ISYE 6501

Homework 6

Artur Cabral, Marta Bras, Pedro Pinto, Katie Price

2019-09-30

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)!)

1. Importing and initial analysis of the data

```
#reading the data
crime_data <- read.table("uscrime.txt", header = TRUE)

# Overall statistics
summary_table <- crime_data %>%
  summarize(number_states = nrow(crime_data),
            total_crime_rate = sum(Crime),
            average_crime_rate = mean(Crime),
            min_crime_rate = min(Crime),
            max_crime_rate = max(Crime))

kable(summary_table, caption = "Crime Rate - overall statistics") %>%
  kable_styling(bootstrap_options = c("striped", "hover", "condensed", "bordered"))
```

```
# Statistics per state
summary_table_state <- crime_data %>% group_by(So) %>%
  summarize(number_states = nrow(crime_data),
            total_crime_rate = sum(Crime),
            average_crime_rate = mean(Crime),
            min_crime_rate = min(Crime),
            max_crime_rate = max(Crime))

kable(summary_table_state, caption = "Crime Rate per state") %>%
  kable_styling(bootstrap_options = c("striped", "hover", "condensed", "bordered"))
```

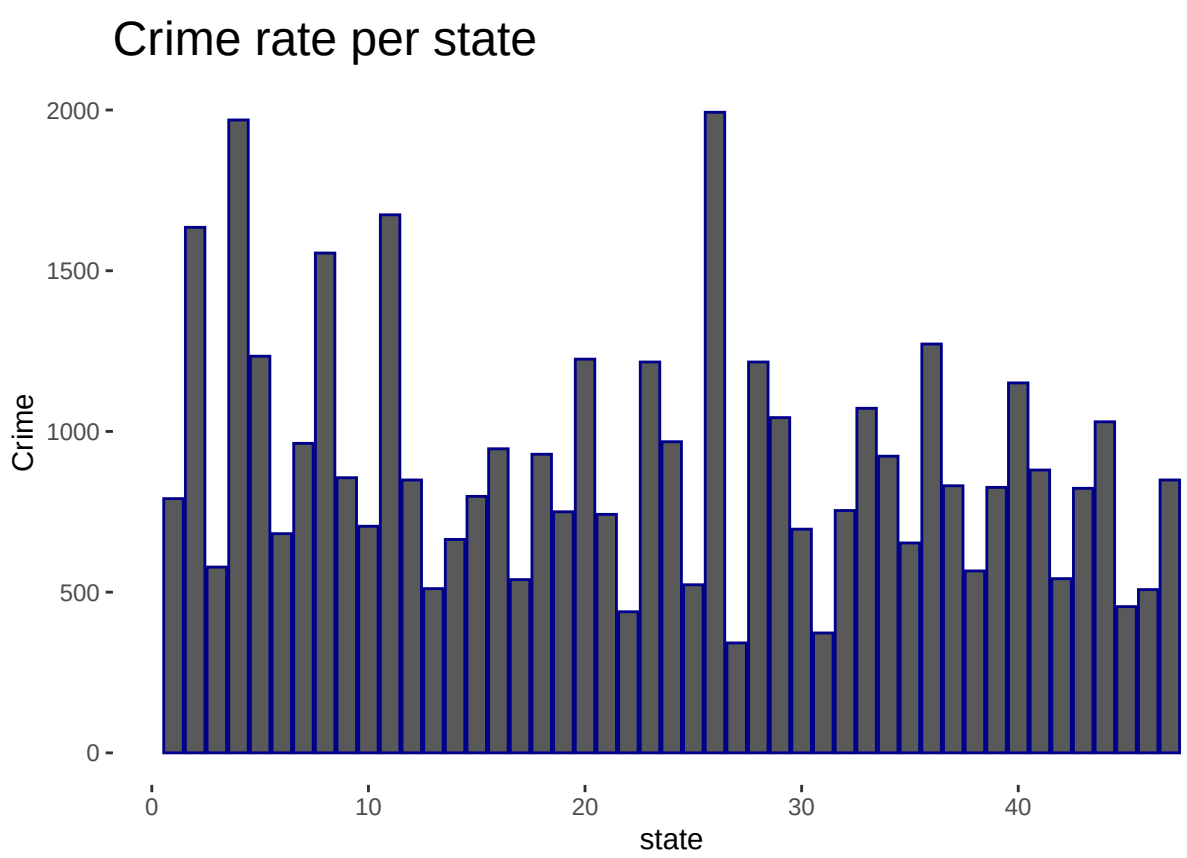
Table 1: Crime Rate - overall statistics

number_states	total_crime_rate	average_crime_rate	min_crime_rate	max_crime_rate
47	42539	905.0851	342	1993

