Assignment 5

Morgan Baccus

Question 1

part a.

```r
#Read in dataset
cars <- read.csv("cars.csv")

#Perform multiple linear regression
lm_mpg <- lm(MPG ~ Origin , data = cars)
summary(lm_mpg)
```

```
##
## Call:
## lm(formula = MPG ~ Origin, data = cars)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -26.7452  -4.6882  -0.6882   3.9440  19.3118
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  26.7452     0.8326  32.122  < 2e-16 ***
## OriginJapan   3.7054     1.1549   3.208  0.00144 **
## OriginUS     -7.0570     0.9447  -7.470 5.02e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.114 on 403 degrees of freedom
## Multiple R-squared:  0.2866, Adjusted R-squared:  0.2831
## F-statistic: 80.96 on 2 and 403 DF,  p-value: < 2.2e-16
```

i)

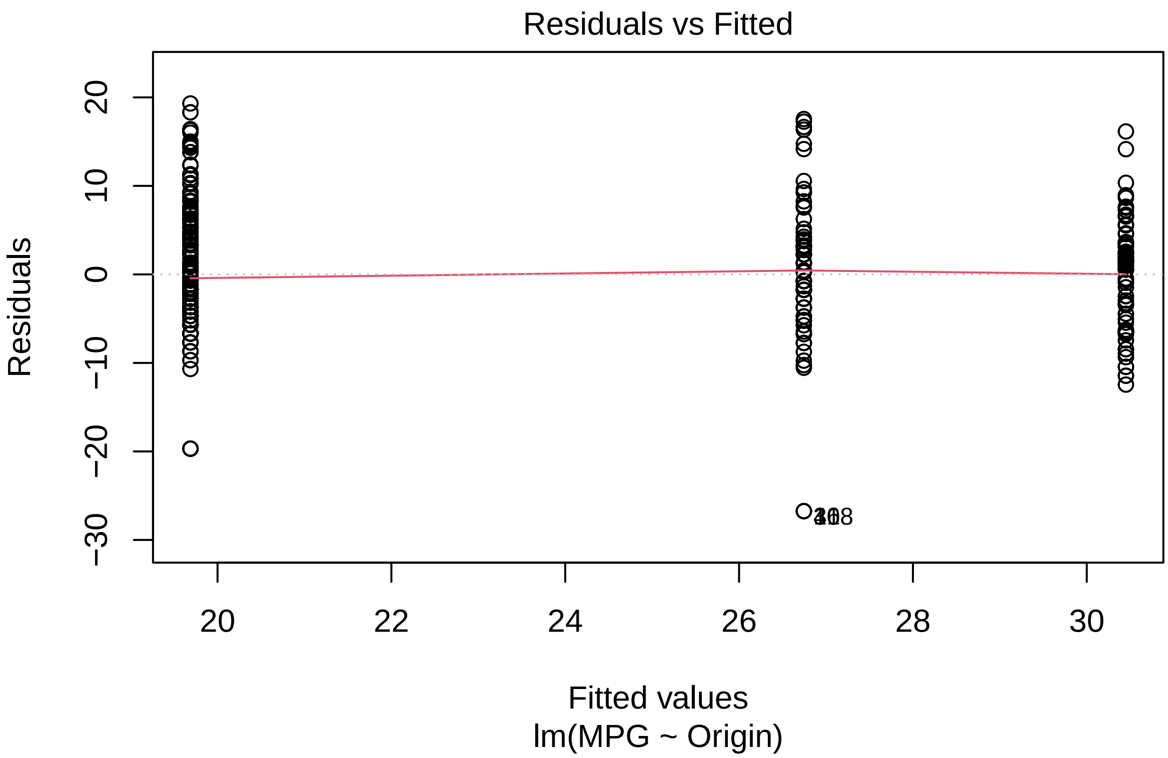
The predictors that appear to have a statistically significant relationship to the response are intercept, weight, and model. OriginUS and displacement also have a less significant relationship. This can be determined by looking at the significance codes in the summary above.

