PCA

student

2023-10-04

Question 9.1

Using the same crime data set uscrime.txt as in Question 8.2, apply Principal Component Analysis and then create a regression model using the first few principal components. Specify your new model in terms of the original variables (not the principal components), and compare its quality to that of your solution to Question 8.2. You can use the R function prcomp for PCA. (Note that to first scale the data, you can include scale. = TRUE to scale as part of the PCA function. Don't forget that, to make a prediction for the new city, you'll need to unscale the coefficients (i.e., do the scaling calculation in reverse)!)

Prediction *I am staying with my original prediction, using scaling from last week due to possible over-fitting this week, and a larger p-value from my log model(though not by much). This most recent prediction, using top 3 predictors, suggests a crime rate of 1,192.321 crimes per 100,000 people, resulting from model with a p-value of 0.004269. Comparatively, two other models from previous homework analyses (using scaling and a logarithmic approach, using top 4 predictors) yielded crime rate predictions of 1,112.678 and 1,037.789 per 100,000 people, respectively. The associated p-values for these models were 0.00317 and 0.00426. Given the proximity of the p-values across all three models, it suggests that each model fits the data reasonably well. When choosing the best model, one might prioritize the model with the lowest p-value (indicating the best fit), along with additional resulting data below.*

Some Result interpretations from the current homework week 6: *After applying PCA, then creating a regression model on top 3 PCs, I specified the new model in terms of the original variables:* * **M** *the result indicates that for every unit increase in the percentage of males aged 14–24 in the total state population, the crime rate tends to decrease by about 19.86 offenses per 100,000 population, holding other variables constant.* **So** *southern state is associated with a decrease in the crime rate by about 10.09 offenses per 100,000 population compared to non-southern states, holding other variables constant* **Ed** *every additional mean year of schooling of the population aged 25 years or over, the crime rate tends to increase by about 8.81 offenses per 100,000 population, holding other variables constant.*

Some things that stand out: *Given the large coefficients in this weeks homework, the translation of the regression coefficients from the principal component space back to the original predictor space, I wonder if this may be indicative of over-fitting or magnification due to the transformation process.*

Variable	Value	Description
M	14.0	percentage of males aged 14–24 in total state population
So	0	indicator variable for a southern state
Ed	10.0	mean years of schooling of the population aged 25 years or over
Po1	12.0	per capita expenditure on police protection in 1960
Po2	15.5	per capita expenditure on police protection in 1959
LF	0.640	labour force participation rate of civilian urban males in the age-group 14-24
M.F	94.0	number of males per 100 females
Pop	150	state population in 1960 in hundred thousands
NW	1.1	percentage of nonwhites in the population

Variable	Value	Description
U1	0.120	unemployment rate of urban males 14–24
U2	3.6	unemployment rate of urban males 35–39
Wealth	3200	wealth: median value of transferable assets or family income
Ineq	20.1	income inequality: percentage of families earning below half the median income
Prob	0.04	probability of imprisonment: ratio of number of commitments to number of offenses
Time	39.	average time in months served by offenders in state prisons before their first release
Crime	-	crime rate: number of offenses per 100,000 population in 1960

Begin Code

Load Libraries

```
library(readr)
```

Load & View Crime Data

```
##      M So   Ed  Po1  Po2    LF   M.F Pop   NW    U1  U2 Wealth Ineq     Prob
## 1 15.1  1  9.1  5.8  5.6 0.510  95.0  33 30.1 0.108 4.1   3940 26.1 0.084602
## 2 14.3  0 11.3 10.3  9.5 0.583 101.2  13 10.2 0.096 3.6   5570 19.4 0.029599
## 3 14.2  1  8.9  4.5  4.4 0.533  96.9  18 21.9 0.094 3.3   3180 25.0 0.083401
## 4 13.6  0 12.1 14.9 14.1 0.577  99.4 157  8.0 0.102 3.9   6730 16.7 0.015801
## 5 14.1  0 12.1 10.9 10.1 0.591  98.5  18  3.0 0.091 2.0   5780 17.4 0.041399
## 6 12.1  0 11.0 11.8 11.5 0.547  96.4  25  4.4 0.084 2.9   6890 12.6 0.034201
##      Time Crime
## 1 26.2011   791
## 2 25.2999  1635
## 3 24.3006   578
## 4 29.9012  1969
## 5 21.2998  1234
## 6 20.9995   682
```

Calculate PCA using sclae predictors and the loadings

```
# remove the response variable 'Crime'
predictor_data <- uscrime_data[,-which(names(uscrime_data) == "Crime")]

# princomp: uses eigenvalue decomposition of the covariance matrix.
#pca_result <- princomp(data[, -16], cor=TRUE)
#summary(pca_result)

# perform PCA, scale the predictors
pca_result <- prcomp(predictor_data, scale. = TRUE)
summary(pca_result)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6     PC7
## Standard deviation     2.4534 1.6739 1.4160 1.07806 0.97893 0.74377 0.56729
## Proportion of Variance 0.4013 0.1868 0.1337 0.07748 0.06389 0.03688 0.02145
## Cumulative Proportion  0.4013 0.5880 0.7217 0.79920 0.86308 0.89996 0.92142
```

```
##                            PC8     PC9    PC10    PC11    PC12    PC13   PC14
## Standard deviation     0.55444 0.48493 0.44708 0.41915 0.35804 0.26333 0.2418
## Proportion of Variance 0.02049 0.01568 0.01333 0.01171 0.00855 0.00462 0.0039
## Cumulative Proportion  0.94191 0.95759 0.97091 0.98263 0.99117 0.99579 0.9997
##                           PC15
## Standard deviation     0.06793
## Proportion of Variance 0.00031
## Cumulative Proportion  1.00000
```

```
pca_result$rotation
```

