Stats_101A_HW_2_Charles_Liu

Charles Liu (304804942)

January 31, 2020

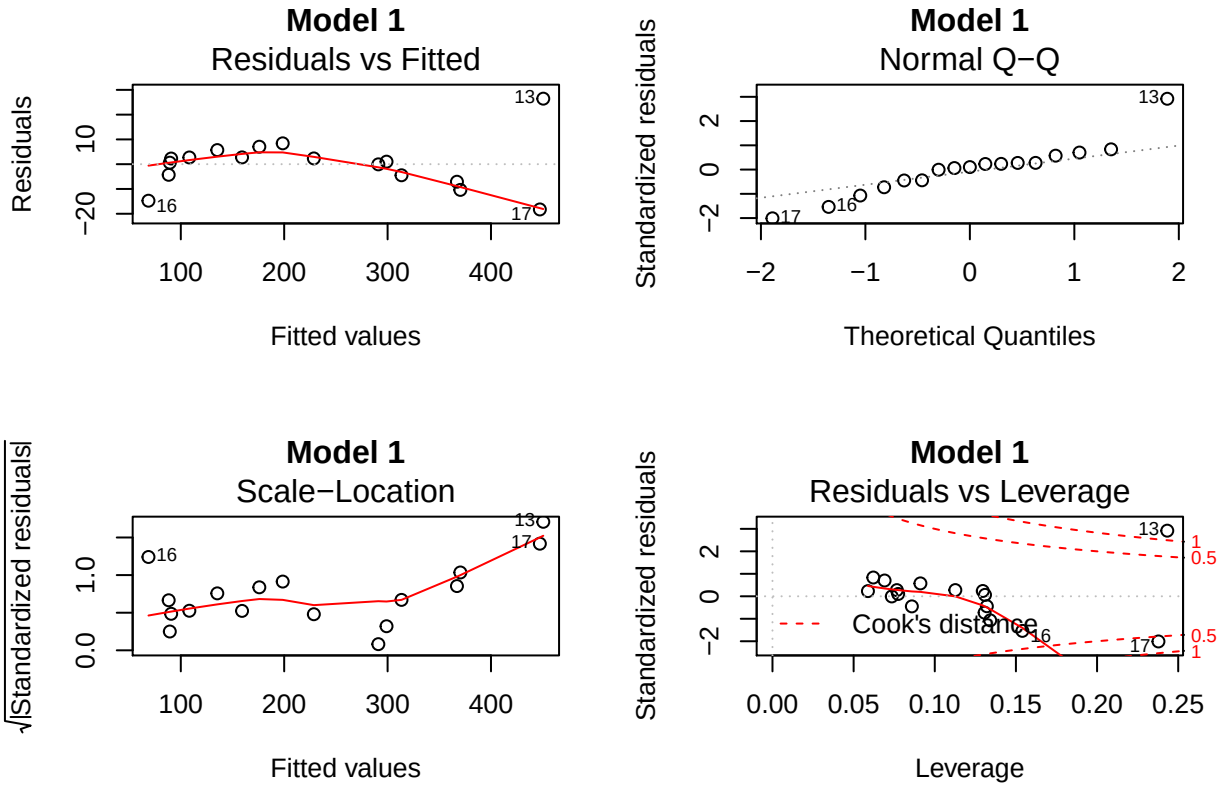
Problem 1

1a)

```
airdata <- read.table("airfares.txt", header = TRUE)
attach(airdata)
m1a <- lm(Fare ~ Distance)
summary(m1a)
```

```
##
## Call:
## lm(formula = Fare ~ Distance)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -18.265  -4.475   1.024   2.745  26.440
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 48.971770   4.405493   11.12 1.22e-08 ***
## Distance     0.219687   0.004421   49.69  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.41 on 15 degrees of freedom
## Multiple R-squared:  0.994,  Adjusted R-squared:  0.9936
## F-statistic:  2469 on 1 and 15 DF,  p-value: < 2.2e-16
```

```
par(mfrow = c(2, 2))
plot(m1a, main = "Model 1")
```



Here the intercept is 48.971. Interpretation is if the Distance travelled is zero also then also the fare is 48.971 and the slope is 0.2196. interpretation is if the distance is increased by one unit then the fare will be increased by 0.2196 unit.

We can see that the R-Squared for both Multiple and Adjusted is aprroximately 99.4% variation of Fare is explained by the Distance. We can therefore say this is a very good model to use numerically.

The Intercept and Distance have p-values of 1.22e-08 & $<$ 2e-16, respectively. We can say that they are statistically significant.

We conclude that we can use Distance to predict Fare. Therefore, we can the critique of the model is correct.

However, We need to transform variables since it doesn't satisfy the linear assumptions.

1b)

From the scatterplot of Distance that measures Fare, we can see there is a strong correlation relationship between these two variables. As Distance increases (x-axis), we can see Fare (y-axis) increase along with our Predictor. From the residual plot, it shows a non-random pattern. If we apply the log transformation to the Distance, we can create a better model.

Problem 2

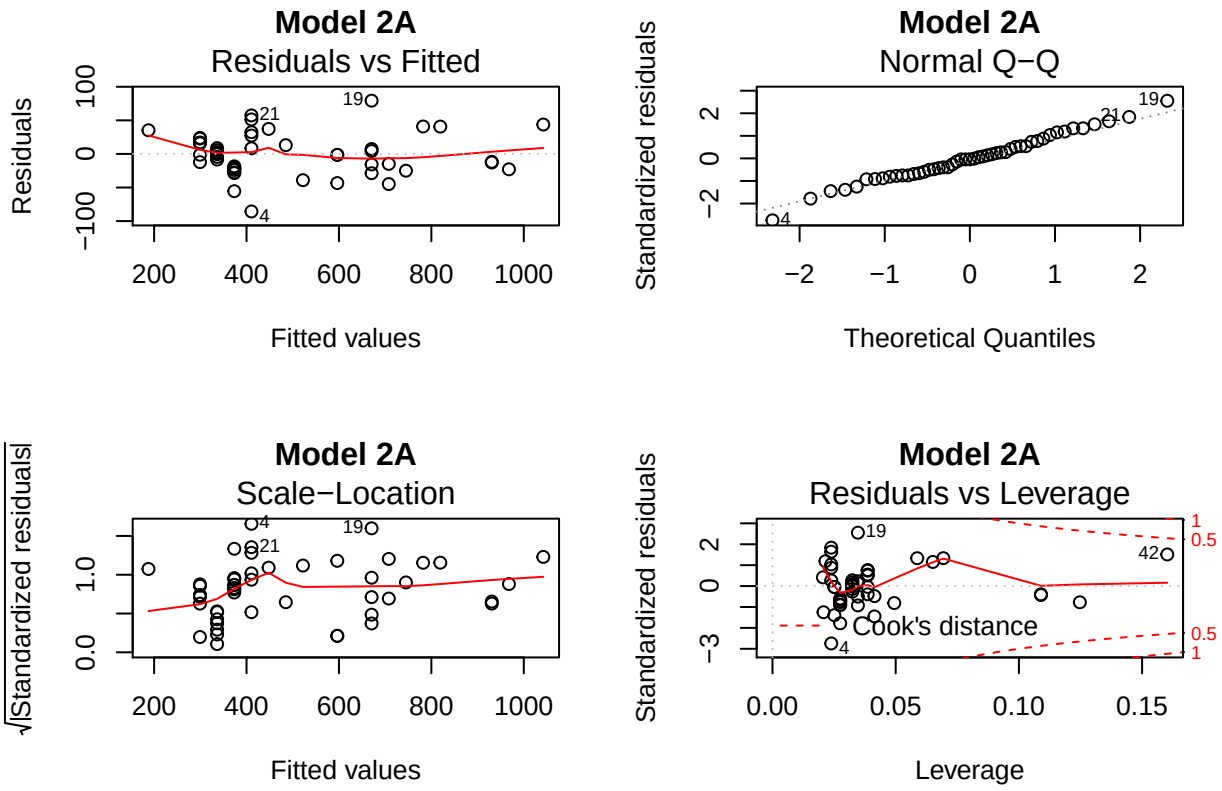
2a.1)

```
diamdata <- read.table("diamonds.txt", header = TRUE)
attach(diamdata)
```