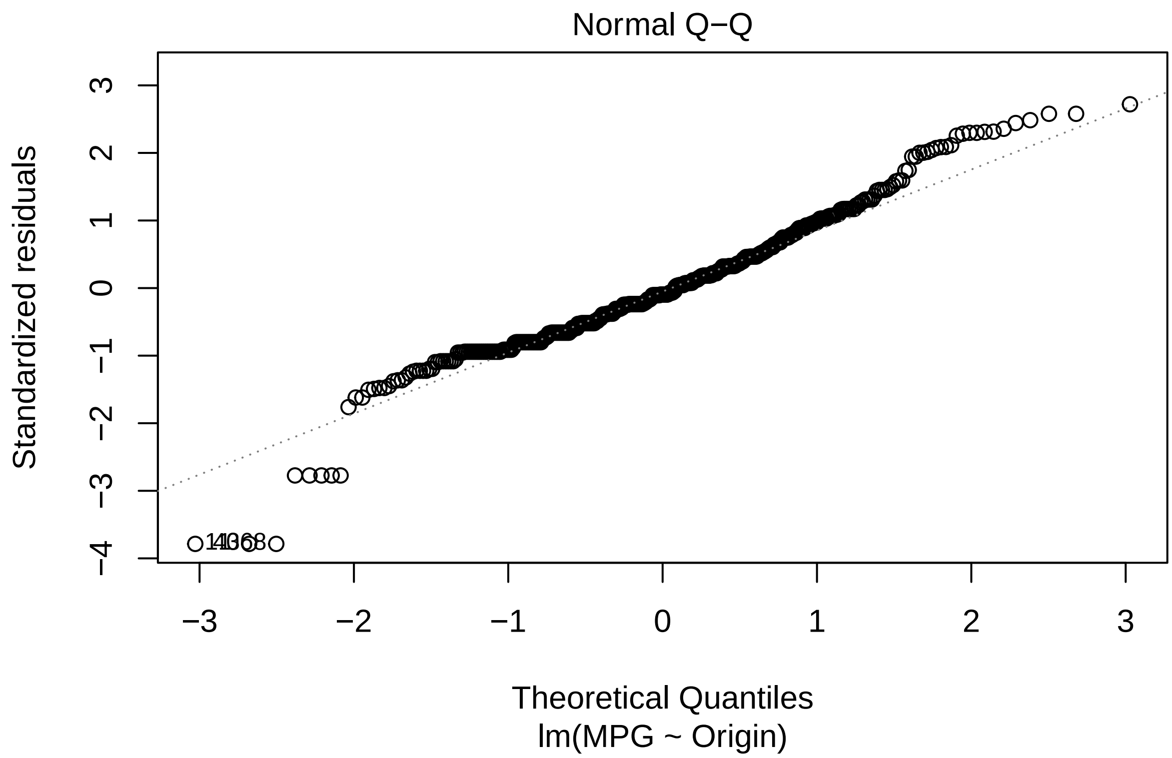
ii)

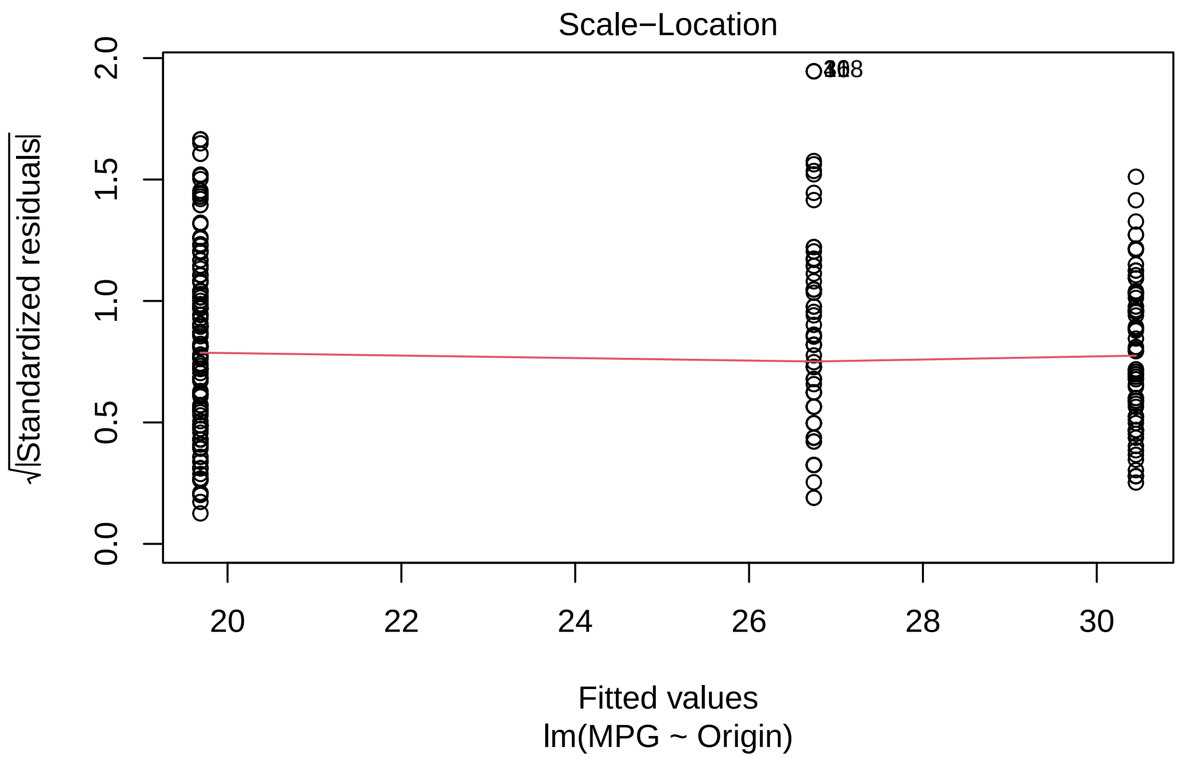
The coefficient for the displacement variable suggests that as displacement increases, MPG increases simultaneously since the coefficient of displacement is positive.

