HW7

Minh Luc, Devin Pham, Kyle Moore

Friday of Week 7, 05/13/2022

Contents

We all contributed equally for this homework.

Question 0

Member 1:

- Name: Minh Luc
- Student ID: A17209607

Member 2:

- Name: Kyle Moore
- Student ID: A14271413

Member 3:

- Name: Devin Pham
- Student ID: A17198936

Question 1

- **(a)**

```r
library(ISLR2)

lm1.fit <- lm(mpg ~ cylinders + displacement + horsepower + weight +
              acceleration + year + origin, data=Auto, subset=1:391)

newdata <- Auto[392,]

predict(lm1.fit, newdata, interval = "confidence")
```

```
##          fit      lwr      upr
## 397 28.67757 27.69303 29.66211
```

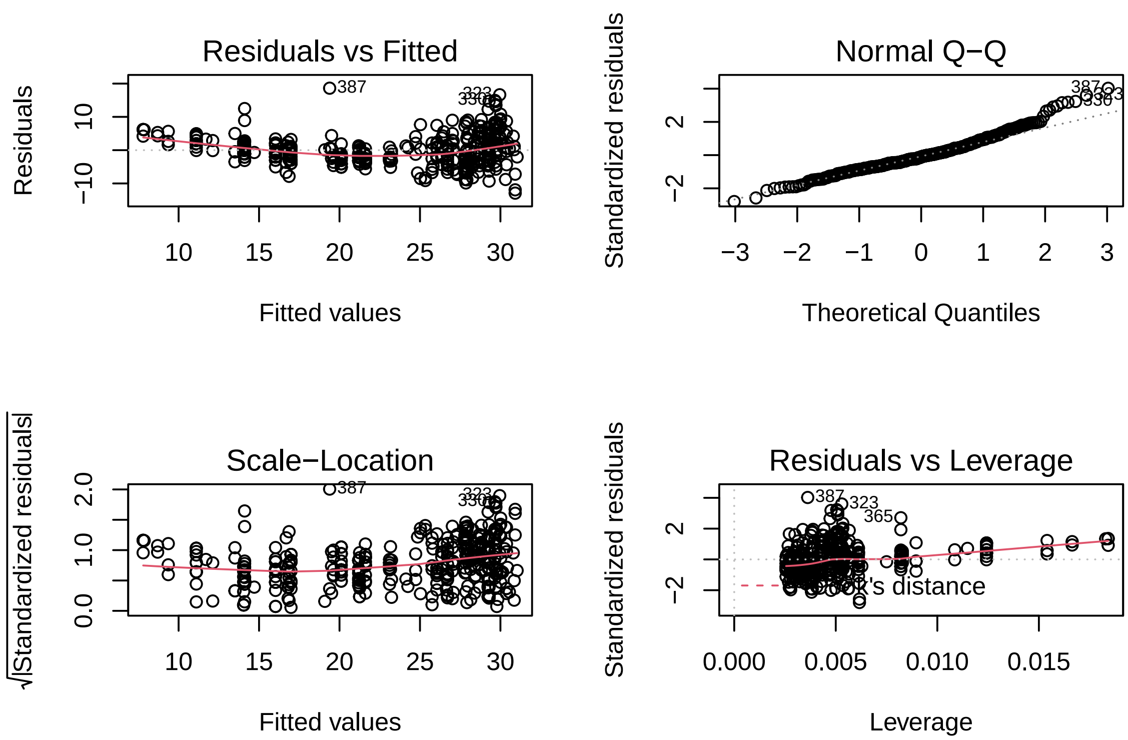
Answer:

The estimated mpg for newdata is 28.67757, and the 95% confidence interval is (27.69303, 29.66211)