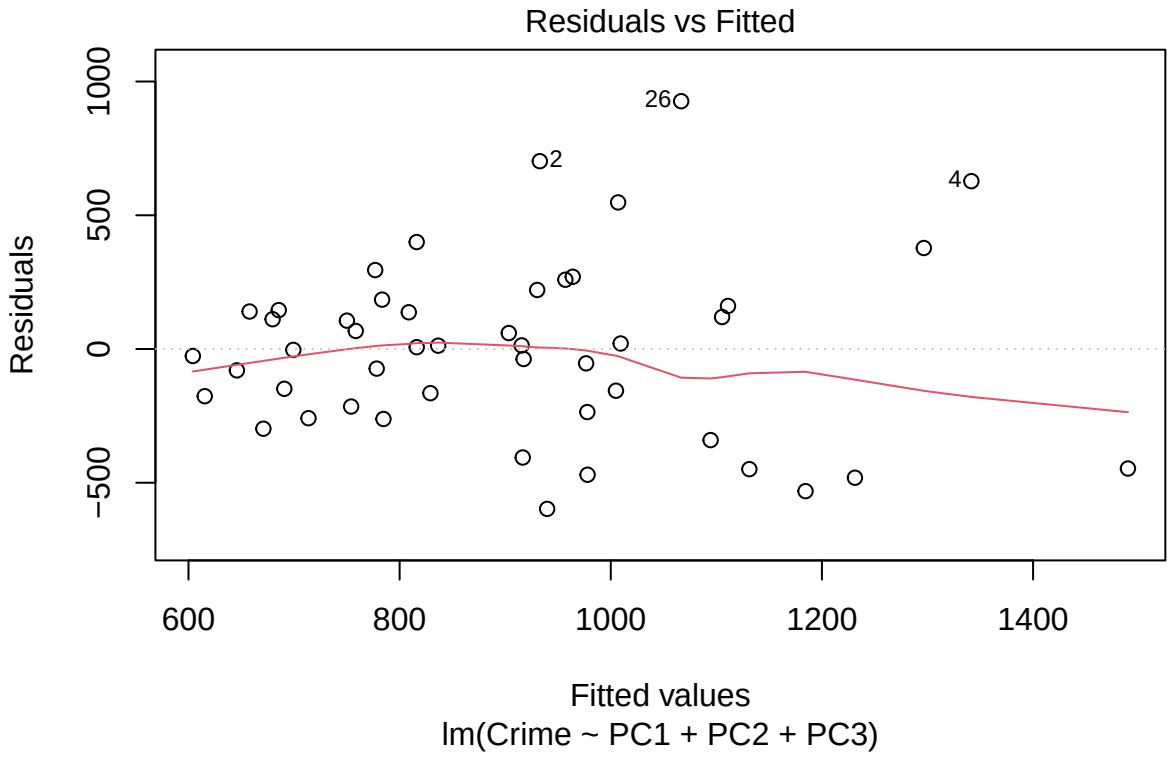
```
##                PC1         PC2           PC3         PC4         PC5
## M      -0.30371194  0.06280357  0.1724199946 -0.02035537 -0.35832737
## So     -0.33088129 -0.15837219  0.0155433104  0.29247181 -0.12061130
## Ed      0.33962148  0.21461152  0.0677396249  0.07974375 -0.02442839
## Po1     0.30863412 -0.26981761  0.0506458161  0.33325059 -0.23527680
## Po2     0.31099285 -0.26396300  0.0530651173  0.35192809 -0.20473383
## LF      0.17617757  0.31943042  0.2715301768 -0.14326529 -0.39407588
## M.F     0.11638221  0.39434428 -0.2031621598  0.01048029 -0.57877443
## Pop     0.11307836 -0.46723456  0.0770210971 -0.03210513 -0.08317034
## NW     -0.29358647 -0.22801119  0.0788156621  0.23925971 -0.36079387
## U1      0.04050137  0.00807439 -0.6590290980 -0.18279096 -0.13136873
## U2      0.01812228 -0.27971336 -0.5785006293 -0.06889312 -0.13499487
## Wealth  0.37970331 -0.07718862  0.0100647664  0.11781752  0.01167683
## Ineq   -0.36579778 -0.02752240 -0.0002944563 -0.08066612 -0.21672823
## Prob   -0.25888661  0.15831708 -0.1176726436  0.49303389  0.16562829
## Time   -0.02062867 -0.38014836  0.2235664632 -0.54059002 -0.14764767
##                 PC6         PC7         PC8         PC9         PC10         PC11
## M      -0.449132706 -0.15707378 -0.55367691  0.15474793 -0.01443093  0.39446657
## So     -0.100500743  0.19649727  0.22734157 -0.65599872  0.06141452  0.23397868
## Ed     -0.008571367 -0.23943629 -0.14644678 -0.44326978  0.51887452 -0.11821954
## Po1    -0.095776709  0.08011735  0.04613156  0.19425472 -0.14320978 -0.13042001
## Po2    -0.119524780  0.09518288  0.03168720  0.19512072 -0.05929780 -0.13885912
## LF      0.504234275 -0.15931612  0.25513777  0.14393498  0.03077073  0.38532827
## M.F    -0.074501901  0.15548197 -0.05507254 -0.24378252 -0.35323357 -0.28029732
## Pop     0.547098563  0.09046187 -0.59078221 -0.20244830 -0.03970718  0.05849643
## NW      0.051219538 -0.31154195  0.20432828  0.18984178  0.49201966 -0.20695666
## U1      0.017385981 -0.17354115 -0.20206312  0.02069349  0.22765278 -0.17857891
## U2      0.048155286 -0.07526787  0.24369650  0.05576010 -0.04750100  0.47021842
## Wealth -0.154683104 -0.14859424  0.08630649 -0.23196695 -0.11219383  0.31955631
## Ineq    0.272027031  0.37483032  0.07184018 -0.02494384 -0.01390576 -0.18278697
## Prob    0.283535996 -0.56159383 -0.08598908 -0.05306898 -0.42530006 -0.08978385
## Time   -0.148203050 -0.44199877  0.19507812 -0.23551363 -0.29264326 -0.26363121
##                PC12        PC13        PC14          PC15
## M       0.16580189  0.05142365  0.04901705 -0.0051398012
## So     -0.05753357  0.29368483 -0.29364512 -0.0084369230
## Ed      0.47786536 -0.19441949  0.03964277  0.0280052040
## Po1     0.22611207  0.18592255 -0.09490151  0.6894155129
## Po2     0.19088461  0.13454940 -0.08259642 -0.7200270100
## LF      0.02705134  0.27742957 -0.15385625 -0.0336823193
## M.F    -0.23925913 -0.31624667 -0.04125321 -0.0097922075
## Pop    -0.18350385 -0.12651689 -0.05326383 -0.0001496323
## NW     -0.36671707 -0.22901695  0.13227774  0.0370783671
## U1     -0.09314897  0.59039450 -0.02335942 -0.0111359325
```