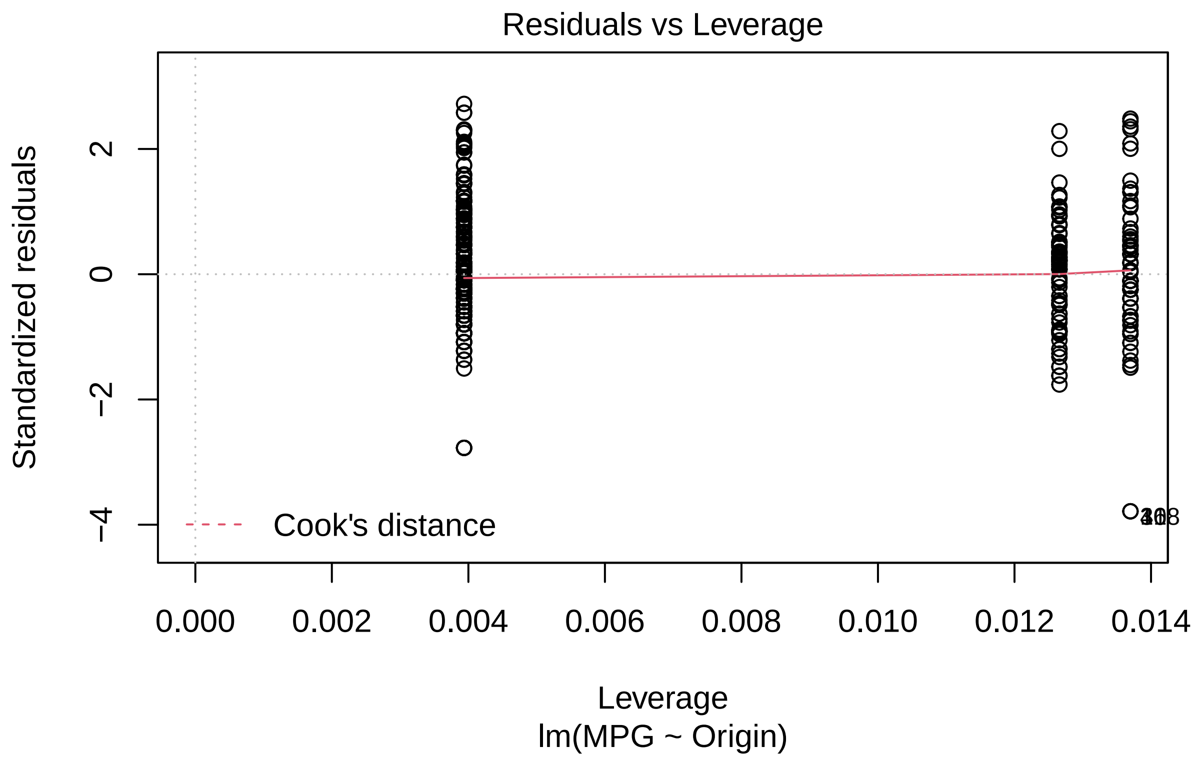
part b.

```
plot(lm_mpg)
```




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(MPG ~ Origin)
−3
−2
−1
0
1
2
3
−4


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
20
22
24
26
28
30
Fitted values
lm(MPG ~ Origin)



A problem with the fit of the graphs is that they are not linear. The residual plots suggest that there are outliers. This is most apparent in the Residual vs. Fitted plot where the majority of the points are within -10 to 10 on the residual scale, but there are quite a few points at or below -20. In the Residuals vs. Leverage plot, most of the points are between 0.00 and 0.05 leverage, but there are several points past 0.012.