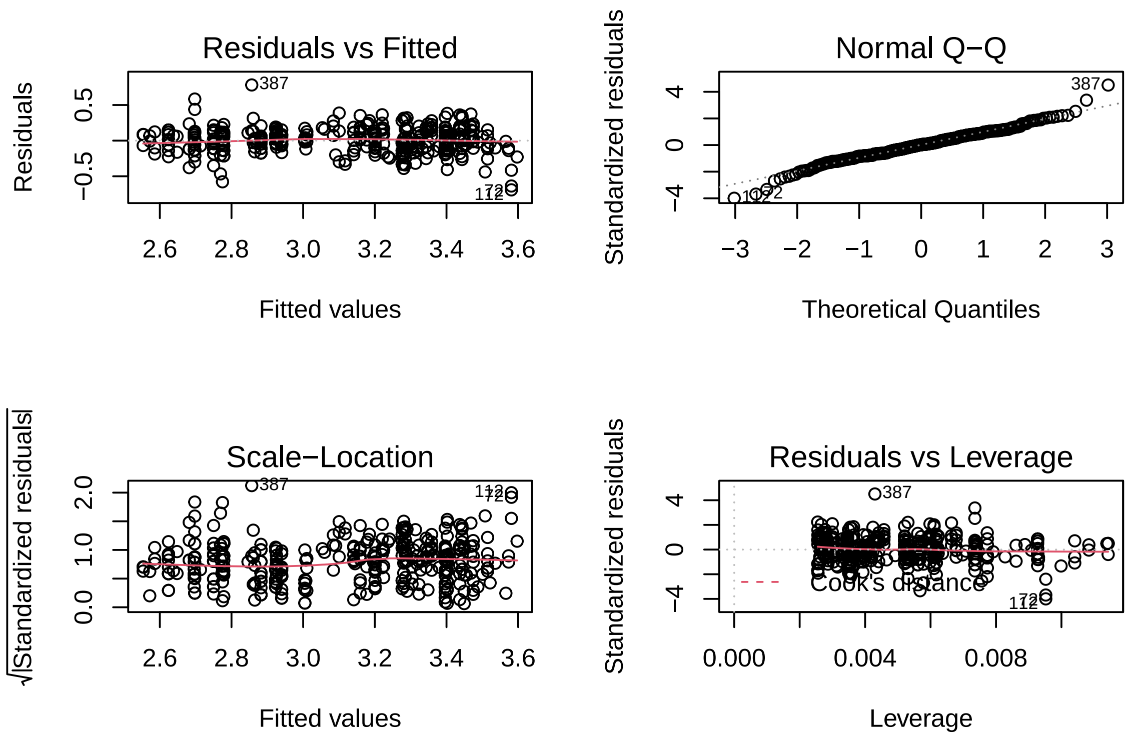
- **(b)**

```r
  model1 <- lm(mpg ~ displacement, data = Auto)
  model2 <- lm(log(mpg) ~ log(displacement), data = Auto)

  opar <- par(mfrow = c(2,2), oma = c(0, 0,0,0))
  plot(model1)
```




```r
  plot(model2)
```



Answer:

Model 2 is more appropriate for the linear model. Its Q-Q plot follows the line better and the residuals vs fitted plot shows heteroscedasticity, which means that the error follows a normal distribution which is better for a linear model.

Question 2

```r
library(ISLR2)
library(boot)
Auto0 <- Auto[,-9] # remove the last variable
p = 7
m = 3
set.seed(m)
glm.full <- glm(mpg~.,data=Auto0)
```

- **(a)**

```r
glm0 <- glm(mpg~1,data=Auto0)
```

- **(b)**

```r
library(qpcR)
```

```
## Loading required package: MASS
```

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

```
## The following object is masked from 'package:ISLR2':
##
##     Boston
```

```
## Loading required package: minpack.lm
```

```
## Loading required package: rgl
```

```
## Loading required package: robustbase
```

```
##
## Attaching package: 'robustbase'
```

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

```
## Loading required package: Matrix
```

```
glm11 <- glm(mpg ~ cylinders,data=Auto0)
glm12 <- glm(mpg ~ displacement,data = Auto0)
glm13 <- glm(mpg ~ horsepower,data = Auto0)
glm14 <- glm(mpg ~ weight,data = Auto0)
glm15 <- glm(mpg ~ acceleration,data = Auto0)
glm16 <- glm(mpg ~ year,data = Auto0)
glm17 <- glm(mpg ~ origin,data = Auto0)

RSS(glm11)
```

```
## [1] 9415.91
```

```
RSS(glm12)
```

```
## [1] 8378.822
```

```
RSS(glm13)
```

```
## [1] 9385.916
```

```
RSS(glm14)
```

```
## [1] 7321.234
```

```
RSS(glm15)
```

```
## [1] 19550.46
```

```
RSS(glm16)
```

```
## [1] 15791.33
```

```
RSS(glm17)
```

```
## [1] 16209.76
```

```
glm1 <- glm14
```

The model glm14 has the lowest RSS, which is for the predictor weight.

- **(c)**

```
library(qpcR)

glm21 <- update(glm1, mpg ~ weight + cylinders)
```

```r
glm22 <- update(glm1, mpg ~ weight + displacement)
glm23 <- update(glm1, mpg ~ weight + horsepower)
glm24 <- update(glm1, mpg ~ weight + acceleration)
glm25 <- update(glm1, mpg ~ weight + year)
glm26 <- update(glm1, mpg ~ weight + origin)

RSS(glm21)
```

```
## [1] 7206.115
```

```r
RSS(glm22)
```

```
## [1] 7170.308
```

```r
RSS(glm23)
```

```
## [1] 6993.845
```

```r
RSS(glm24)
```

```
## [1] 7152.893
```

```r
RSS(glm25)
```

```
## [1] 4568.952
```

```r
RSS(glm26)
```

```
## [1] 7098.988
```

```r
glm2 <- glm25
```

The model glm25 has the lowest RSS, which is for the predictor year.