Table 2: Crime Rate per state

So	number_states	total_crime_rate	average_crime_rate	min_crime_rate	max_crime_rate
0	47	28830	930.0000	342	1993
1	47	13709	856.8125	439	1555

```
# Crime rate per state
## adding an index variable
state <- seq(1, length(crime_data$So))
crime_data_2 <- cbind(state, crime_data)
state <- factor(crime_data_2$state)

ggplot(crime_data_2, aes(x=state, y = Crime)) +
  geom_col(color='darkblue') +
  ggtitle("Crime rate per state") +
  theme_bw() + theme(panel.border = element_blank(), panel.grid.major = element_blank(),
  panel.grid.minor = element_blank()) +
  theme(plot.title = element_text(size=18))
```

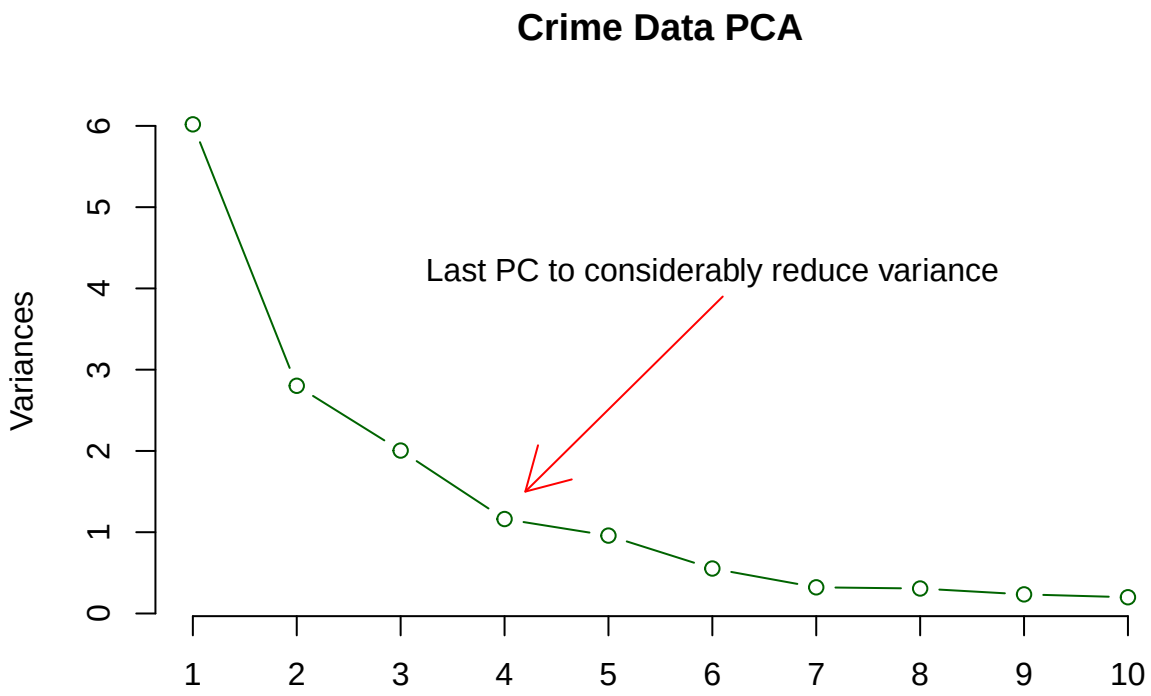


2. Principal Component Analysis:

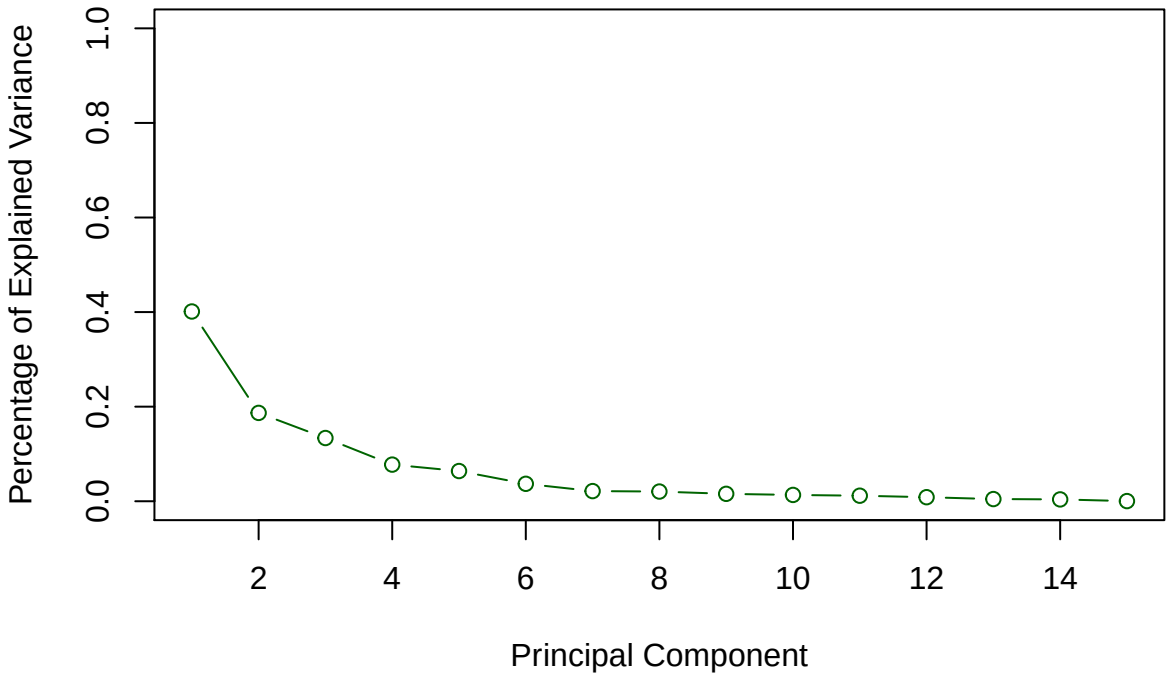
```r
# Running the Principal Component Analysis (PCA):

crime_pca <- prcomp(crime_data[,1:15], scale. = TRUE)
summary(crime_pca)
```

```
## Importance of components:
##                            PC1    PC2    PC3     PC4     PC5     PC6
## Standard deviation     2.4534 1.6739 1.4160 1.07806 0.97893 0.74377
## Proportion of Variance 0.4013 0.1868 0.1337 0.07748 0.06389 0.03688
## Cumulative Proportion  0.4013 0.5880 0.7217 0.79920 0.86308 0.89996
##                            PC7     PC8     PC9    PC10    PC11    PC12
## Standard deviation     0.56729 0.55444 0.48493 0.44708 0.41915 0.35804
## Proportion of Variance 0.02145 0.02049 0.01568 0.01333 0.01171 0.00855
## Cumulative Proportion  0.92142 0.94191 0.95759 0.97091 0.98263 0.99117
##                           PC13   PC14    PC15
## Standard deviation     0.26333 0.2418 0.06793
## Proportion of Variance 0.00462 0.0039 0.00031
## Cumulative Proportion  0.99579 0.9997 1.00000
```