part c.

```
lm_temp <- lm(formula = MPG ~ Cylinders * Displacement, data = cars)
summary(lm_temp)
```

```
##
## Call:
## lm(formula = MPG ~ Cylinders * Displacement, data = cars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -29.771  -2.409  -0.053   2.544  21.211
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)            46.671226   2.771542  16.839  < 2e-16 ***
## Cylinders              -2.108696   0.631204  -3.341 0.000914 ***
## Displacement           -0.130301   0.019039  -6.844 2.89e-11 ***
## Cylinders:Displacement  0.010756   0.002442   4.404 1.36e-05 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.307 on 402 degrees of freedom
## Multiple R-squared:  0.6039, Adjusted R-squared:  0.601
## F-statistic: 204.3 on 3 and 402 DF,  p-value: < 2.2e-16
```

```
lm_temp <- lm(formula = MPG ~ Displacement * Horsepower, data = cars)
summary(lm_temp)
```

```
##
## Call:
## lm(formula = MPG ~ Displacement * Horsepower, data = cars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -33.036  -1.960  -0.152   2.355  18.985
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)              4.850e+01  1.638e+00  29.604  < 2e-16 ***
## Displacement            -9.647e-02  8.294e-03 -11.632  < 2e-16 ***
## Horsepower              -1.728e-01  1.993e-02  -8.671  < 2e-16 ***
## Displacement:Horsepower  4.708e-04  6.015e-05   7.827 4.47e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.976 on 402 degrees of freedom
## Multiple R-squared:  0.6518, Adjusted R-squared:  0.6492
## F-statistic: 250.9 on 3 and 402 DF,  p-value: < 2.2e-16
```

```
lm_temp <- lm(formula = MPG ~ Horsepower * Weight, data = cars)
summary(lm_temp)
```

```
##
## Call:
## lm(formula = MPG ~ Horsepower * Weight, data = cars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -34.036  -1.947   0.016   2.130  15.508
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        6.014e+01  2.681e+00  22.431  < 2e-16 ***
## Horsepower        -2.010e-01  2.911e-02  -6.905 1.97e-11 ***
## Weight            -1.042e-02  9.358e-04 -11.138  < 2e-16 ***
## Horsepower:Weight  4.381e-05  7.589e-06   5.773 1.56e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.9 on 402 degrees of freedom
## Multiple R-squared:  0.6624, Adjusted R-squared:  0.6598
## F-statistic: 262.9 on 3 and 402 DF,  p-value: < 2.2e-16
```

```r
lm_temp <- lm(formula = MPG ~ Weight * Acceleration, data = cars)
summary(lm_temp)
```

```
##
## Call:
## lm(formula = MPG ~ Weight * Acceleration, data = cars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -32.471  -2.559  -0.081   2.780  16.135
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)         33.7051643  5.7934738   5.818 1.22e-08 ***
## Weight              -0.0053370  0.0017328  -3.080  0.00221 **
## Acceleration         0.7367719  0.3677195   2.004  0.04578 *
## Weight:Acceleration -0.0001368  0.0001149  -1.191  0.23429
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.126 on 402 degrees of freedom
## Multiple R-squared:  0.6305, Adjusted R-squared:  0.6277
## F-statistic: 228.7 on 3 and 402 DF,  p-value: < 2.2e-16
```

```r
lm_temp <- lm(formula = MPG ~ Acceleration * Model, data = cars)
summary(lm_temp)
```

```
##
## Call:
## lm(formula = MPG ~ Acceleration * Model, data = cars)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -28.8119  -4.7550  -0.4205   5.0054  17.9553
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -115.52674   34.50378  -3.348 0.000890 ***
## Acceleration          3.42269    2.20292   1.554 0.121041
## Model                 1.66546    0.46103   3.612 0.000342 ***
## Acceleration:Model   -0.03469    0.02927  -1.185 0.236705
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.461 on 402 degrees of freedom
## Multiple R-squared:  0.4131, Adjusted R-squared:  0.4087
## F-statistic: 94.32 on 3 and 402 DF,  p-value: < 2.2e-16
```

```r
lm_temp <- lm(formula = MPG ~ Model * Origin, data = cars)
summary(lm_temp)
```

```
##
```

```
## Call:
## lm(formula = MPG ~ Model * Origin, data = cars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -32.488  -3.628  -0.240   3.565  13.887
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -55.9386    14.3599  -3.895 0.000115 ***
## Model                1.0917     0.1894   5.764 1.64e-08 ***
## OriginJapan         12.9220    19.8942   0.650 0.516365
## OriginUS           -16.7921    16.1127  -1.042 0.297964
## Model:OriginJapan   -0.1430     0.2596  -0.551 0.582032
## Model:OriginUS       0.1324     0.2126   0.623 0.533846
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.726 on 400 degrees of freedom
## Multiple R-squared:  0.5413, Adjusted R-squared:  0.5355
## F-statistic:  94.4 on 5 and 400 DF,  p-value: < 2.2e-16
```

It appears that the cylinders and displacement, displacement and horsepower, horsepower and weight, and weight and acceleration interactions are statistically significant.

Question 2

part a.

```r
#Install libraries

library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```r
attach(Boston)

#View dataset explanation
?Boston
```

```
## starting httpd help server ...
```

```
##  done
```

```r
#age
lm_age <- lm(crim ~ age , data = Boston)

#black
lm_black <- lm(crim ~ black , data = Boston)

#chas
lm_chas <- lm(crim ~ chas , data = Boston)

#dis
lm_dis <- lm(crim ~ dis , data = Boston)

#indus
lm_indus <- lm(crim ~ indus , data = Boston)

#lstat
lm_lstat <- lm(crim ~ lstat , data = Boston)

#medv
lm_medv <- lm(crim ~ medv , data = Boston)

#nox
lm_nox <- lm(crim ~ nox , data = Boston)

#pratio
lm_ptratio <- lm(crim ~ ptratio , data = Boston)

#rad
lm_rad <- lm(crim ~ rad , data = Boston)

#rm
lm_rm <- lm(crim ~ rm , data = Boston)

#tax
lm_tax <- lm(crim ~ tax , data = Boston)

#zn
lm_zn <- lm(crim ~ zn , data = Boston)
```

part b.

There is a statistically significant association between the predictor and the response in every model with the exception of chas. The vairables are defined as:
- crim is the per capita crime rate by town.
- nox is the nitrogen oxides concentration.
- chas is the Charles River dummy variable (= 1 if tract bounds river; 0 otherwise).
- rm is the average number of rooms per dwelling.
- dis is the weighted mean of distances to five Boston
employment centers.
- medv is the median value of owner-occupied homes in $1000s.