- **(d)**

```r
library(qpcR)

glm31 <- update(glm2, mpg ~ weight + year + cylinders)
glm32 <- update(glm2, mpg ~ weight + year + displacement)
glm33 <- update(glm2, mpg ~ weight + year + horsepower)
glm34 <- update(glm2, mpg ~ weight + year + acceleration)
glm35 <- update(glm2, mpg ~ weight + year + origin)

RSS(glm31)
```

```
## [1] 4563.994
```

```r
RSS(glm32)
```

```
## [1] 4568.91
```

```r
RSS(glm33)
```

```
## [1] 4565.651
```

```r
RSS(glm34)
```

```
## [1] 4558.502
```

```r
RSS(glm35)
```

```
## [1] 4348.105
```

```r
glm3 <- glm35
```

The model glm33 has the lowest RSS, which is for the predictor origin.

- **(e)**

```r
library(qpcR)

glm41 <- update(glm3, mpg ~ weight + year + origin + cylinders)
glm42 <- update(glm3, mpg ~ weight + year + origin + displacement)
glm43 <- update(glm3, mpg ~ weight + year + origin + horsepower)
glm44 <- update(glm3, mpg ~ weight + year + origin + acceleration)

RSS(glm41)
```

```
## [1] 4347.958
```

```r
RSS(glm42)
```

```
## [1] 4332.729
```

```r
RSS(glm43)
```

```
## [1] 4333.7
```

```r
RSS(glm44)
```

```
## [1] 4333.073
```

```r
glm4 <- glm42
```

The model glm42 has the lowest RSS, which is for the predictor displacement.

```r
library(qpcR)

glm51 <- update(glm1, mpg ~ weight + year + origin + displacement + cylinders)
glm52 <- update(glm1, mpg ~ weight + year + origin + displacement + horsepower)
glm53 <- update(glm1, mpg ~ weight + year + origin + displacement + acceleration)

RSS(glm51)
```

```
## [1] 4313.373
```

```r
RSS(glm52)
```

```
## [1] 4286.842
```

```r
RSS(glm53)
```

```
## [1] 4290.224
```

```r
glm5 <- glm52
```

The model glm52 has the lowest RSS, which is for the predictor horsepower.

```r
library(qpcR)

glm61 <- update(glm1, mpg ~ weight + year + origin + displacement + horsepower + cylinders)
glm62 <- update(glm1, mpg ~ weight + year + origin + displacement + horsepower + acceleration)
```

```r
RSS(glm61)
```

```
## [1] 4259.571
```

```r
RSS(glm62)
```

```
## [1] 4278.004
```

```r
glm6 <- glm61
```

The model glm61 has the lowest RSS, which is for the predictor cylinders.

```r
library(qpcR)

glm71 <- update(glm1, mpg ~ weight + year + origin + displacement + horsepower + cylinders + acce

RSS(glm71)
```

```
## [1] 4252.213
```

```r
glm7 <- glm71
```

The model glm71 has the lowest RSS and is the last predictor, which is for the predictor acceleration.

- **(f)**

```r
cv.error0 = cv.glm(Auto0, glm0)
cv.error1 = cv.glm(Auto0, glm1)
cv.error2 = cv.glm(Auto0, glm2)
cv.error3 = cv.glm(Auto0, glm3)
cv.error4 = cv.glm(Auto0, glm4)
cv.error5 = cv.glm(Auto0, glm5)
cv.error6 = cv.glm(Auto0, glm6)
cv.error7 = cv.glm(Auto0, glm7)

cv.error0$delta[1]
```

```
## [1] 61.07394
```

```r
cv.error1$delta[1]
```

```
## [1] 18.85161
```

```r
cv.error2$delta[1]
```

```
## [1] 11.83564
```

```r
cv.error3$delta[1]
```

```
## [1] 11.32646
```

```r
cv.error4$delta[1]
```

```
## [1] 11.35661
```

```r
cv.error5$delta[1]
```

```
## [1] 11.30361
```

```r
cv.error6$delta[1]
```

```
## [1] 11.28777
```

```
cv.error7$delta[1]
```

```
## [1] 11.37113
```

The model glm6 has the lowest value of LOOCV. ***

Question 3

- **(e)** You do not need to compare your answer with part b.

This problem expects you to realize that when there is no simple formula to compute, you can try to estimate using the bootstrap.