Homework 10

Miles Tweed

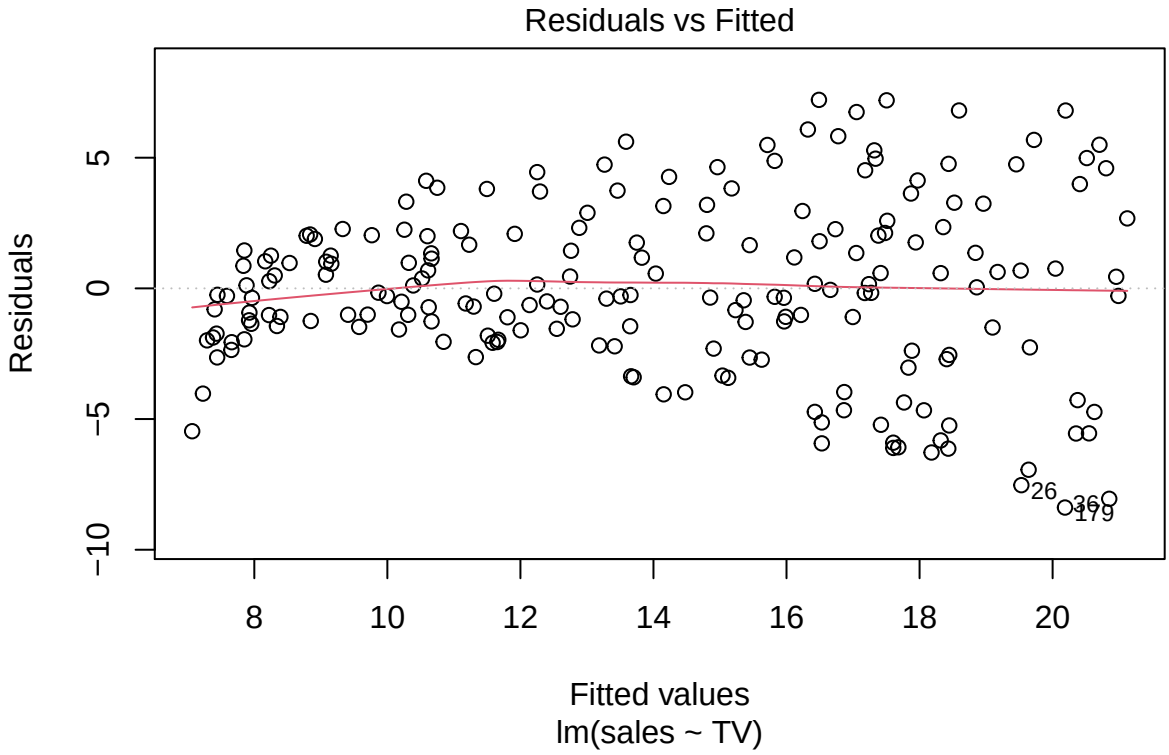
4/25/2021

Problem 1

Part 1

a.

```
ads <- read.csv('../../Data/Advertising.csv')
lm.obj <- lm(sales ~ TV, data=ads)
plot(lm.obj, which = 1)
```

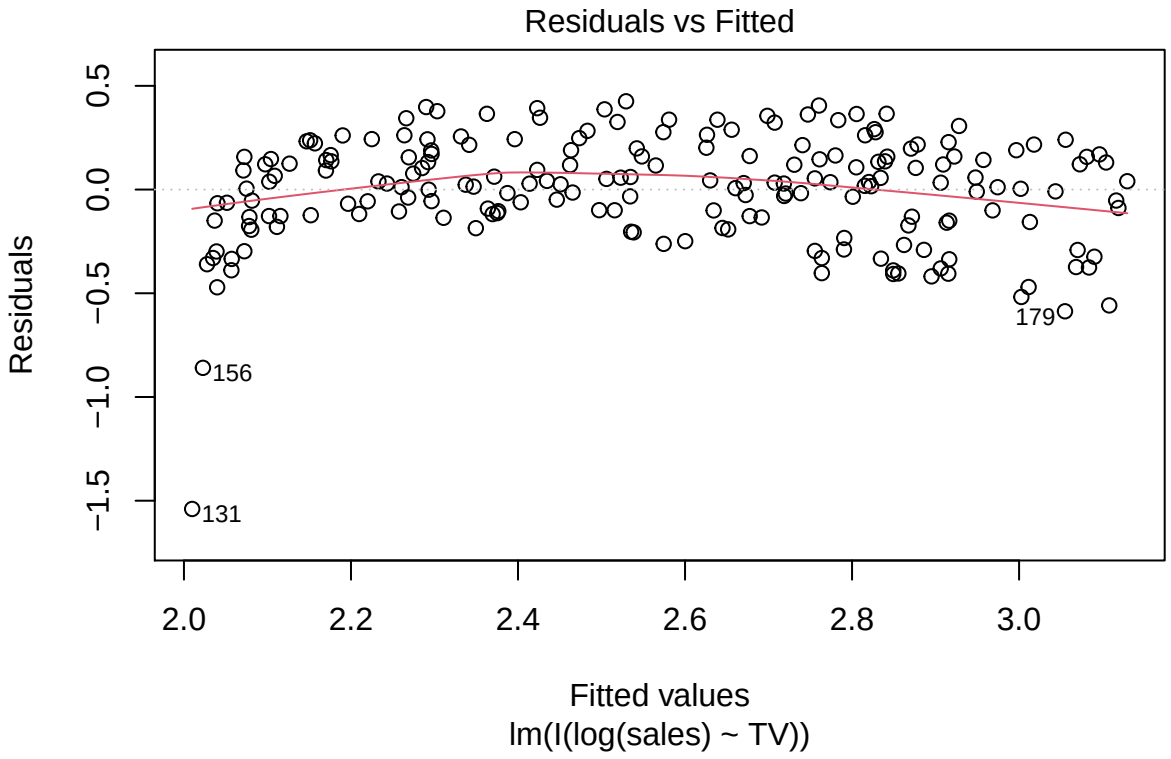


The residuals have heteroscedasticity or non-constant variance. We can fix this by fitting the predictor variable to the log or square root of the response.

```
# Modifying the model to use the log of the response
lm.obj <- lm(I(log(sales) ~ TV), data=ads)
```

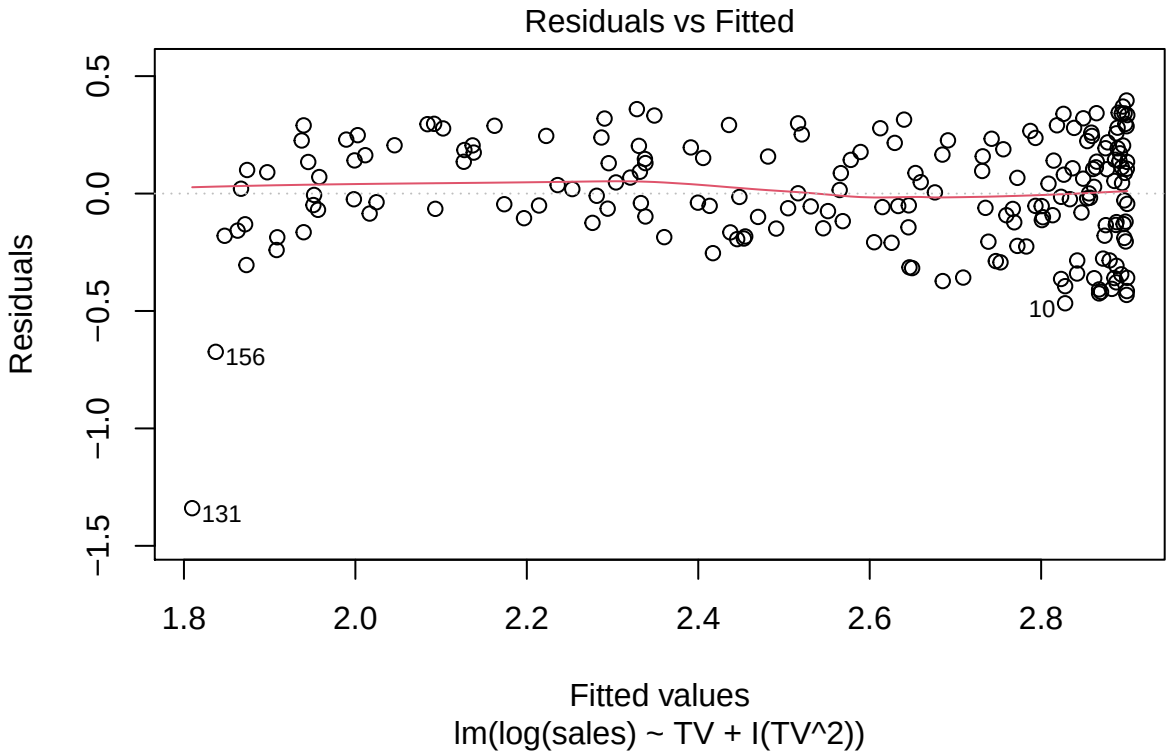
b.

```
# Residual Vs Fitted for modifies model
plot(lm.obj, which = 1)
```



