STAT 435 HW 2

Pat McCornack

2022-10-14

1. Chapter 3: Question 14 of ISLR Book:

a) What is the form of the linear model y?

The linear model y is of the form $y = \beta_1 x_1 + \beta_2 x_2 + \epsilon$ where $\beta_1 = 2$ and $\beta_2 = 0.3$ and the intercept is equal to 2.

```
library(MASS)
set.seed(1)
x1 <- runif(100)
x2 <- 0.5 * x1 + rnorm(100) / 10
y <- 2 + 2 * x1 + 0.3 * x2 + rnorm(100)
```

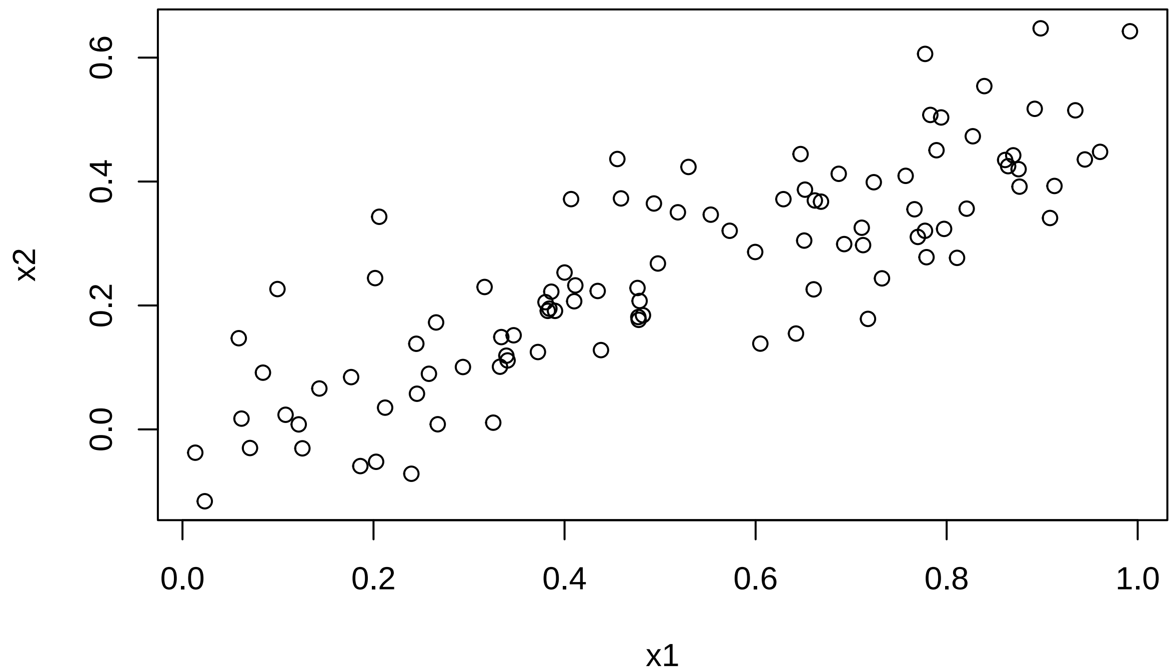
b) What is the correlation between x1 and x2?

The correlation coefficient between x2 and x2 is $r = 0.835$. This is strong positive corre

```
cor(x1, x2)
```

```
## [1] 0.8351212
```

```
plot(x1,x2)
```



c) Describe results of least squares regression.

The correlation coefficients for the linear model using x1 and x2 for least squares regression are: $\hat{\beta}_0 = 2.13, \hat{\beta}_1 = 1.44, \hat{\beta}_2 = 1.01$. The intercept is close, but both $\hat{\beta}_1 and \hat{\beta}_2$ are significantly different than the true coefficients of $\beta_1 = 2 and \beta_2 = 0.3$.

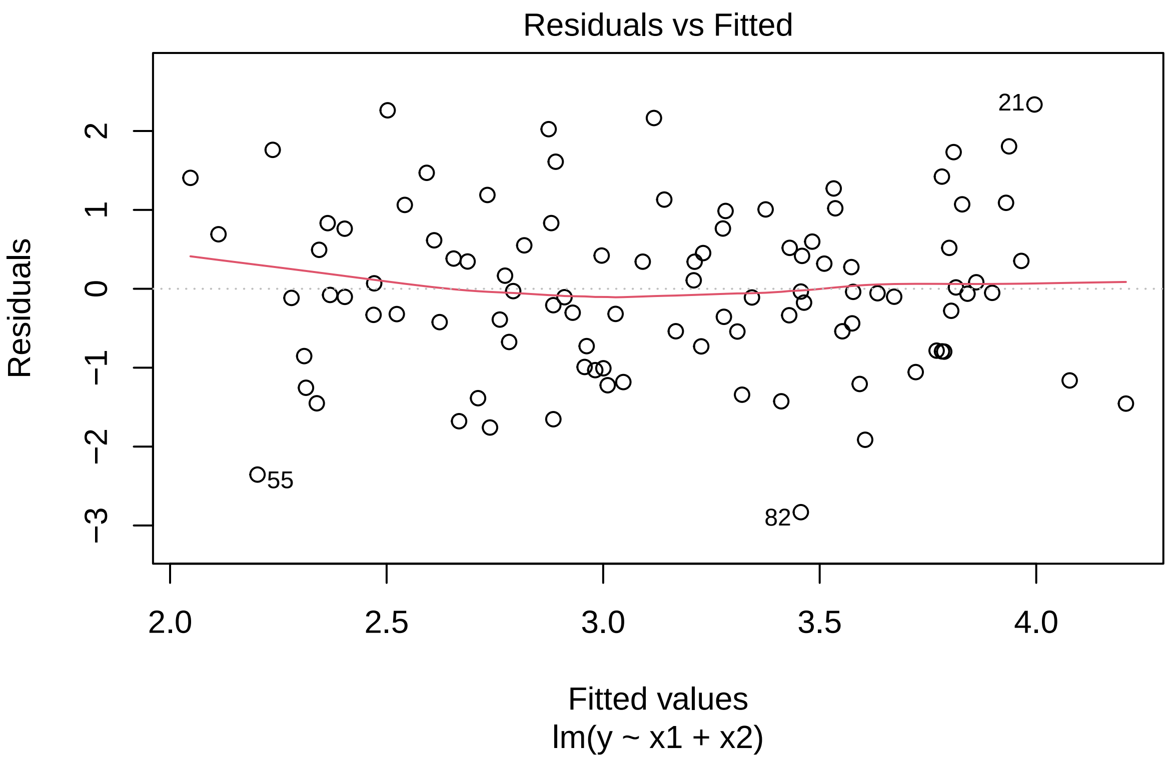
Since the p-value of $\hat{\beta}_1 = .049$ we can reject the null hypothesis $H_0 : \beta_1 = 0$ at a significance level $\alpha = 0.05$. We cannot reject the null hypothesis $\beta_2 = 0$ however.

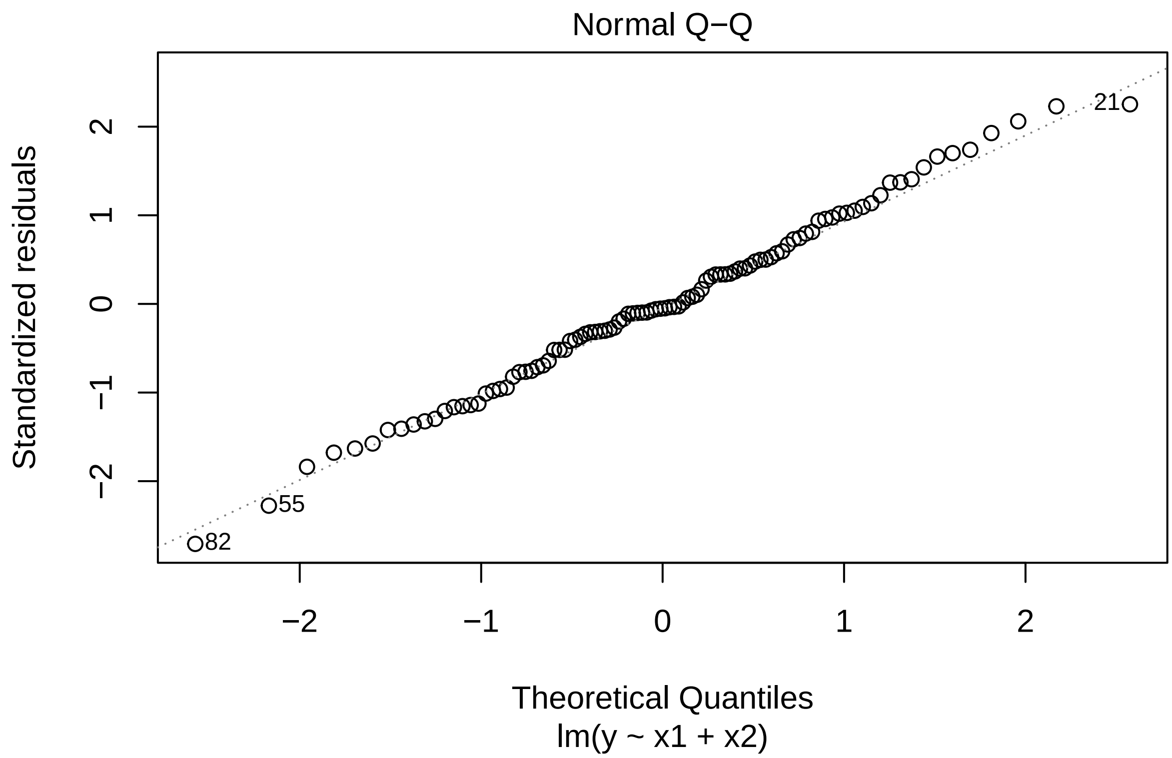
```
lm.model <- lm(y ~ x1 + x2)
summary(lm.model)
```