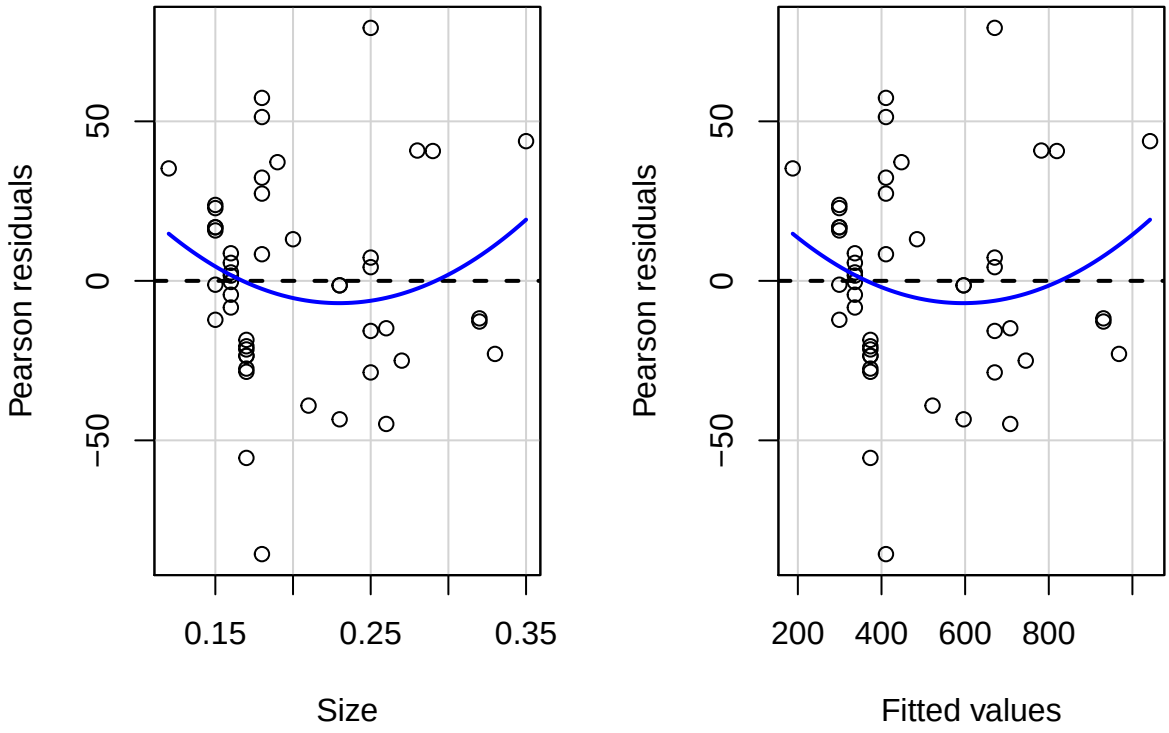
```r
m2a1 <- lm(Price ~ Size)
summary(m2a1)
```

```
##
## Call:
## lm(formula = Price ~ Size)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -85.654 -21.503  -1.203  16.797  79.295
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -258.05      16.94  -15.23   <2e-16 ***
## Size         3715.02      80.41   46.20   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 31.6 on 47 degrees of freedom
## Multiple R-squared:  0.9785, Adjusted R-squared:  0.978
## F-statistic:  2135 on 1 and 47 DF,  p-value: < 2.2e-16
```

```r
par(mfrow = c(2, 2))
plot(m2a1, main = "Model 2A")
```




```r
residualPlots(m2a1)
```




```
##            Test stat Pr(>|Test stat|)
## Size          1.2506           0.2174
## Tukey test    1.2506           0.2111
```

2a.2)

Some weaknesses include that the data it is also seen that the model is not perfectly linear, thus it may not give an accurate representation of the model. Price can depend on many things besides Size, and this may lead to some misleading results.