The plot shows a curved fit around the mean=0 line. We can assess a quadratic model to remedy this/

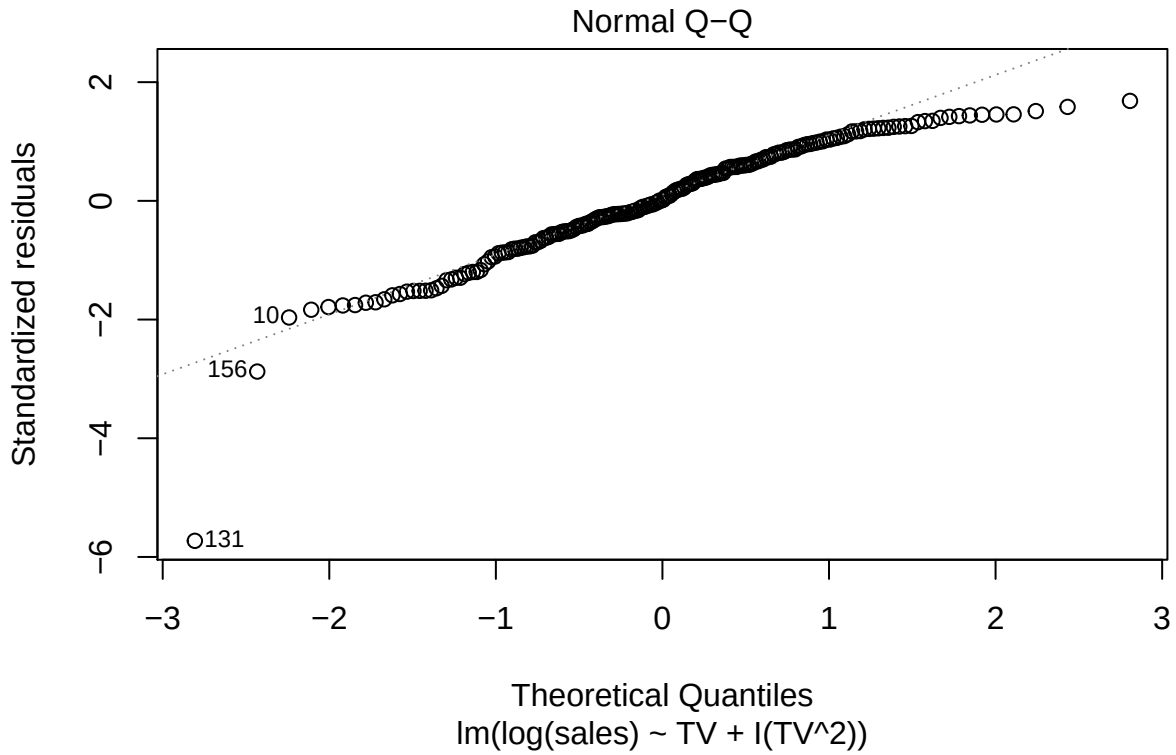
```
# Modifying the model to be quadratic
lm.obj <- lm(log(sales)~TV+I(TV^2), data=ads)
plot(lm.obj, which = 1)
```



The fit of the residuals is now much more close to the mean=0 line.

c.

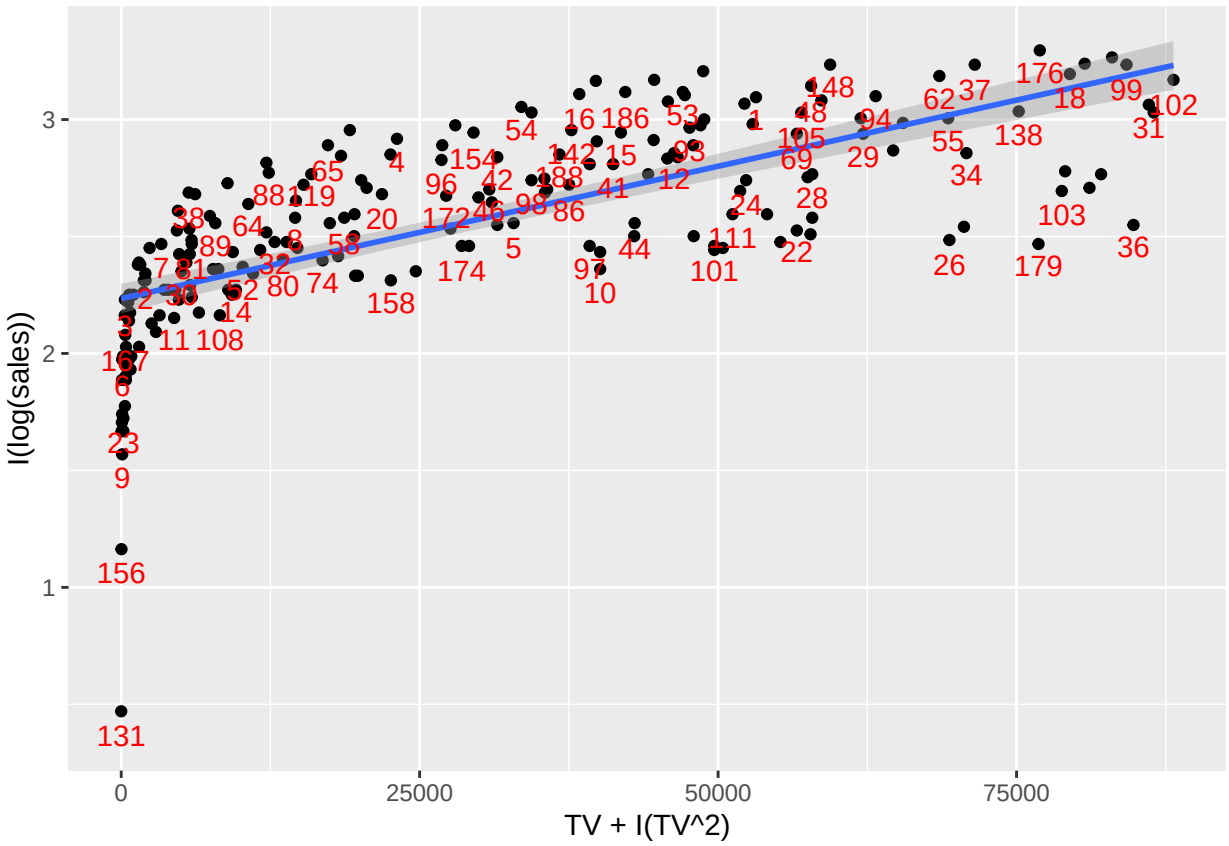
```
plot(lm.obj, which=2)
```



The residuals are not quite normally distributed since it deviated from the regression at the edges. This is not a big problem in this case since the sample size is large enough (>50).

d

```
library(tidyverse)
ggplot(data = ads, aes(y=I(log(sales)), x = TV + I(TV^2))) +
  geom_point() +
  geom_smooth(method = 'lm') +
  geom_text(aes(y = jitter(I(log(sales)), amount = 0.01),
                label = rownames(ads)),
                color = 'red',
                check_overlap = TRUE,
                nudge_y = -0.1)
```