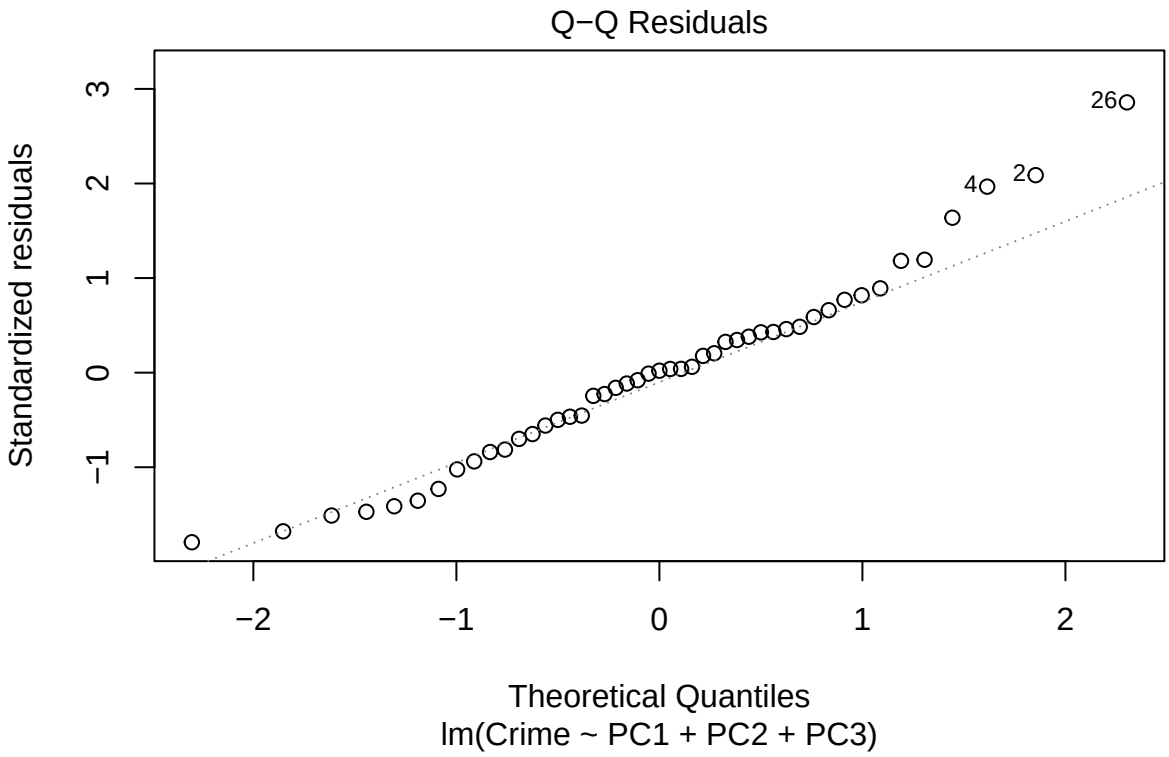
```
## U2      0.28440496 -0.43292853 -0.03985736 -0.0073618948
## Wealth -0.32172821  0.14077972  0.70031840  0.0025685109
## Ineq    0.43762828  0.12181090  0.59279037 -0.0177570357
## Prob    0.15567100  0.03547596  0.04761011 -0.0293376260
## Time    0.13536989  0.05738113 -0.04488401 -0.0376754405
```

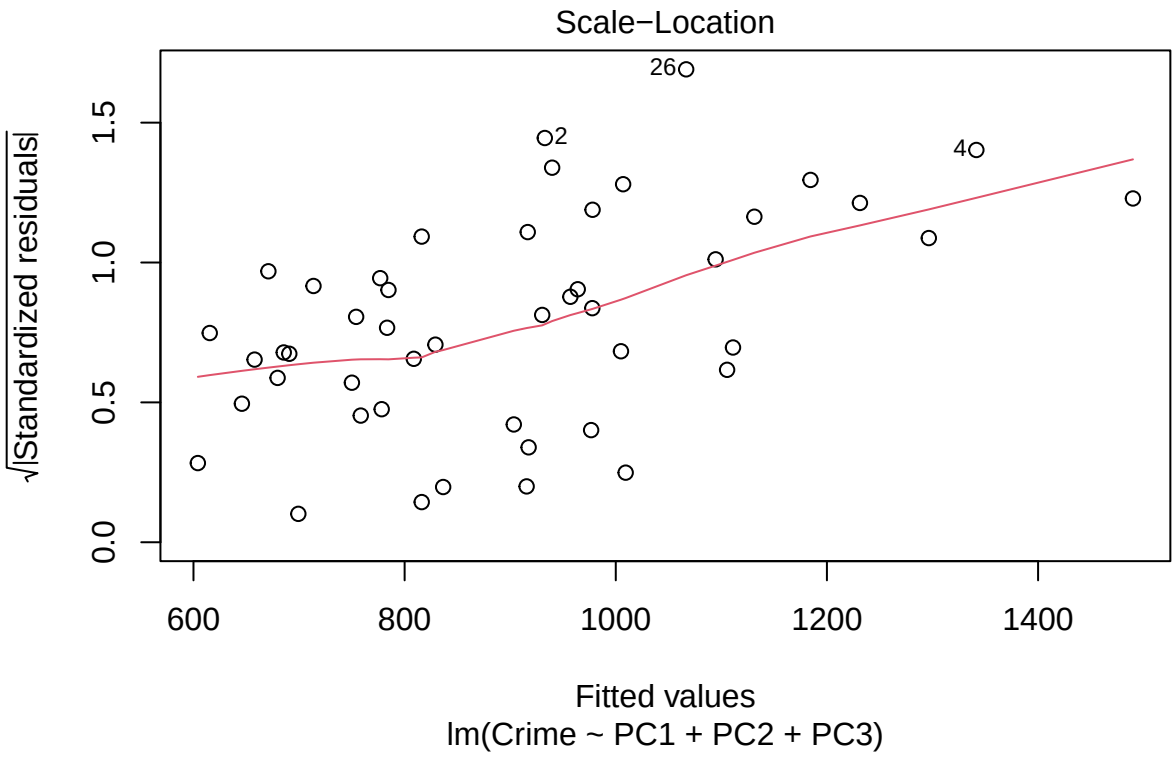
Create a regression model using the principal components, top 3 PCA results that together capture 70%