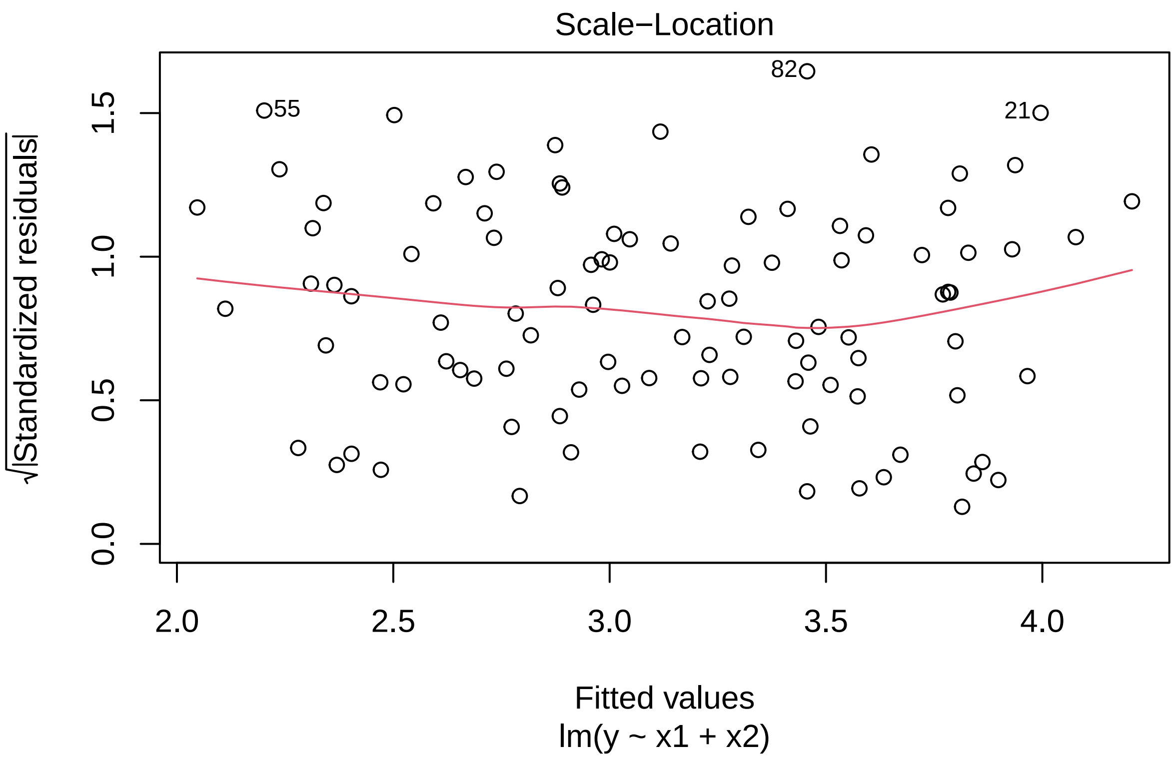
```
##
## Call:
## lm(formula = y ~ x1 + x2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8311 -0.7273 -0.0537  0.6338  2.3359
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.1305     0.2319   9.188 7.61e-15 ***
## x1            1.4396     0.7212   1.996   0.0487 *
## x2            1.0097     1.1337   0.891   0.3754
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 1.056 on 97 degrees of freedom
## Multiple R-squared:  0.2088, Adjusted R-squared:  0.1925
## F-statistic:  12.8 on 2 and 97 DF,  p-value: 1.164e-05
```

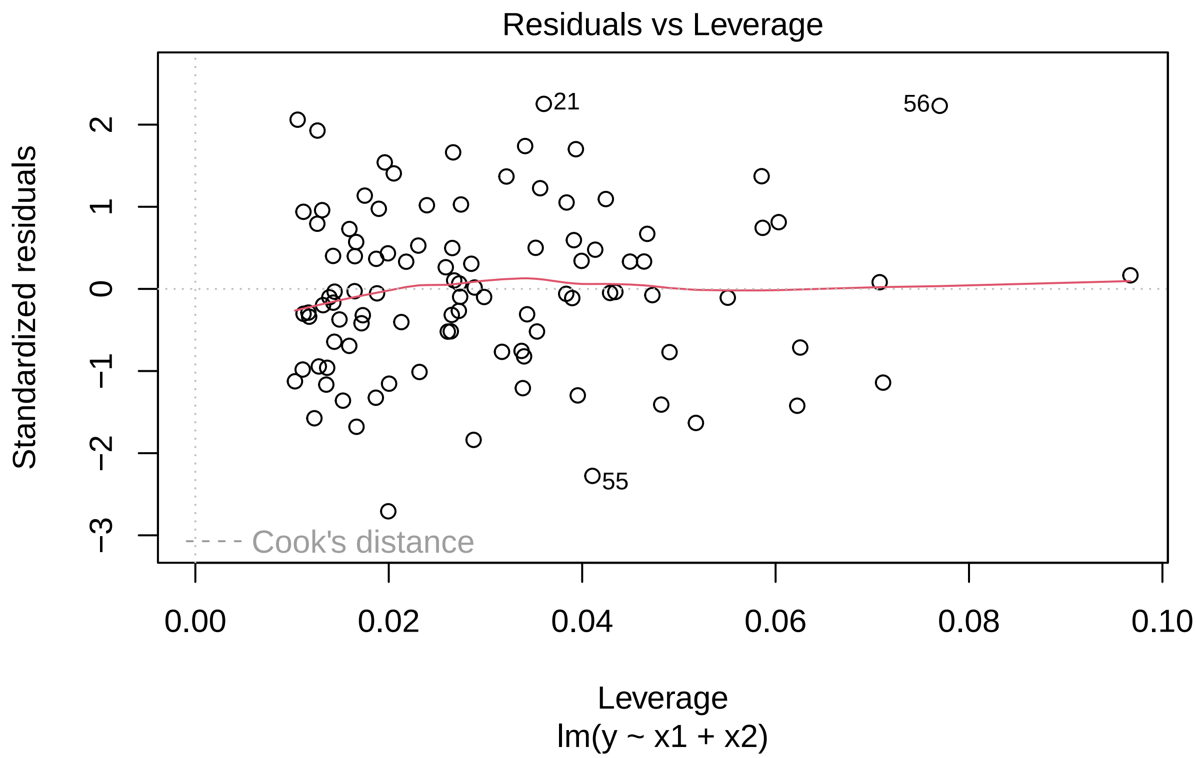
```
plot(lm.model)
```




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(y ~ x1 + x2)
2
1
0
−1
−2
−2
−1
0
1
2
21
55
82



Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
55
82
21
Fitted values
lm(y ~ x1 + x2)



d) Predict y using only x1

This model has a coefficient of $\hat{\beta}_1 = 1.98$ which is very close to the true value of 2. With a p-value of close to 0, we can reject the null hypothesis $\beta_1 = 0$.

```
x1_model <- lm(y ~ x1)
summary(x1_model)
```

```
##
## Call:
## lm(formula = y ~ x1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.89495 -0.66874 -0.07785  0.59221  2.45560
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.1124     0.2307   9.155 8.27e-15 ***
## x1            1.9759     0.3963   4.986 2.66e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.055 on 98 degrees of freedom
## Multiple R-squared:  0.2024, Adjusted R-squared:  0.1942
## F-statistic: 24.86 on 1 and 98 DF,  p-value: 2.661e-06
```

e) Predict y using only x2

The coefficient $\hat{\beta}_2$ for this model is even further from the true value than the multivariate model at a value of 2.9 vs. the true value of 0.3, but the p-value **can** be used to reject the null hypothesis $\beta_2 = 0$ at a value of close to 0.

```
x2_model <- lm(y ~ x2)
summary(x2_model)
```

```
##
## Call:
## lm(formula = y ~ x2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.62687 -0.75156 -0.03598  0.72383  2.44890
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.3899     0.1949   12.26  < 2e-16 ***
## x2            2.8996     0.6330    4.58 1.37e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.072 on 98 degrees of freedom
## Multiple R-squared:  0.1763, Adjusted R-squared:  0.1679
## F-statistic: 20.98 on 1 and 98 DF,  p-value: 1.366e-05
```

f) Do these results contradict each other.

For x1: The coefficient $\hat{\beta}_1$ from (d) is closer to the true value than that from (c), but the results aren't necessarily contradictory.

For x2: The results for coefficient $\hat{\beta}_2$ from (c) and (e) are contradictory. In (c) the result for $\hat{\beta}_2$ did not allow us to reject the null hypothesis as it was distant from the true value of β_2. The result for $\hat{\beta}_2$ from (e) did allow us to reject the null hypothesis despite being more distant from the true value.

g) Refit the model with an additional (mismeasured) observation

For the multivariate model the new observation made the coefficient less close to the true values and flipped the significance of the variables using the p-values. This model has that the null hypothesis can be rejected using the second coefficient but not the first.

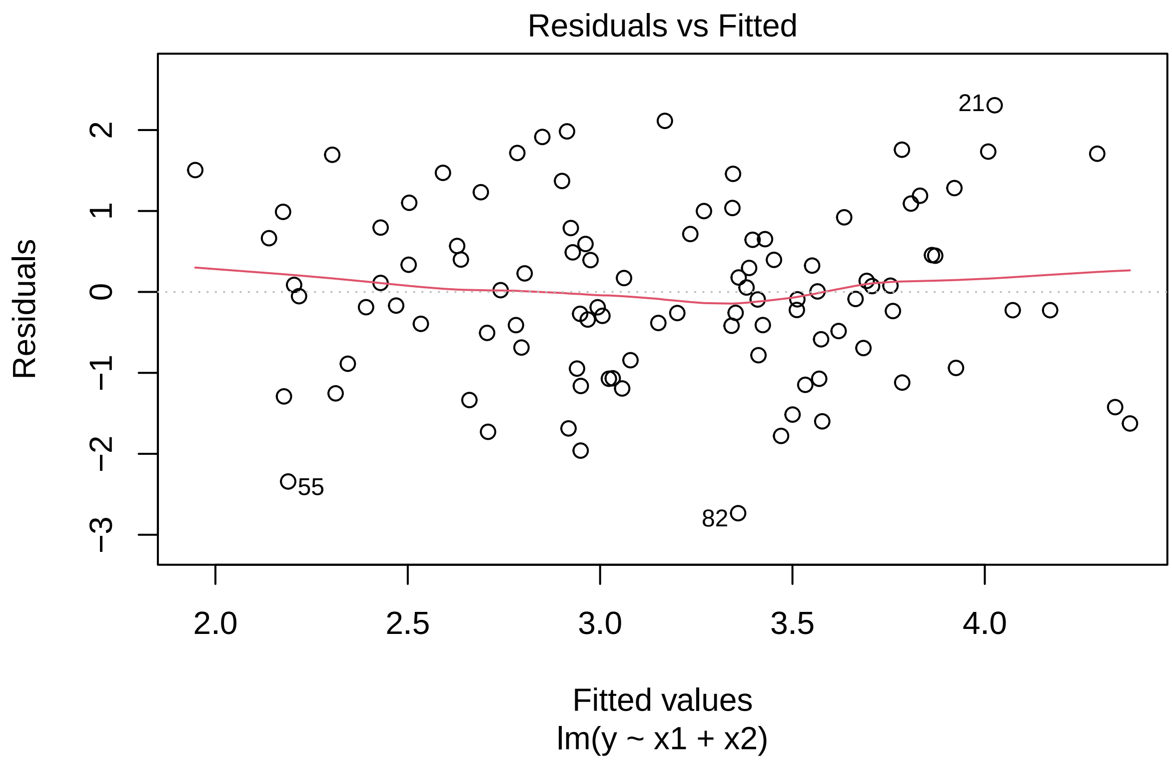
In this model the observation is neither an outlier (defined as having a studentized residual > abs(3)) nor a high leverage point (based on Cook's Distance).