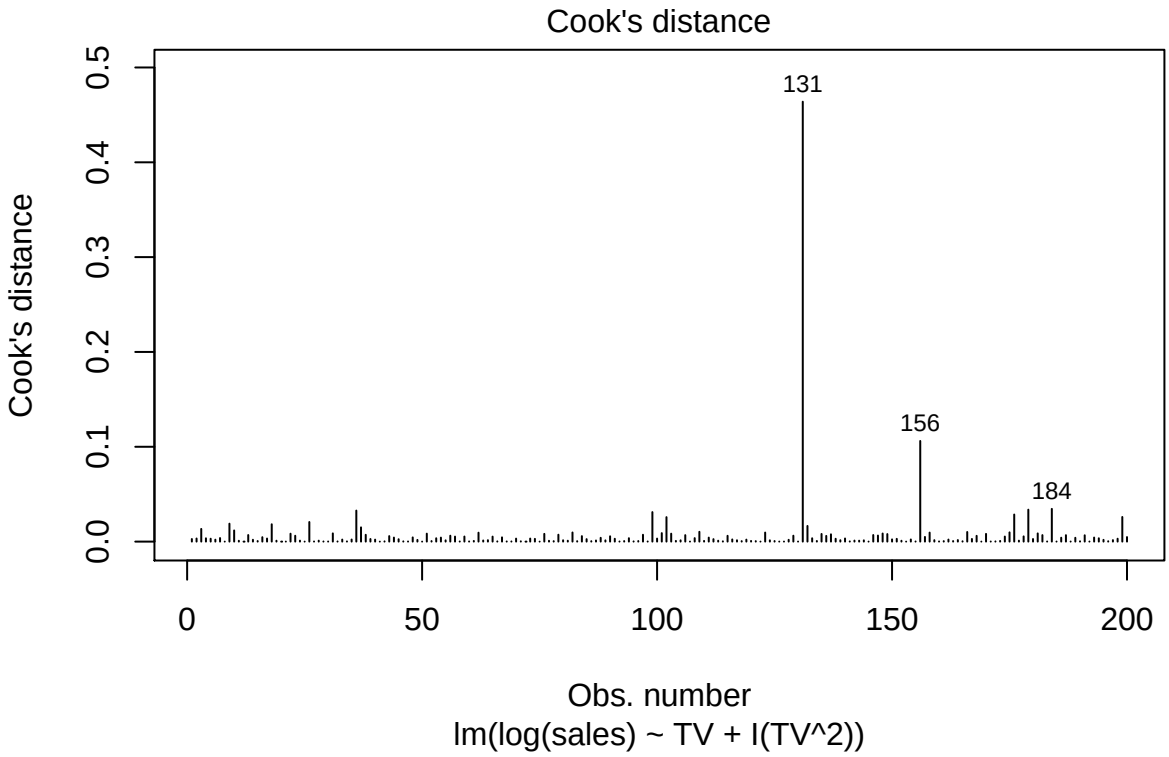
1) regression outliers: any point that is located far from the regression line, especially points 131, 156, 9, 23

2) high leverage points: any point that is towards the upper and lower ends of the x range, especially the points 131, 156, 9 23

3 Influential outliers: any point that is both far from the regression line and also a high leverage point, 131, 156, 9, 23

e. I am choosing the point 131 as the most influential regression outlier.

```
plot(lm.obj, which = 4)
```




```
# With observation
lm.obj.1 <- lm(log(sales)~TV+I(TV^2), data=ads)
sum.one <- summary(lm.obj.1)
coefs1 <- lm.obj.1$coefficients
r2.1 <- sum.one$r.squared
rse1 <- sqrt(deviance(lm.obj.1)/df.residual(lm.obj.1))

print(coefs1)
```

```
##   (Intercept)            TV       I(TV^2)
##  1.803910e+00  8.153267e-03 -1.515409e-05
```

```
print(paste("R^2 without:", r2.1, "RSE without:", rse1))
```

```
## [1] "R^2 without: 0.671225818597951 RSE without: 0.238786824966898"
```

```
# Without observation
lm.obj.2 <- lm(log(sales)~TV+I(TV^2), data=ads[-131,])
sum.two <- summary(lm.obj.2)
coefs2 <- lm.obj.2$coefficients
r2.2 <- sum.two$r.squared
rse2 <- sqrt(deviance(lm.obj.2)/df.residual(lm.obj.2))

print(coefs2)
```

```
##   (Intercept)            TV       I(TV^2)
##  1.861285e+00  7.371516e-03 -1.292457e-05
```

```
print(paste("R^2 without:", r2.2, "RSE without:", rse2))
```

```
## [1] "R^2 without: 0.685400643737174 RSE without: 0.218553909372783"
```

Fitted Equation With Outlier

$$\log(\hat{sales}) = 1.804 + 0.00815 \cdot TV - 1.52 \cdot 10^{-5} \cdot TV^2$$

Fitted Equation Without Outlier

$$\log(\hat{sales}) = 1.861 + 0.00737 \cdot TV - 1.29 \cdot 10^{-5} \cdot TV^2$$

The model did not change much by removing this one outlier. The R^2 changed from 0.67 to 0.69 which means that the model without the outlier explains two percent more variance in the response than the model with the outlier. The RSE changed from 0.239 to 0.219.

f. I think it would be fine to include this observation since it does not seem to change the model much. Potentially it would be more effective to remove other influential outliers as well though.

Part 2

a.

```
lm.obj <- lm(sales ~ TV + radio + newspaper, data = ads)

step(lm.obj)
```

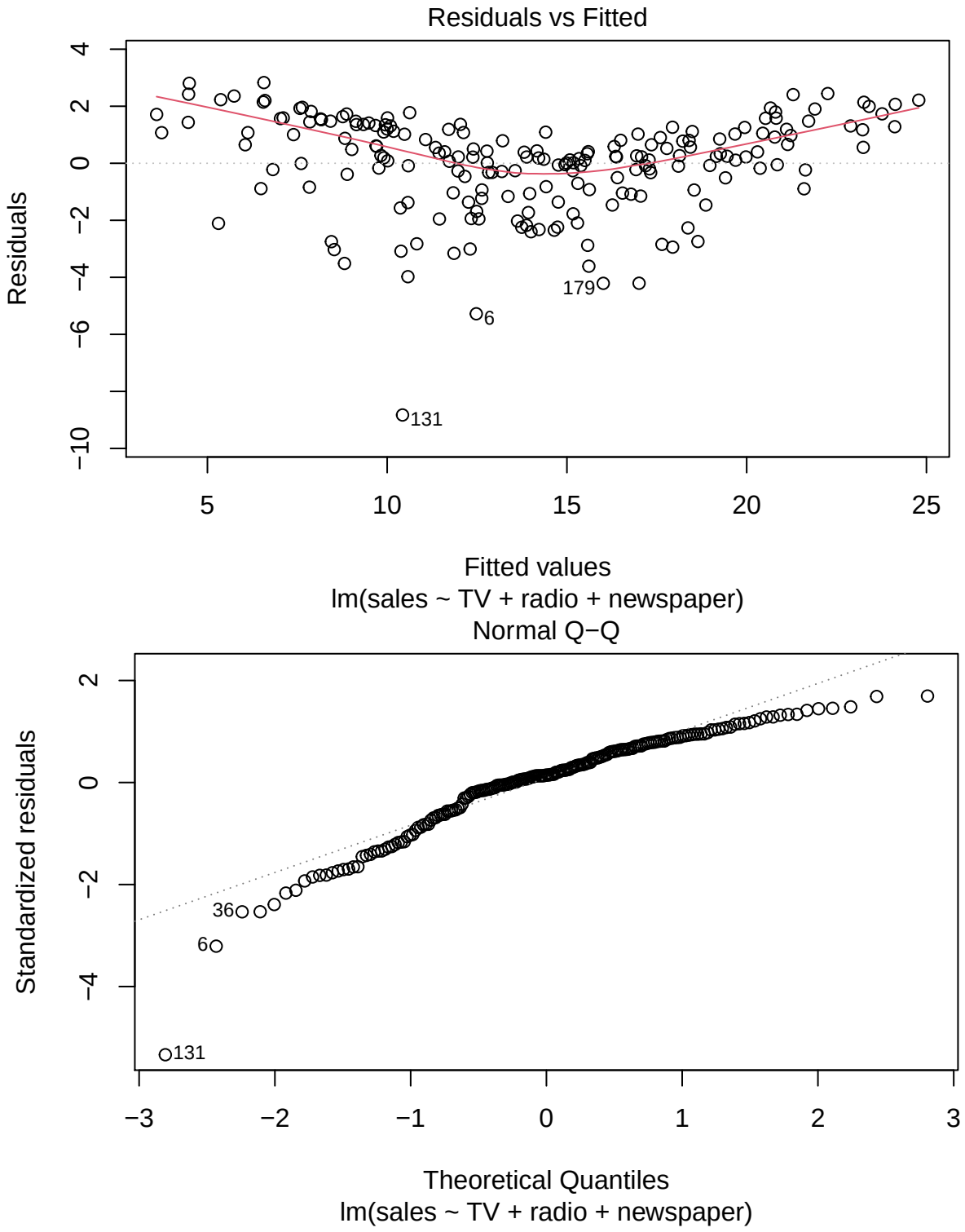
```
## Start:  AIC=212.79
## sales ~ TV + radio + newspaper
##
##              Df Sum of Sq    RSS    AIC
## - newspaper  1      0.09  556.9 210.82
## <none>                     556.8 212.79
## - radio      1   1361.74 1918.6 458.20
## - TV         1   3058.01 3614.8 584.90
##
## Step:  AIC=210.82
## sales ~ TV + radio
##
##         Df Sum of Sq    RSS    AIC
## <none>                556.9 210.82
## - radio  1    1545.6 2102.5 474.52
## - TV     1    3061.6 3618.5 583.10
```

```
##
## Call:
## lm(formula = sales ~ TV + radio, data = ads)
##
## Coefficients:
## (Intercept)           TV        radio
##     2.92110      0.04575      0.18799
```