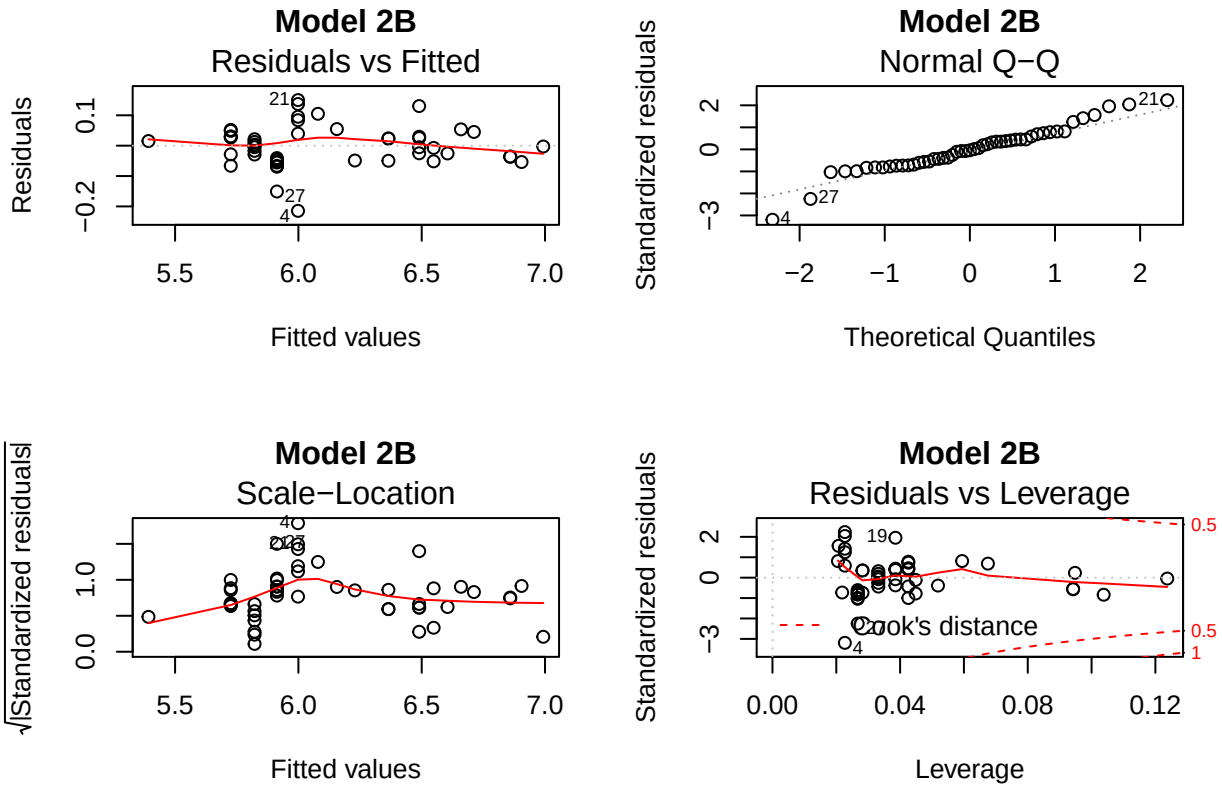
2b.1)

```
m2b1 <- lm(log(Price) ~ log(Size))
summary(m2b1)
```

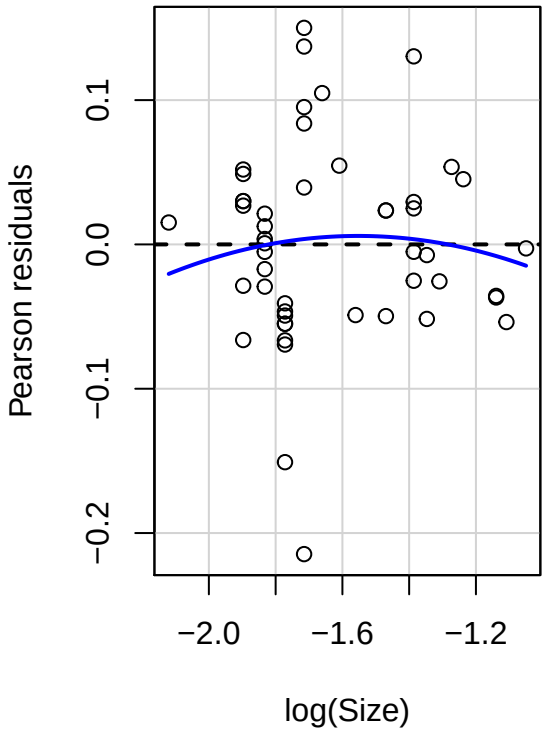
```
##
## Call:
## lm(formula = log(Price) ~ log(Size))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.21460 -0.04646 -0.00274  0.03001  0.15005
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.56317    0.06221  137.65   <2e-16 ***
```

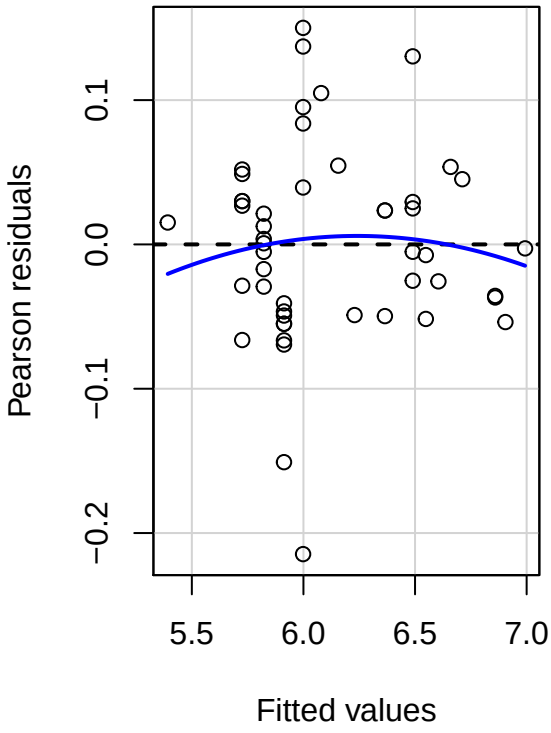
```
## log(Size)    1.49566    0.03772   39.65   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.06796 on 47 degrees of freedom
## Multiple R-squared:  0.971,  Adjusted R-squared:  0.9704
## F-statistic:  1572 on 1 and 47 DF,  p-value: < 2.2e-16
```

```
par(mfrow = c(2, 2))
plot(m2b1, main = "Model 2B")
```




```
residualPlots(m2b1)
```






```
##            Test stat Pr(>|Test stat|)
## log(Size)    -0.5356           0.5948
## Tukey test   -0.5356           0.5922
```

2b.2)

Some weaknesses include that the data after the log transformation has a much lower R-Squared, both Multiple and Adjusted, are slightly lower than Part A's model. Part B's model has a much more non-random pattern compared to Part A's model. Another possible weakness is that if this log model is done by hand rather than using RStudio, it will be much more complicated to find the results. Another weakness is that the data utilizes Price to be predicted by Size. This is not always the best case as there are other factors that could be used to predict Price.

2c)

We see that the the Multiple R-Squared (97.85%) and Adjusted R-Squared (97.8%) for the model Part A is better compared to the model Part B of Multiple R-Squared (97.1%) and Adjusted R-Squared (97.04%). Part A's model has a better explanation of variation in its model compared to Part B's model. I used the log transformation for both variables in Part B's model, but it appears Part A's model is better.