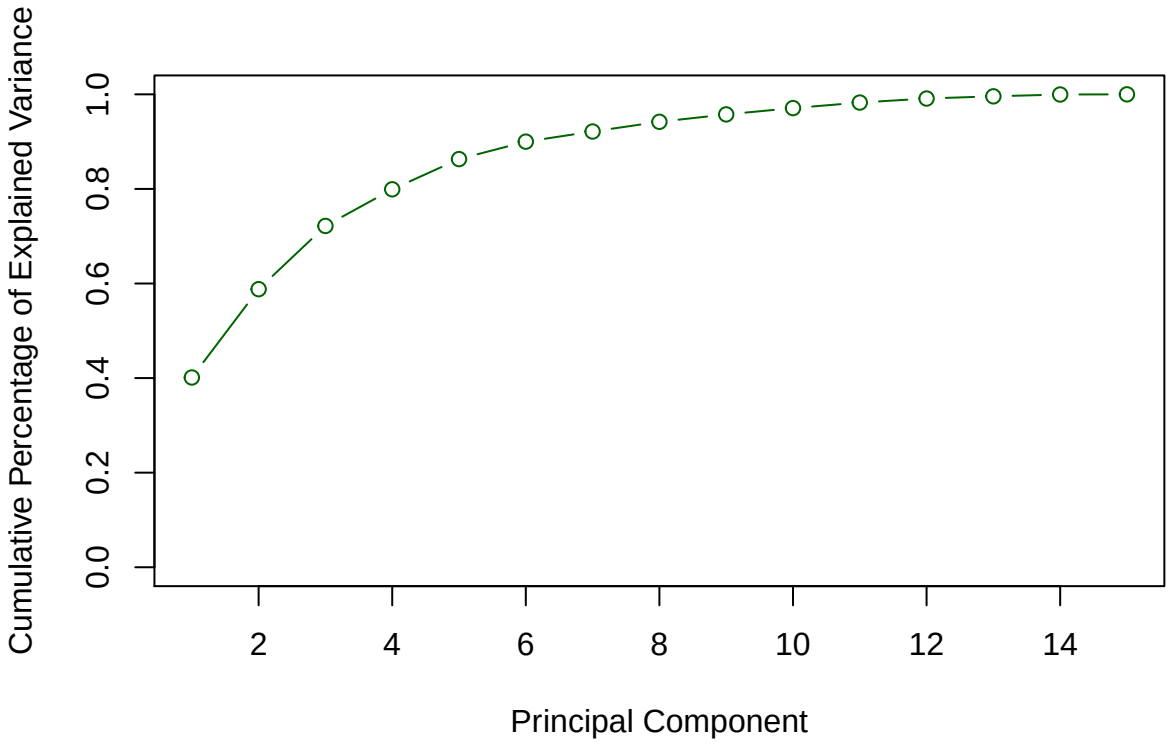
```r
# Plotting the cost graph of different number of principal components
# (commonly known as the "elbow graph"):

screeplot(crime_pca, main="Crime Data PCA", type="lines", col="darkgreen")
arrows(x0 = 6.1, y0 = 3.9, x1 = 4.2, y1 = 1.5, col = "red")
text(x = 6, y = 4.2, labels = "Last PC to considerably reduce variance")
```




```
# Based on the elbow graph, we should use the first 4 principal components.+


exp_variance <- crime_pca$sdev^2
prop_exp_variance <- exp_variance/sum(exp_variance)
cum_sum = cumsum(prop_exp_variance)

plot(prop_exp_variance, xlab = "Principal Component",
     ylab = "Percentage of Explained Variance",
     type = "b",
     ylim = c(0,1),
     col="darkgreen")
```




```
plot((cum_sum), xlab = "Principal Component",
     ylab = "Cumulative Percentage of Explained Variance",
     type = "b",