Other than all the variables being about Boston, there does not exist a strong relationship between any of the variables. The variables are all either environment, housing, or employment statistics and have correlations to crime rates with the exception of chas.

part c.

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

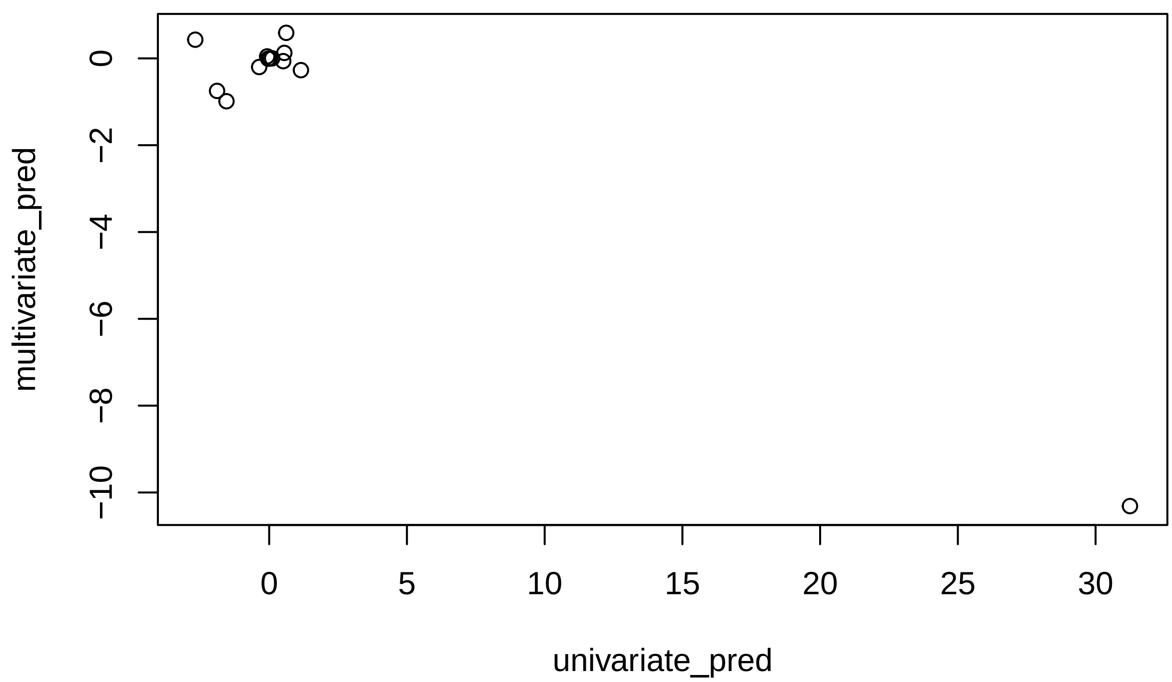
```
##
## Call:
## lm(formula = crim ~ ., data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.924 -2.120 -0.353  1.019 75.051
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  17.033228   7.234903   2.354 0.018949 *
## zn            0.044855   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
```

We can reject the null hypothesis for the predictors intercept, zn, dis, rad, black, and medv.

part d.

```
univariate_pred <- c(
  lm_zn$coefficients[2],
  lm_indus$coefficients[2],
  lm_chas$coefficients[2],
  lm_nox$coefficients[2],
  lm_rm$coefficients[2],
  lm_age$coefficients[2],
  lm_dis$coefficients[2],
  lm_rad$coefficients[2],
  lm_tax$coefficients[2],
  lm_ptratio$coefficients[2],
  lm_black$coefficients[2],
  lm_lstat$coefficients[2],
  lm_medv$coefficients[2]
  )

multivariate_pred <- lm_all$coefficients[2:14]
plot(univariate_pred, multivariate_pred)
```