Problem 3

3a)

```
echodata <- read_csv("echo1.csv")
```

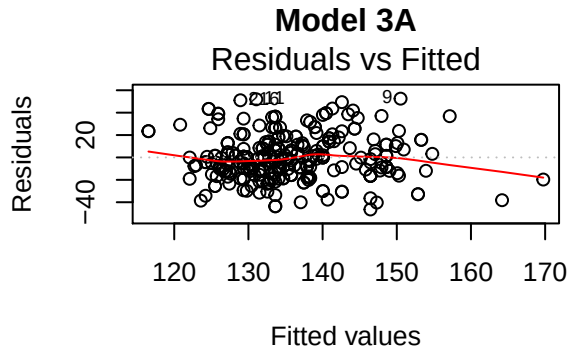
```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   gender = col_character(),
##   hxofCig = col_character(),
##   ecg = col_character()
## )
```

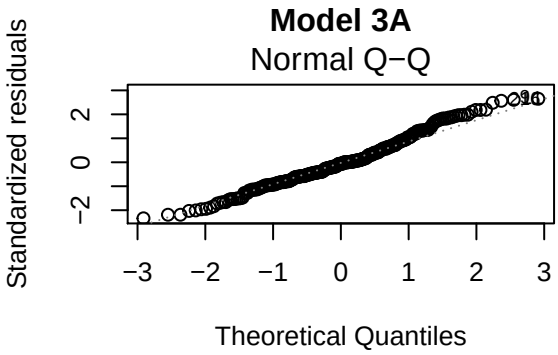
```
## See spec(...) for full column specifications.
```

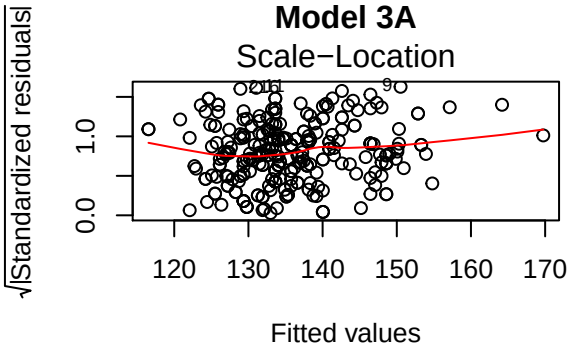
```
attach(echodata)
m3a <- lm(basebp ~ sbp)
summary(m3a)
```

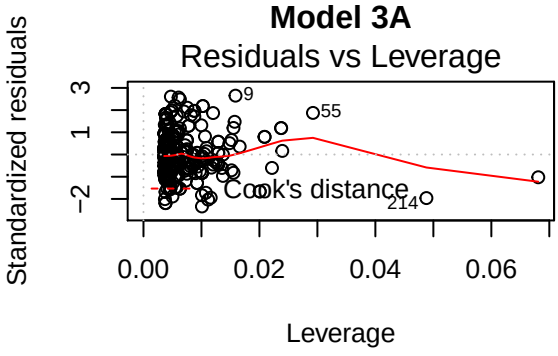
```
##
## Call:
## lm(formula = basebp ~ sbp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -46.449 -12.456  -1.273  11.444  52.490
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 103.70036    4.88943   21.21  < 2e-16 ***
## sbp           0.21374    0.03176    6.73 9.71e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 19.97 on 277 degrees of freedom
## Multiple R-squared:  0.1405, Adjusted R-squared:  0.1374
## F-statistic:  45.3 on 1 and 277 DF,  p-value: 9.705e-11
```

```
par(mfrow = c(2, 2))
plot(m3a, main = "Model 3A")
```

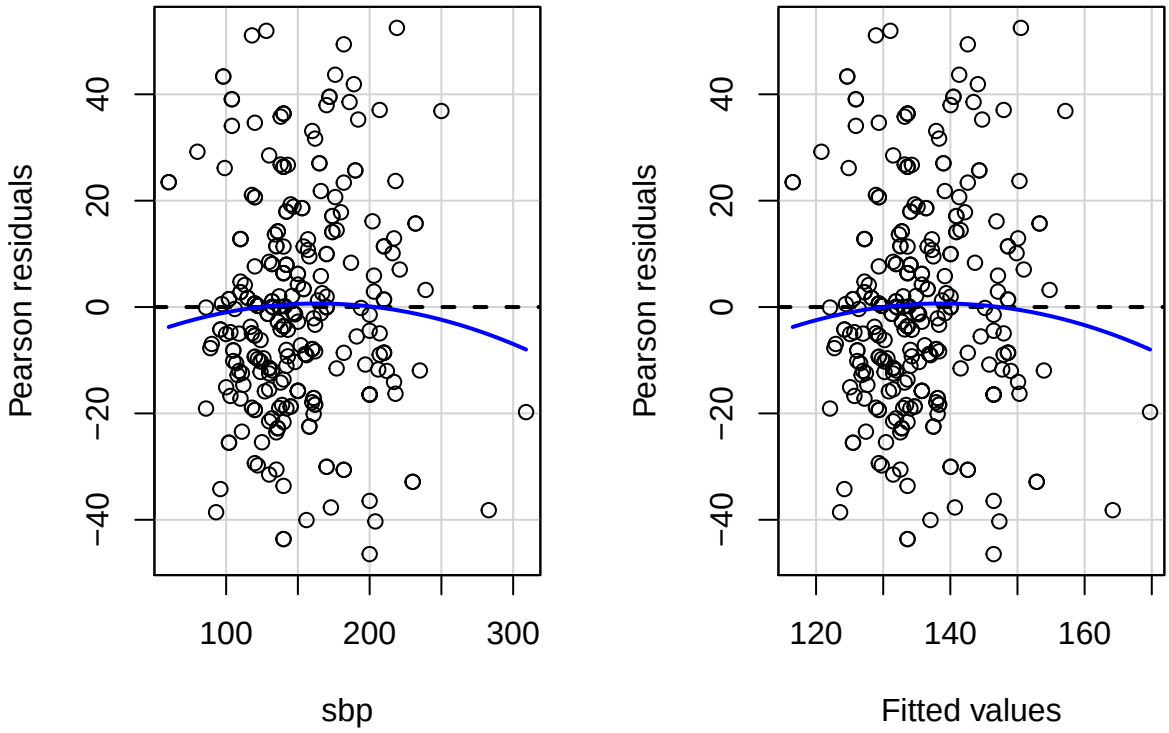









```
residualPlots(m3a)
```




```
##            Test stat Pr(>|Test stat|)
## sbp          -0.7452           0.4568
## Tukey test   -0.7452           0.4562
```

We can see from the Normal Q-Q plot that it is mostly linear, but not perfectly linear, indicating that it is not perfectly random. We can say it is more or less of Normally Distributed given our model, and it is somewhat linear model.

