HW5

Adhrit Srivastav, eid: Ams22362

10/11/2021

Problem 1 Predicting tip from bill

```
tip = c(3, 4, 5, 6, 7, 8, 9, 11)
bill = c(18, 24, 26, 32, 35, 39, 47, 58)
toy = data.frame(tip, bill)

# a
set.seed(1789)
#train data
train=sample(length(tip), length(tip)/2, replace=F)

modelToy=lm(tip~bill, data = toy, subset=train)

modelNOINT = lm(tip~ 0 + bill, data = toy, subset=train)

#Compute MSE for the testing set
mean((tip-predict(modelToy, toy))[-train]^2)
```

```
## [1] 0.0918386
```

```
mean((tip-predict(modelNOINT, toy))[-train]^2)
```

```
## [1] 0.1215984
```

```
# b
library(boot)
# simple lin reg model
glm.fit=glm(tip~bill, data = toy)
cv.err=cv.glm(toy,glm.fit)
cv.err$delta[1]
```

```
## [1] 0.1972437
```

```
# no-int model
glm.fitNOINT=glm(tip~ 0 + bill, data = toy)
cv.errNOINT=cv.glm(toy,glm.fitNOINT)
cv.errNOINT$delta[1]
```

```
## [1] 0.1432629
```

```
# c
# simple lin reg model
glm.fitP = glm(tip ~ bill, data = toy)
cv.errorP = cv.glm(toy, glm.fitP,K=4)$delta[1]
cv.errorP
```

```
## [1] 0.1632529
```

```
# no-int model
glm.fitNOINT = glm(tip ~ 0 + bill, data = toy)
cv.errorNOINT = cv.glm(toy, glm.fitNOINT,K=4)$delta[1]
cv.errorNOINT
```

```
## [1] 0.125475
```

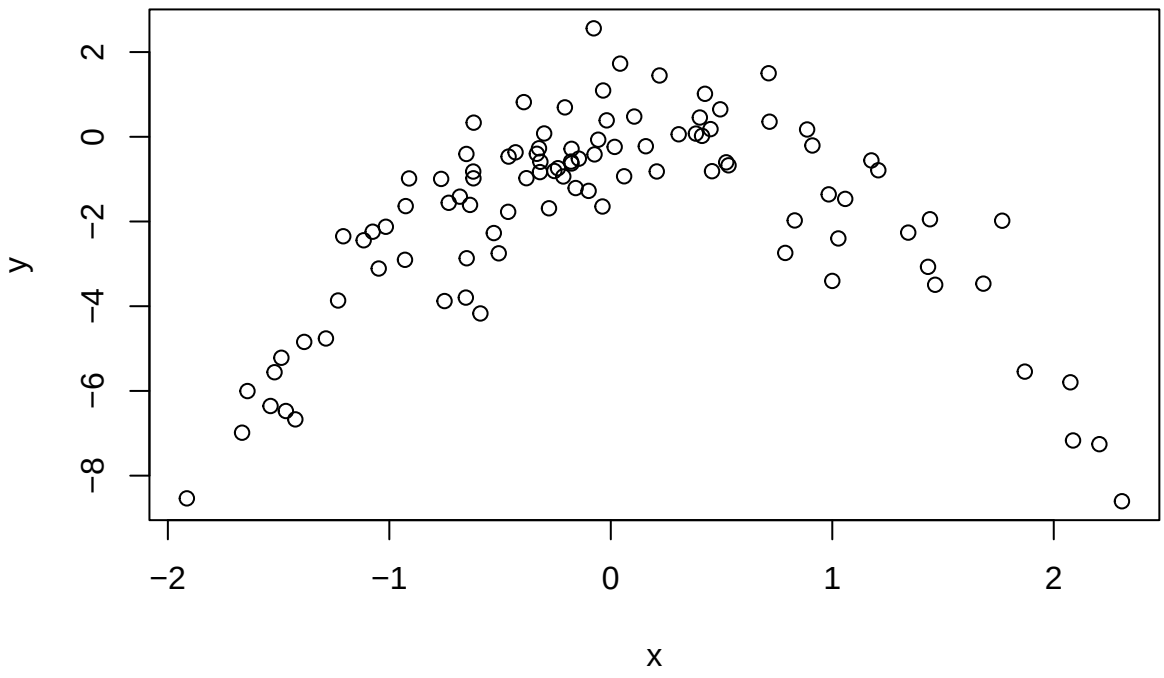
a) The Validation Set approach splits the data into training and testing sets that we can use to further validate the model's efficacy. It gives insight on how the model will perform when applied to new data. The simple regression model has an MSE of 0.0918386 while the no-int model has a larger MSE of 0.1215984.

b) Leave One Out Cross Validation (LOOCV) is when (n-1) observations are considered as the training set and the remainder are used as the test/validation set. The MSE for the simple regression model is 0.1972437 while the MSE of the no-int model is 0.1432629. Simple linear regression has higher MSE.

c) The 4-fold Cross Validation basically runs the training and validation data sets four times. The MSE for the simple regression model is 0.1632529 while the MSE of the no-int model is 0.125475. MSE of simple linear regression has higher MSE.

Problem 2

```
set.seed(1)
y=rnorm(100)
x=rnorm(100)
y=x-2*x^2+rnorm(100)

# b
plot(y ~ x)
```




```
# c
set.seed(1)
randData = data.frame(x, y)
fit.glm1 <- glm(y ~ x)
cv.glm(randData, fit.glm1)$delta[1]
```

```
## [1] 5.890979
```

```
fit.glm2 <- glm(y ~ poly(x, 2))
cv.glm(randData, fit.glm2)$delta[1]
```

```
## [1] 1.086596
```

```
fit.glm3 <- glm(y ~ poly(x, 3))
cv.glm(randData, fit.glm3)$delta[1]
```

```
## [1] 1.102585
```

```
fit.glm4 <- glm(y ~ poly(x, 4))
cv.glm(randData, fit.glm4)$delta[1]
```

```
## [1] 1.114772
```

```r
# d
set.seed(1986)
randData = data.frame(x, y)
fit.glm1 <- glm(y ~ x)
cv.glm(randData, fit.glm1)$delta[1]
```

```
## [1] 5.890979
```

```r
fit.glm2 <- glm(y ~ poly(x, 2))
cv.glm(randData, fit.glm2)$delta[1]
```

```
## [1] 1.086596
```

```r
fit.glm3 <- glm(y ~ poly(x, 3))
cv.glm(randData, fit.glm3)$delta[1]
```

```
## [1] 1.102585
```

```r
fit.glm4 <- glm(y ~ poly(x, 4))
cv.glm(randData, fit.glm4)$delta[1]
```

```
## [1] 1.114772
```

```r
# e
summary(fit.glm1)
```

```
##
## Call:
## glm(formula = y ~ x)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -7.3469  -0.9275   0.8028   1.5608   4.3974
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.8185     0.2364  -7.692 1.14e-11 ***
## x             0.2430     0.2479   0.981    0.329
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 5.580018)
##
##     Null deviance: 552.21  on 99  degrees of freedom
## Residual deviance: 546.84  on 98  degrees of freedom
## AIC: 459.69
##
## Number of Fisher Scoring iterations: 2
```

```
summary(fit.glm2)
```

```
##
## Call:
## glm(formula = y ~ poly(x, 2))
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.89884  -0.53765   0.04135   0.61490   2.73607
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.8277     0.1032 -17.704   <2e-16 ***
## poly(x, 2)1   2.3164     1.0324   2.244   0.0271 *
## poly(x, 2)2 -21.0586     1.0324 -20.399   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 1.06575)
##
##     Null deviance: 552.21  on 99  degrees of freedom
## Residual deviance: 103.38  on 97  degrees of freedom
## AIC: 295.11
##
## Number of Fisher Scoring iterations: 2
```

```
summary(fit.glm3)
```

```
##
## Call:
## glm(formula = y ~ poly(x, 3))
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.87250  -0.53881   0.02862   0.59383   2.74350
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.8277     0.1037 -17.621   <2e-16 ***
## poly(x, 3)1   2.3164     1.0372   2.233   0.0279 *
## poly(x, 3)2 -21.0586     1.0372 -20.302   <2e-16 ***
## poly(x, 3)3  -0.3048     1.0372  -0.294   0.7695
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 1.075883)
##
##     Null deviance: 552.21  on 99  degrees of freedom
## Residual deviance: 103.28  on 96  degrees of freedom
## AIC: 297.02
##
## Number of Fisher Scoring iterations: 2
```

```
summary(fit.glm4)
```

```
##
## Call:
## glm(formula = y ~ poly(x, 4))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8914  -0.5244   0.0749   0.5932   2.7796
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.8277     0.1041 -17.549   <2e-16 ***
## poly(x, 4)1   2.3164     1.0415   2.224   0.0285 *
## poly(x, 4)2 -21.0586     1.0415 -20.220   <2e-16 ***
## poly(x, 4)3  -0.3048     1.0415  -0.293   0.7704
## poly(x, 4)4  -0.4926     1.0415  -0.473   0.6373
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 1.084654)
##
##     Null deviance: 552.21  on 99  degrees of freedom
## Residual deviance: 103.04  on 95  degrees of freedom
## AIC: 298.78
##
## Number of Fisher Scoring iterations: 2
```

a) n is 100. p is 2. y = -2x^2 + x

b) The data is curved and parabolic.

c) In the code above

d) Yes, they are the same. Because changing random seed doesn't change overall approach.

e) The second model had the smallest LOOCV error because it was a quadratic model.

f) Intercept, degree 1, and degree 2 are all statistically significant. This is consistent with the conclusions drawn from the cross-validation results.