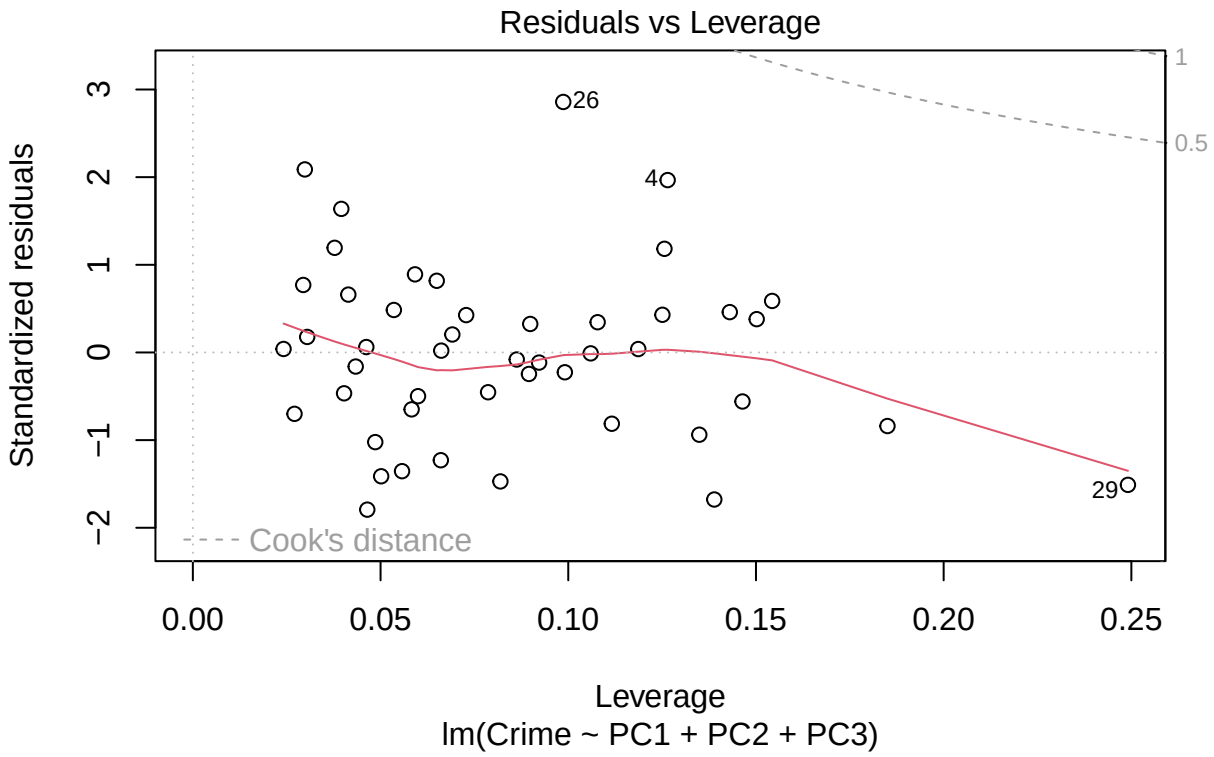
PC1: 40.13% PC1 + PC2: 58.80% (40.13% + 18.68%) PC1 + PC2 + PC3: 72.17% (58.80% + 13.37%)...so on.

```r
# get scores for the first three PCs
PC_scores <- pca_result$x[,1:3]

# build regression model using the selected PCs
model_pca <- lm(Crime ~ PC1 + PC2 + PC3, data = data.frame(Crime = uscrime_data$Crime, PC_scores))

plot(model_pca)
```




Q−Q Residuals
Standardized residuals
−1
0
1
2
3
26
4
2
−2
−1
0
1
2
Theoretical Quantiles
lm(Crime ~ PC1 + PC2 + PC3)


Scale−Location
26
2
4
√|Standardized residuals|
0.0
0.5
1.0
1.5
600
800
1000
1200
1400
Fitted values
lm(Crime ~ PC1 + PC2 + PC3)



Get the implied regression coefficients for each of the original predictors in terms of their relationship with the Crime variable after taking into account the PCA.

```
# using only the first 3 PCs
loadings <- pca_result$rotation[, 1:3]
# excluding intercept
coefficients <- coef(model_pca)[-1]

# compute a_j for each predictor
a_j <- loadings %*% coefficients

print(a_j)
```

```
##              [,1]
## M      -19.864364
## So     -10.087915
## Ed       8.814723
## Po1     40.313495
## Po2     40.117965
## LF      -4.056143
## M.F    -25.165386
## Pop     42.060241
## NW      -1.181092
## U1     -14.528175
## U2       6.210256
## Wealth  30.425896
## Ineq   -21.934406
```

```
## Prob   -30.943539
## Time    30.929216
```

Get Predicted Value for City

```
# define the new observation using the original variables.
new_observation <- data.frame(M = 14.0,
                              So = 0,
                              Ed = 10.0,
                              Po1 = 12.0,
                              Po2 = 15.5,
                              LF = 0.640,
                              M.F = 94.0,
                              Pop = 150,
                              NW = 1.1,
                              U1 = 0.120,
                              U2 = 3.6,
                              Wealth = 3200,
                              Ineq = 20.1,
                              Prob = 0.04,
                              Time = 39.)

# transform the new observation into the PCA space
new_observation_PC <- as.matrix(scale(new_observation,
                                center = pca_result$center,
                                scale = pca_result$scale)) %*% pca_result$rotation[,1:3]

# predict using the model
predicted_value <- predict(model_pca, newdata = data.frame(PC1 = new_observation_PC[,1],
                                                           PC2 = new_observation_PC[,2],
                                                           PC3 = new_observation_PC[,3]))

print(predicted_value)
```

```
##      PC1
## 1192.321
```

Original: p-value

```
# F-statistic
f_stat <- summary(model_pca)$fstatistic[1]

# degrees of freedom
df1 <- summary(model_pca)$fstatistic[2]
df2 <- summary(model_pca)$fstatistic[3]

# calculate p-value
p_value <- 1 - pf(f_stat, df1, df2)

# print the results
print(paste("p-value:", p_value))
```

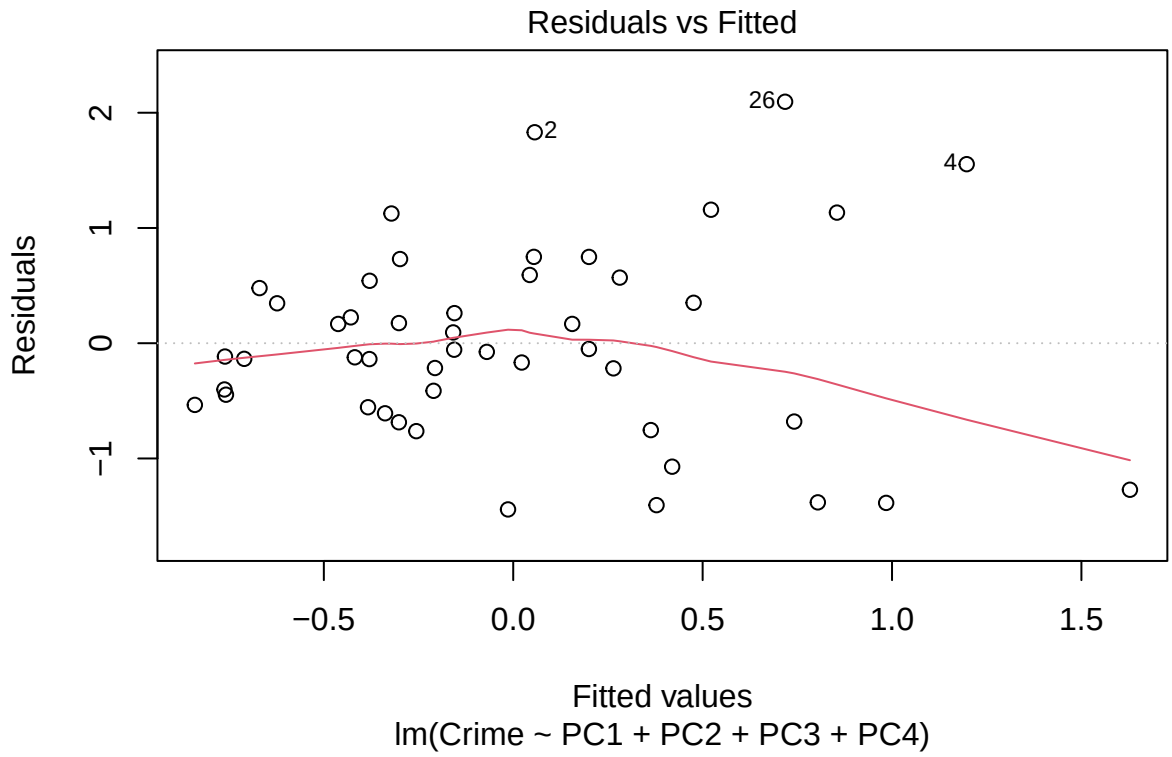
```
## [1] "p-value: 0.00321719282092137"
```