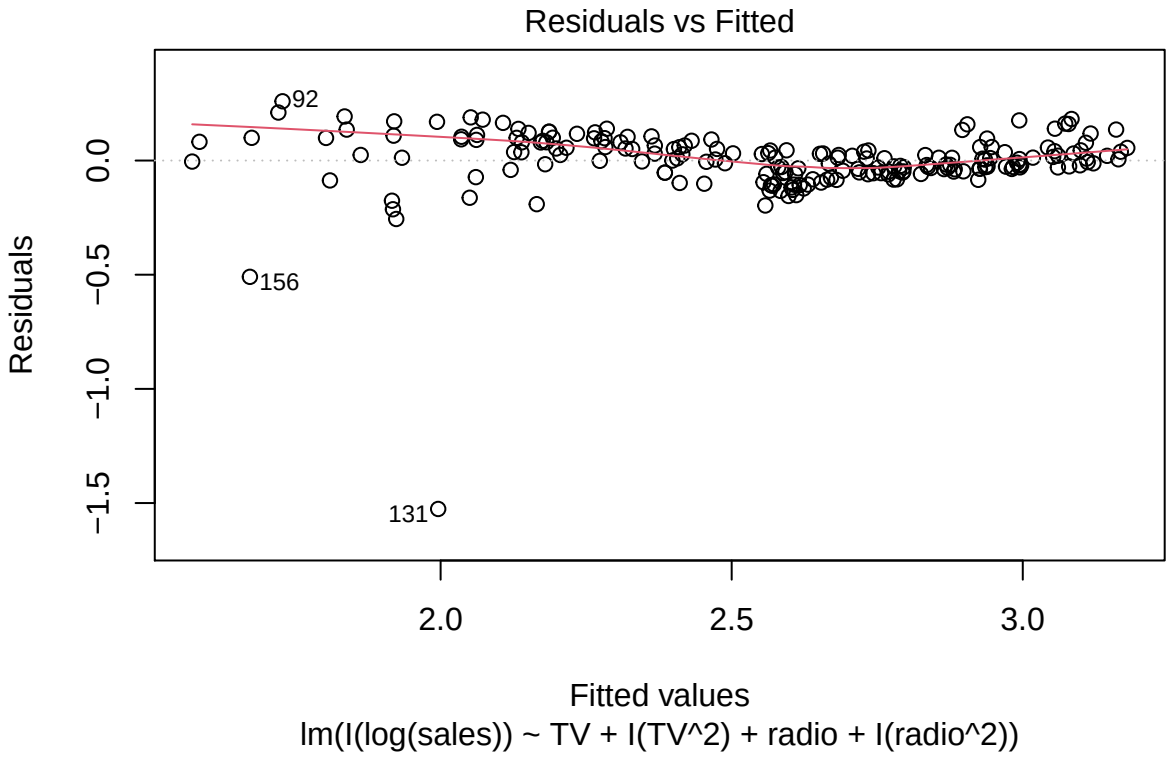
The newspaper variable was dropped using backward selection.

b.

```
plot(lm.obj, which = c(1,2))
```




```
lm.obj <- lm(I(log(sales)) ~ TV + I(TV^2)  + radio + I(radio^2) , data = ads)
plot(lm.obj, which = 1)
```



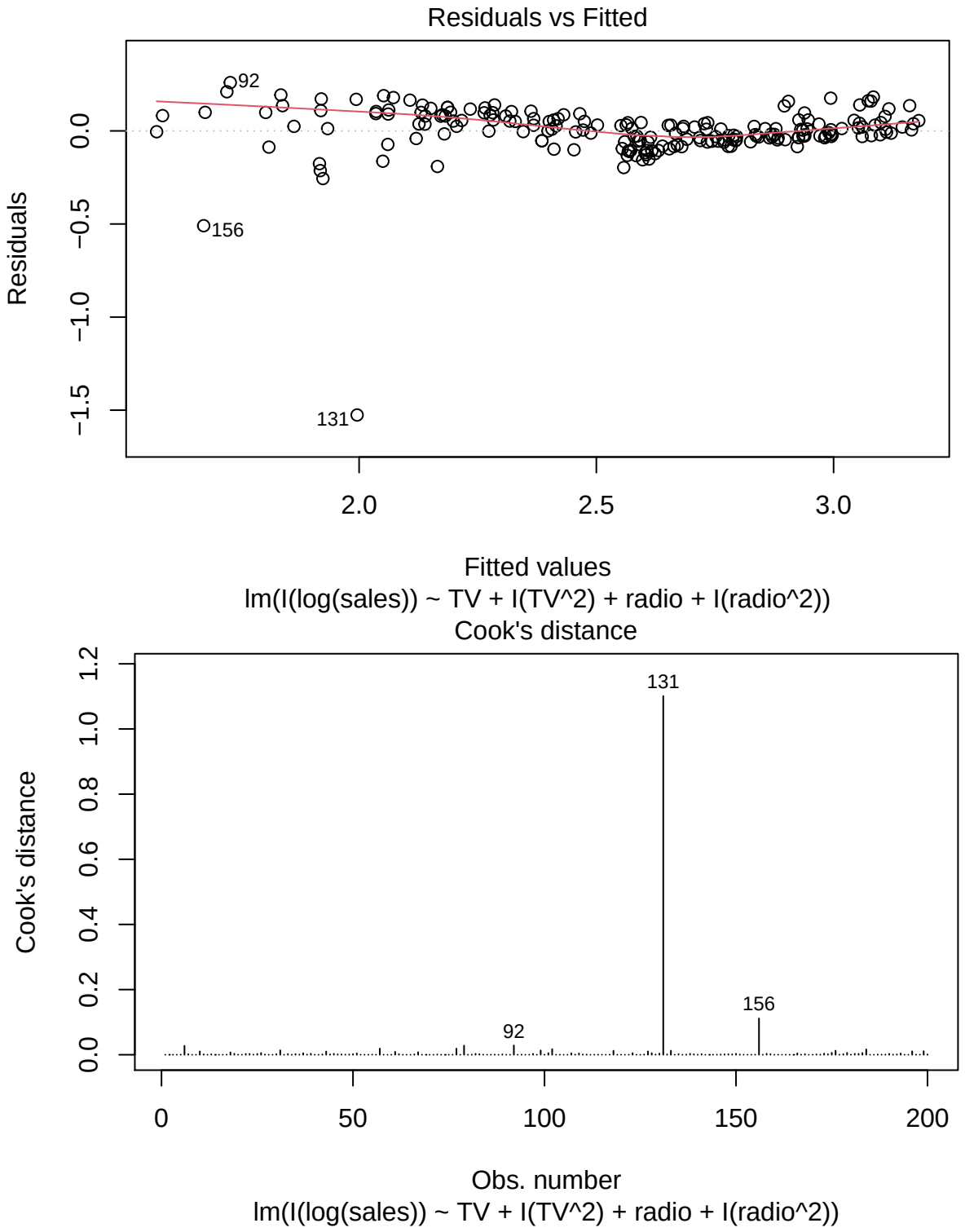
c.