part e.

```
poly_fit_zn <- lm(formula = crim ~ poly(zn, 3), data = Boston)

# compare fit
anova(lm_zn, poly_fit_zn)
```

```
## Analysis of Variance Table
##
## Model 1: crim ~ zn
## Model 2: crim ~ poly(zn, 3)
##   Res.Df   RSS Df Sum of Sq      F   Pr(>F)
## 1    504 35862
## 2    502 35187  2    674.56 4.8118 0.008512 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
poly_fit_indus <- lm(formula = crim ~ poly(indus, 3), data = Boston)
anova(lm_indus, poly_fit_indus)
```

```
## Analysis of Variance Table
##
## Model 1: crim ~ indus
## Model 2: crim ~ poly(indus, 3)
##   Res.Df   RSS Df Sum of Sq      F    Pr(>F)
## 1    504 31187
## 2    502 27662  2    3525.1 31.987 8.409e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
poly_fit_nox <- lm(formula = crim ~ poly(nox, 3), data = Boston)
anova(lm_nox, poly_fit_nox)
```

```
## Analysis of Variance Table
##
## Model 1: crim ~ nox
## Model 2: crim ~ poly(nox, 3)
##   Res.Df   RSS Df Sum of Sq      F    Pr(>F)
## 1    504 30742
## 2    502 26267  2    4474.6 42.758 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
poly_fit_rm <- lm(formula = crim ~ poly(rm, 3), data = Boston)
anova(lm_rm, poly_fit_rm)
```

```
## Analysis of Variance Table
##
## Model 1: crim ~ rm
## Model 2: crim ~ poly(rm, 3)
```

```
##   Res.Df   RSS Df Sum of Sq      F   Pr(>F)
## 1    504 35567
## 2    502 34831  2    736.69 5.3088 0.005229 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
poly_fit_age <- lm(formula = crim ~ poly(age, 3), data = Boston)
anova(lm_age, poly_fit_age)
```

```
## Analysis of Variance Table
##
## Model 1: crim ~ age
## Model 2: crim ~ poly(age, 3)
##   Res.Df   RSS Df Sum of Sq     F    Pr(>F)
## 1    504 32714
## 2    502 30853  2      1861 15.14 4.125e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
poly_fit_dis <- lm(formula = crim ~ poly(dis, 3), data = Boston)
anova(lm_dis, poly_fit_dis)
```

```
## Analysis of Variance Table
##
## Model 1: crim ~ dis
## Model 2: crim ~ poly(dis, 3)
##   Res.Df   RSS Df Sum of Sq     F    Pr(>F)
## 1    504 31977
## 2    502 26983  2    4994.5 46.46 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
poly_fit_tax <- lm(formula = crim ~ poly(tax, 3), data = Boston)
anova(lm_tax, poly_fit_tax)
```

```
## Analysis of Variance Table
##
## Model 1: crim ~ tax
## Model 2: crim ~ poly(tax, 3)
##   Res.Df   RSS Df Sum of Sq     F    Pr(>F)
## 1    504 24674
## 2    502 23581  2    1093.5 11.64 1.144e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
poly_fit_rad <- lm(formula = crim ~ poly(rad, 3), data = Boston)
anova(lm_rad, poly_fit_rad)
```

```
## Analysis of Variance Table
##
## Model 1: crim ~ rad
```

```
## Model 2: crim ~ poly(rad, 3)
##   Res.Df   RSS Df Sum of Sq      F  Pr(>F)
## 1    504 22745
## 2    502 22417  2    328.06 3.6733 0.02608 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
poly_fit_ptratio <- lm(formula = crim ~ poly(ptratio, 3), data = Boston)
anova(lm_ptratio, poly_fit_ptratio)
```

```
## Analysis of Variance Table
##
## Model 1: crim ~ ptratio
## Model 2: crim ~ poly(ptratio, 3)
##   Res.Df   RSS Df Sum of Sq      F    Pr(>F)
## 1    504 34222
## 2    502 33112  2    1110.2 8.4155 0.0002542 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
poly_fit_black <- lm(formula = crim ~ poly(black, 3), data = Boston)
anova(lm_black, poly_fit_black)
```

```
## Analysis of Variance Table
##
## Model 1: crim ~ black
## Model 2: crim ~ poly(black, 3)
##   Res.Df   RSS Df Sum of Sq      F Pr(>F)
## 1    504 31823
## 2    502 31765  2    58.495 0.4622 0.6302
```

```
poly_fit_lstat <- lm(formula = crim ~ poly(lstat, 3), data = Boston)
anova(lm_lstat, poly_fit_lstat)
```

```
## Analysis of Variance Table
##
## Model 1: crim ~ lstat
## Model 2: crim ~ poly(lstat, 3)
##   Res.Df   RSS Df Sum of Sq     F  Pr(>F)
## 1    504 29607
## 2    502 29221  2    386.39 3.319 0.03698 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
poly_fit_medv <- lm(formula = crim ~ poly(medv, 3), data = Boston)
anova(lm_medv, poly_fit_medv)
```

```
## Analysis of Variance Table
##
## Model 1: crim ~ medv
## Model 2: crim ~ poly(medv, 3)
```

```
##   Res.Df   RSS Df Sum of Sq      F    Pr(>F)
## 1    504 31730
## 2    502 21663  2     10066 116.63 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Question 3

part a.

image:

X_1 = hours studied

X_2 = Undergrad GPA

X_3 = PSQI score

Y = receive an A

$\beta_0 = -7$

$\beta_1 = 0.1$

$\beta_2 = 1$

$\beta_3 = -0.04$

$X_1 = 30$

$X_2 = 3.0$

$X_3 = 11$

$Y = ?$

$$\hat{p}(X) = \frac{e^{\hat{\beta}_0} + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \hat{\beta}_3 x_3}{1 + e^{\hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \hat{\beta}_3 x_3}}$$

$$Y = \frac{e^{-7 + (.1 \times 30) + (3.0 \times 1) + (11 \times -.04)}}{1 + e^{-7 + (.1 \times 30) + (1 \times 3.0) + (11 \times -.04)}}$$

$$\boxed{Y = 0.191545}$$

part b.

image:

$x_1 = ?$

$x_2 = 3.0$

$x_3 = 11$

$Y = .6$

$$\hat{P}(x) = \frac{e^{\hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \hat{\beta}_3 x_3}}{1 + e^{\hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \hat{\beta}_3 x_3}}$$

$$.6 = \frac{e^{-7 + (.1x_1) + (1 \times 3.0) + (11 \times -.04)}}{1 + e^{-7 + (.1x) + (1 \times 3.0) + (11 \times -.04)}}$$

$$.6 = \frac{e^{(.1x_1) - 4.44}}{1 + e^{(.1x_1) - 4.44}}$$

$$e^{(.1x_1) - 4.44} = .6\left(1 + e^{(.1x_1) - 4.44}\right)$$

$$e^{(.1x_1) - 4.44} = .6 + .6e^{(.1x_1) - 4.44}$$

$$.4e^{(.1x_1) - 4.44} = .6$$

$$e^{(.1x_1) - 4.44} = 1.5$$

$$.1x_1 - 4.44 = 0.40546$$

$$.1x_1 = 4.84546$$

$$\boxed{x_1 = 48.45 \text{ hours}}$$

part c.

image:

$x_1 = ?$

$x_2 = 3.0$

$x_3 = 5$

$Y = .5$

$$\hat{p}(x) = \frac{e^{\hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \hat{\beta}_3 x_3}}{1 + e^{\hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \hat{\beta}_3 x_3}}$$

$$.5 = \frac{e^{-7 + (.1x_1) + (1 \times 3.0) + (-.04 \times 5)}}{1 + e^{-7 + (.1x_1) + (1 \times 3.0) + (-.04 \times 5)}}$$

$$.5 = \frac{e^{(.1x_1) - 4.2}}{1 + e^{(.1x_1) - 4.2}}$$

$$e^{(.1x_1) - 4.2} = .5\left(1 + e^{(.1x_1) - 4.2}\right)$$

$$e^{(.1x_1) - 4.2} = .5 + .5e^{(.1x_1) - 4.2}$$

$$.5e^{(.1x_1) - 4.2} = .5$$

$$e^{(.1x_1) - 4.2} = 1$$

$$.1x_1 - 4.2 = 0$$

$$.1x_1 = 4.2$$

$$\boxed{x_1 = 42 \text{ hours}}$$

Question 4

part a.