Problem 3

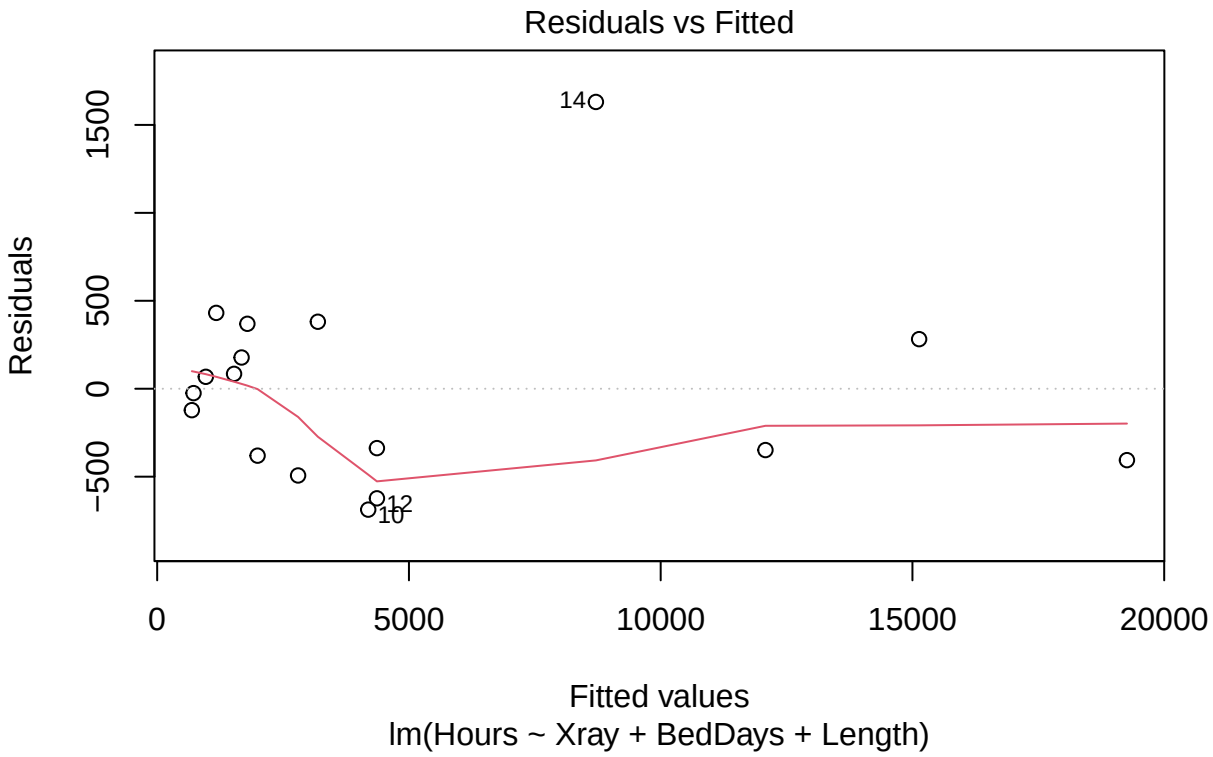
```
library(olsrr)
```

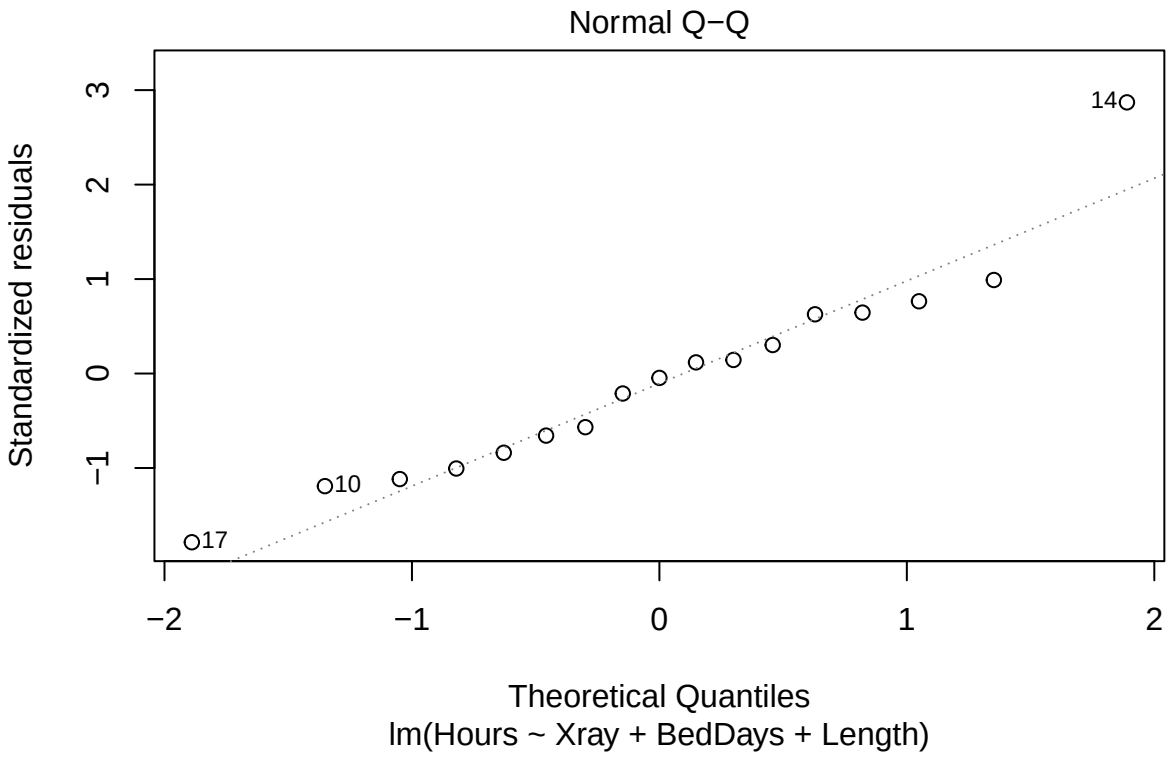
```
##
## Attaching package: 'olsrr'

## The following object is masked from 'package:datasets':
##
##     rivers
```

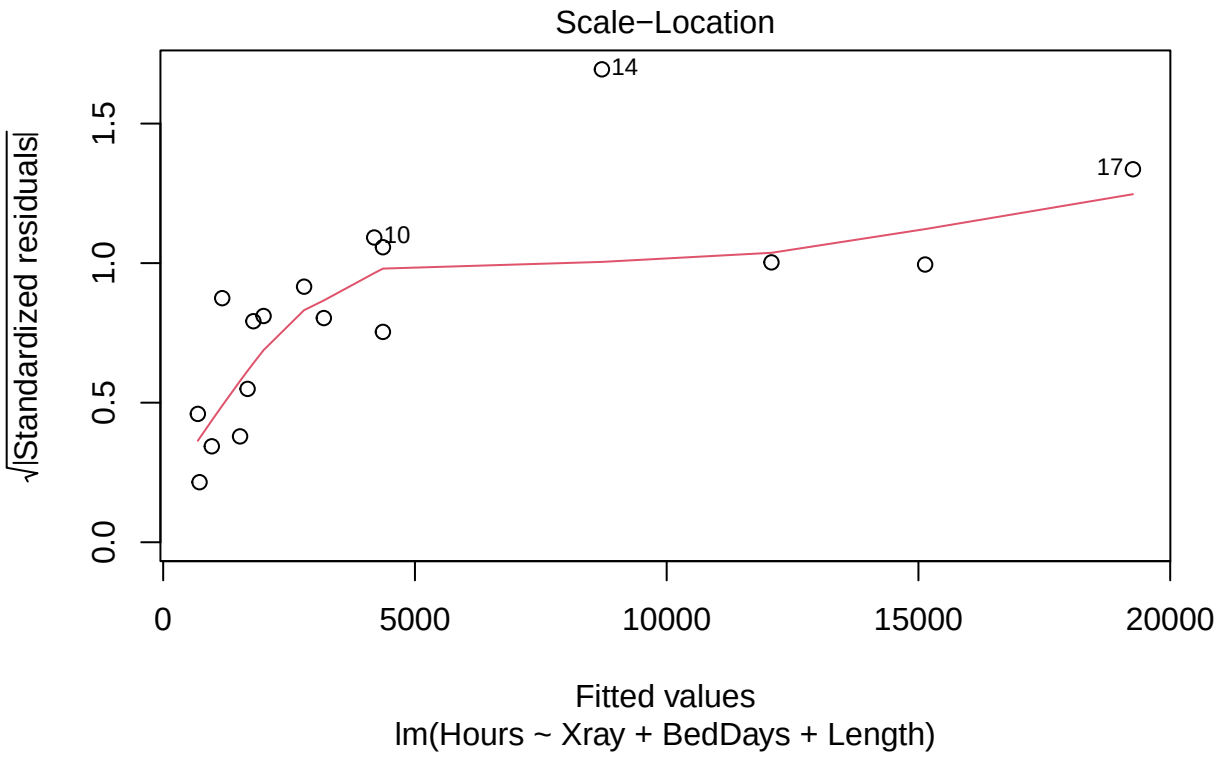
```r
hospital = read.csv("C:/Users/adhri/OneDrive/Documents/R/App_Reg_and_Time_Series/chpt4/hospitaldata.csv
attach(hospital)

modelH = lm(Hours ~ Xray + BedDays + Length)
summary(modelH)
```