1) For regression outliers we can observe the points that have the largest value greater than or less than zero in the regression vs fitted plot (92,131,6 in this case)

2) For the high leverage point we would choose points that are at the extremes of the x ranges 0.7 to 296.4 for TV and 0.0 to 49.6 for radio.

3) For the influential outliers we could identify points that are both regression outliers and high leverage or we can look at the Cook's distance in the plot below. In this case the most influential outlier is observation 131

```
plot(lm.obj, which = c(1,4))
```



d.

The observation 131 is the most influential outlier according to the Cook's distance.

```
# With observation
lm.obj.1 <- lm(I(log(sales)) ~ TV + I(TV^2)  + radio + I(radio^2), data=ads)
sum.one <- summary(lm.obj.1)
```

```
coefs1 <- lm.obj.1$coefficients
r2.1 <- sum.one$r.squared
rse1 <- sqrt(deviance(lm.obj.1)/df.residual(lm.obj.1))

print(coefs1)
```

```
##   (Intercept)            TV       I(TV^2)         radio    I(radio^2)
##  1.466391e+00  8.887187e-03 -1.812498e-05  1.524304e-02 -5.126426e-05
```

```
print(paste("R^2 without:", r2.1, "RSE without:", rse1))
```

```
## [1] "R^2 without: 0.878911207615597 RSE without: 0.145656305408956"
```

```
# Without observation
lm.obj.2 <- lm(I(log(sales)) ~ TV + I(TV^2)  + radio + I(radio^2), data=ads[-131,])
sum.two <- summary(lm.obj.2)
coefs2 <- lm.obj.2$coefficients
r2.2 <- sum.two$r.squared
rse2 <- sqrt(deviance(lm.obj.2)/df.residual(lm.obj.2))

print(coefs2)
```

```
##  (Intercept)           TV      I(TV^2)        radio   I(radio^2)
##  1.519968022  0.008023864 -0.000015700  0.015589803 -0.000047217
```

```
print(paste("R^2 without:", r2.2, "RSE without:", rse2))
```

```
## [1] "R^2 without: 0.9429600463634 RSE without: 0.0935397957222308"
```

Fitted Equation With Outlier

$$\log(\hat{sales}) = 1.466 + 0.00889 \cdot TV - 1.81 \cdot 10^{-5} \cdot TV^2 + 0.0152 \cdot radio - 5.13 \cdot 10^{-5} \cdot radio^2$$

Fitted Equation Without Outlier

$$\log(\hat{sales}) = 1.520 + 0.00802 \cdot TV - 0.0000157 \cdot 10^{-5} \cdot TV^2 + 0.0156 \cdot radio - 0.0000472 \cdot 10^{-5} \cdot radio^2$$

The R^2 changed from 0.88 to 0.94 which means that the model without the outlier explains six percent more variance in the response than the model with the outlier. The RSE changed from 0.146 to 0.094.

e.

Since removing this outlier resulted in a more substantial change in the model I think it would be best to exclude this observation.

Problem 2

Part 1

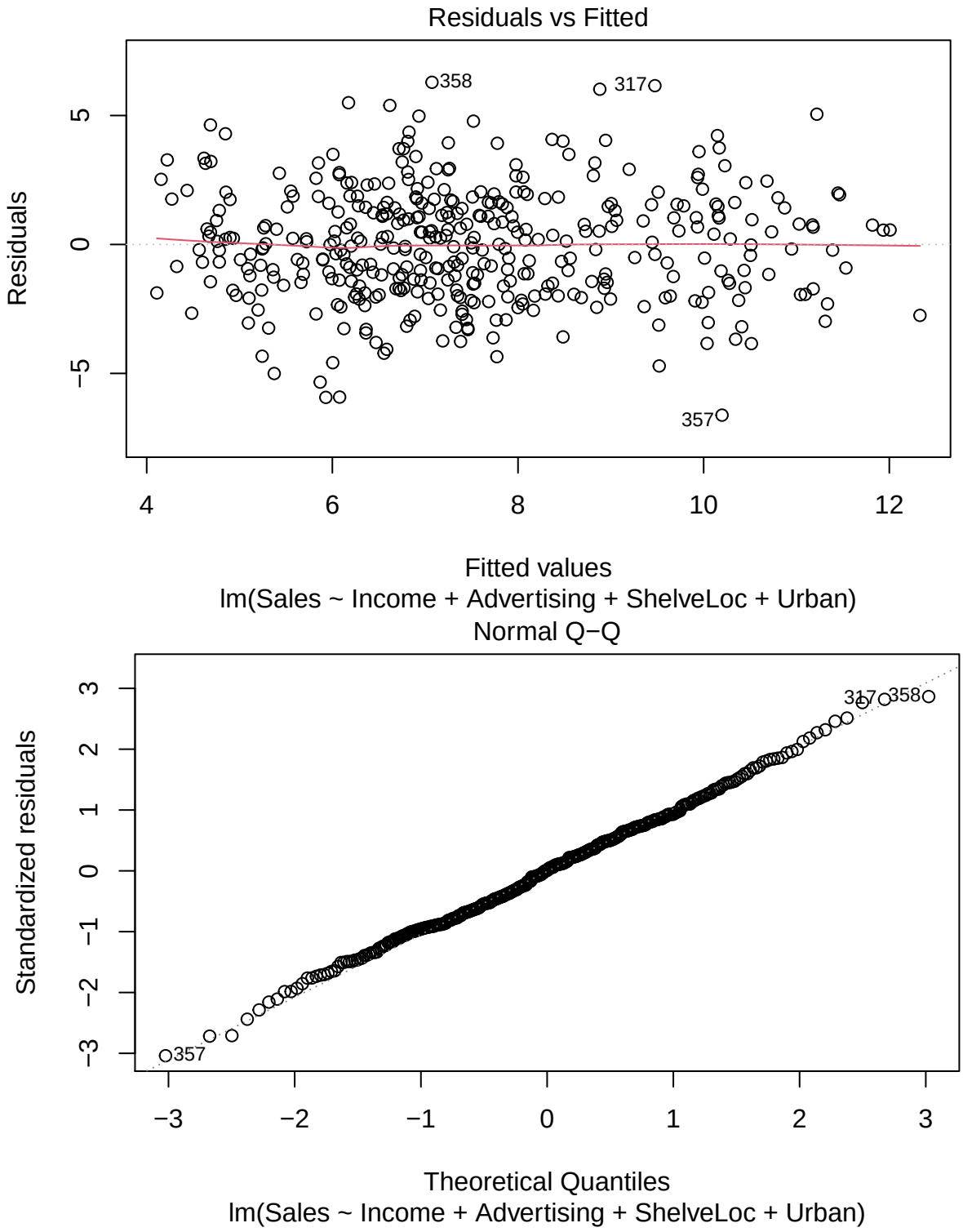
```
library(ISLR)

lm.obj <- lm(Sales ~ Income + Advertising + ShelveLoc + Urban, data = Carseats)
```

a.

$$Sales_i = \beta_0 + \beta_1 Income_i + \beta_2 Advertising_i + \beta_3 ShelveLoc_i + \beta_4 Urban_i + \epsilon_i,\ \epsilon_i \sim \mathbb{N}(0, \sigma^2)$$

b.

```
plot(lm.obj, which = c(1,2))
```



Yes, the modelling assumptions do hold in this case.

c.

We can rely on the p-values provided by classic inference here because the modelling assumption $\epsilon_i \sim \mathbb{N}(0, \sigma^2)$

holds true.

```
summary(lm.obj)
```

```
##
## Call:
## lm(formula = Sales ~ Income + Advertising + ShelveLoc + Urban,
##     data = Carseats)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.6178 -1.5492  0.0182  1.4956  6.2868
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.696673   0.411685   8.979  < 2e-16 ***
## Income          0.016499   0.003958   4.169 3.77e-05 ***
## Advertising     0.096308   0.016650   5.784 1.49e-08 ***
## ShelveLocGood   4.656799   0.329846  14.118  < 2e-16 ***
## ShelveLocMedium 1.837221   0.270946   6.781 4.39e-11 ***
## UrbanYes        0.045990   0.242571   0.190     0.85
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.202 on 394 degrees of freedom
## Multiple R-squared:  0.3999, Adjusted R-squared:  0.3923
## F-statistic: 52.51 on 5 and 394 DF,  p-value: < 2.2e-16
```

The two most significant predictors are the ones that indicate the categorical quality of the shelving location for the car seat. Holding income, advertising, and Urban location constant, a car seat with a good shelve location will sell $4.656 \cdot 1000 = 4,656$ more units than one with a bad shelving location on average. Holding income, advertising, and Urban location constant, a car seat with a medium shelve location will sell $1.837 \cdot 1000 = 1,837$ more units than one with a bad shelving location on average.