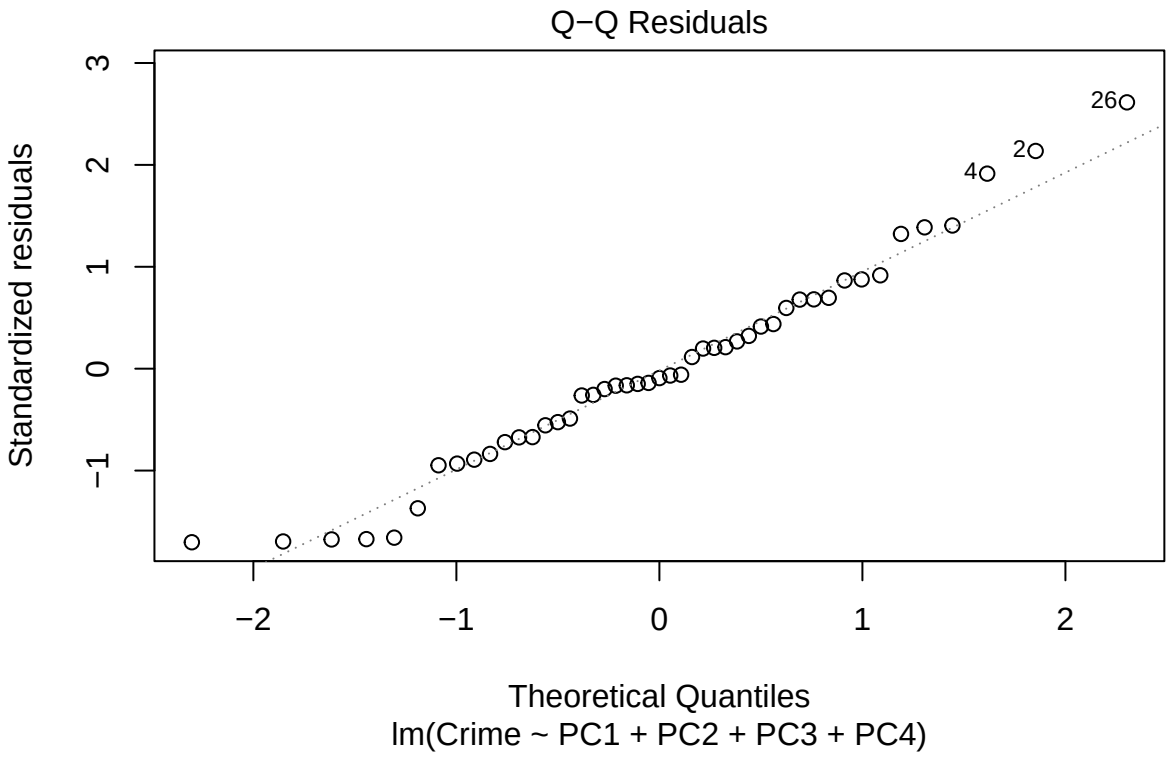
Previous Homework: Response Transformation: Scaling

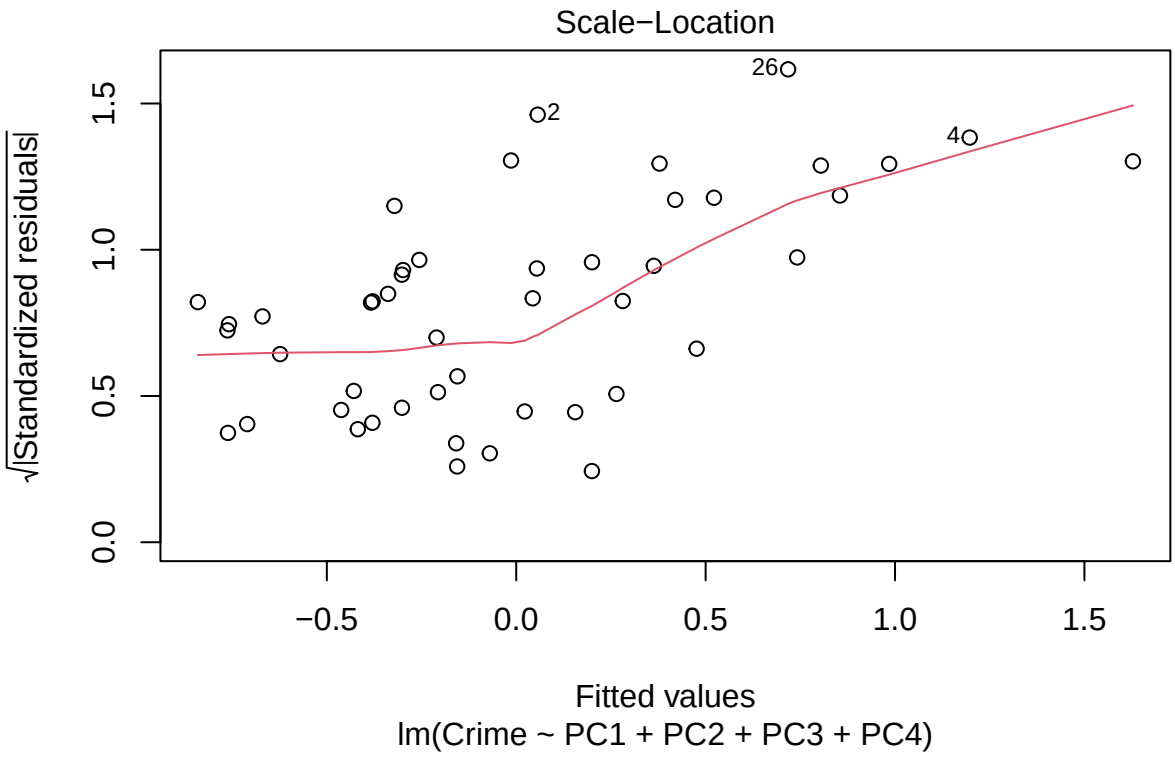
```
uscrime_data <- read.csv("C:\\Users\\Public\\Documents\\gatech\\crime.csv")

# scaling the response
uscrime_data$Crime_scaled <- scale(uscrime_data$Crime)
```