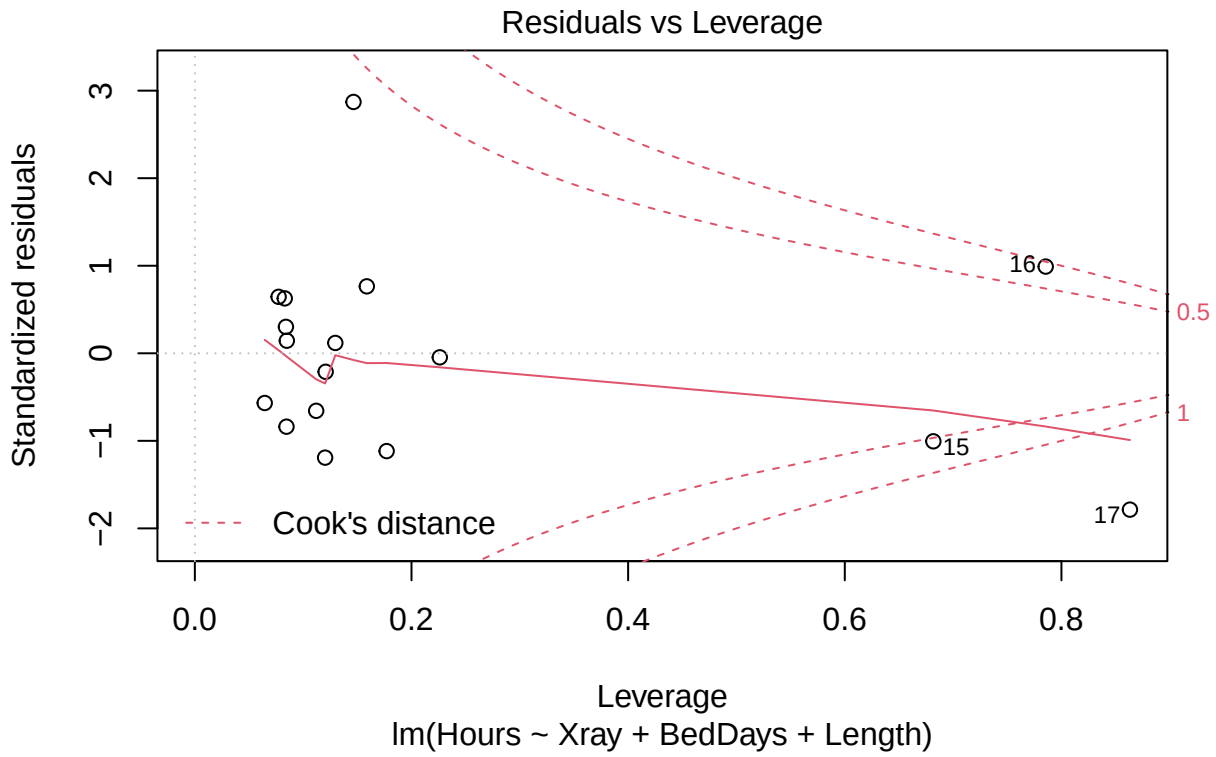
```
##
## Call:
## lm(formula = Hours ~ Xray + BedDays + Length)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -687.40 -380.60  -25.03  281.91 1630.50
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1523.38924  786.89772   1.936   0.0749 .
## Xray           0.05299    0.02009   2.637   0.0205 *
## BedDays        0.97848    0.10515   9.305 4.12e-07 ***
## Length      -320.95083  153.19222  -2.095   0.0563 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 614.8 on 13 degrees of freedom
## Multiple R-squared:  0.9901, Adjusted R-squared:  0.9878
## F-statistic:   432 on 3 and 13 DF,  p-value: 2.894e-13
```

```r
plot(modelH)
```


Residuals vs Fitted
Residuals
1500
500
0
−500
14
12
10
0
5000
10000
15000
20000
Fitted values
lm(Hours ~ Xray + BedDays + Length)


Normal Q−Q
Standardized residuals
3
2
1
0
−1
14
10
17
−2
−1
0
1
2
Theoretical Quantiles
lm(Hours ~ Xray + BedDays + Length)

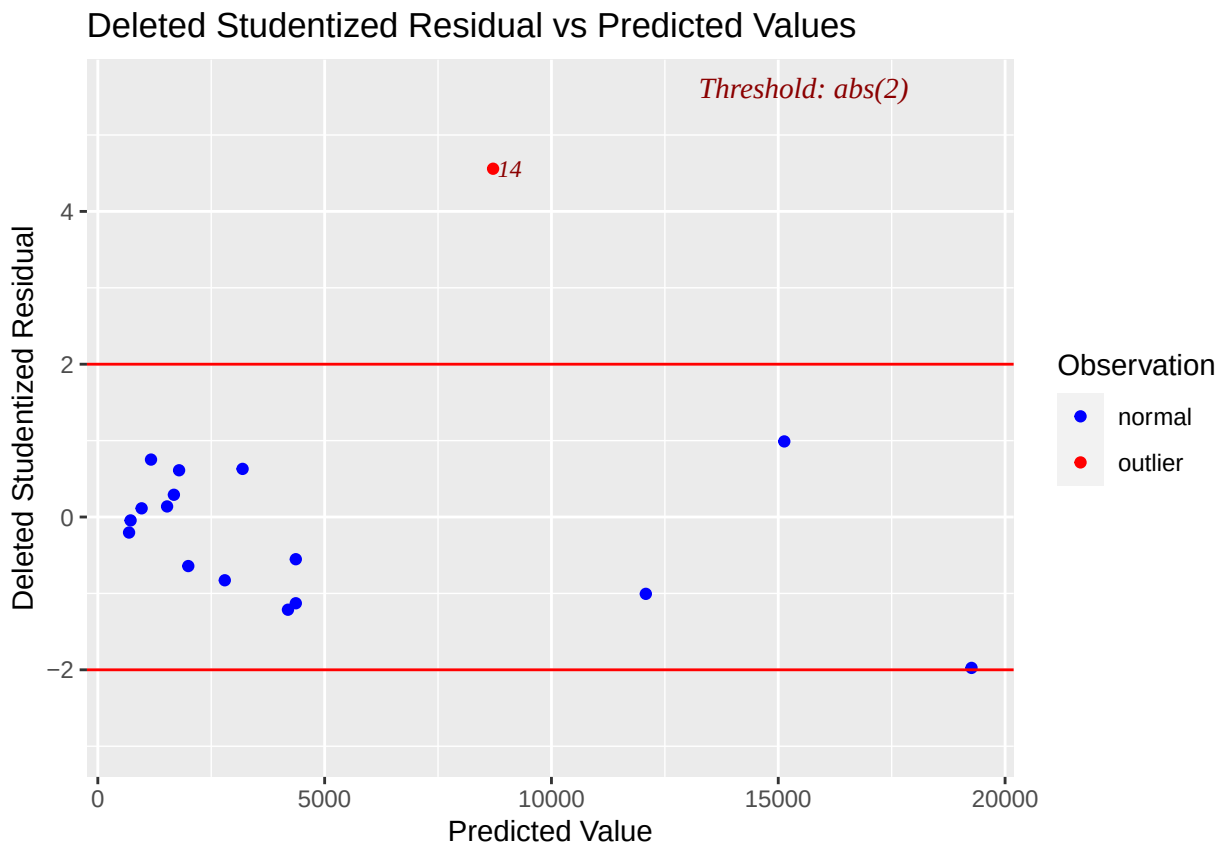

Scale−Location
14
17
10
√|Standardized residuals|
0.0
0.5
1.0
1.5
0
5000
10000
15000
20000
Fitted values
lm(Hours ~ Xray + BedDays + Length)



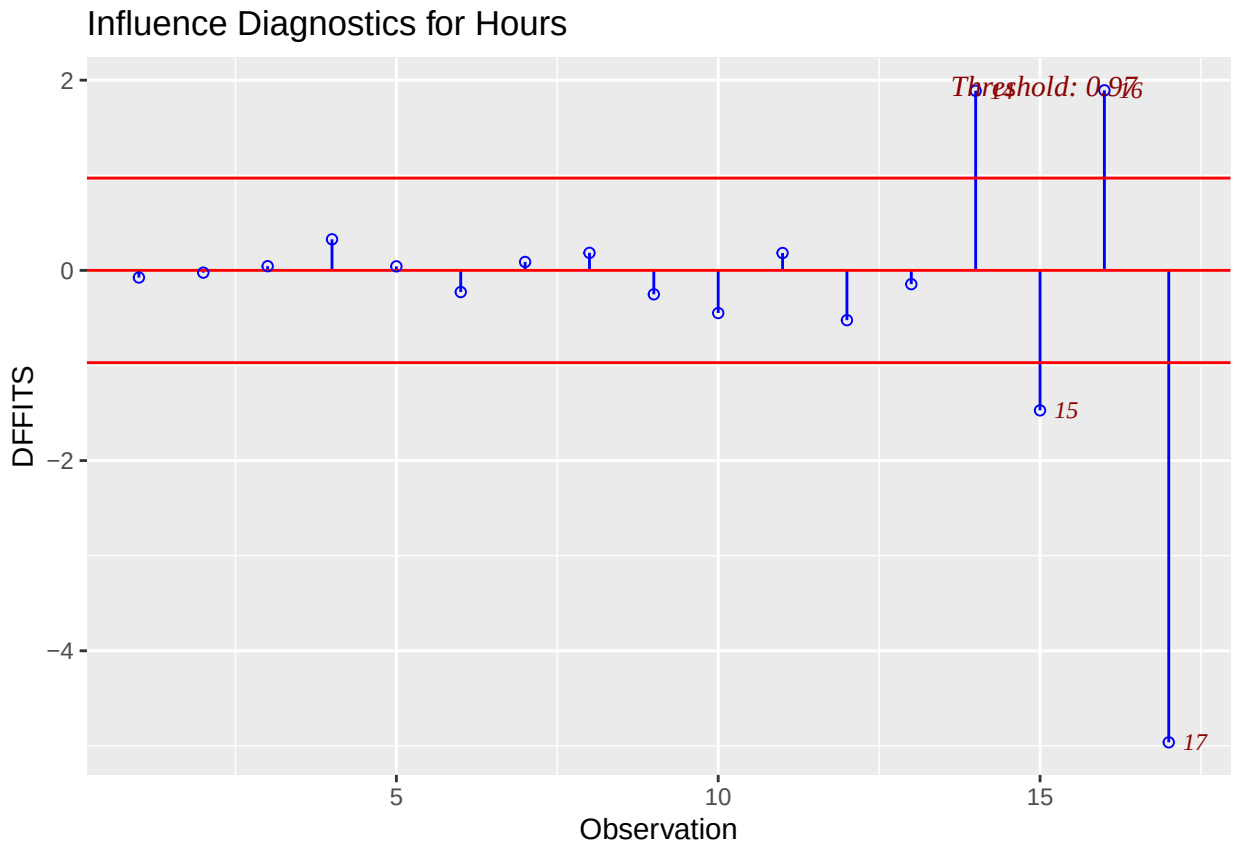

```
# c
lm.influence(modelH)$hat
```

```
##          1          2          3          4          5          6          7
## 0.12074859 0.22612778 0.12966448 0.15876175 0.08491374 0.11201066 0.08407803
##          8          9         10         11         12         13         14
## 0.08300469 0.08459615 0.12026226 0.07733452 0.17705815 0.06449836 0.14645059
##         15         16         17
## 0.68176286 0.78548020 0.86324719
```