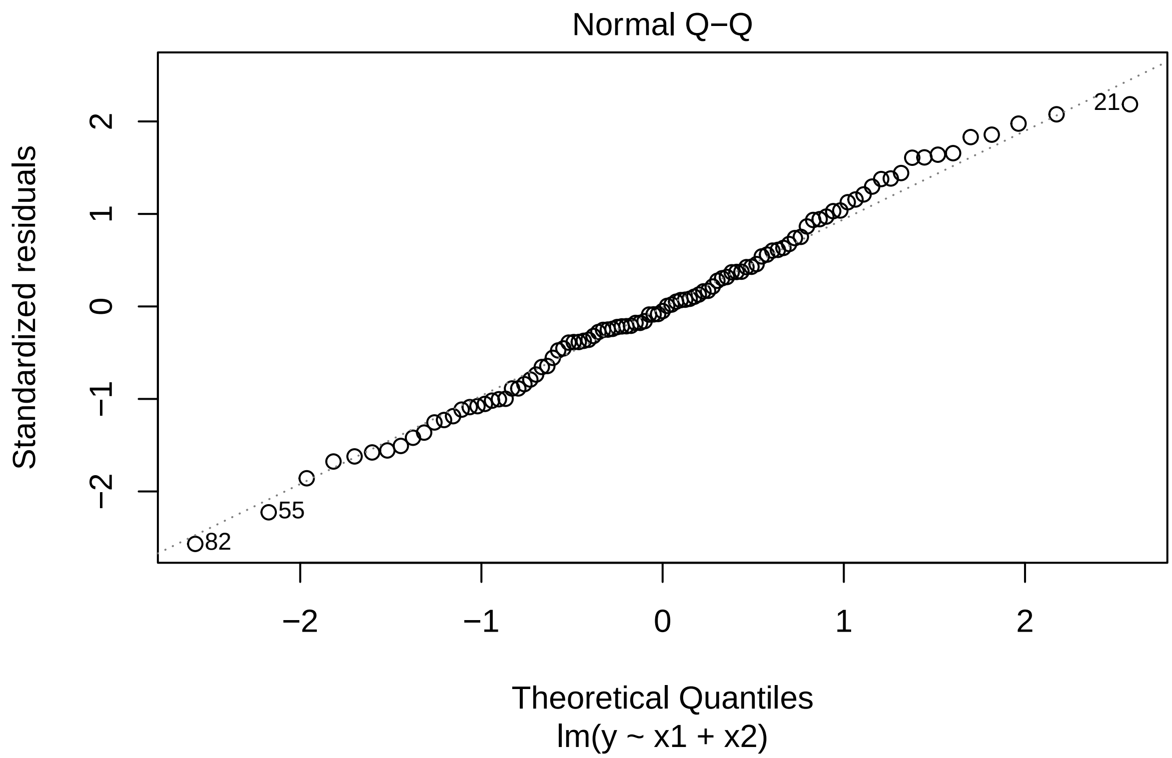
```
x1 <- c(x1, 0.1)
x2 <- c(x2, 0.8)
y <- c(y, 6)

lm.model <- lm(y ~ x1 + x2)
summary(lm.model)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2)
##
```

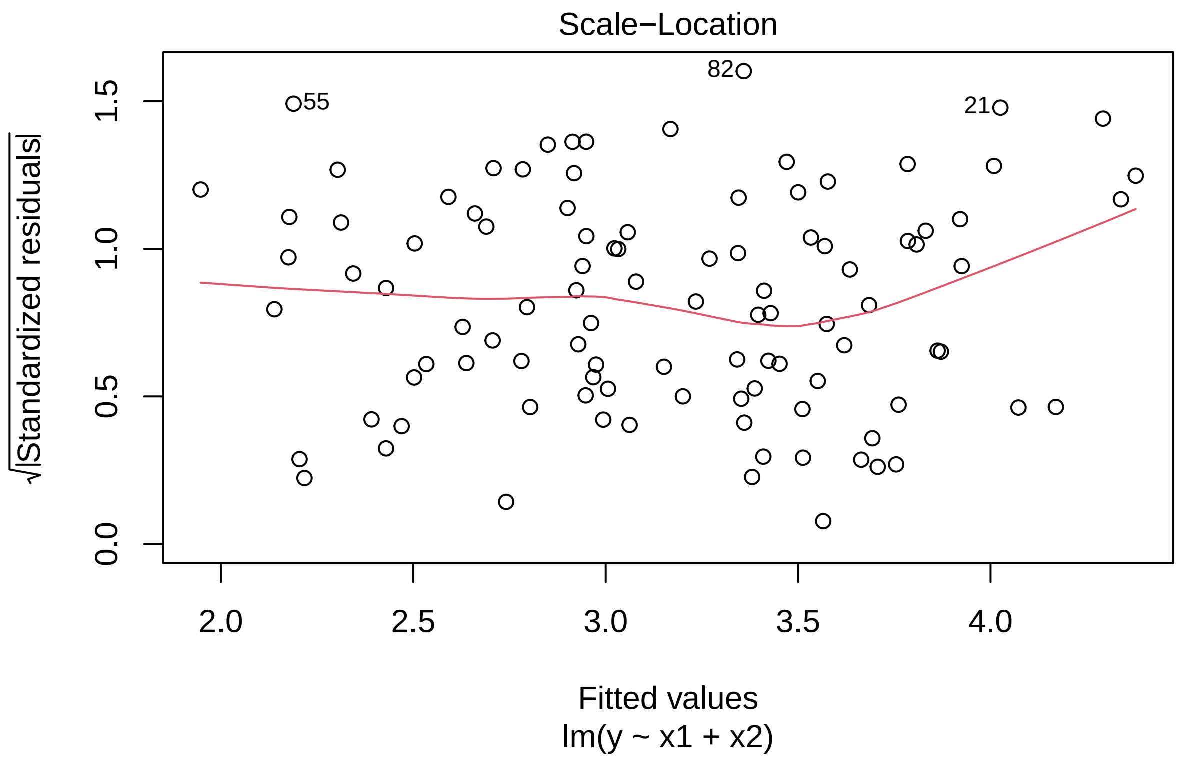
```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.73348 -0.69318 -0.05263  0.66385  2.30619
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.2267     0.2314   9.624 7.91e-16 ***
## x1            0.5394     0.5922   0.911  0.36458
## x2            2.5146     0.8977   2.801  0.00614 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.075 on 98 degrees of freedom
## Multiple R-squared:  0.2188, Adjusted R-squared:  0.2029
## F-statistic: 13.72 on 2 and 98 DF,  p-value: 5.564e-06
```

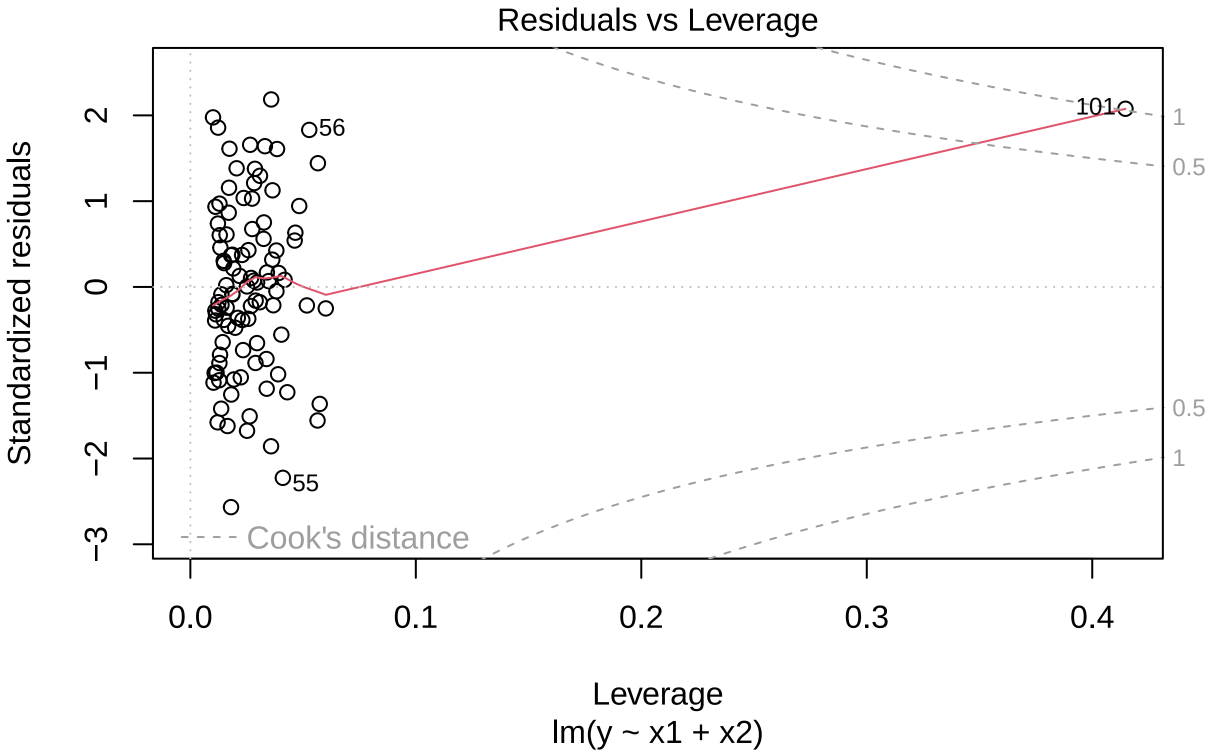
```
plot(lm.model)
```



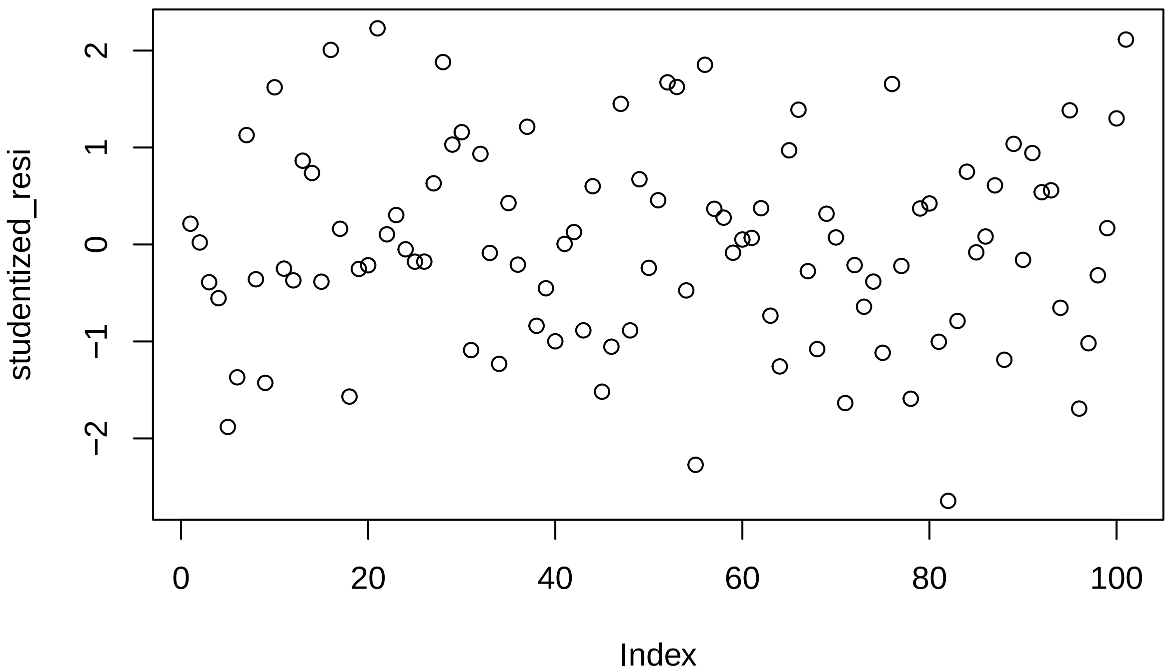


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(y ~ x1 + x2)
21
55
82
−2
−1
0
1
2