3b)

```
anovaecho <- aov(basebp ~ sbp)
summary(anovaecho)
```

```
##              Df Sum Sq Mean Sq F value   Pr(>F)
## sbp           1  18065   18065    45.3 9.71e-11 ***
## Residuals   277 110466     399
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We find R^2 = 1 - (RSS/SST), and SST = RSS + SSreg –>SST = 18065 + 110466 = 128531 & R^2 = 1 - (110466/128531) = 0.1405

```
RSS <- 110466
SST <- (110466 + 18065)
R_sq <- (1 - (RSS/SST))
R_sq
```

```
## [1] 0.1405498
```

Null Hypothesis: the coefficient equals to zero –> REJECT (for p-value < 0.05). For F-tests, the NULL HYPOTHESIS: All coefficients equal to zero. We see that the p-value < 0.05. We can see that the p-value under the F-statistic is 9.71e-11. Since the p-value is less that 0.05, we therefore REJECT the Null Hypothesis.

F-Stat calculation: F = MSR/MSE = (SSR/1)/(SSE/227) = SSR/SSE * 277 = SSR/(SST - SSR)*277 R^2 = SSR/SST F = (SST*R^2)/ (SST - SST*R^2) 277 = $R^{2/(1-R}2)$*277 F = R^2/(1 - R^2)*277

```
n <- nrow(echodata)
F_stat <- (0.1405)/((1-0.1405)/(n - 2))
F_stat
```

```
## [1] 45.2804
```

We also see that the F-value is 45.3. We see the F-value & F-table both are the same. From this, we can see we REJECT the Null Hypothesis.

Se^2 = Var(Y)*(1 - r^2) SSE = (1 - R^2)*SST –> SSE = (1 - r^2)*SST –> divide by (n - 1) –> SSE/(n - 1) = (1 - r^2)*(SST/(n - 1)) –> SSE/(n - 2) = Se^2 –> but SSE/(n - 1) = (1 - r^2)*Var(Y)

```
SSE <- (1 - R_sq)*SST
Se2_true <- (SSE/(n - 2))
Se2_estim <- var(basebp)*(1 - R_sq)
Se2_true
```

```
## [1] 398.7942
```

```
Se2_estim
```

```
## [1] 397.3579
```

We see that it is approximately the same for both Se^2 because the difference is from (n - 2) vs. (n - 1). This essentially true for Se^2.

3c)

(Adjusted) R^2 = 1 - (1 - R^2)*((n-1)/(n - p - 1)) (Adjusted) R^2 = 1 - [RSS / (n - p - 1)] / [SST / (n-1)] –> (Adjusted) R^2 = 0.1374

```
R_sq_adj <- 1 - (1 - 0.1405)*((n - 1)/(n - 2))
R_sq_adj
```

```
## [1] 0.1373971
```

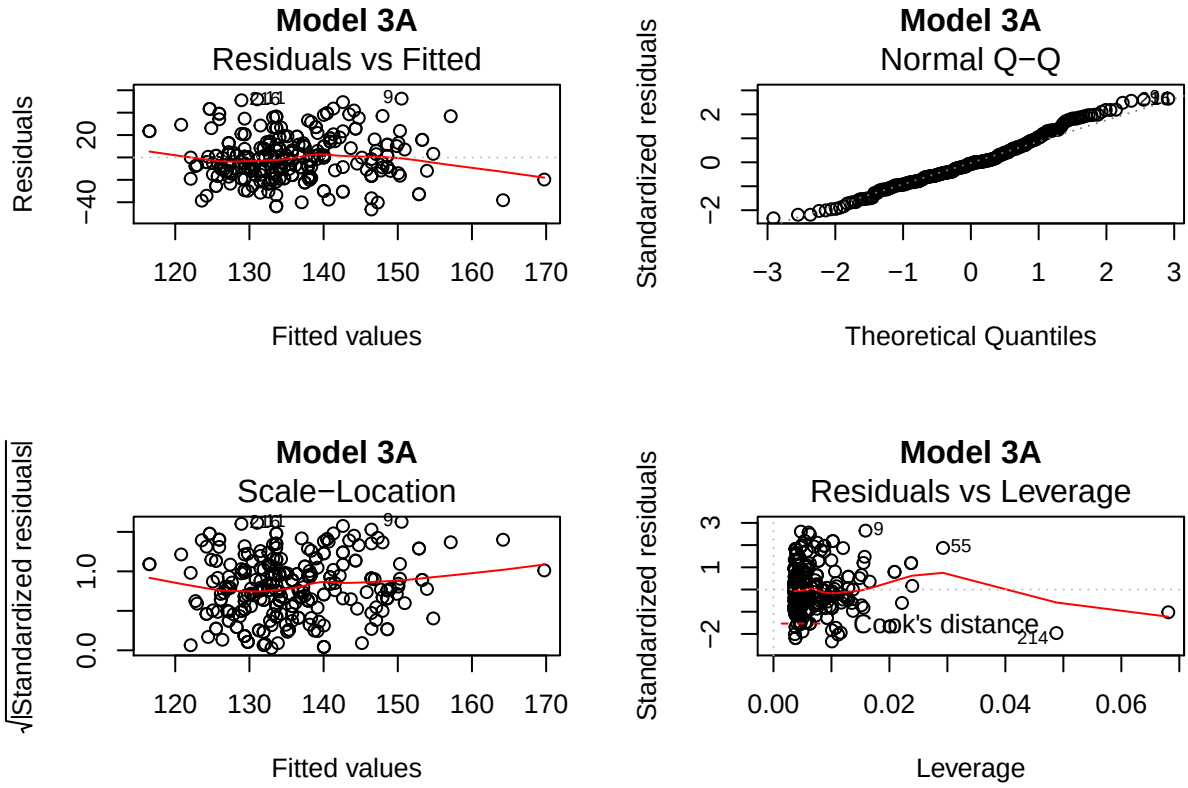
```
R_sq
```

```
## [1] 0.1405498
```

The reason why R^2 (Adjusted) is lower than R^2 (Multiple) is because it takes into account for the Parameter. The more Parameters you have, the lower the R^2 (Adjusted) is. If there was no Parameter, which is not possible for any linear model, then both R^2 (Multiple) and R^2 (Adjusted) will be approximately equal.

3d)

```
par(mfrow = c(2, 2))
plot(m3a, main = "Model 3A")
```



We can see from the Residuals vs. Fitted plot has no distinct pattern across the horizontal line, indicating there is not a non-linear relationship and has linearity. The Normal Q-Q plot shows us the residuals are Normally distributed, which is a good sign for our model. The Scale-Location plot tells us the assumption of equal variance (homoscedasticity). We see from this plot that it does satisfy our assumption for homoscedasticity because the points are equally spread across the horizontal line. Lastly, the Residuals vs. Leverage plot helps us find possible outliers/good leverages/bad leverages, such as 9, 55, 214.

3e)

Find hatvalues

```
hatv <- hatvalues(m3a)
LV <- ifelse(hatv >= 2*mean(hatv), "YES", "NO")
table(LV)
```