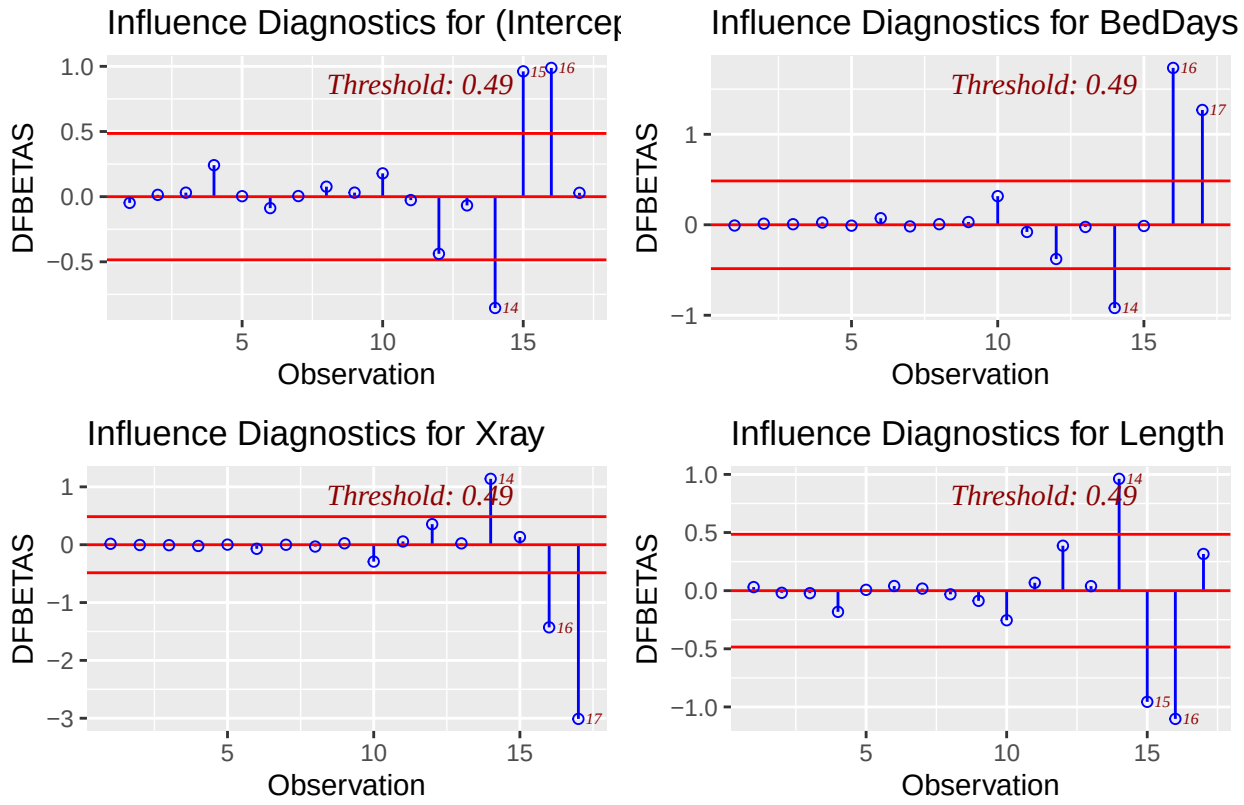
```
# d
ols_plot_resid_stud_fit(modelH)
```




```
ols_plot_dffits(modelH)
```




```
ols_plot_dfbetas(modelH)
```



a) $\widehat{Hours} = 1523.38924 + Xray$*0.05299 + $BedDays$*0.97848 + $Length$*-320.95083

b) There are some points of possible influential observations.

c) 15, 16, and 17 have high leverage because their hat values are large.

d) We can see there is a singular outlier outside the thresholds.

e) From dffits we can see there are 4 outliers (14, 15, 16, and 17).

f) From dfbetas we can see there are outliers in all of the features. All of the features are responsible for some outliers.

g) Constructed a new model by removing outliers.

Problem 4

```
cocaine = read.csv("C:/Users/adhri/OneDrive/Documents/R/App_Reg_and_Time_Series/chpt4/cocainedata.csv")
attach(cocaine)

modelC = lm(price~quant+qual+trend)
summary(modelC)
```

```
##
## Call:
```

```
## lm(formula = price ~ quant + qual + trend)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -43.479 -12.014  -3.743  13.969  43.754
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 90.84681    8.58024  10.588 1.39e-14 ***
## quant       -0.05997    0.01018  -5.892 2.85e-07 ***
## qual         0.11620    0.20326   0.572   0.5700
## trend       -2.35459    1.38612  -1.699   0.0954 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.06 on 52 degrees of freedom
## Multiple R-squared:  0.5097, Adjusted R-squared:  0.4814
## F-statistic: 18.02 on 3 and 52 DF,  p-value: 3.806e-08
```

```
confint(modelC)
```

```
##                    2.5 %       97.5 %
## (Intercept) 73.62929549 108.06432810
## quant       -0.08039395  -0.03954557
## qual        -0.29167529   0.52408371
## trend       -5.13604466   0.42685868
```

a) I would expect beta2 to be negative and beta3 to be positive.

b) $\widehat{price}$ = 90.84681 + *quant**-0.05997 + *qual**.11620 + *trend**-2.35459. The signs were as I expected. For every additional gram of cocaine, the price decreases by approximately on the average -0.05997 when controlling for qual and trend. For every additional percentage of purity, the price increases by approximately on the average .11620 when controlling for quant and trend. For every additional unit of time, the price decreases by approximately on the average -2.35459. when controlling for qual and quant.

c) The R-Squared value tells us that 50.97% of the variation in cocaine price is explained by the model shown above.

d) H0: the number of sales has no impact on the risk of getting caught. HA: the greater the number of sales, the higher the risk of getting caught. We can see that the p-value of quant is 2.85e-07 which is less than .05 which means we can reject the null hypothesis.

e) H0: the quality of cocaine has no influence on price. HA: the quality of cocaine does have an influence on price. We see that the p-value of qual is .57 which is greater than the alpha which indicates that we fail to reject the null hypothesis.

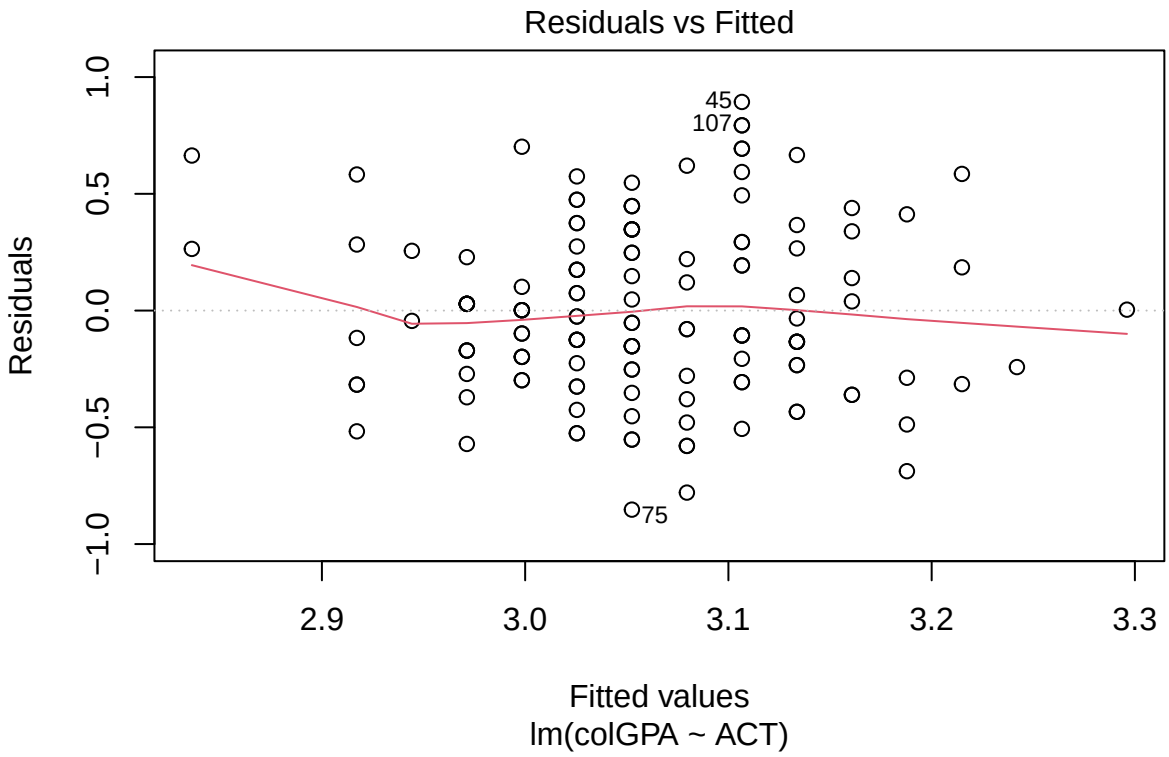
f) -2.3546. The price is decreasing because the quantity and quality of cocaine is decreasing as smugglers get caught and as they mass produce.

