754e-14
## 6         age   15.1400633 4.125056e-07
## 9         tax   11.6400227 1.144238e-05
## 10    ptratio    8.4155300 2.541647e-04
## 5          rm    5.3088168 5.229427e-03
## 1          zn    4.8118205 8.511995e-03
## 8         rad    3.6732699 2.607832e-02
## 12      lstat    3.3190437 3.698322e-02
## 11      black    0.4622222 6.301501e-01
## 3        chas           NA           NA
```

To compare the results between a simple linear regression model and a third order model suggested by the question, we fit the two models separately and use ANOVA test to see if the third order fit is sufficiently larger than the model we get from a simple linear regression.

From the results, we can see that variables 'medv', 'dis', 'nox', 'indus', 'age', 'tax', 'ptratio', etc., have significant P values which suggests a significant non-linear relationship.

Question 5 (4.7-1)

We work from **Eq. (4.3)** back to **Eq. (4.2)** to show that they are equivalent:

$$\frac{p(X)}{1-p(X)} = e^{\beta_0+\beta_1 X} \Rightarrow p(X) = e^{\beta_0+\beta_1 X}(1-p(X)) \Rightarrow p(X)(1+e^{\beta_0+\beta_1 X}) = e^{\beta_0+\beta_1 X} \Rightarrow p(X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}.$$

Question 6 (4.7-8)

Note that for the KNN with $N = 1$, it will have a zero error rate for the training data set. So we must have

$$\frac{0 + \text{ test error rate}}{2} = 0.18 \Rightarrow \text{ test error rate} = 0.36.$$

We have the information that logistic regression achieves a test error rate of 0.30, so we would prefer logistic regression because of small error rate.