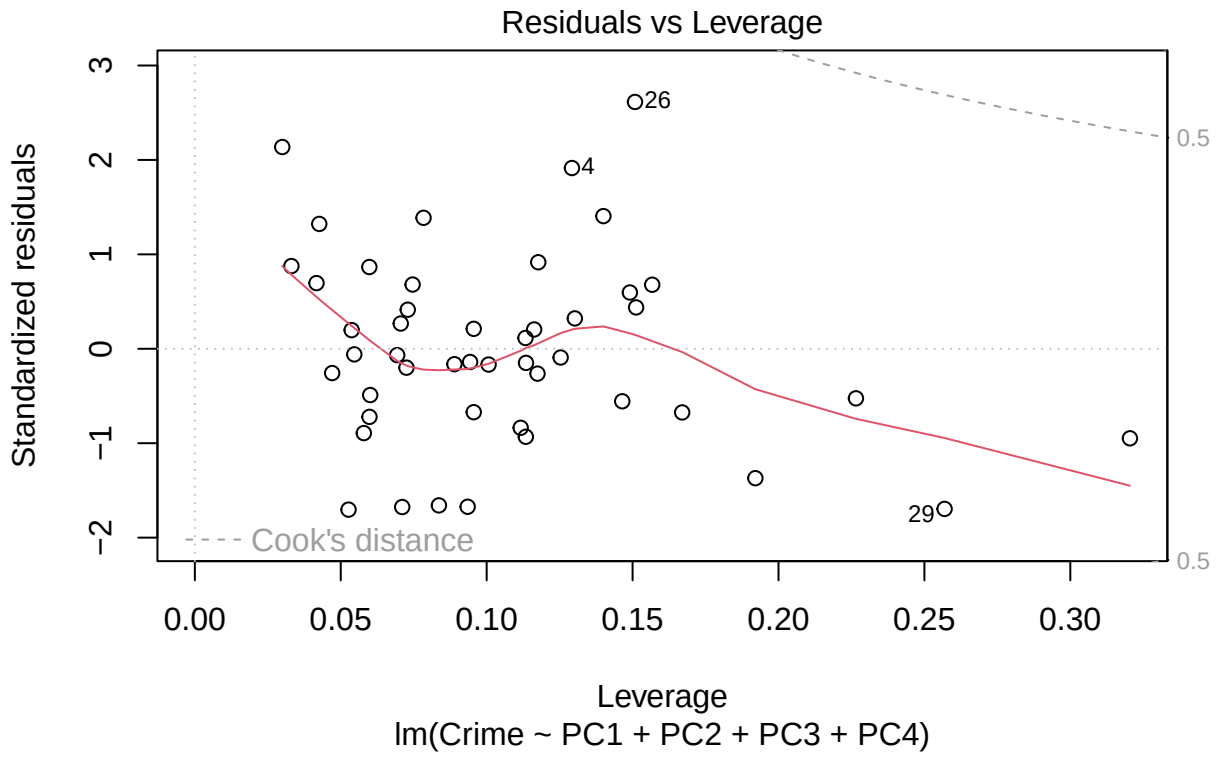
Regression with Scaled Response

```
# get scores for the first two PCs
PC_scores <- pca_result$x[,1:4]

# build regression model using the selected PCs
model_pca <- lm(Crime ~ PC1 + PC2 + PC3 + PC4,
                data = data.frame(Crime = uscrime_data$Crime_scaled,
                                  PC_scores))

# extract coefficients in terms of the original predictors
original_coeff <- as.matrix(pca_result$rotation[, 1:4]) %*% coef(model_pca)[-1]

plot(model_pca)
```




Q−Q Residuals
Standardized residuals
3
2
1
0
−1
26
2
4
−2
−1
0
1
2
Theoretical Quantiles
lm(Crime ~ PC1 + PC2 + PC3 + PC4)


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
−0.5
0.0
0.5
1.0
1.5
26
2
4
Fitted values
lm(Crime ~ PC1 + PC2 + PC3 + PC4)




```
print(original_coeff)
```

```
##                [,1]
## M       -0.055015551
## So       0.026432464
## Ed       0.037109603
## Po1      0.164070698
## Po2      0.166918823
## LF      -0.036211737
## M.F     -0.063184924
## Pop      0.102984769
## NW       0.039907016
## U1      -0.070384971
## U2       0.003686762
## Wealth   0.099823110
## Ineq    -0.071197011
## Prob     0.008521265
## Time    -0.017097346
```

New Data: scaled response

```
# transform the new observation into the PCA space
new_observation_PC <- as.matrix(scale(new_observation,
                                      center = pca_result$center,
                                      scale = pca_result$scale)) %*% pca_result$rotation[,1:4]
```

```r
# predict using the model
predicted_value <- predict(model_pca, newdata = data.frame(PC1 = new_observation_PC[,1],
                                                           PC2 = new_observation_PC[,2],
                                                           PC3 = new_observation_PC[,3],
                                                           PC4 = new_observation_PC[,4]))

original_scale_prediction <- (predicted_value * sd(uscrime_data$Crime)) + mean(uscrime_data$Crime)

print(original_scale_prediction)
```

```
##      PC1
## 1112.678
```

Previous Homework

Scaling: p-value

```r
# extract F-statistic
f_stat <- summary(model_pca)$fstatistic[1]

# degrees of freedom
df1 <- summary(model_pca)$fstatistic[2]
df2 <- summary(model_pca)$fstatistic[3]

# calculate p-value
p_value <- 1 - pf(f_stat, df1, df2)

# results
print(paste("p-value:", p_value))
```

```
## [1] "p-value: 0.00317775214807403"
```