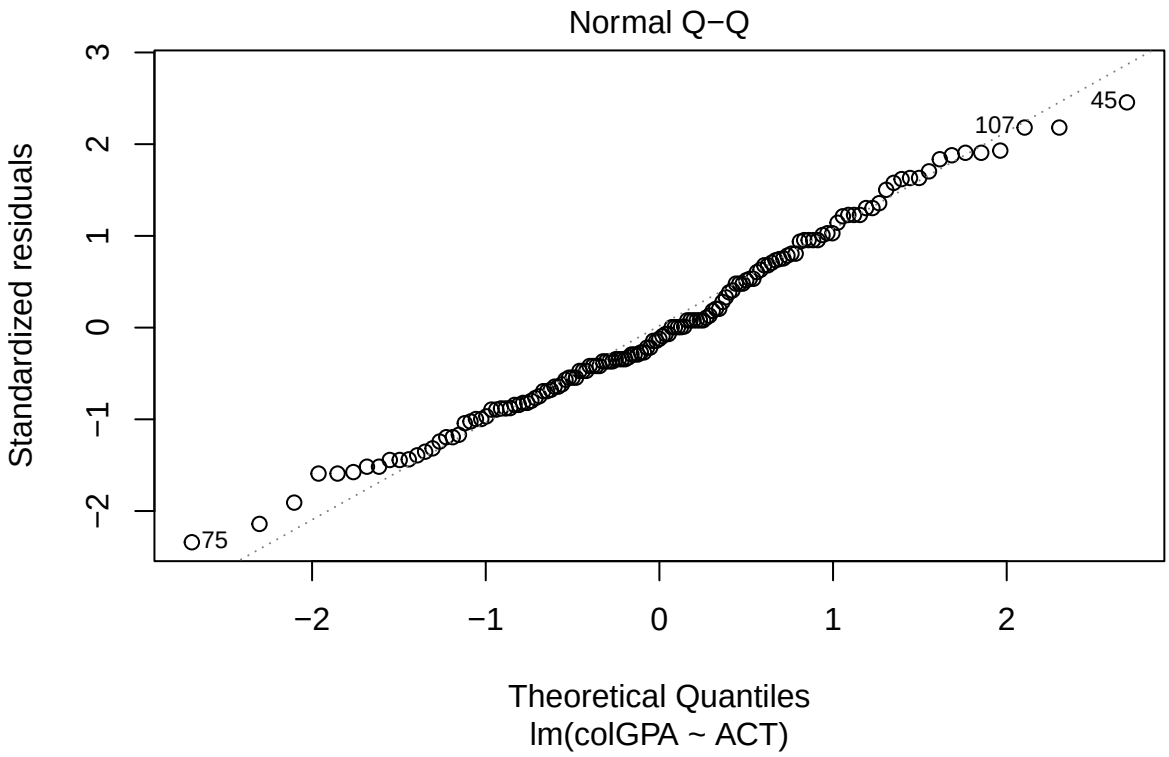
Problem 5

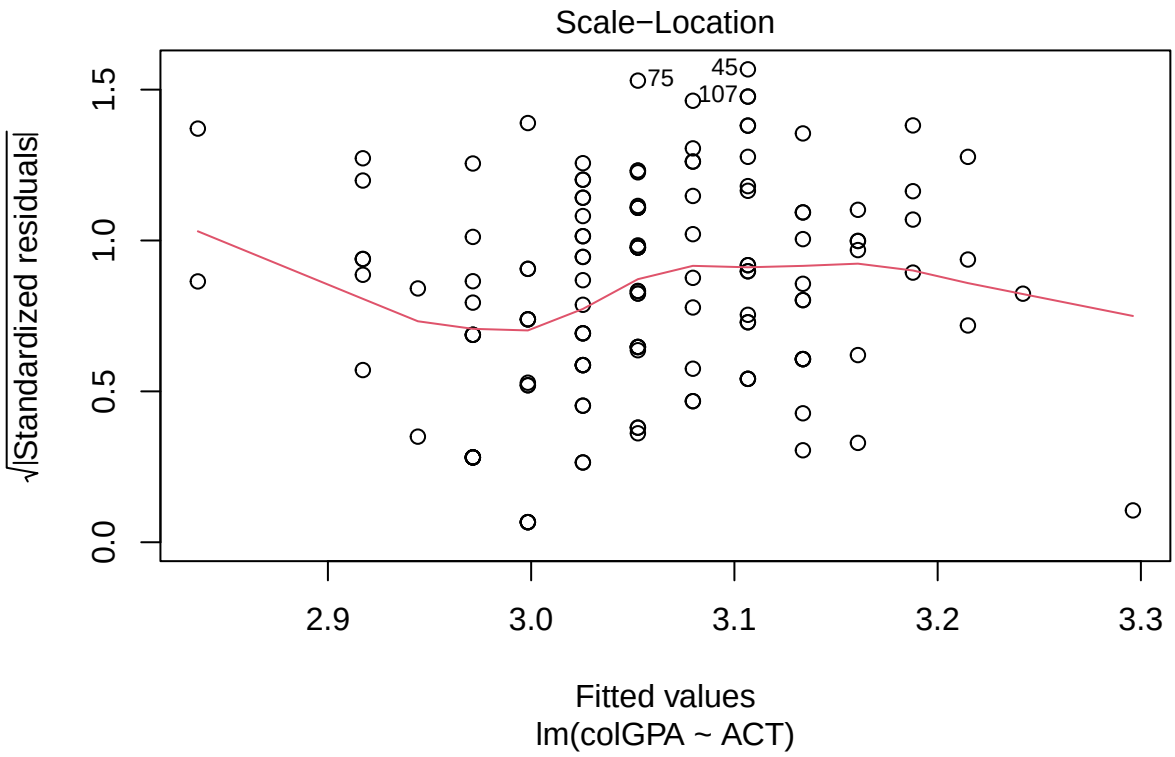
```r
gpa = read.csv("C:/Users/adhri/OneDrive/Documents/R/App_Reg_and_Time_Series/chpt4/gpa.csv")
attach(gpa)

lower = sqrt(((139)*0.3656**2)/qchisq(.9,139))
upper = sqrt(((139)*0.3656**2)/qchisq(.1,139))

modelGA = lm(colGPA ~ ACT)
summary(modelGA)
```

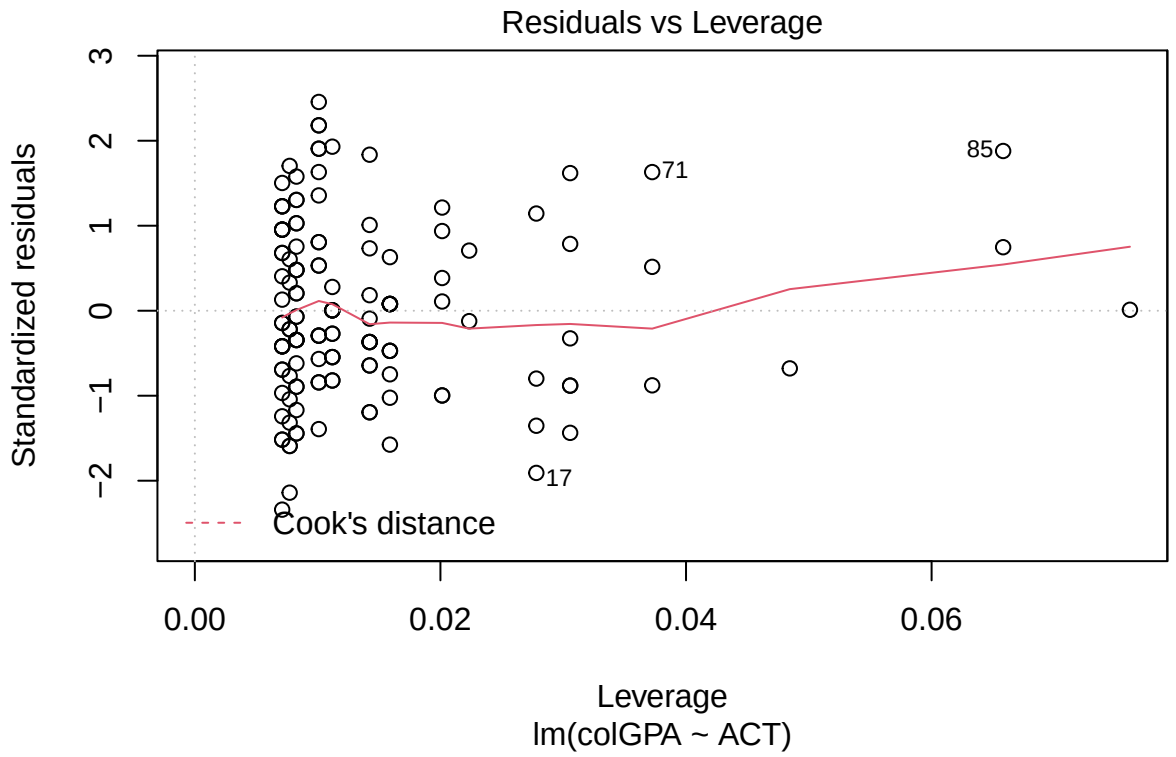
```
##
## Call:
## lm(formula = colGPA ~ ACT)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.85251 -0.25251 -0.04426  0.26400  0.89336
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.40298    0.26420   9.095  8.8e-16 ***
## ACT          0.02706    0.01086   2.491   0.0139 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3656 on 139 degrees of freedom
##   (184 observations deleted due to missingness)
## Multiple R-squared:  0.04275,    Adjusted R-squared:  0.03586
## F-statistic: 6.207 on 1 and 139 DF,  p-value: 0.0139
```

```r
plot(modelGA)
```