Part 2

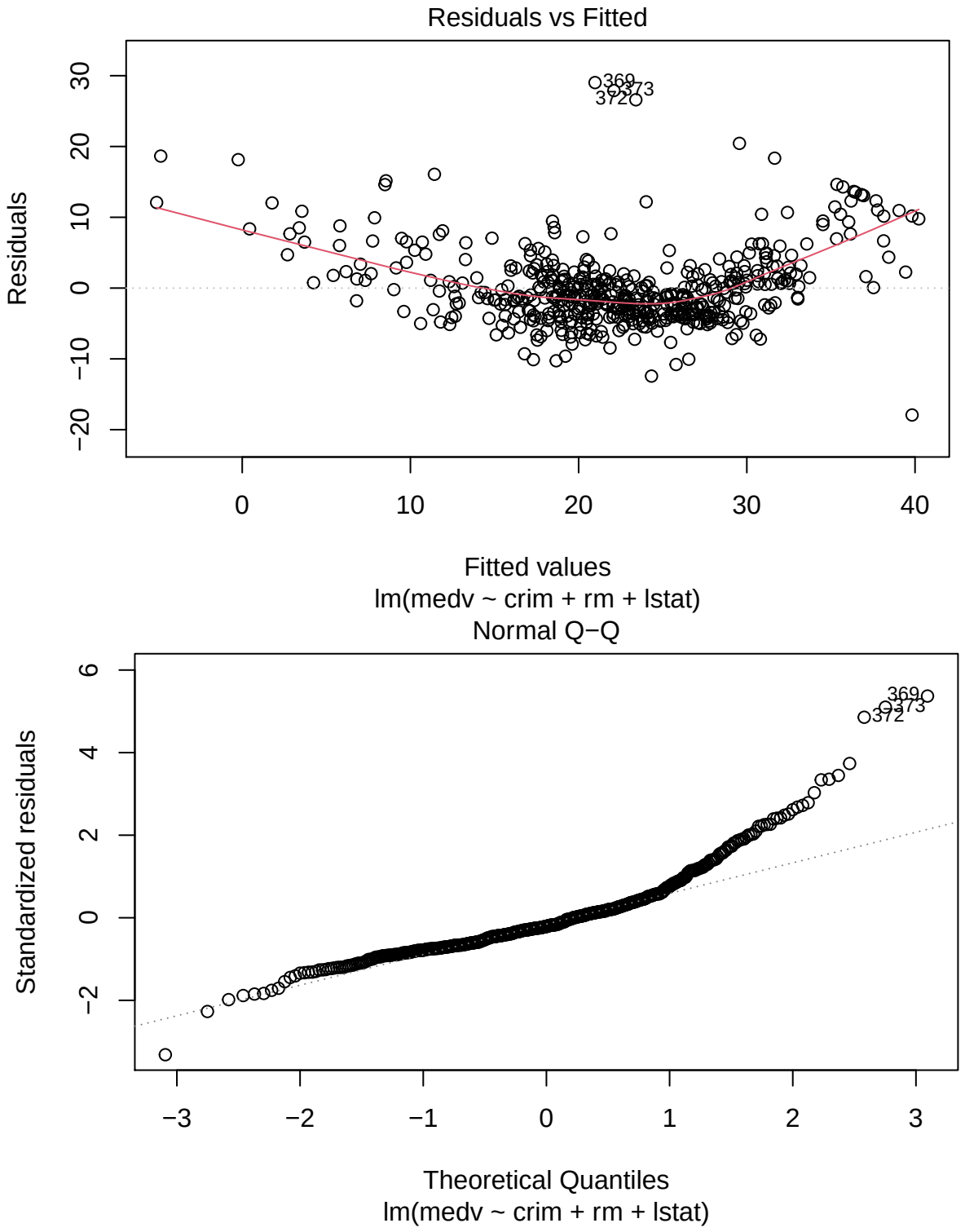
```
library(MASS)
lm.obj <- lm(medv~crim + rm + lstat, data = Boston)
```

a.

$$medv_i = \beta_0 + \beta_1 crim_i + \beta_2 rm_i + \beta_3 lstat_i + \epsilon_i, \ \mathbb{N}(0, \sigma^2)$$

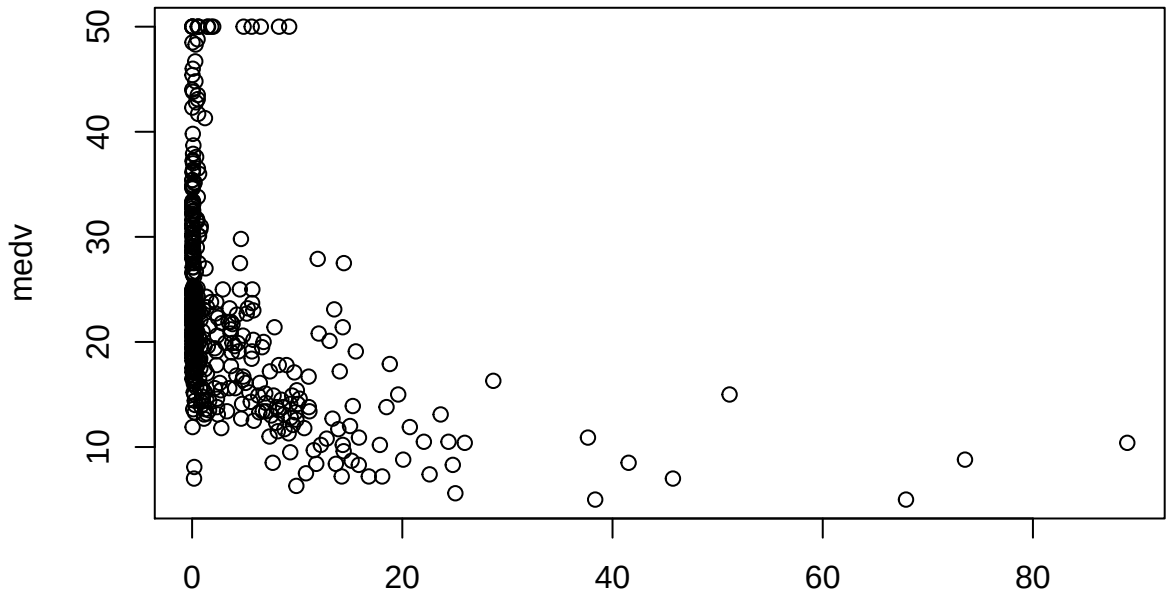
b.