Scale−Location
82
55
21
1.5
1.0
0.5
0.0
√|Standardized residuals|
2.0
2.5
3.0
3.5
4.0
Fitted values
lm(y ~ x1 + x2)




```
# Check for outliers
studentized_resi = studres(lm.model)
plot(studentized_resi)
```




```r
# Check for leverage points
cd = cooks.distance(lm.model)
leverage_indices = which(cd > 4/nrow(lm.model))
leverage_indices
```

```
## integer(0)
```

In the model using only x1 the new observation made the coefficient more distant from the true value and raised the p-value, although the p-value is still significant enough to reject the null hypothesis.

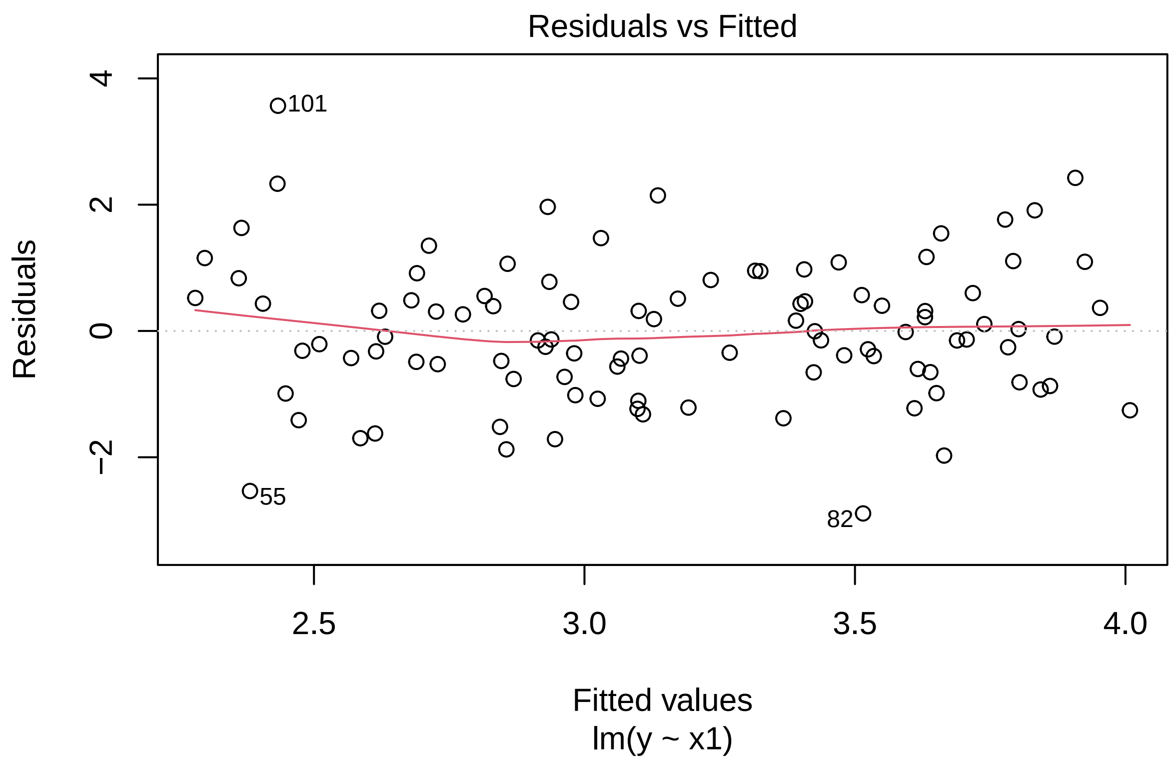
The new observation is neither an outlier nor a high leverage point in this model either based on studentized residuals and cooks distance.

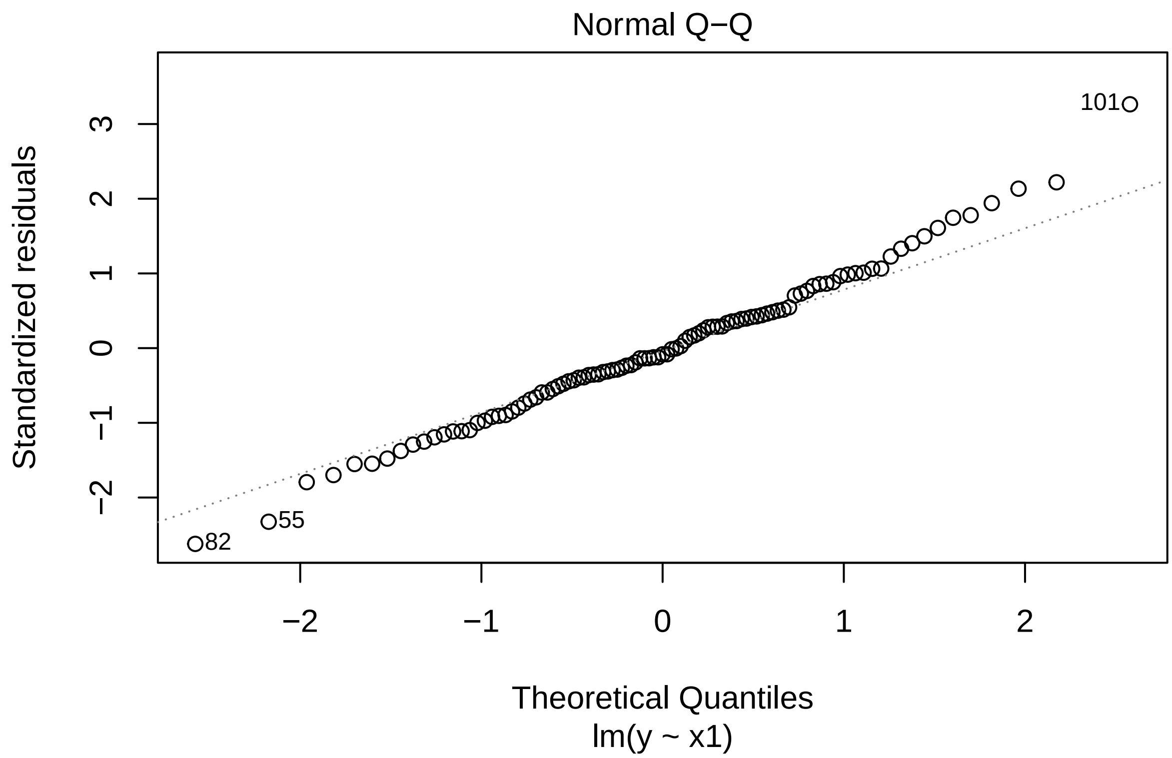
```r
x1_model <- lm(y ~ x1)
summary(x1_model)
```

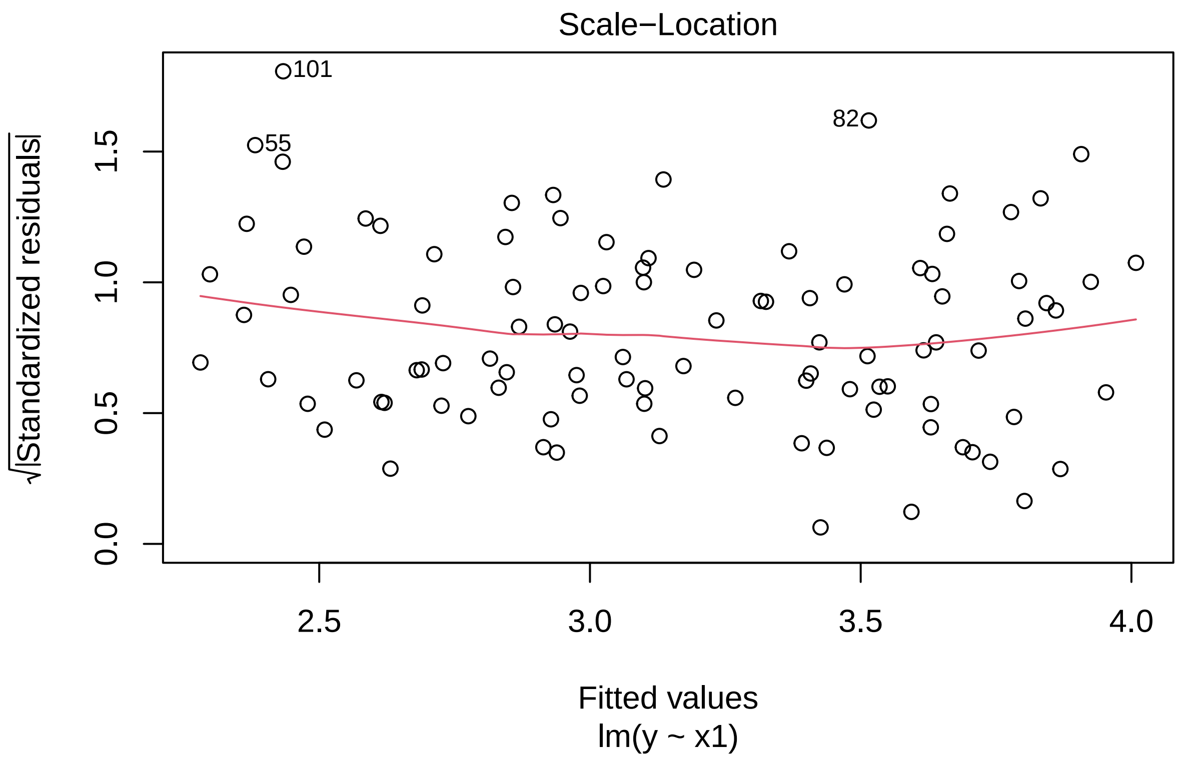
```
##
## Call:
## lm(formula = y ~ x1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8897 -0.6556 -0.0909  0.5682  3.5665
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.2569     0.2390   9.445 1.78e-15 ***
## x1            1.7657     0.4124   4.282 4.29e-05 ***
```