     ylim = c(0,1),
     col="darkgreen")
```



3. Linear Regression Model using the first four Principal Components:

The goal here is to use less variables and decrease the complexity of the model. Now, only utilizing the 4 main principal components, we will create a linear regression model.

```
#Selecting the 4 main principal components:
#Number of principal components we want to test = k
k  = 4

main_pcs = crime_pca$x[,1:k]
```

Now we combine PCs 1:k with the crime data from our original data set to create the linear regression model. Binding reduces the complexity of the model while making it more robust.

```
pc_df <- data.frame(cbind(main_pcs, crime_data[,16]))

# Performing the linear regression:

linear_reg <- lm(V5~.,data = pc_df)
summary(linear_reg)
```

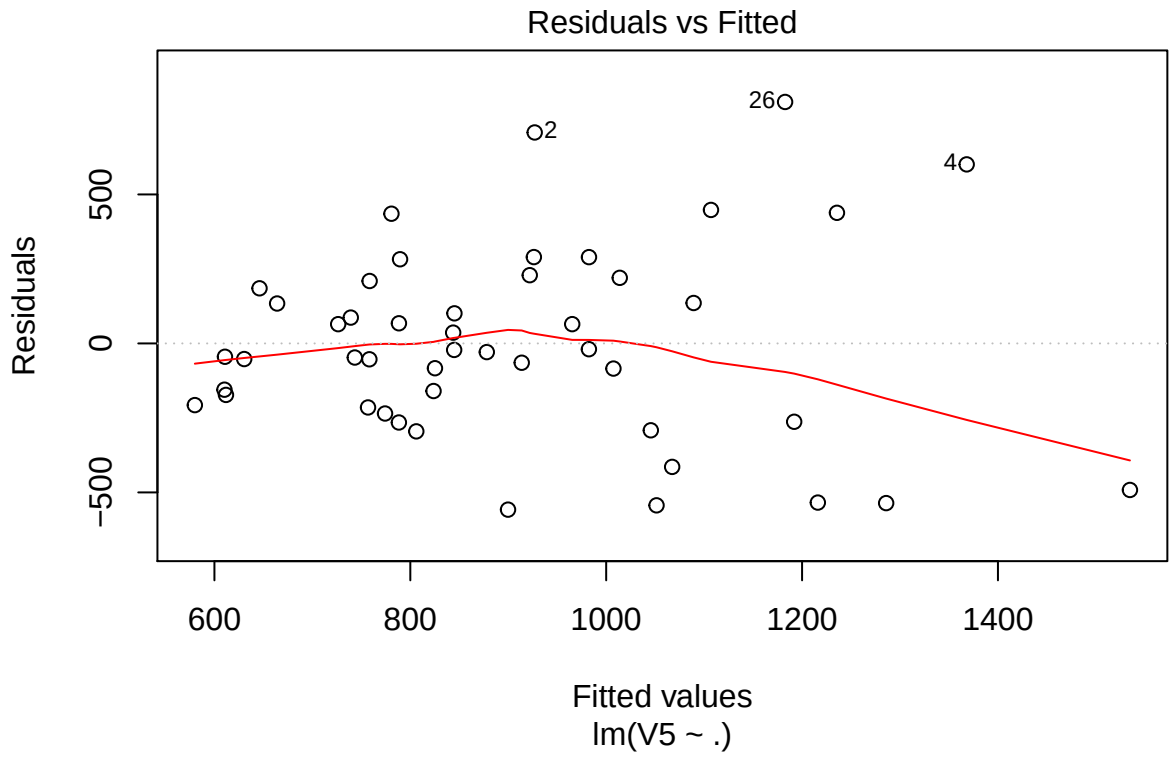
```
##
## Call:
## lm(formula = V5 ~ ., data = pc_df)
```