Residuals vs Fitted
Residuals
1.0
0.5
0.0
−0.5
−1.0
45
107
75
2.9
3.0
3.1
3.2
3.3
Fitted values
lm(colGPA ~ ACT)


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(colGPA ~ ACT)
75
107
45
−2
−1
0
1
2
3

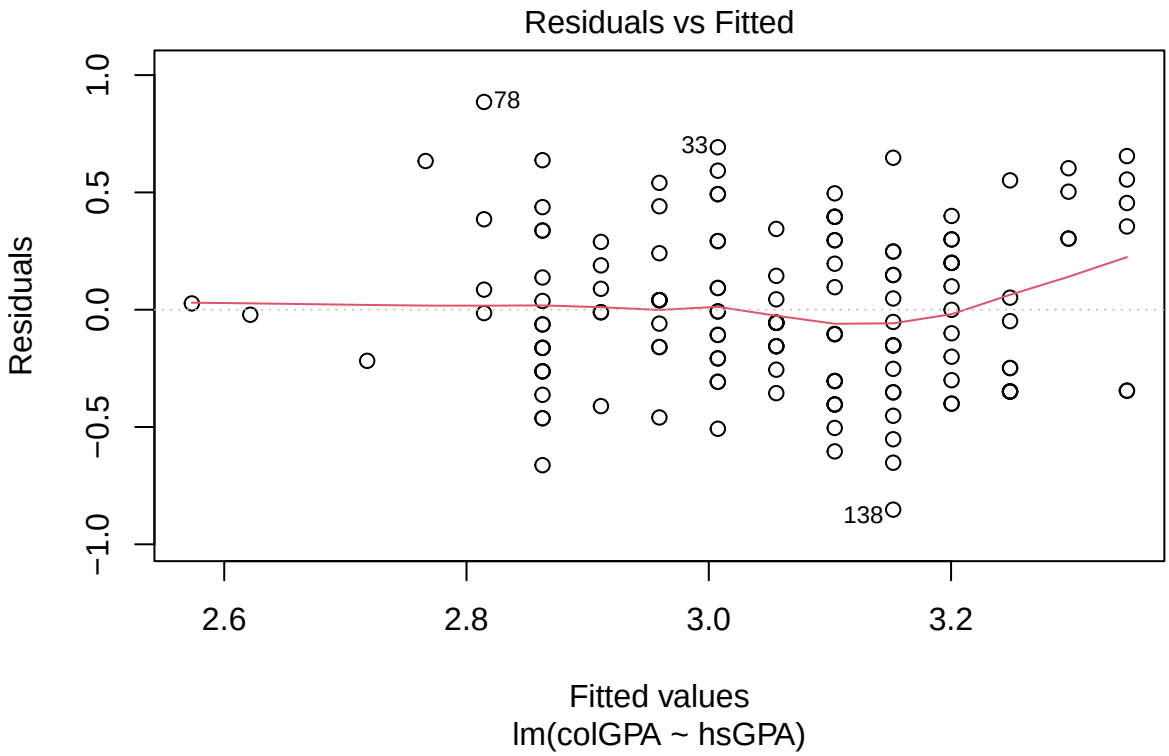

Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
75
45
107
2.9
3.0
3.1
3.2
3.3
Fitted values
lm(colGPA ~ ACT)

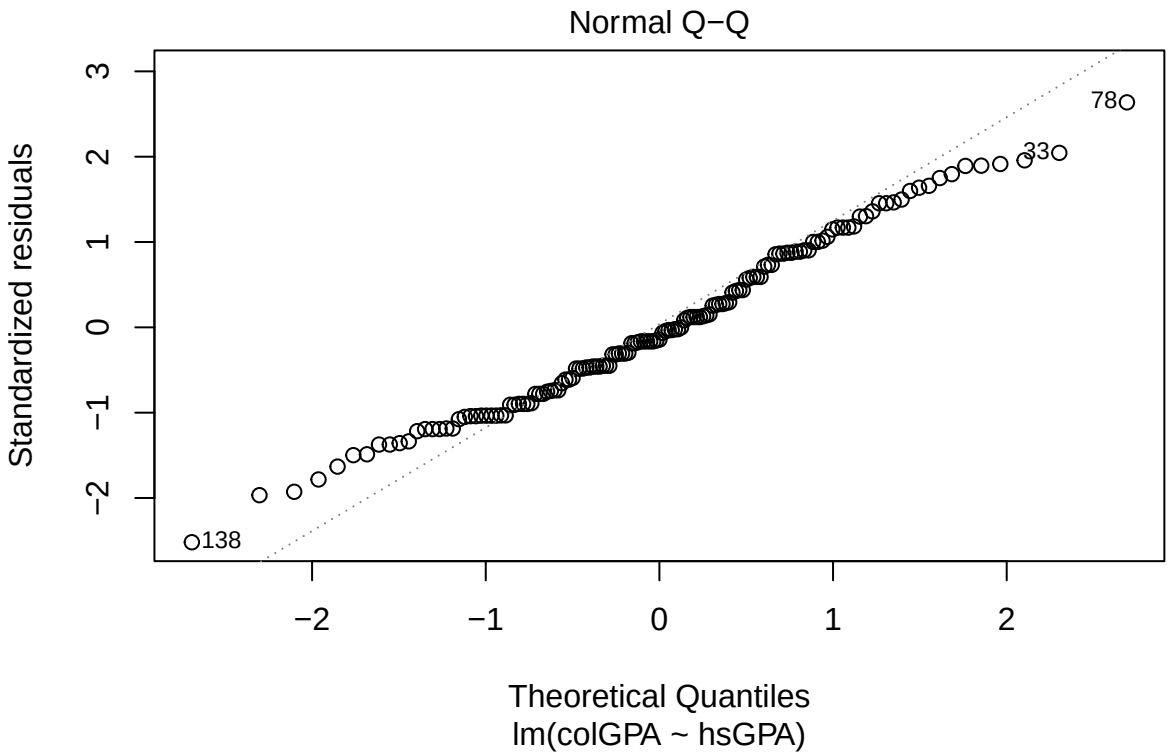


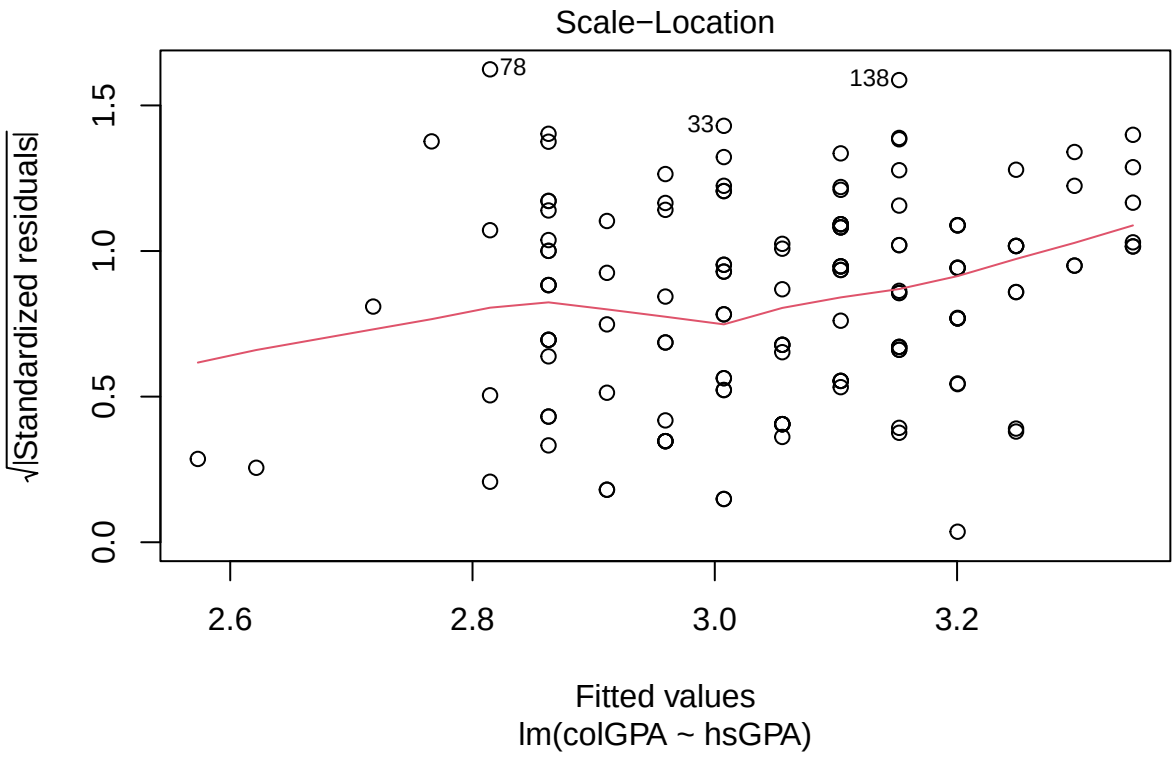

```
modelGH = lm(colGPA ~ hsGPA)
summary(modelGH)
```