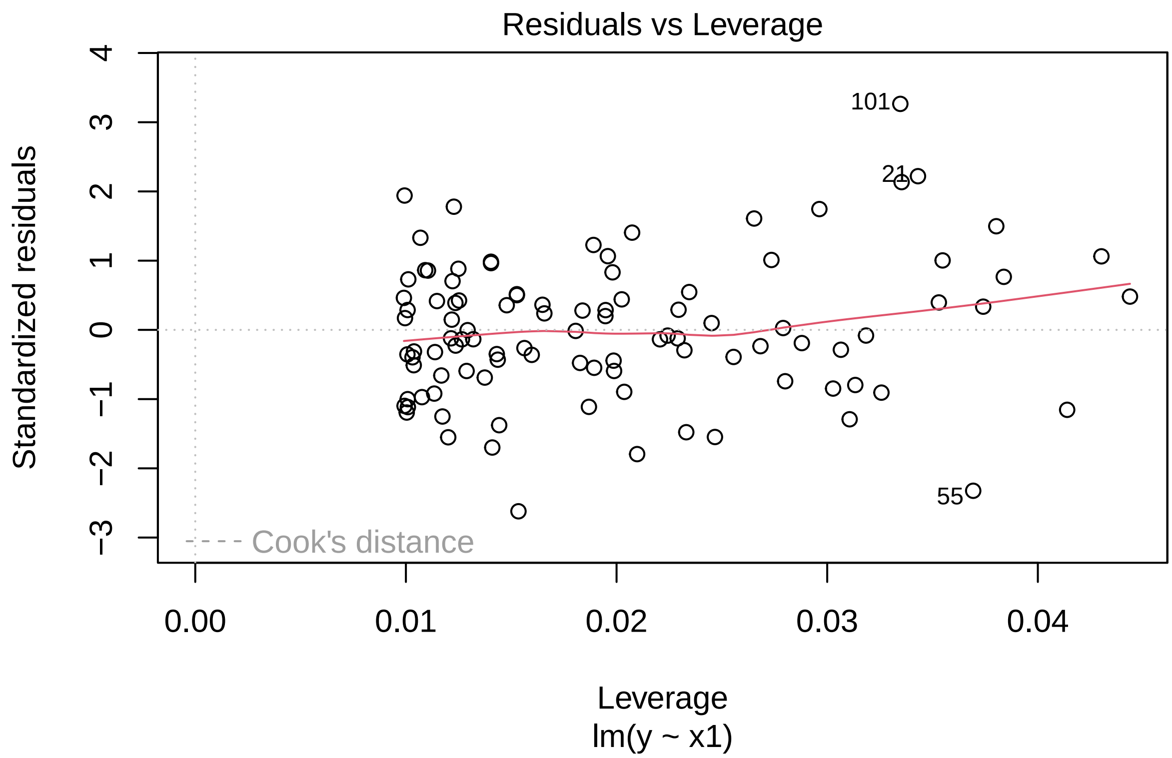
```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.111 on 99 degrees of freedom
## Multiple R-squared:  0.1562, Adjusted R-squared:  0.1477
## F-statistic: 18.33 on 1 and 99 DF,  p-value: 4.295e-05
```

```
plot(x1_model)
```

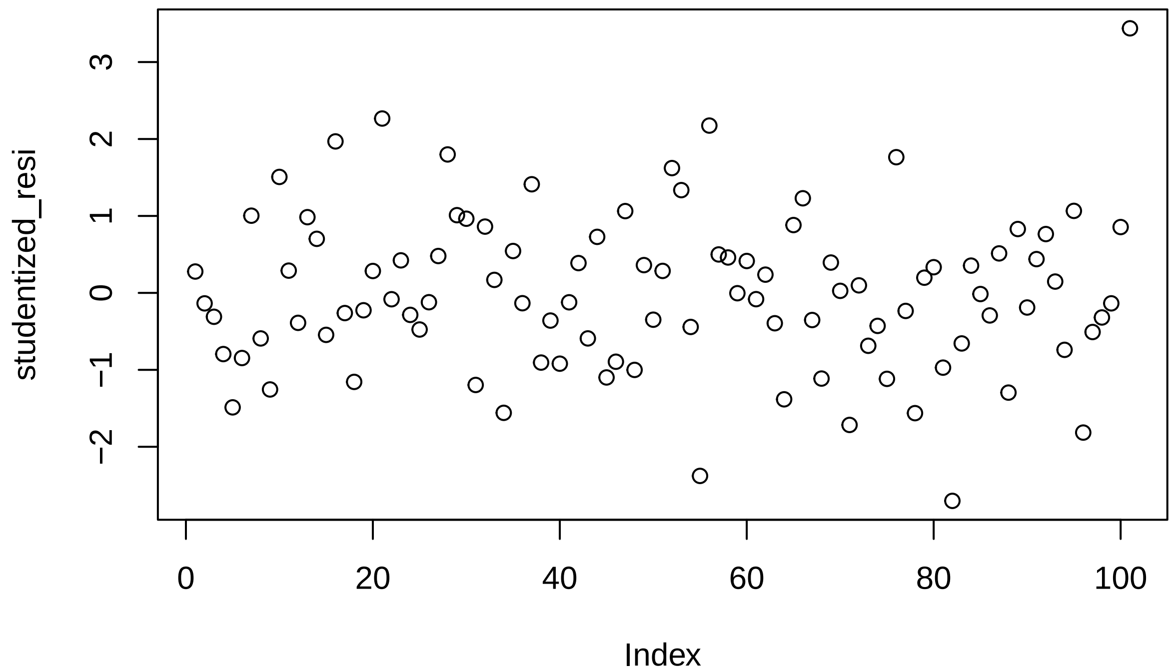



Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(y ~ x1)
101
55
82
3
2
1
0
−1
−2
−2
−1
0
1
2


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.5
3.0
3.5
4.0
Fitted values
lm(y ~ x1)
101
55
82




```
# Check for outliers
studentized_resi = studres(x1_model)
plot(studentized_resi)
```




```
outliers = which(studentized_resi > abs(3))
outliers
```

```
## 101
## 101
```

```
# Check for leverage points
cd = cooks.distance(x1_model)
leverage_indices = which(cd > 4/nrow(x1_model))
leverage_indices
```

```
## integer(0)
```

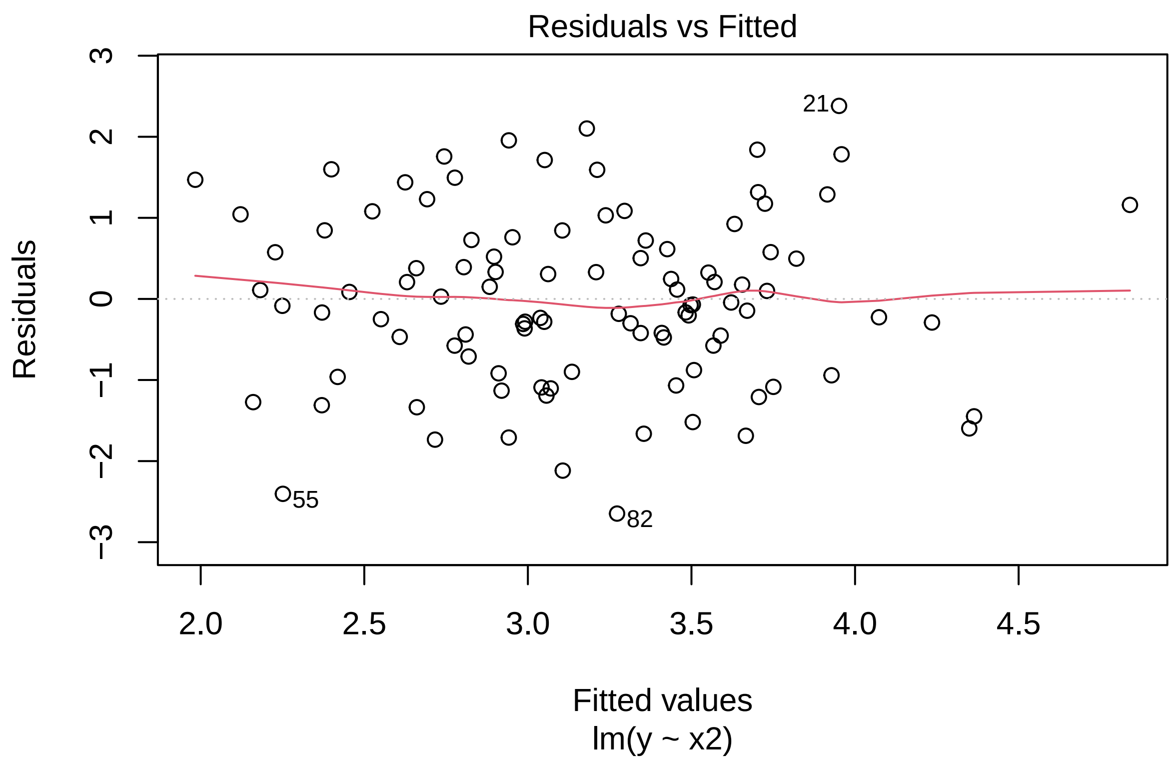
The results for the model based on only x2 are the same. The coefficient was shifted further from its true value and the new observation is not an outlier or high leverage point based on studentized residuals and cooks distance.