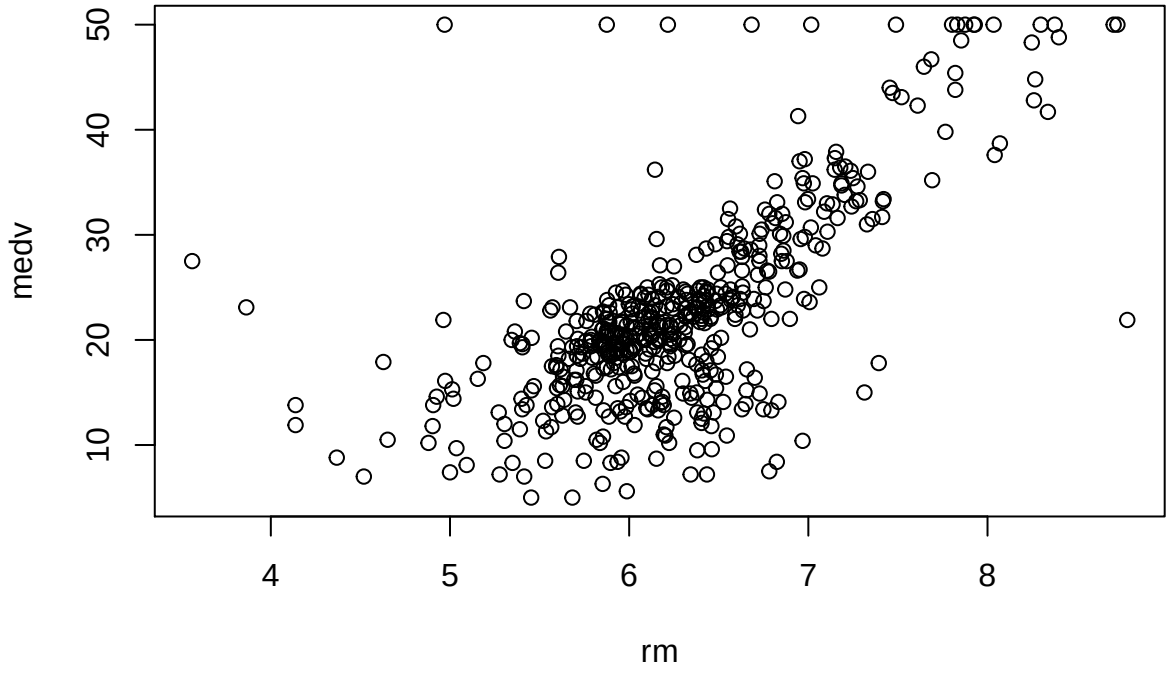
```
plot(lm.obj, which = c(1,2))
```

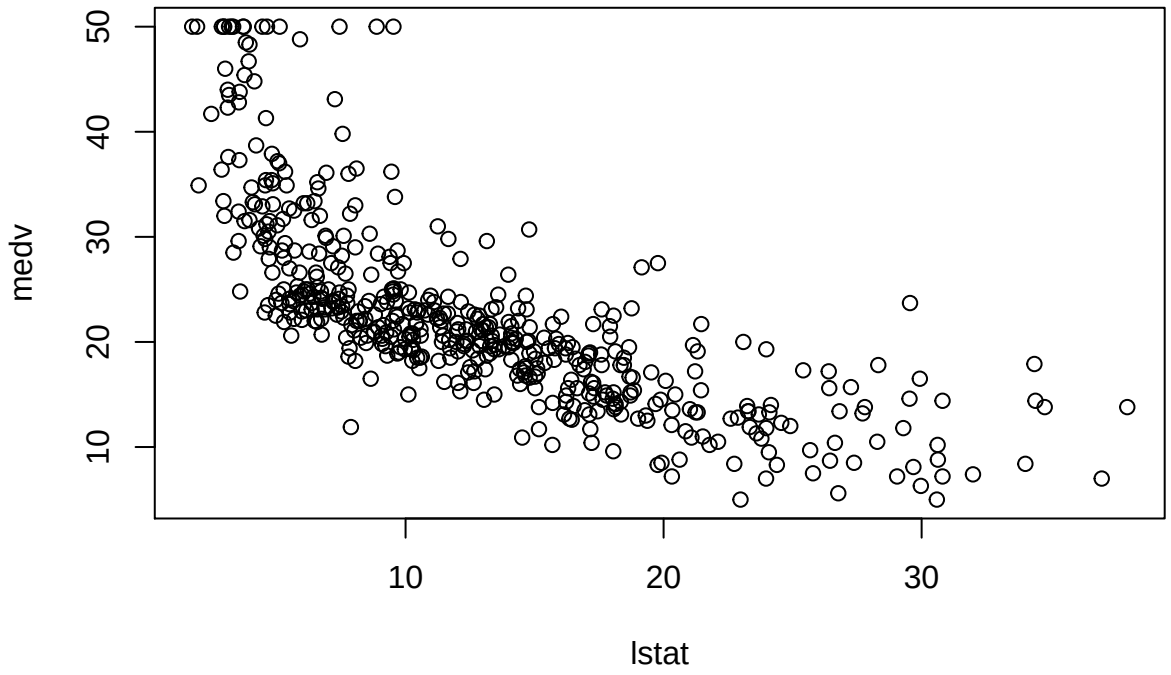


Here the modelling assumptions do not hold true. The residual vs fitted shows a curved fit line that tends to sit above the mean=0 line and the Q-Q plot strays a lot at the end of the distribution.

```
plot(medv ~ crim + rm + lstat, data=Boston)
```