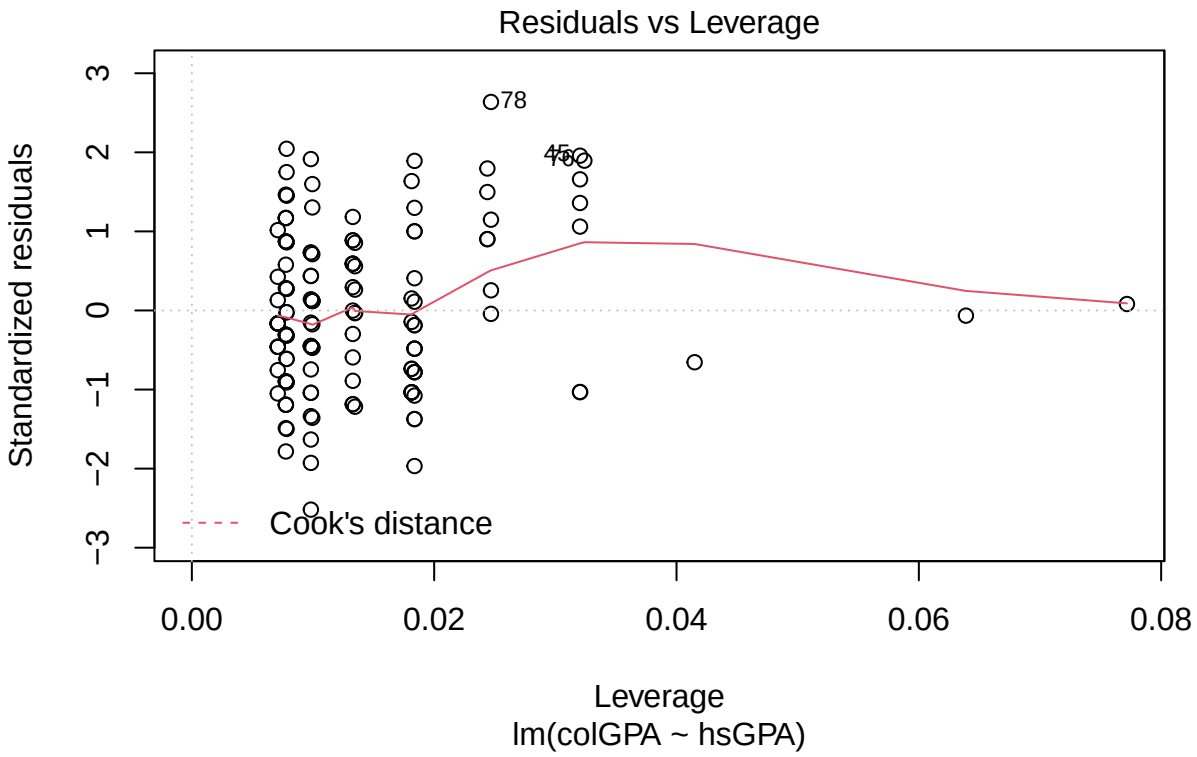
```
##
## Call:
## lm(formula = colGPA ~ hsGPA)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.85220 -0.26274 -0.04868  0.28902  0.88551
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.41543    0.30694   4.611 8.98e-06 ***
## hsGPA        0.48243    0.08983   5.371 3.21e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.34 on 139 degrees of freedom
##   (184 observations deleted due to missingness)
## Multiple R-squared:  0.1719, Adjusted R-squared:  0.1659
## F-statistic: 28.85 on 1 and 139 DF,  p-value: 3.211e-07
```

```
plot(modelGH)
```


Residuals vs Fitted
Residuals
1.0
0.5
0.0
−0.5
−1.0
78
33
138
2.6
2.8
3.0
3.2
Fitted values
lm(colGPA ~ hsGPA)


Normal Q−Q
Standardized residuals
3
2
1
0
−1
−2
78
33
138
−2
−1
0
1
2
Theoretical Quantiles
lm(colGPA ~ hsGPA)


Scale−Location
78
138
33
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.6
2.8
3.0
3.2
Fitted values
lm(colGPA ~ hsGPA)




```
lower1 = sqrt(((139)*0.34**2)/qchisq(.9,139))
upper1 = sqrt(((139)*0.34**2)/qchisq(.1,139))
```

a) The 80% confidence interval suggests that ACT is not an adequate predictor. The confidence interval is .3399679 to .3966225.

b)

i. For every increase in unit of hsGPA, colGPA increases by approximately on the average .48.

ii. We are testing if H0: beta_1 = 0 because we want to see if there is any difference between hsGPA and colGPA. We reject the null hypothesis bc p-value is less than alpha. Thus there is a significant difference between hsGPA and colGPA.

iii. The residual plot passes the linearity test and the equal variance condition. It also passes normality and randomness conditions.

iv. Yes hsGPA is an adequate predictor for colGPA using the admission committee's guidelines because the upper bound of the confidence interval is less than .5/1.28.

v. The hsGPA seems like a better predictor because its R-Squared value is 17.19% which is greater than the ACT predictor's R-Squared value of 4.275%. Upper bound for hsGPA also less than .5/1.28 so it's an adequate predictor, whereas ACT is not even an adequate because the upper bound is greater than .5/1.28.

4.2 Adirondack High Peaks

```
library(car)
```

```
## Loading required package: carData
```

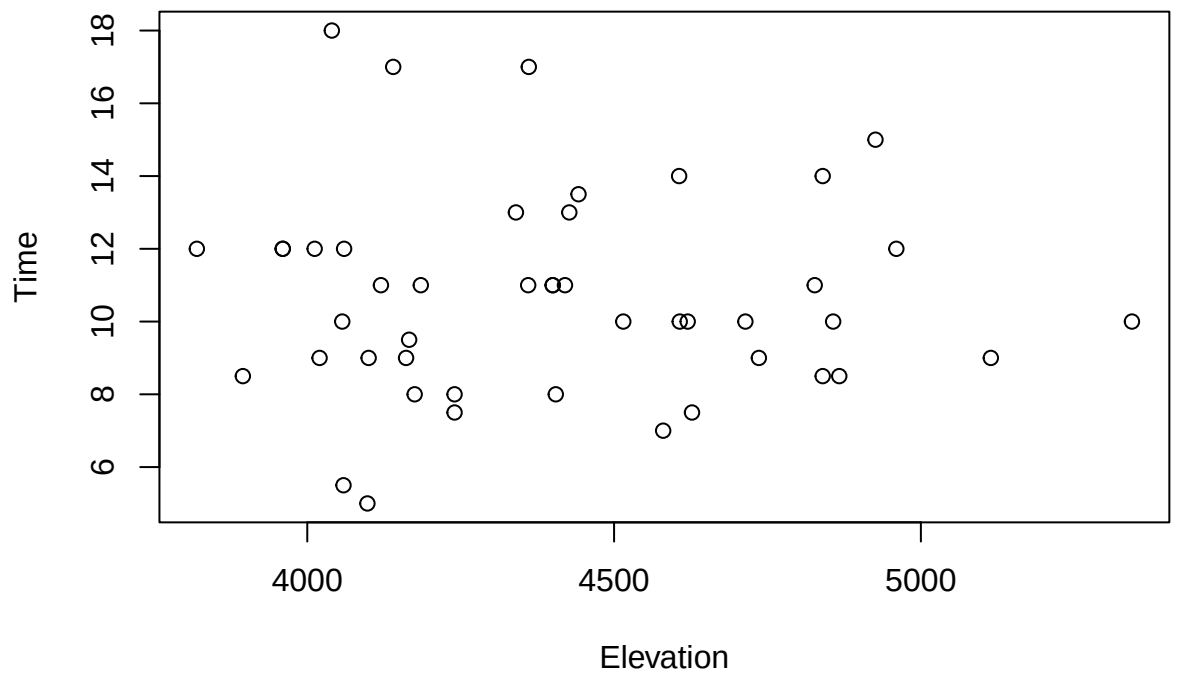
```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:boot':
##
##     logit
```

```
peaks = read.csv("C:/Users/adhri/OneDrive/Documents/R/App_Reg_and_Time_Series/chpt4/datasets/HighPeaks.
attach(peaks)
```

```
## The following object is masked from hospital:
##
##     Length
```

```
plot(Time ~ Elevation)
```



```r
modelP = lm(Time ~ Elevation + Length)
summary(modelP)
```

```
##
## Call:
## lm(formula = Time ~ Elevation + Length)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.5924 -0.8050 -0.1959  0.6380  3.8432
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.0753787  2.5327132   3.188  0.00267 **
## Elevation   -0.0014483  0.0005805  -2.495  0.01653 *
## Length       0.7123344  0.0593330  12.006 2.54e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.37 on 43 degrees of freedom
## Multiple R-squared:  0.7703, Adjusted R-squared:  0.7596
## F-statistic: 72.09 on 2 and 43 DF,  p-value: 1.844e-14
```

```r
modelL = lm(Time ~ Length)
summary(modelL)
```