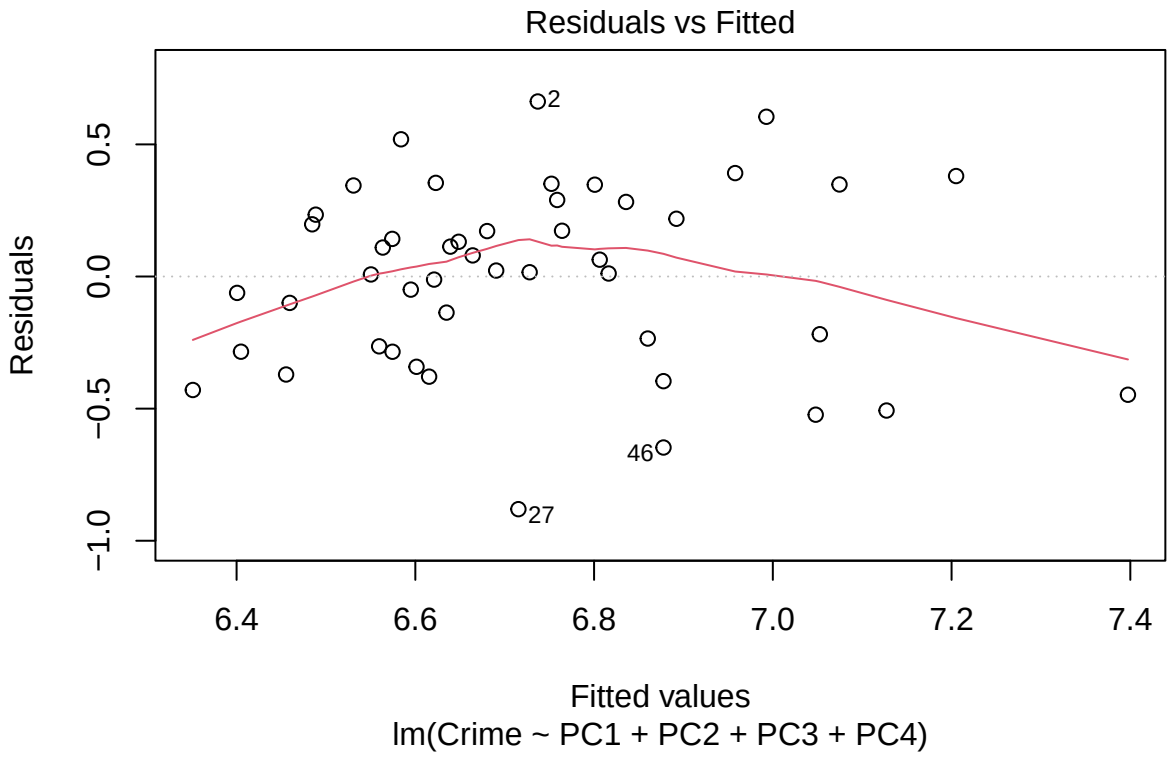
Log

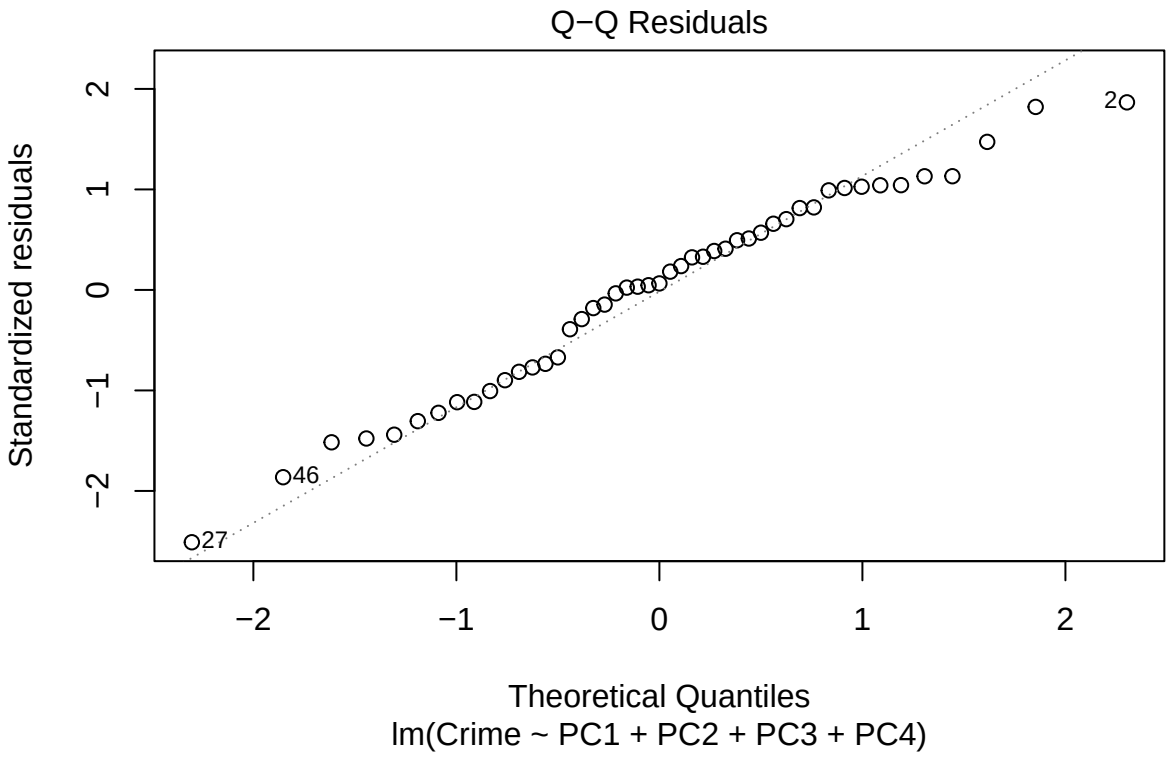
```r
# crime data
uscrime_data <- read.csv("C:\\Users\\Public\\Documents\\gatech\\crime.csv")
# log-transforming the response
uscrime_data$Crime_log <- log(uscrime_data$Crime)
```

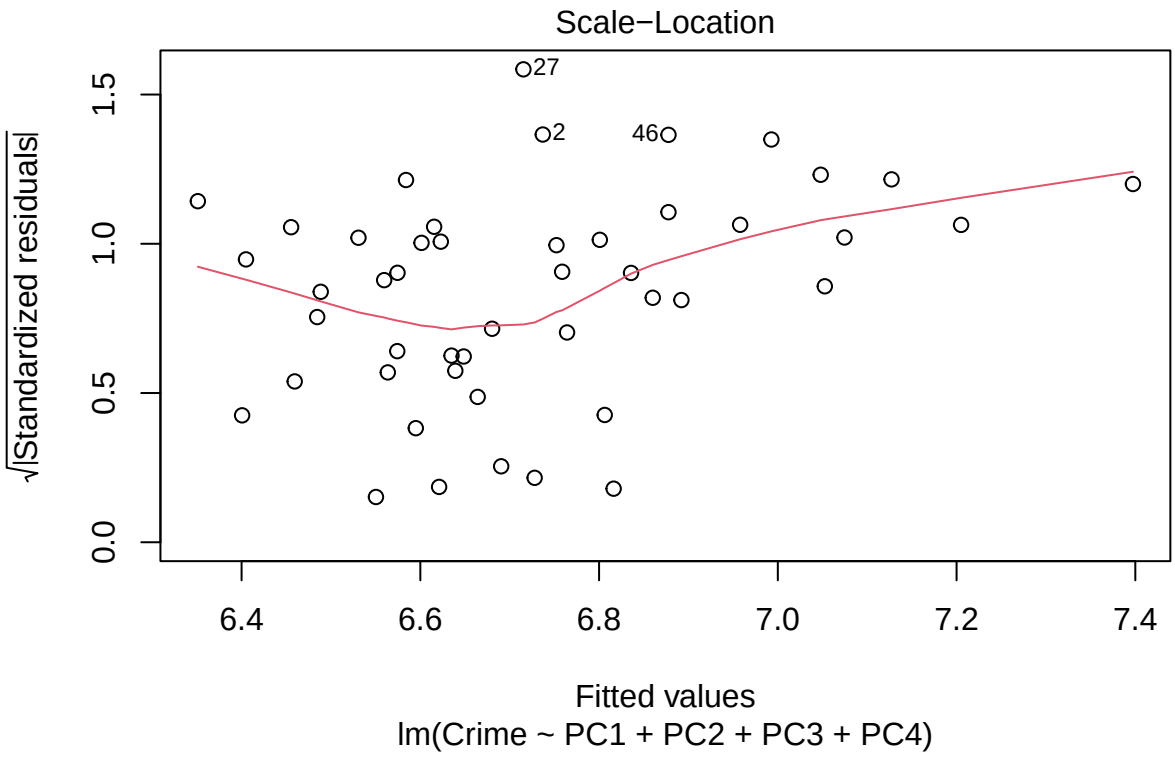
Regression with Log-Transformed Response