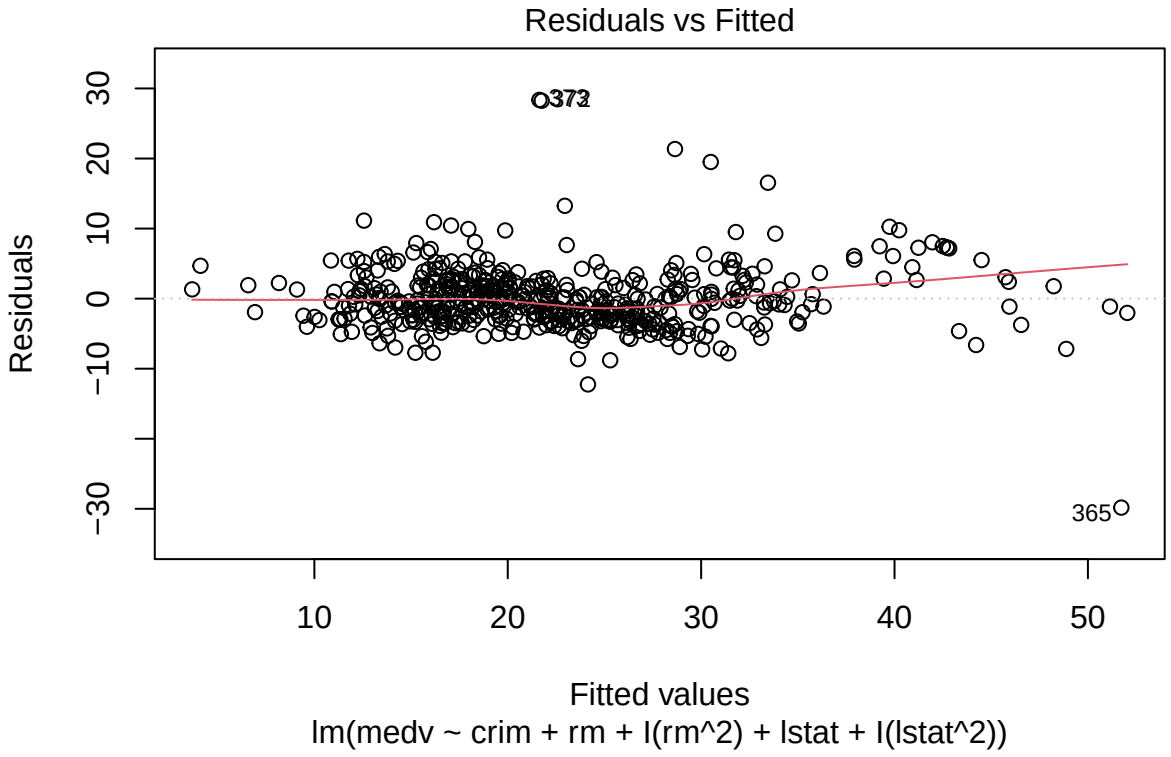

medv
10
20
30
40
50
0
20
40
60
80
crim

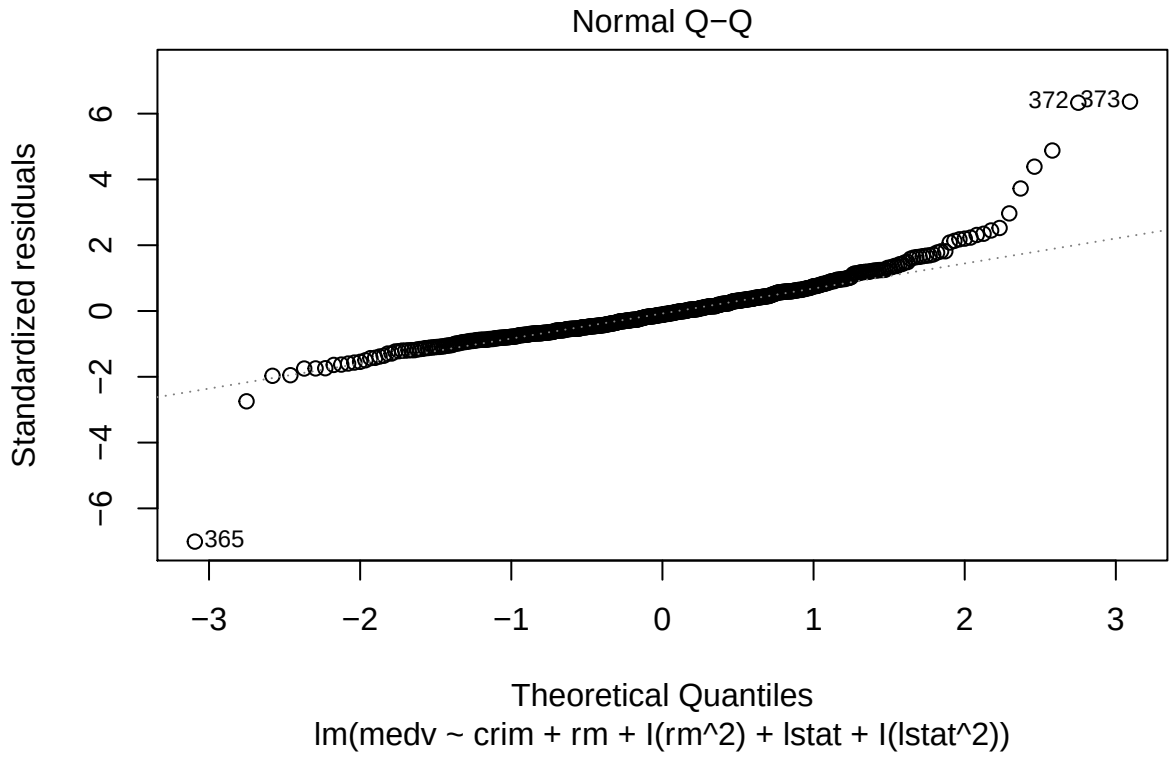

medv
10
20
30
40
50
4
5
6
7
8
rm



There seems to be a non-linear relationship with the lstat variable and also possibly the rm variable. Adding quadratic terms for these variable leads to a better result.

```
lm.obj <- lm(medv ~ crim + rm + I(rm^2) + lstat + I(lstat^2), data = Boston)

plot(lm.obj, which = c(1,2))
```





c.

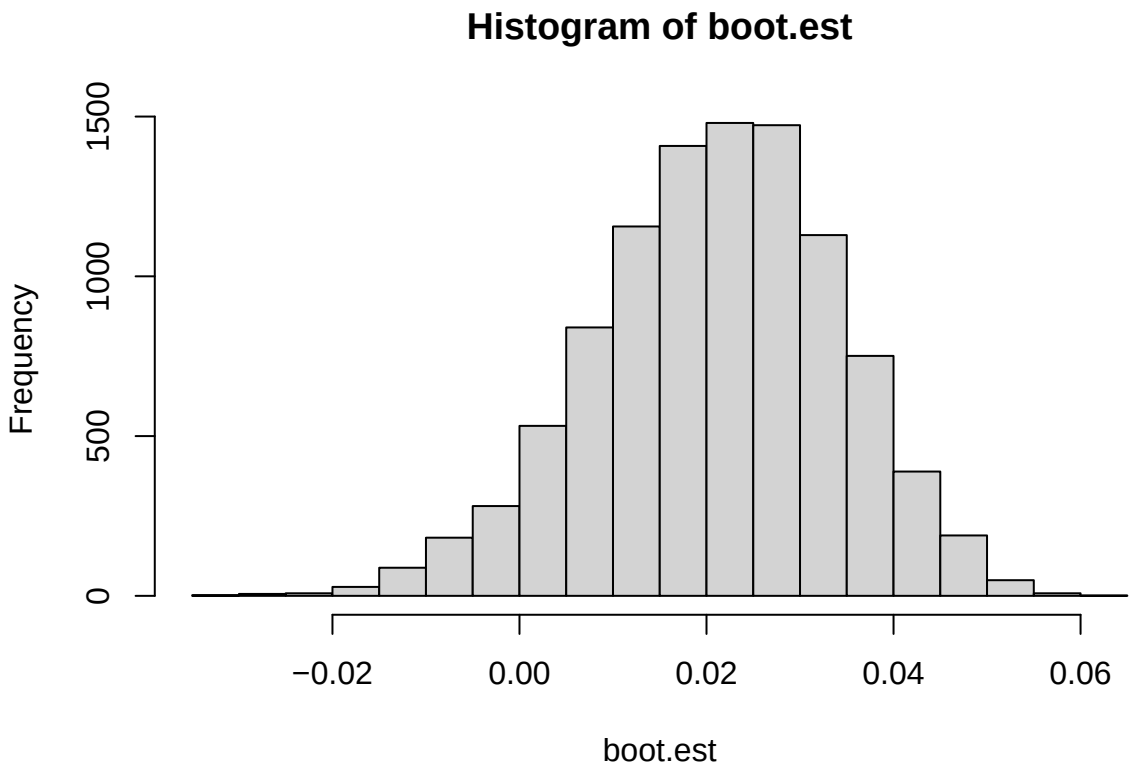
```
summary(lm.obj)
```

```
##
## Call:
## lm(formula = medv ~ crim + rm + I(rm^2) + lstat + I(lstat^2),
##     data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -29.8272  -2.6141  -0.4946   1.9367  28.3637
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 107.719490   9.659127  11.152  < 2e-16 ***
## crim         -0.147737   0.026242  -5.630 3.01e-08 ***
## rm          -27.122219   3.018635  -8.985  < 2e-16 ***
## I(rm^2)       2.453939   0.233615  10.504  < 2e-16 ***
## lstat        -1.351325   0.117355 -11.515  < 2e-16 ***
## I(lstat^2)    0.022595   0.003414   6.618 9.41e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.473 on 500 degrees of freedom
## Multiple R-squared:  0.7658, Adjusted R-squared:  0.7635
## F-statistic:   327 on 5 and 500 DF,  p-value: < 2.2e-16
```

All of the coefficients for each predictor were statistically significant at the $\alpha = 0.001$ level as was the model over all.

Problem 3

```
library(ISwR)
df <- thuesen

boot.est <- numeric(10000)

for (i in c(1:10000)) {
  inds <- sample(c(1:24), 24, replace = TRUE)
  data = df[inds,]
  lm.obj <- lm(short.velocity~blood.glucose, data = data)
  boot.est[i] <- lm.obj$coefficients[2]
}

hist(boot.est)
```




```
se <- sd(boot.est)/sqrt(length(boot.est))
conf.int <- c(mean(boot.est) - 1.96*se, mean(boot.est + 1.96*se))
print(paste("beta1:",round(mean(boot.est),5)))
```

```
## [1] "beta1: 0.02099"
```

```
print(paste("SE:", round(se, 5)))
```

```
## [1] "SE: 0.00013"
```

```
print(paste("95% confidence interval:", round(conf.int[1], 5),'-',round(conf.int[2], 5)))
```

```
## [1] "95% confidence interval: 0.02074 - 0.02125"
```

```
t.test(boot.est, numeric(10000))
```

```
##
##  Welch Two Sample t-test
##
## data:  boot.est and numeric(10000)
## t = 161.6, df = 9999, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.02073703 0.02124628
## sample estimates:
##  mean of x  mean of y
## 0.02099165 0.00000000
```

```
lm.obj <- lm(short.velocity~blood.glucose, data = df)
summary(lm.obj)
```

```
##
## Call:
## lm(formula = short.velocity ~ blood.glucose, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.40141 -0.14760 -0.02202  0.03001  0.43490
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.09781    0.11748   9.345 6.26e-09 ***
## blood.glucose  0.02196    0.01045   2.101   0.0479 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2167 on 21 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.1737, Adjusted R-squared:  0.1343
## F-statistic: 4.414 on 1 and 21 DF,  p-value: 0.0479
```

Since the bootstrap estimate is over 10000 replicates the standard error is very small (about 1% of the standard error for the classical approach) so the 95 confidence interval is narrow. The significance for the blood.glucose predictor would not change at the $\alpha = 0.05$ level since the classical approach was significant at this level but the bootstrap estimate is more significant.