```
## LV
##  NO YES
## 260  19
```

Find standardized residuals

```
std.error <- rstandard(m3a)
OL <- ifelse(abs(std.error) >= 2, "YES", "NO")
table(OL)
```

```
## OL
##  NO YES
## 266  13
```

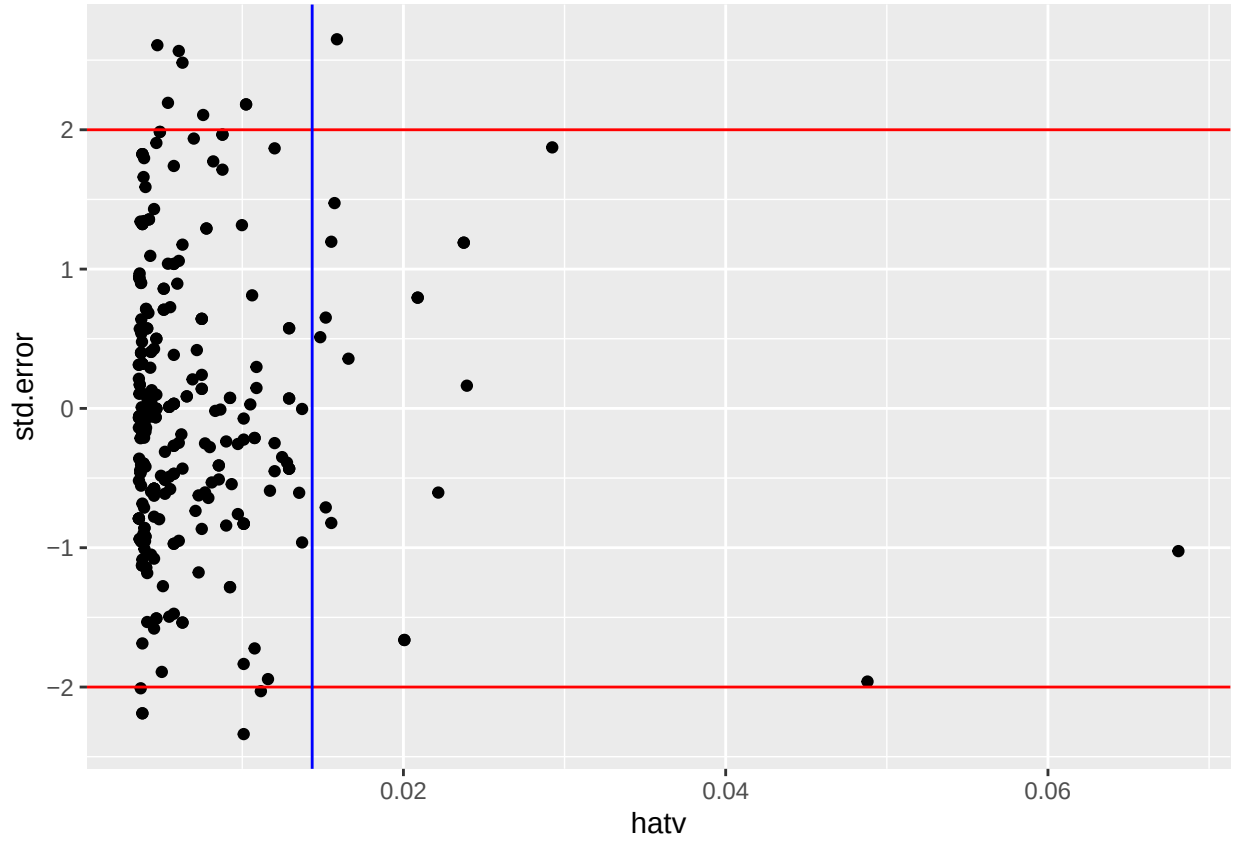
Combine the table

```
table(LV, OL)
```

```
##       OL
## LV      NO YES
##   NO   248  12
##   YES  18   1
```

3f)

```
gg <- data.frame(hatv, std.error)
ggplot(gg, aes(x = hatv, y = std.error)) +
  geom_point() +
  geom_hline(yintercept = 2, color = "red") +
  geom_hline(yintercept = -2, color = "red") +
  geom_vline(xintercept = 2*mean(hatv), color = "blue")
```

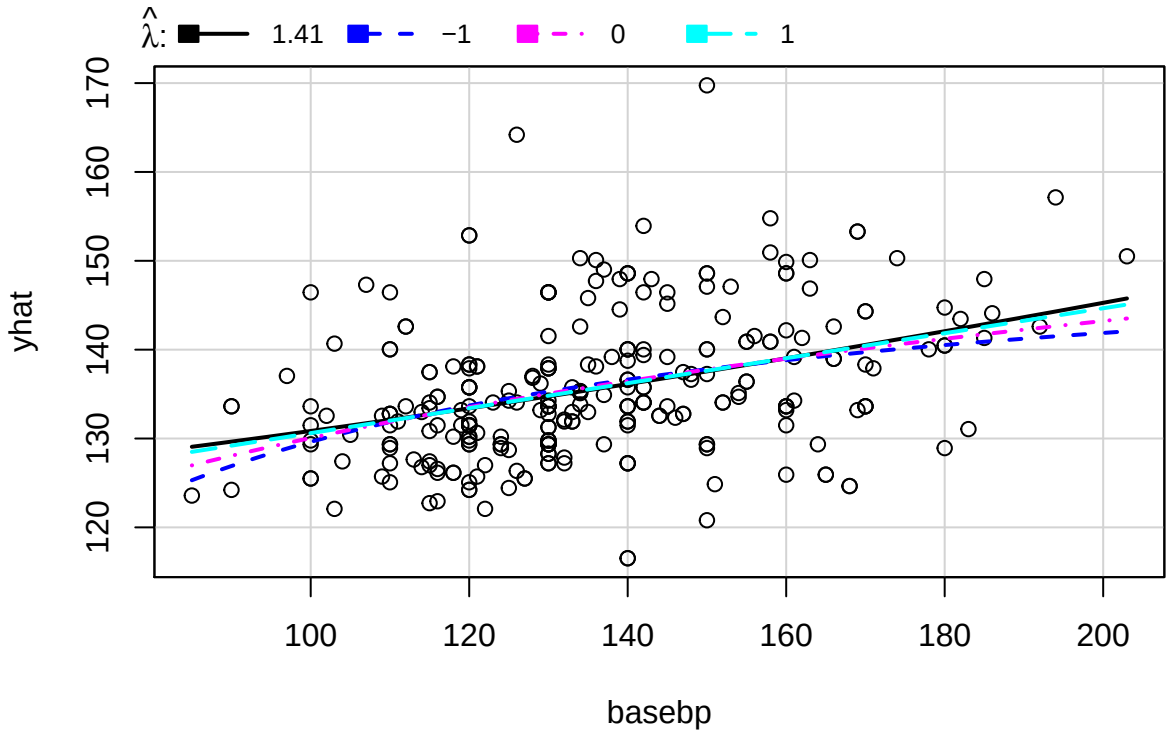


Problem 4

4a)

Use inverse response plot