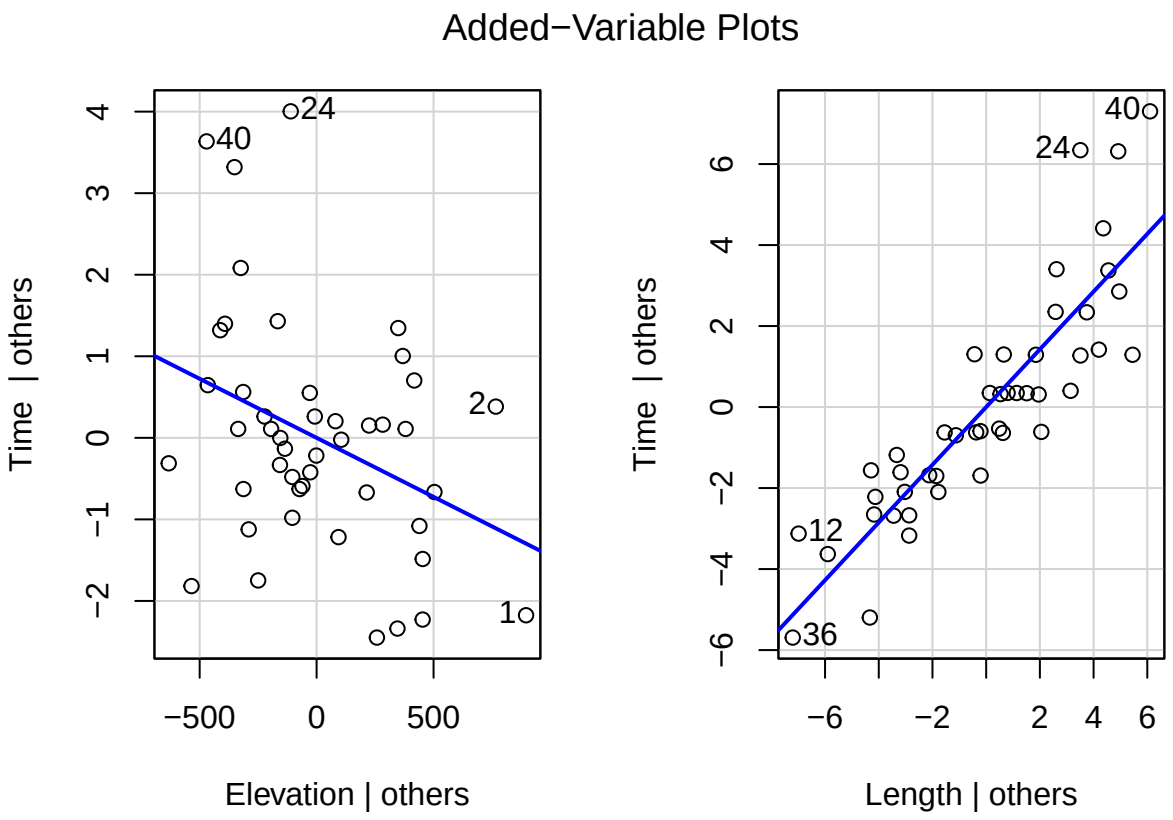
```
##
## Call:
## lm(formula = Time ~ Length)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.4491 -0.6687 -0.0122  0.5590  4.0034
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.04817    0.80371   2.548   0.0144 *
## Length       0.68427    0.06162  11.105 2.39e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.449 on 44 degrees of freedom
## Multiple R-squared:  0.737,  Adjusted R-squared:  0.7311
## F-statistic: 123.3 on 1 and 44 DF,  p-value: 2.39e-14
```

```r
anova(modelL, modelP, test = 'F')
```

```
## Analysis of Variance Table
##
## Model 1: Time ~ Length
## Model 2: Time ~ Elevation + Length
##   Res.Df    RSS Df Sum of Sq      F  Pr(>F)
```

```
## 1     44 92.415
## 2     43 80.731  1    11.684 6.2234 0.01653 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#c
avPlots(modelP)
```



a) It does not look like elevation would be a good predictor for time because the data is very scattered and random.

b) The t-test and nested F-test indicate that elevation is a relevant predictor for the multiple regression model. The two predictor model is better at explaining time because it has an R-Squared value of 77.03% whereas the R-Squared of the Length predictor alone is 73.7%. We already know Elevation is a poor predictor for Time.

c) There is a clear negative trend with added variables plot for elevation.

4.18 GPA by Verbal SAT slope

```
library(caTools)
satgpa = read.csv("C:/Users/adhri/OneDrive/Documents/R/App_Reg_and_Time_Series/chpt4/datasets/SATGPA.csv
attach(satgpa)
```

```r
# set.seed(1)
# data1 = sample.split(GPA, SplitRatio = 0.80)
# train = subset(satgpa, data1 == TRUE)
# test = subset(satgpa, data1 == FALSE)

modelS = lm(GPA~VerbalSAT)
summary(modelS)
```

```
##
## Call:
## lm(formula = GPA ~ VerbalSAT)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.62002 -0.25932  0.03885  0.20502  0.51621
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2.6042036  0.4377919   5.948  5.5e-06 ***
## VerbalSAT   0.0009056  0.0007659   1.182     0.25
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3154 on 22 degrees of freedom
## Multiple R-squared:  0.05976,    Adjusted R-squared:  0.01702
## F-statistic: 1.398 on 1 and 22 DF,  p-value: 0.2496
```

```r
rsquare = summary(modelS)$r.squared
vsat = satgpa[c("VerbalSAT")]

count = 0

for (i in 1:1000){
  randomized = cbind(vsat, satgpa$GPA[sample(nrow(satgpa))])
  colnames(randomized) = c("VerbalSAT", "GPA")
  model = lm(randomized$GPA~randomized$VerbalSAT)
  cur = summary(model)$r.squared

  if (cur >= rsquare){
    count = count + 1
  }
}

count/1000
```

```
## [1] 0.235
```

The p-value is close to .25 so we fail to reject the null hypothesis. We cannot conclude that the coefficient is significantly different from zero. The t-value is 1.156 which correlates to the p-value of .2635 which both reject the null hypothesis and we conclude there is not a significant correlation between VerbalSAT and GPA.

4.20 Bootstrapping Adirondack hikes

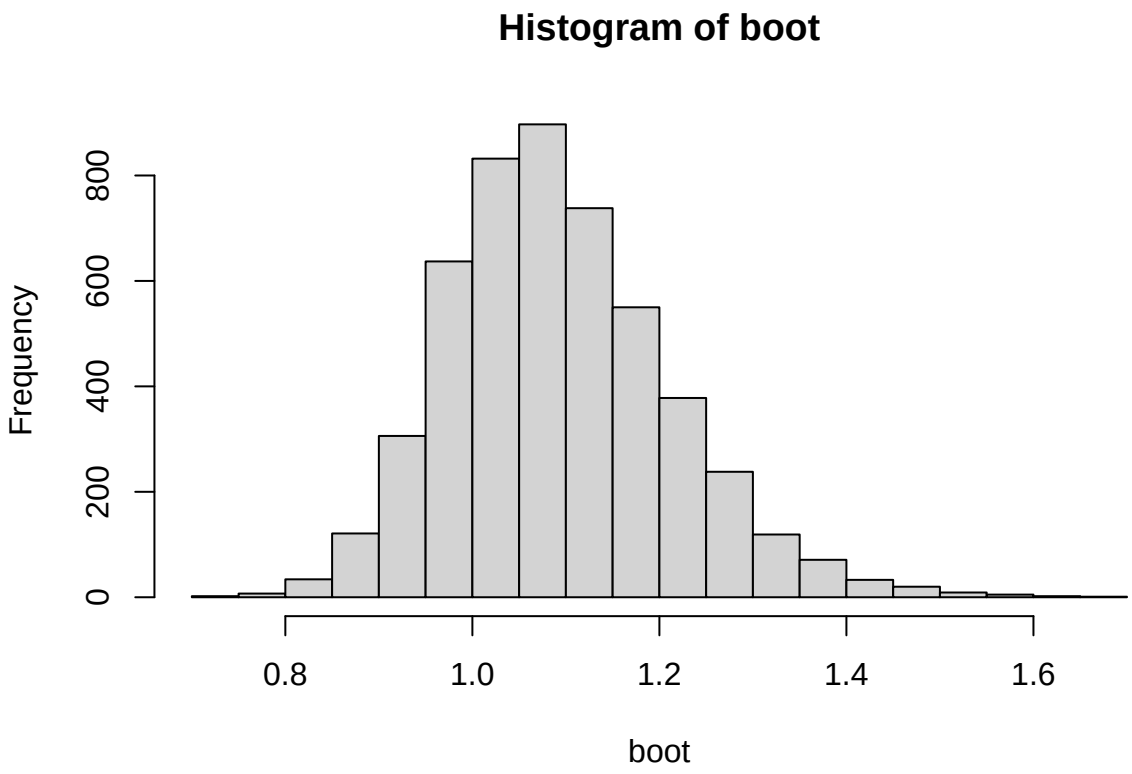
```
peaks = read.csv("C:/Users/adhri/OneDrive/Documents/R/App_Reg_and_Time_Series/chpt4/datasets/HighPeaks.
attach(peaks)
```

```
## The following objects are masked from peaks (pos = 5):
##
##     Ascent, Difficulty, Elevation, Length, Peak, Time

## The following object is masked from hospital:
##
##     Length
```

```
modelLT = lm(Length ~ Time)

# a
confint(modelLT, level = 0.90)
```

```
##                    5 %     95 %
## (Intercept) -0.6930744 2.893854
## Time         0.9141373 1.240075
```

```
# b
  N = 5000
  boot = c()
  for (i in 1:N){
    df = peaks[sample(nrow(peaks), nrow(peaks), replace = TRUE),]
    boot_model = lm(df$Length~df$Time)
    slope = boot_model$coefficients[2]
    boot = append(boot, slope)
  }

  hist(boot)
```




```
# c
avg = mean(boot)
stdev = sd(boot)

# d
avg - qt(.95, length(Time))*stdev
```

```
## [1] 0.8928055
```

```
avg + qt(.95, length(Time))*stdev
```

```
## [1] 1.293049
```

```
# e
sorted = sort(boot)
l = sorted[N*0.05]
r = sorted[N*0.95]

# f
upper_dist = r - 1.07711
lower_bound = 1.07711 - upper_dist
lower_dist = 1.07711 - l
upper_bound = 1.07711 + lower_dist
```

a) We are 90% confident that the mean time taken to hike is between 0.9141373 and 1.240075.

b) The histogram is slightly skewed right and appears to be centered around 1.1.

c) Mean and SD of bootstrap slopes are roughly equal to that of the un-bootstrapped model.

d) CI 0.8909787 to 1.295605.

e) CI .919 to 1.31

f) New lower and upper bounds are .844 to 1.24.

g) Intervals are similar. Differences can be attributed to slight right skewed bootstrap.