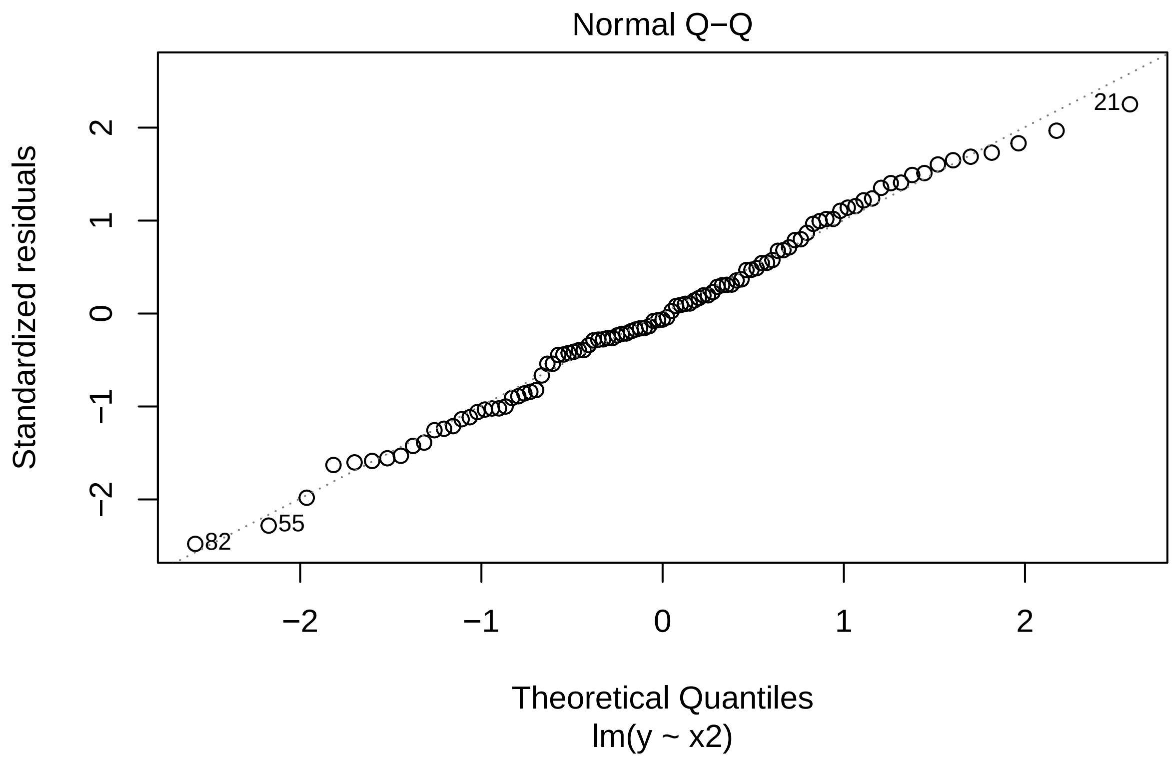
```
x2_model <- lm(y ~ x2)
summary(x2_model)
```

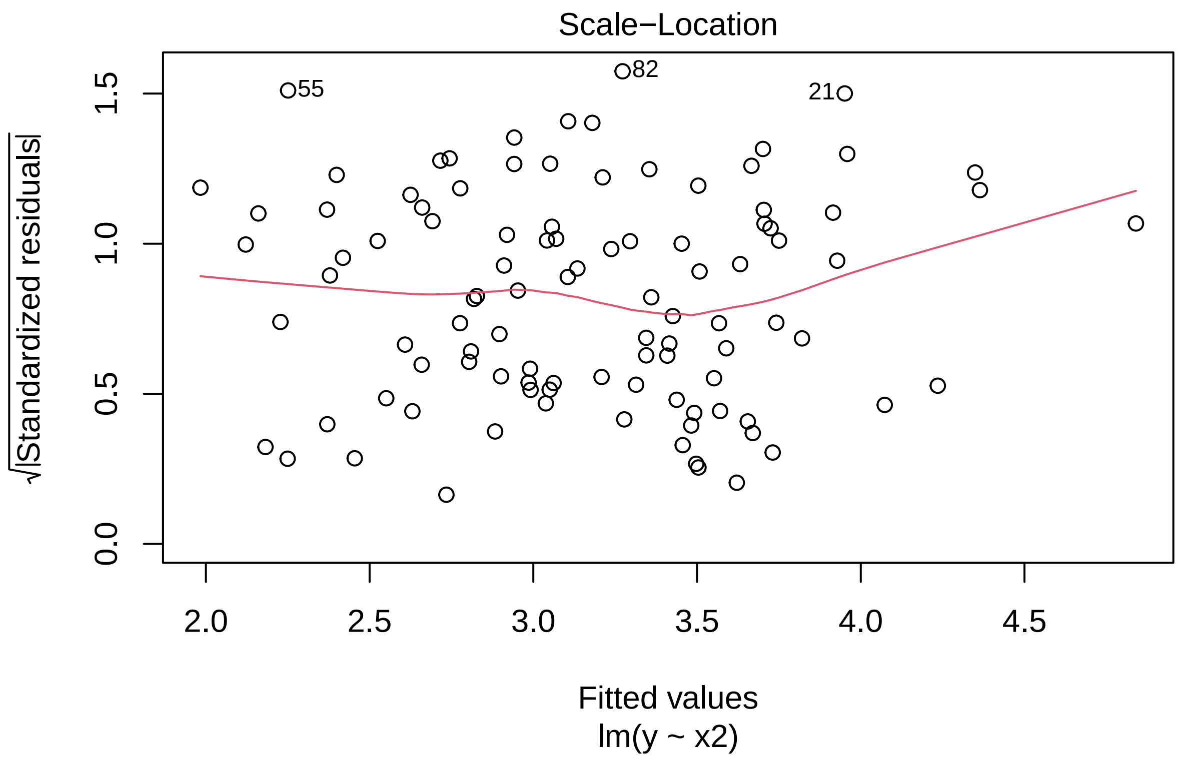
```
##
## Call:
## lm(formula = y ~ x2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.64729 -0.71021 -0.06899  0.72699  2.38074
##
## Coefficients:
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.3451     0.1912  12.264  < 2e-16 ***
## x2            3.1190     0.6040   5.164 1.25e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.074 on 99 degrees of freedom
## Multiple R-squared:  0.2122, Adjusted R-squared:  0.2042
## F-statistic: 26.66 on 1 and 99 DF,  p-value: 1.253e-06
```

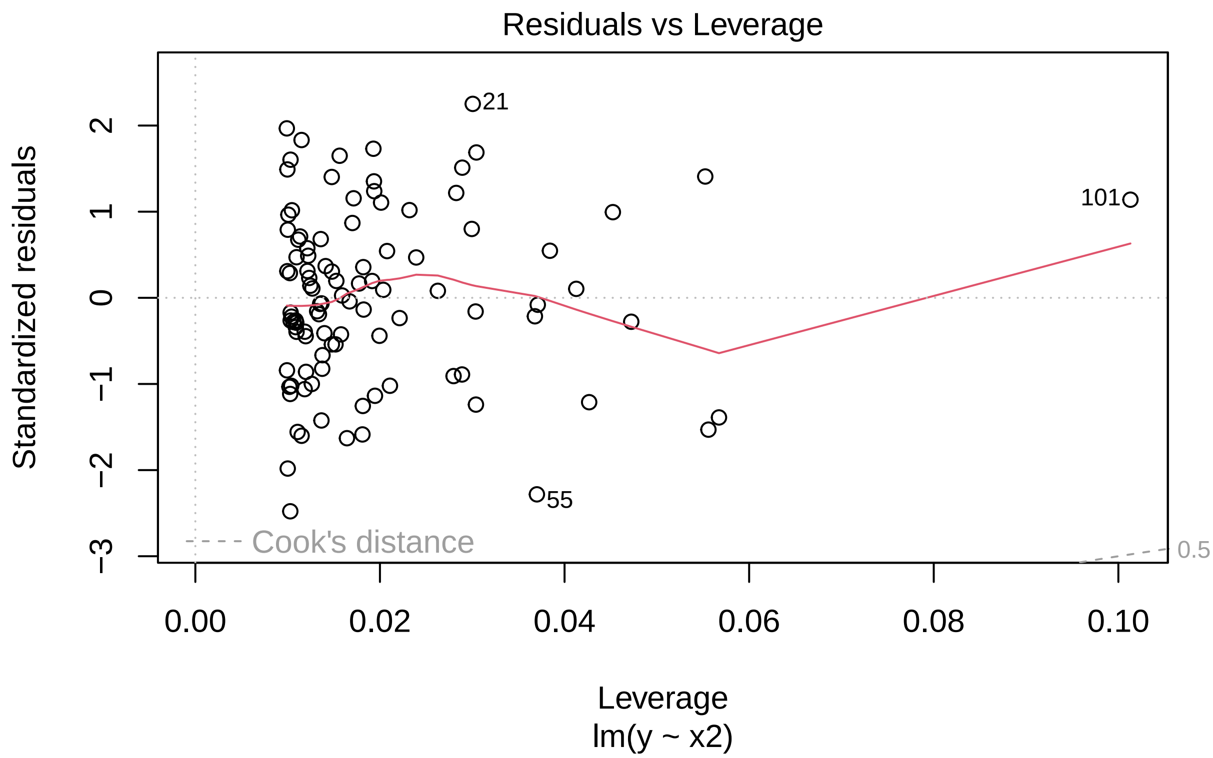
```
plot(x2_model)
```



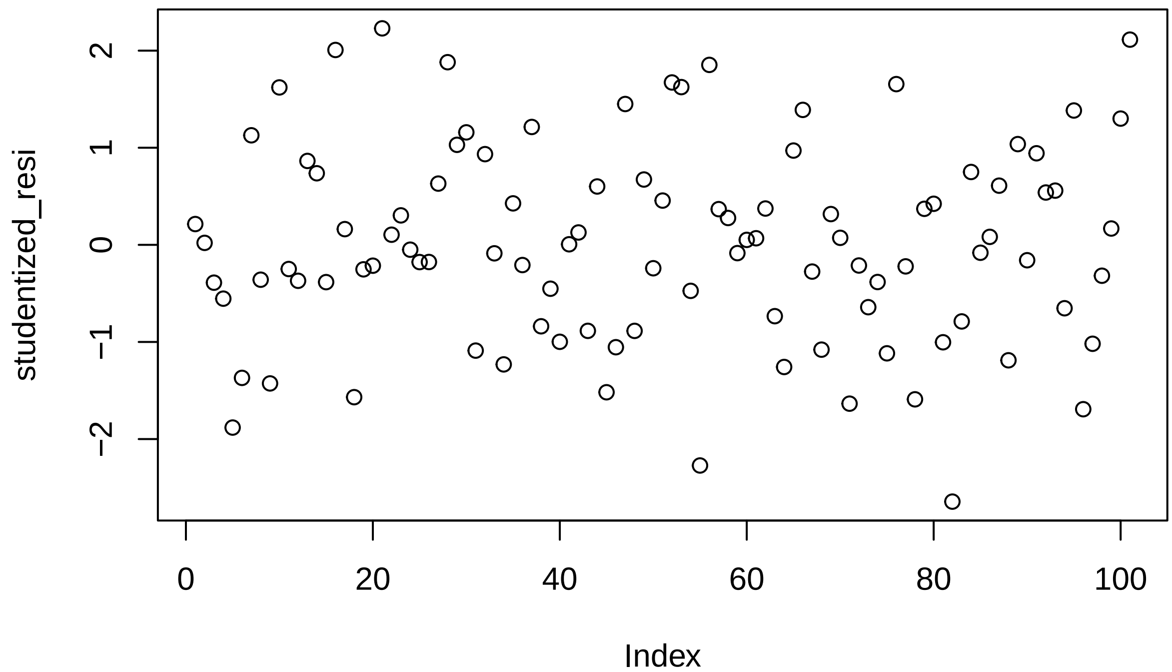

Normal Q−Q
Standardized residuals
2
1
0
−1
−2
21
55
82
−2
−1
0
1
2
Theoretical Quantiles
lm(y ~ x2)



Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
55
82
21
2.0
2.5
3.0
3.5
4.0
4.5
Fitted values
lm(y ~ x2)




```
# Check for outliers
studentized_resi = studres(lm.model)
plot(studentized_resi)
```




```r
# Check for leverage points
cd = cooks.distance(lm.model)
leverage_indices = which(cd > 4/nrow(lm.model))
leverage_indices
```

```
## integer(0)
```

Q2

```r
set.seed(1)
```

a)

Simulate the training data set.

```r
n = 25

x <- rnorm(n,0,1)
eps <- rnorm(n,0,1)
y = exp(x) + eps
```

b)

Fit four regression models

```
p = 4

y1 <- lm(y ~ x)
y2 <- lm(y ~ x + x^2)
y3 <- lm(y ~ x + x^2 + x^3)
y4 <- lm(y ~ x + x^2 + x^3 + x^4)
```

c)

Create a training dataset with 500 observations

```
n = 500
x.test <- rnorm(n,0,1)
eps.test <- rnorm(n,0,1)
y.test <- exp(x.test) + eps.test
```

d)

Compute the test error for each of the four models.

```
MSE = c()

fitted.values <- coef(y1)[1] + x.test * coef(y1)[2]
MSE[1] <- mean((y.test - fitted.values)^2)

fitted.values <- coef(y2)[1] + x.test * coef(y2)[2]
MSE[2] <- mean((y.test - fitted.values)^2)

fitted.values <- coef(y3)[1] + x.test * coef(y3)[2]
MSE[3] <- mean((y.test - fitted.values)^2)

fitted.values <- coef(y4)[1] + x.test * coef(y4)[2]
MSE[4] <- mean((y.test - fitted.values)^2)
```

e)

Which model is the 'best fit' model?

The model with the lowest MSE value and therefore best fit is the first model y ~ x. It's surprising to me that the linear model is the best fit as I would have expected the polynomial function x + x^2 + x^3 to be the best fit.

```
which.min(MSE)
```

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

Q3

Using the Hitters dataset from the ISLR package:

```
library(ISLR)
library(glmnet)
```

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

```
## Loaded glmnet 4.1-4
```

```r
library(dplyr)
```

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

## The following object is masked from 'package:MASS':
##
##     select

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

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

```r
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --

## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.8     v stringr 1.4.1
## v tidyr   1.2.0     v forcats 0.5.2
## v readr   2.1.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x tidyr::expand() masks Matrix::expand()
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x tidyr::pack()   masks Matrix::pack()
## x dplyr::select() masks MASS::select()
## x tidyr::unpack() masks Matrix::unpack()
```

```r
data("Hitters")
attach(Hitters)
```

a.

Split the data into training/testing sets

```r
df <- data.frame(filter(Hitters, !is.na(Hitters$Salary))) # Remove NAs
df <- df[,-c(14,15,20)]  # remove factor variables for regression

train_n <- ceiling(nrow(df) * 3/4)
test_n <- nrow(df) - train_n

data.train <- df[1:train_n,]
data.test <- df[train_n+1:nrow(df),]
```

b.

Fit the model using least squares regression and report the test error.

The mean test error of the linear model is -45.84.

```r
lm.model <- lm(data.train$Salary ~ ., data = data.train)
lm.fitted <- as.matrix(cbind(1, data.test[-17])) %*% coef(lm.model)
```

```r
test_error <- data.test$Salary - lm.fitted
mean(test_error, na.rm = TRUE)
```

```
## [1] -45.83773
```

```r
test_error
```

```
##                            [,1]
## -Ron Hassey           -34.97217
## -Rickey Henderson     851.34713
## -Reggie Jackson     -1277.27213
## -Ron Kittle           183.92963
## -Ray Knight           -99.66930
## -Rick Leach            14.03376
## -Rick Manning         155.65622
## -Rance Mulliniks       63.03653
## -Ron Oester           170.21510
## -Rey Quinones          17.21533
## -Rafael Ramirez       731.76379
## -Ronn Reynolds        131.61764
## -Ron Roenicke        -191.58751
## -Ryne Sandberg       -330.55230
## -Rafael Santana        35.64796
## -Rick Schu            -55.42647
## -Ruben Sierra         -63.52250
## -Roy Smalley          309.92911
## -Robby Thompson      -384.36189
## -Rob Wilfong           73.43356
## -Robin Yount         -367.17417
## -Steve Balboni       -520.29508
## -Scott Bradley       -237.48535
## -Sid Bream           -687.11739
## -Steve Buechele       -51.50404
## -Shawon Dunston       -95.08424
## -Scott Fletcher      -186.22611
## -Steve Garvey         -49.01759
## -Steve Jeltz         -232.87837
## -Steve Lombardozzi   -326.22622
## -Spike Owen            30.49317
## -Steve Sax          -1000.94415
## -Tony Bernazard      -353.82798
## -Tom Brookens         -67.88535
## -Tom Brunansky        473.28443
## -Tony Fernandez      -390.50857
## -Tim Flannery        -231.07392
## -Tom Foley            -62.72669
## -Tony Gwynn           -63.72088
## -Terry Harper         204.29391
## -Tommy Herr           111.48830
## -Tim Hulett           260.48883
## -Terry Kennedy        516.41635
## -Tito Landrum         149.88776
## -Tim Laudner          -63.34817
## -Tom Paciorek        -251.14170
## -Tony Pena            329.91081
```

```
## -Terry Pendleton     -77.19860
## -Tony Phillips      -178.65899
## -Terry Puhl          609.59721
## -Ted Simmons        -407.12015
## -Tim Teufel          -55.13832
## -Tim Wallach         367.94431
## -Vince Coleman        31.62912
## -Von Hayes            49.81745
## -Vance Law           134.89628
## -Wally Backman       -63.20466
## -Wade Boggs          265.00286
## -Will Clark         -504.20045
## -Wally Joyner       -780.19729
## -Willie McGee        240.88200
## -Willie Randolph     -32.44173
## -Wayne Tolleson       82.78513
## -Willie Upshaw      -179.09485
## -Willie Wilson       376.70912
## NA                          NA
## NA.1                        NA
## NA.2                        NA
## NA.3                        NA
## NA.4                        NA
## NA.5                        NA
## NA.6                        NA
## NA.7                        NA
## NA.8                        NA
## NA.9                        NA
## NA.10                       NA
## NA.11                       NA
## NA.12                       NA
## NA.13                       NA
## NA.14                       NA
## NA.15                       NA
## NA.16                       NA
## NA.17                       NA
## NA.18                       NA
## NA.19                       NA
## NA.20                       NA
## NA.21                       NA
## NA.22                       NA
## NA.23                       NA
## NA.24                       NA
## NA.25                       NA
## NA.26                       NA
## NA.27                       NA
## NA.28                       NA
## NA.29                       NA
## NA.30                       NA
## NA.31                       NA
## NA.32                       NA
## NA.33                       NA
## NA.34                       NA
## NA.35                       NA
```

```
## NA.36                           NA
## NA.37                           NA
## NA.38                           NA
## NA.39                           NA
## NA.40                           NA
## NA.41                           NA
## NA.42                           NA
## NA.43                           NA
## NA.44                           NA
## NA.45                           NA
## NA.46                           NA
## NA.47                           NA
## NA.48                           NA
## NA.49                           NA
## NA.50                           NA
## NA.51                           NA
## NA.52                           NA
## NA.53                           NA
## NA.54                           NA
## NA.55                           NA
## NA.56                           NA
## NA.57                           NA
## NA.58                           NA
## NA.59                           NA
## NA.60                           NA
## NA.61                           NA
## NA.62                           NA
## NA.63                           NA
## NA.64                           NA
## NA.65                           NA
## NA.66                           NA
## NA.67                           NA
## NA.68                           NA
## NA.69                           NA
## NA.70                           NA
## NA.71                           NA
## NA.72                           NA
## NA.73                           NA
## NA.74                           NA
## NA.75                           NA
## NA.76                           NA
## NA.77                           NA
## NA.78                           NA
## NA.79                           NA
## NA.80                           NA
## NA.81                           NA
## NA.82                           NA
## NA.83                           NA
## NA.84                           NA
## NA.85                           NA
## NA.86                           NA
## NA.87                           NA
## NA.88                           NA
## NA.89                           NA
```

```
## NA.90                            NA
## NA.91                            NA
## NA.92                            NA
## NA.93                            NA
## NA.94                            NA
## NA.95                            NA
## NA.96                            NA
## NA.97                            NA
## NA.98                            NA
## NA.99                            NA
## NA.100                           NA
## NA.101                           NA
## NA.102                           NA
## NA.103                           NA
## NA.104                           NA
## NA.105                           NA
## NA.106                           NA
## NA.107                           NA
## NA.108                           NA
## NA.109                           NA
## NA.110                           NA
## NA.111                           NA
## NA.112                           NA
## NA.113                           NA
## NA.114                           NA
## NA.115                           NA
## NA.116                           NA
## NA.117                           NA
## NA.118                           NA
## NA.119                           NA
## NA.120                           NA
## NA.121                           NA
## NA.122                           NA
## NA.123                           NA
## NA.124                           NA
## NA.125                           NA
## NA.126                           NA
## NA.127                           NA
## NA.128                           NA
## NA.129                           NA
## NA.130                           NA
## NA.131                           NA
## NA.132                           NA
## NA.133                           NA
## NA.134                           NA
## NA.135                           NA
## NA.136                           NA
## NA.137                           NA
## NA.138                           NA
## NA.139                           NA
## NA.140                           NA
## NA.141                           NA
## NA.142                           NA
## NA.143                           NA
```

```
## NA.144                          NA
## NA.145                          NA
## NA.146                          NA
## NA.147                          NA
## NA.148                          NA
## NA.149                          NA
## NA.150                          NA
## NA.151                          NA
## NA.152                          NA
## NA.153                          NA
## NA.154                          NA
## NA.155                          NA
## NA.156                          NA
## NA.157                          NA
## NA.158                          NA
## NA.159                          NA
## NA.160                          NA
## NA.161                          NA
## NA.162                          NA
## NA.163                          NA
## NA.164                          NA
## NA.165                          NA
## NA.166                          NA
## NA.167                          NA
## NA.168                          NA
## NA.169                          NA
## NA.170                          NA
## NA.171                          NA
## NA.172                          NA
## NA.173                          NA
## NA.174                          NA
## NA.175                          NA
## NA.176                          NA
## NA.177                          NA
## NA.178                          NA
## NA.179                          NA
## NA.180                          NA
## NA.181                          NA
## NA.182                          NA
## NA.183                          NA
## NA.184                          NA
## NA.185                          NA
## NA.186                          NA
## NA.187                          NA
## NA.188                          NA
## NA.189                          NA
## NA.190                          NA
## NA.191                          NA
## NA.192                          NA
## NA.193                          NA
## NA.194                          NA
## NA.195                          NA
## NA.196                          NA
## NA.197                          NA
```

c.