```r
#Install libraries
library(dplyr)
library(SnowballC)
library(tm)
```

```
## Loading required package: NLP
```

```r
library(tidytext)
library(stringr)

#Read in dataset
cc <- read.csv("consumer_complaints.csv", encoding = "UTF-8")

#Remove [X+] from consumer_complaint column
cc$Consumer_complaint = gsub("[X+]", " ", cc$Consumer_complaint)

#Remove punctuation from consumer_complaint column
cc$Consumer_complaint = gsub("[[:punct:]]", " ", cc$Consumer_complaint)

corpus <- Corpus(VectorSource(cc$Consumer_complaint))

dtm <- DocumentTermMatrix(corpus, control = list(
  removeNumbers = TRUE,
  stemming = TRUE,
  stopwords = TRUE
  ))

dtm <- removeSparseTerms(dtm, 0.99)
print(cc$Product[1])
```

```
## [1] "Vehicle loan or lease"
```

```r
tidy(dtm[1, ])
```

```
## # A tibble: 67 x 3
##    document term      count
##    <chr>    <chr>     <dbl>
##  1 1        accept        1
##  2 1        account       3
##  3 1        advis         4
##  4 1        agreement     1
##  5 1        amount        1
##  6 1        anoth         1
##  7 1        ask           1
##  8 1        back          1
##  9 1        bill          2
## 10 1        case          1
## # ... with 57 more rows
```

part b.

```r
#Install libraries
library(caret)
```

```
## Loading required package: ggplot2

##
## Attaching package: 'ggplot2'

## The following object is masked from 'package:NLP':
##
##     annotate

## Loading required package: lattice
```

```r
library(tidyr)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v tibble  3.1.4     v purrr   0.3.4
## v readr   2.0.1     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x ggplot2::annotate() masks NLP::annotate()
## x dplyr::filter()     masks stats::filter()
## x dplyr::lag()        masks stats::lag()
## x purrr::lift()       masks caret::lift()
## x MASS::select()      masks dplyr::select()
```

```r
library(broom)
library(naivebayes)
```

```
## naivebayes 0.9.7 loaded
```

```r
complaints_df <- tidy(dtm)
colnames(complaints_df)[1] <- "doc"
complaints_df$doc <- as.numeric(complaints_df$doc)
complaints_df <- complaints_df %>%
  pivot_wider(values_from = count, names_from = term, values_fill = 0,
              names_repair="unique") %>%
  mutate(doc = cc$Product[doc])
complaints_df$doc <- as.factor(complaints_df$doc)

features <- complaints_df %>% dplyr::select(-doc)
labels <- complaints_df$doc

cor.features <- cor(features)
```

```r
hc <- findCorrelation(cor.features, cutoff=0.3)
hc <- sort(hc)
reduced_features <- features[,-c(hc)]

train_index <- createDataPartition(labels, p = .8,
                                   list = FALSE,
                                   times = 1)

train_features <- as.matrix(features[train_index,])
train_labels <- as.matrix(labels[train_index])
test_features <- as.matrix(features[-train_index,])
test_labels <- labels[-train_index]

nb.fit <- multinomial_naive_bayes(train_features, train_labels)
nb.class <- predict(nb.fit, test_features)
mean(nb.class == as.matrix(test_labels))
```