```
par(mfrow = c(1,1))
inverseResponsePlot(m3a)
```

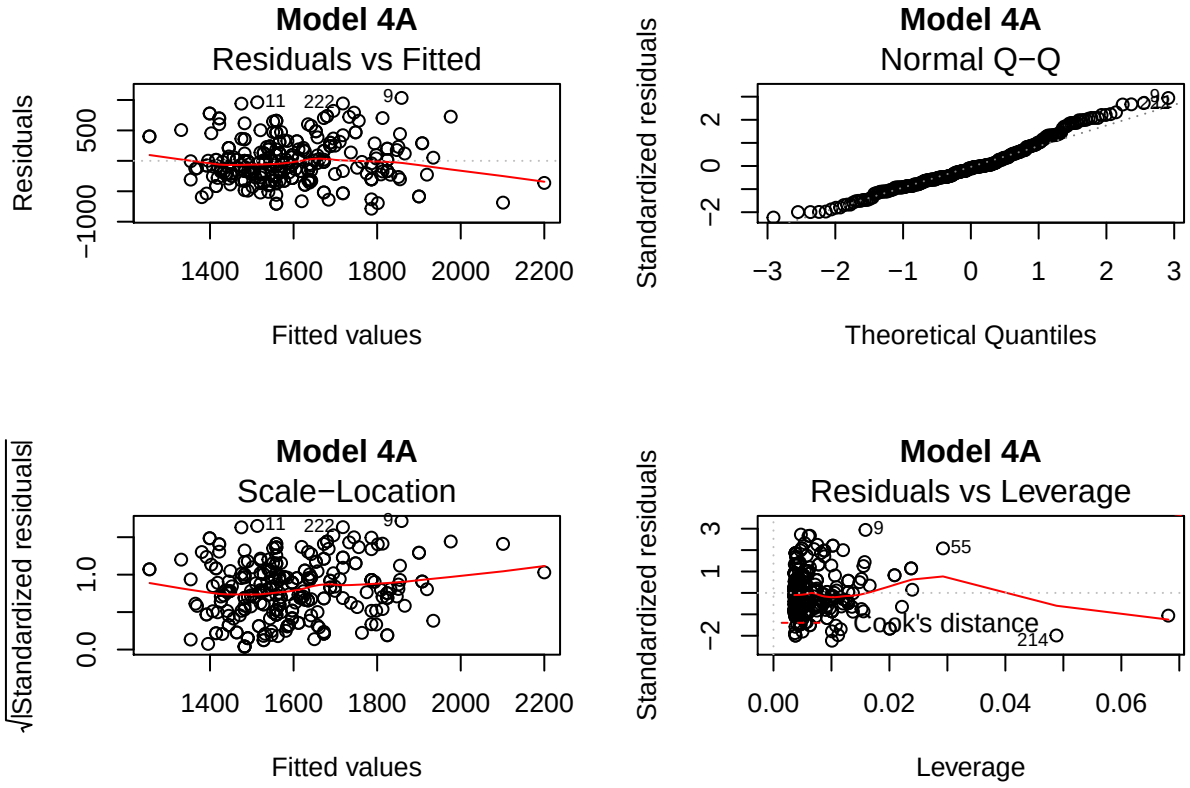



```
##      lambda      RSS
## 1  1.413234 15521.43
## 2 -1.000000 15677.80
## 3  0.000000 15574.03
## 4  1.000000 15525.77
```

```
m4a <- lm(basebp^(1.5) ~ sbp)
summary(m4a)
```

```
##
## Call:
## lm(formula = basebp^(1.5) ~ sbp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -786.35 -225.06  -34.72  199.56 1033.86
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1027.4596    86.7166  11.848  < 2e-16 ***
## sbp            3.7944     0.5632   6.737 9.35e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 354.2 on 277 degrees of freedom
## Multiple R-squared:  0.1408, Adjusted R-squared:  0.1377
## F-statistic: 45.38 on 1 and 277 DF,  p-value: 9.345e-11
```

```
par(mfrow = c(2,2))
plot(m4a, main = "Model 4A")
```



We can see that Model 4A in Problem 4a is slightly better because you can see the Multiple R^2 is 0.1408 & Adjusted R^2 is 0.1377, compared to 0.1405 & 0.1374 respectively for Model 3 in Problem 3a. Overall, they are pretty similar, except that Model 4A is 0.003 better in terms of R^2.

4b)

```
summary(powerTransform(cbind(basebp, sbp) ~ 1))
```

```
## bcPower Transformations to Multinormality
##        Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd
## basebp    0.0179           0      -0.5951       0.6309
## sbp       0.1426           0      -0.1982       0.4835
##
## Likelihood ratio test that transformation parameters are equal to 0
##  (all log transformations)
##                                LRT df    pval
## LR test, lambda = (0 0) 0.6807068  2 0.71152
##
## Likelihood ratio test that no transformations are needed
##                                LRT df       pval
## LR test, lambda = (1 1) 32.60354  2 8.3221e-08
```