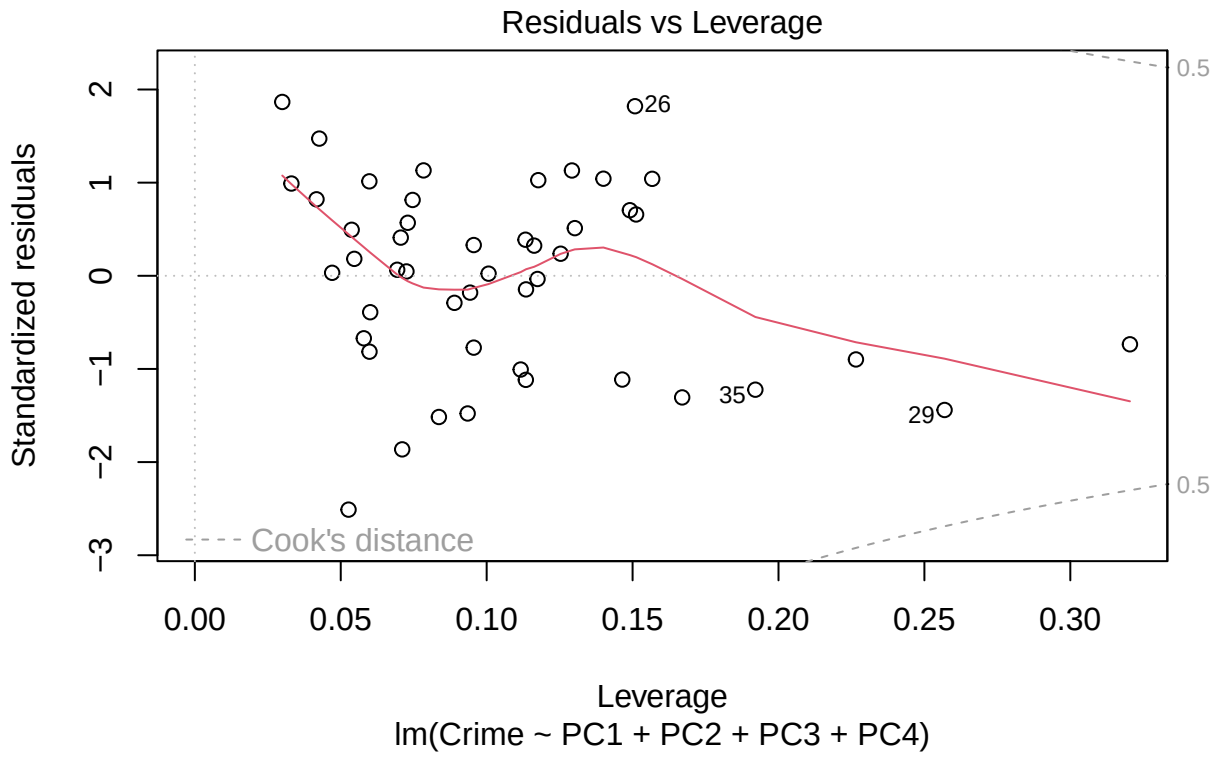
```r
# get scores for the first two PCs
PC_scores <- pca_result$x[,1:4]

# build regression model using the selected PCs
model_pca <- lm(Crime ~ PC1 + PC2 + PC3 + PC4,
                data = data.frame(Crime = uscrime_data$Crime_log,
                                  PC_scores))

# extract coefficients in terms of the original predictors
```

```r
original_coeff <- as.matrix(pca_result$rotation[, 1:4]) %*% coef(model_pca)[-1]

plot(model_pca)
```




Q−Q Residuals
Standardized residuals
Theoretical Quantiles
lm(Crime ~ PC1 + PC2 + PC3 + PC4)
2
46
27
−2
−1
0
1
2


Scale−Location
27
2
46
√|Standardized residuals|
0.0
0.5
1.0
1.5
6.4
6.6
6.8
7.0
7.2
7.4
Fitted values
lm(Crime ~ PC1 + PC2 + PC3 + PC4)




```
print(original_coeff)
```

```
##                [,1]
## M       -0.019819845
## So       0.015464890
## Ed       0.013084055
## Po1      0.068681152
## Po2      0.069937915
## LF      -0.016320770
## M.F     -0.029523955
## Pop      0.043682383
## NW       0.021128738
## U1      -0.034619214
## U2      -0.001736897
## Wealth   0.039550711
## Ineq    -0.027197648
## Prob     0.006702271
## Time    -0.007102780
```

Transform

```
# transform the new observation into the PCA space
new_observation_PC <- as.matrix(scale(new_observation,
                                      center = pca_result$center,
                                      scale = pca_result$scale)) %*% pca_result$rotation[,1:4]
```

```r
# predict using the model
predicted_value <- predict(model_pca, newdata = data.frame(PC1 = new_observation_PC[,1],
                                                           PC2 = new_observation_PC[,2],
                                                           PC3 = new_observation_PC[,3],
                                                           PC4 = new_observation_PC[,4]))

original_scale_prediction <- exp(predicted_value)

print(original_scale_prediction)
```

```
##      PC1
## 1037.789
```

R-Squared

```r
summary(model_pca)$adj.r.squared
```

```
## [1] 0.2314751
```

P-value

```r
# extract F-statistic
f_stat <- summary(model_pca)$fstatistic[1]

# degrees of freedom
df1 <- summary(model_pca)$fstatistic[2]
df2 <- summary(model_pca)$fstatistic[3]

# p-value
p_value <- 1 - pf(f_stat, df1, df2)

# results
print(paste("p-value:", p_value))
```

```
## [1] "p-value: 0.00426909291234767"
```