```
## [1] 0.6788365
```

```r
confusionMatrix(data = nb.class, reference = test_labels)
```

```
## Confusion Matrix and Statistics
##
##                               Reference
## Prediction                      Bank account or service
##   Bank account or service                          1199
##   Checking or savings account                      1380
##   Consumer Loan                                     119
##   Money transfers                                   121
##   Other financial service                            33
##   Payday loan                                        29
##   Student loan                                       34
##   Vehicle loan or lease                              56
##   Virtual currency                                    5
##                               Reference
## Prediction                      Checking or savings account Consumer Loan
##   Bank account or service                              1431            55
##   Checking or savings account                          5312            66
##   Consumer Loan                                          44           705
##   Money transfers                                       332            17
##   Other financial service                                40            23
##   Payday loan                                            44           202
##   Student loan                                           39           139
##   Vehicle loan or lease                                 105           687
##   Virtual currency                                       10             0
##                               Reference
## Prediction                      Money transfers Other financial service
##   Bank account or service                    22                       4
##   Checking or savings account                63                       5
##   Consumer Loan                               4                       8
##   Money transfers                           193                       9
##   Other financial service                     3                      15
##   Payday loan                                 4                       5
```

```
##   Student loan                           5                       11
##   Vehicle loan or lease                  4                        1
##   Virtual currency                       1                        0
##                              Reference
## Prediction                    Payday loan Student loan Vehicle loan or lease
##   Bank account or service               9           25                    35
##   Checking or savings account          17           45                    75
##   Consumer Loan                        36          179                   707
##   Money transfers                       2           16                    25
##   Other financial service               6           72                    15
##   Payday loan                         223          104                    56
##   Student loan                         47         5182                   151
##   Vehicle loan or lease                 9          172                  1896
##   Virtual currency                      0            2                     0
##                              Reference
## Prediction                    Virtual currency
##   Bank account or service                    0
##   Checking or savings account                2
##   Consumer Loan                              0
##   Money transfers                            0
##   Other financial service                    0
##   Payday loan                                0
##   Student loan                               0
##   Vehicle loan or lease                      0
##   Virtual currency                           1
##
## Overall Statistics
##
##                Accuracy : 0.6788
##                  95% CI : (0.6726, 0.685)
##     No Information Rate : 0.3391
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5871
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: Bank account or service
## Sensitivity                                 0.40289
## Specificity                                 0.91553
## Pos Pred Value                              0.43129
## Neg Pred Value                              0.90604
## Prevalence                                  0.13719
## Detection Rate                              0.05527
## Detection Prevalence                        0.12815
## Balanced Accuracy                           0.65921
##                      Class: Checking or savings account Class: Consumer Loan
## Sensitivity                                      0.7220              0.37223
## Specificity                                      0.8847              0.94459
## Pos Pred Value                                   0.7627              0.39123
## Neg Pred Value                                   0.8611              0.94022
## Prevalence                                       0.3391              0.08731
```

```
## Detection Rate                                   0.2449          0.03250
## Detection Prevalence                             0.3211          0.08307
## Balanced Accuracy                                0.8034          0.65841
##                      Class: Money transfers Class: Other financial service
## Sensitivity                        0.645485                      0.2586207
## Specificity                        0.975601                      0.9911255
## Pos Pred Value                     0.269930                      0.0724638
## Neg Pred Value                     0.994947                      0.9979987
## Prevalence                         0.013783                      0.0026737
## Detection Rate                     0.008897                      0.0006915
## Detection Prevalence               0.032960                      0.0095422
## Balanced Accuracy                  0.810543                      0.6248731
##                      Class: Payday loan Class: Student loan
## Sensitivity                     0.63897              0.8939
## Specificity                     0.97920              0.9732
## Pos Pred Value                  0.33433              0.9240
## Neg Pred Value                  0.99401              0.9618
## Prevalence                      0.01609              0.2672
## Detection Rate                  0.01028              0.2389
## Detection Prevalence            0.03075              0.2585
## Balanced Accuracy               0.80908              0.9336
##                      Class: Vehicle loan or lease Class: Virtual currency
## Sensitivity                                0.6405               0.3333333
## Specificity                                0.9448               0.9991701
## Pos Pred Value                             0.6471               0.0526316
## Neg Pred Value                             0.9433               0.9999077
## Prevalence                                 0.1364               0.0001383
## Detection Rate                             0.0874               0.0000461
## Detection Prevalence                       0.1351               0.0008759
## Balanced Accuracy                          0.7927               0.6662517
```