```
# We accept the assumption lamda = c(0, 0), and the LRT shows the parameters are equal
# to zero for all log transformations.
```

Choose lambda = (0, 0), which means log(x) and log(y)

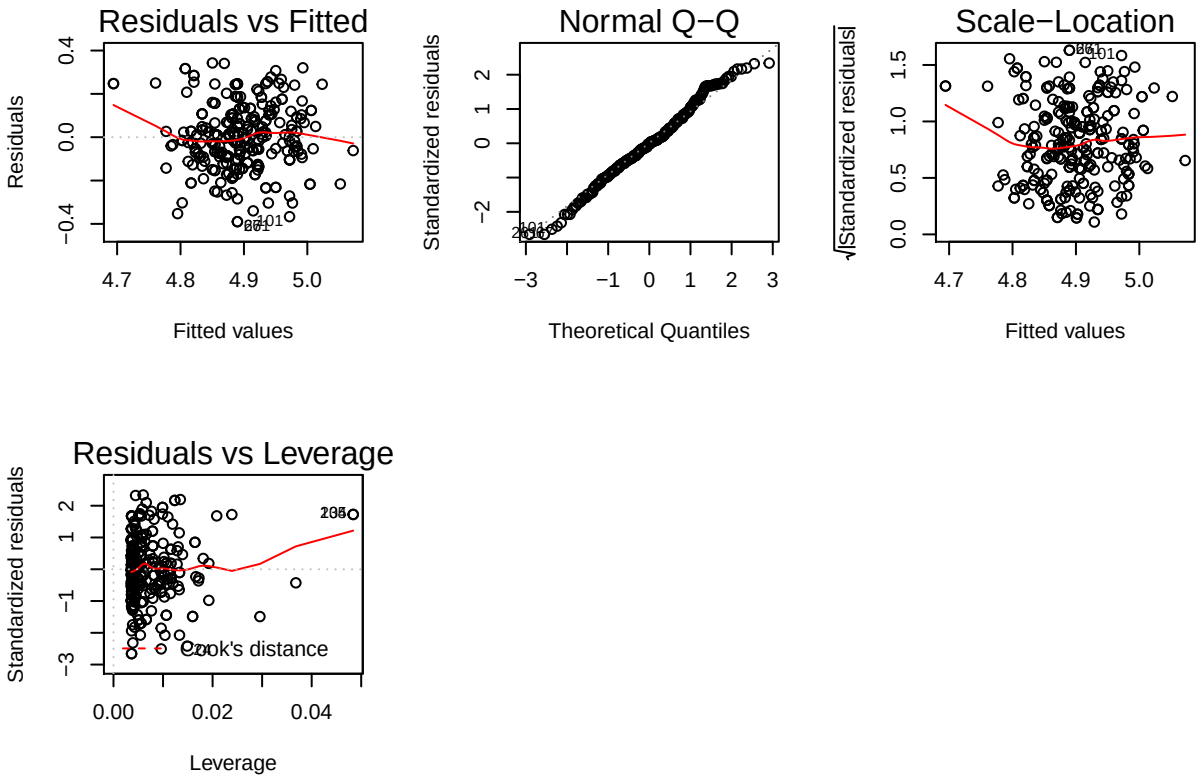
```
m4b <- lm(log(basebp) ~ log(sbp))
summary(m4b)
```

```
##
## Call:
## lm(formula = log(basebp) ~ log(sbp))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.38990 -0.08984 -0.00312  0.09195  0.34268
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.74996    0.17598  21.309  < 2e-16 ***
## log(sbp)     0.23064    0.03533   6.528 3.16e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.147 on 277 degrees of freedom
## Multiple R-squared:  0.1333, Adjusted R-squared:  0.1302
## F-statistic: 42.62 on 1 and 277 DF,  p-value: 3.163e-10
```

```
par(mfrow = c(2, 3))
plot(m4b, Main = "Model 4B")
```

```
## Warning in plot.window(...): "Main" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "Main" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "Main" is not
## a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "Main" is not
## a graphical parameter

## Warning in box(...): "Main" is not a graphical parameter

## Warning in title(...): "Main" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "Main" is not a
## graphical parameter

## Warning in plot.window(...): "Main" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "Main" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "Main" is not
## a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "Main" is not
## a graphical parameter

## Warning in box(...): "Main" is not a graphical parameter

## Warning in title(...): "Main" is not a graphical parameter
```

```
## Warning in plot.window(...): "Main" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "Main" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "Main" is not
## a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "Main" is not
## a graphical parameter

## Warning in box(...): "Main" is not a graphical parameter

## Warning in title(...): "Main" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "Main" is not a
## graphical parameter

## Warning in plot.window(...): "Main" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "Main" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "Main" is not
## a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "Main" is not
## a graphical parameter

## Warning in box(...): "Main" is not a graphical parameter

## Warning in title(...): "Main" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "Main" is not a
## graphical parameter
```



We can see that Model 4B in Problem 4b is worse because you can see the Multiple R^2 is 0.1333 & Adjusted R^2 is 0.1302, compared to 0.1405 & 0.1374 respectively for Model 3 in Problem 3a. Overall, Model 3 is a better choice for the than Model 4B.

Problem 5

5a)

```
salmondata <- read.table("salmon.txt", header = TRUE)

# From the table we can get the F-value and Pr(>F)
# SST = (n - 1)*Var(Y)
# SSR = R2 * SST
# SSE = SST - SSR
# MSR = SSR/df1
# MSE = SSE/df2

n <- nrow(salmondata)
df1 <- 1
df2 <- 98
R2 <- 0.2915
variance_y <- 2138.142


SST = (n - 1)*variance_y
```

```
SSR = R2 * SST
SSE = SST - SSR
MSR = SSR/df1
MSE = SSE/df2

SST
```

```
## [1] 211676.1
```

```
SSR
```

```
## [1] 61703.57
```

```
SSE
```

```
## [1] 149972.5
```

```
MSR
```

```
## [1] 61703.57
```

```
MSE
```

```
## [1] 1530.332
```

F-value: 40.32

Pr(>F): 6.747e-09

SST: 211676.1

SSR: 61703.57

SSE: 149972.5

MSR: 61703.57

MSE: 1530.332

```
anova(lm(salmondata$Marine ~ salmondata$Freshwater))
```

```
## Analysis of Variance Table
##
## Response: salmondata$Marine
##                        Df Sum Sq Mean Sq F value    Pr(>F)
## salmondata$Freshwater   1  61706   61706  40.323 6.747e-09 ***
## Residuals              98 149970    1530
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

5b)

```
S_xx <- (100 - 1)*676.0541
x_bar <- 117.9
```

```
h_4 <- (1/100) + ((86 - x_bar)^2 / S_xx)
h_4 > (4/100)
```

```
## [1] FALSE
```

```
h_41 <- (1/100) + ((84 - x_bar)^2 / S_xx)
h_41 > (4/100)
```

```
## [1] FALSE
```

```
h_53 <- (1/100) + ((179 - x_bar)^2 / S_xx)
h_53 > (4/100)
```

```
## [1] TRUE
```

We see that the Observation 53 is a Leverage Point.

5c)

Se = sqrt(SSE/(n-2))

```
Se <- sqrt(SSE/(n - 2))
Se
```

```
## [1] 39.11945
```

```
e_4 <- 506 - (511.3656 - (0.9602 * 86))
r_4 <- e_4 / (Se * sqrt(1 - h_4))
r_4 >= 2
```

```
## [1] FALSE
```

```
e_41 <- 511 - (511.3656 - (0.9602 * 84))
r_41 <- e_41 / (Se * sqrt(1 - h_41))
r_41 >= 2
```

```
## [1] TRUE
```

```
e_53 <- 407 - (511.3656 - (0.9602 * 179))
r_53 <- e_53 / (Se * sqrt(1 - h_53))
r_53 >= 2
```

```
## [1] FALSE
```

We see that Observation 41 is an Outlier.

5d)

4 - (iv) Not a leverage point nor an outlier (ordinary)

41 - (ii) An outlier but Not a leverage point

53 - (iii) A good leverage point