Fit a ridge regression model on the training set, with λ chosen by cross-validation. Report test error obtained.

The best lambda value ends up being .001 selected using cross validation. The mean test error of the ridge regression model is -46.2 and the vector of test errors are as follows:

```r
# Cross-validation for lambda

lam <- seq(0.001, 2, length.out = 100)
k = 5
ncv = ceiling(nrow(data.train)/k)
cv.ind = rep(1:k, ncv)
cv.ind.rand = sample(cv.ind, nrow(data.train), replace = F)

cv.error <- c(); MSE.cv <- c()
for(i in 1:100){
    for(j in 1:k){
      l.train <- data.train[cv.ind.rand != j, ]
      y.train <- data.train$Salary
      ridge.model <- glmnet(data.train[-17], y.train, lambda = lam[i], alpha = 0)

      l.test <- data.train[cv.ind.rand == j, ]
      l.test.values <- l.test$Salary
      test.response <- as.matrix(cbind(1, l.test[-17])) %*% coef(ridge.model, s = lam[i])
      MSE.cv[j] = mean((l.test.values - test.response)^2)
    }
  cv.error[i] = mean(MSE.cv)
}
lam_index = which.min(cv.error)

ridge.model <- glmnet(data.train[-17], data.train$Salary, lambda = lam[lam_index], alpha = 0)

ridge.fitted <- as.matrix(cbind(1, data.test[-17])) %*% coef(ridge.model, s = lam[lam_index])
test_error <- data.test$Salary - as.numeric(ridge.fitted)
mean(test_error, na.rm = TRUE)
```

```
## [1] -46.19122
```

```r
test_error
```

```
##   [1]   -36.79078   851.24480 -1278.96748   184.54370  -100.06800    13.41332
##   [7]   161.75241    60.24510   167.24070    16.03406   730.21029   130.90995
##  [13]  -192.41663  -329.50197    31.78446   -55.79786   -63.33625   311.96959
##  [19]  -387.19654    73.55683  -360.22149  -520.10163  -236.81900  -686.47443
##  [25]   -52.80200   -97.42345  -186.83451   -52.70070  -234.61467  -329.86623
##  [31]    30.62521  -999.40037  -352.31646   -63.52127   473.31713  -393.40336
##  [37]  -232.17255   -63.14417   -67.41617   203.55431   114.75830   257.59755
##  [43]   517.53392   147.76785   -61.84191  -254.86225   328.94896   -77.68033
##  [49]  -179.32041   608.56397  -408.76122   -56.24395   366.05661    30.61383
##  [55]    54.01019   134.83320   -65.19952   258.77512  -503.86038  -777.69172
##  [61]   241.47031   -28.11561    82.24028  -177.88565   378.76999          NA
##  [67]          NA          NA          NA          NA          NA          NA
##  [73]          NA          NA          NA          NA          NA          NA
##  [79]          NA          NA          NA          NA          NA          NA
##  [85]          NA          NA          NA          NA          NA          NA
##  [91]          NA          NA          NA          NA          NA          NA
```

```
##  [97]          NA          NA          NA          NA          NA          NA
## [103]          NA          NA          NA          NA          NA          NA
## [109]          NA          NA          NA          NA          NA          NA
## [115]          NA          NA          NA          NA          NA          NA
## [121]          NA          NA          NA          NA          NA          NA
## [127]          NA          NA          NA          NA          NA          NA
## [133]          NA          NA          NA          NA          NA          NA
## [139]          NA          NA          NA          NA          NA          NA
## [145]          NA          NA          NA          NA          NA          NA
## [151]          NA          NA          NA          NA          NA          NA
## [157]          NA          NA          NA          NA          NA          NA
## [163]          NA          NA          NA          NA          NA          NA
## [169]          NA          NA          NA          NA          NA          NA
## [175]          NA          NA          NA          NA          NA          NA
## [181]          NA          NA          NA          NA          NA          NA
## [187]          NA          NA          NA          NA          NA          NA
## [193]          NA          NA          NA          NA          NA          NA
## [199]          NA          NA          NA          NA          NA          NA
## [205]          NA          NA          NA          NA          NA          NA
## [211]          NA          NA          NA          NA          NA          NA
## [217]          NA          NA          NA          NA          NA          NA
## [223]          NA          NA          NA          NA          NA          NA
## [229]          NA          NA          NA          NA          NA          NA
## [235]          NA          NA          NA          NA          NA          NA
## [241]          NA          NA          NA          NA          NA          NA
## [247]          NA          NA          NA          NA          NA          NA
## [253]          NA          NA          NA          NA          NA          NA
## [259]          NA          NA          NA          NA          NA
```

d.

Fit the lasso model and report the test erro along with the number of non-zero coefficients.

The mean error for the lasso regression model is -46.19. However, I believe there's a mistake in the cross-validation code for lambda as the lasso error should not be the same as the ridge error. I cannot figure out where the error is. All variables are non-zero with the chosen lambda, which is likely not to be the case.

```
# Cross-validation for lambda

lam <- seq(0.001, 2, length.out = 100)
k = 5
ncv = ceiling(nrow(data.train)/k)
cv.ind = rep(1:k, ncv)
cv.ind.rand = sample(cv.ind, nrow(data.train), replace = F)

cv.error <- c(); MSE.cv <- c()
for(i in 1:100){
    for(j in 1:k){
      l.train <- data.train[cv.ind.rand != j, ]
      y.train <- data.train$Salary
      lasso.model <- glmnet(data.train[-17], y.train, lambda = lam[i], alpha = 1)

      l.test <- data.train[cv.ind.rand == j, ]
      l.test.values <- l.test$Salary
      test.response <- as.matrix(cbind(1, l.test[-17])) %*% coef(lasso.model, s = lam[i])
```

```
      MSE.cv[j] = mean((l.test.values - test.response)^2)
    }
  cv.error[i] = mean(MSE.cv)
}
lam_index = which.min(cv.error)  # which value of lambda

lasso.model <- glmnet(data.train[-17], data.train$Salary, lambda = lam[lam_index], alpha = 1)

lasso.fitted <- as.matrix(cbind(1, data.test[-17])) %*% coef(lasso.model, s = lam[lam_index])
test_error <- data.test$Salary - as.numeric(lasso.fitted)
mean(test_error, na.rm = TRUE)
```

```
## [1] -46.19173
```

```
test_error
```

```
##   [1]   -36.79955   851.23784 -1278.95716   184.54736  -100.07141    13.41706
##   [7]   161.75804    60.24214   167.22721    16.02155   730.17474   130.90619
##  [13]  -192.40581  -329.46812    31.77112   -55.80791   -63.34646   311.95799
##  [19]  -387.20906    73.57332  -360.19466  -520.09559  -236.81069  -686.45748
##  [25]   -52.81484   -97.43921  -186.83005   -52.71564  -234.60490  -329.87339
##  [31]    30.62946  -999.39629  -352.30814   -63.50388   473.30693  -393.41548
##  [37]  -232.16825   -63.14119   -67.45451   203.55331   114.80419   257.57301
##  [43]   517.53739   147.76996   -61.83148  -254.86133   328.93473   -77.67423
##  [49]  -179.31176   608.55263  -408.78901   -56.24795   366.04420    30.59123
##  [55]    54.03272   134.84826   -65.20677   258.74488  -503.87019  -777.67355
##  [61]   241.46705   -28.08873    82.24204  -177.86735   378.78292          NA
##  [67]          NA          NA          NA          NA          NA          NA
##  [73]          NA          NA          NA          NA          NA          NA
##  [79]          NA          NA          NA          NA          NA          NA
##  [85]          NA          NA          NA          NA          NA          NA
##  [91]          NA          NA          NA          NA          NA          NA
##  [97]          NA          NA          NA          NA          NA          NA
## [103]          NA          NA          NA          NA          NA          NA
## [109]          NA          NA          NA          NA          NA          NA
## [115]          NA          NA          NA          NA          NA          NA
## [121]          NA          NA          NA          NA          NA          NA
## [127]          NA          NA          NA          NA          NA          NA
## [133]          NA          NA          NA          NA          NA          NA
## [139]          NA          NA          NA          NA          NA          NA
## [145]          NA          NA          NA          NA          NA          NA
## [151]          NA          NA          NA          NA          NA          NA
## [157]          NA          NA          NA          NA          NA          NA
## [163]          NA          NA          NA          NA          NA          NA
## [169]          NA          NA          NA          NA          NA          NA
## [175]          NA          NA          NA          NA          NA          NA
## [181]          NA          NA          NA          NA          NA          NA
## [187]          NA          NA          NA          NA          NA          NA
## [193]          NA          NA          NA          NA          NA          NA
## [199]          NA          NA          NA          NA          NA          NA
## [205]          NA          NA          NA          NA          NA          NA
## [211]          NA          NA          NA          NA          NA          NA
## [217]          NA          NA          NA          NA          NA          NA
## [223]          NA          NA          NA          NA          NA          NA
## [229]          NA          NA          NA          NA          NA          NA
```

```
## [235]          NA          NA          NA          NA          NA          NA
## [241]          NA          NA          NA          NA          NA          NA
## [247]          NA          NA          NA          NA          NA          NA
## [253]          NA          NA          NA          NA          NA          NA
## [259]          NA          NA          NA          NA          NA
```

```
coef(lasso.model, s = lam[lam_index])
```

```
## 17 x 1 sparse Matrix of class "dgCMatrix"
##                      s1
## (Intercept) 120.0626842
## AtBat        -3.0760596
## Hits         10.9313223
## HmRun        -3.8202376
## Runs         -2.4366967
## RBI           0.7222072
## Walks         6.6604145
## Years         1.0011724
## CAtBat       -0.1547079
## CHits        -0.1705178
## CHmRun        0.9420960
## CRuns         1.6651054
## CRBI          0.8621874
## CWalks       -0.8744750
## PutOuts       0.4108652
## Assists       0.7001926
## Errors       -4.9637317
```

e

Comment on the obtained results.

The test error, with a mean of approximately -46, is fairly low given the scale of the numbers we're working with. The can pretty accurately predict salary using the various observations. The difference in test error between the linear and lasso/ridge regression was marginal.