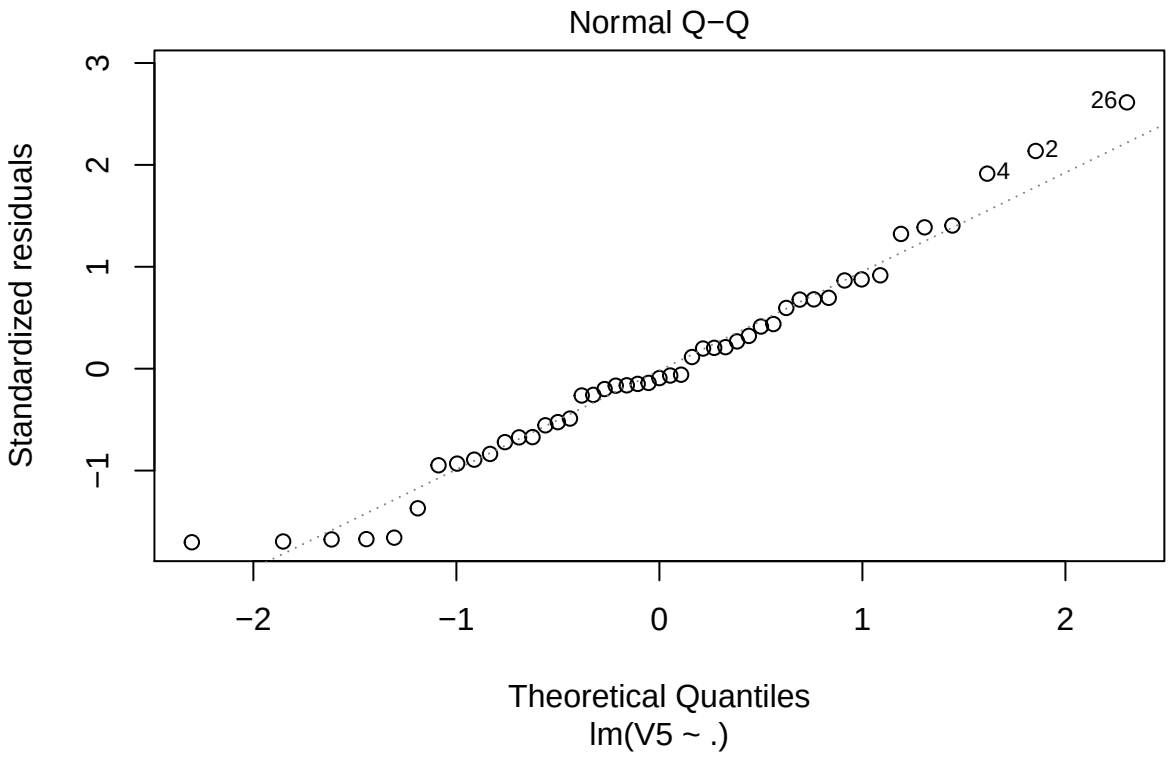
```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -557.76 -210.91  -29.08  197.26  810.35
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   905.09      49.07  18.443  < 2e-16 ***
## PC1            65.22      20.22   3.225  0.00244 **
## PC2           -70.08      29.63  -2.365  0.02273 *
## PC3            25.19      35.03   0.719  0.47602
## PC4            69.45      46.01   1.509  0.13872
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 336.4 on 42 degrees of freedom
## Multiple R-squared:  0.3091, Adjusted R-squared:  0.2433
## F-statistic: 4.698 on 4 and 42 DF,  p-value: 0.003178
```

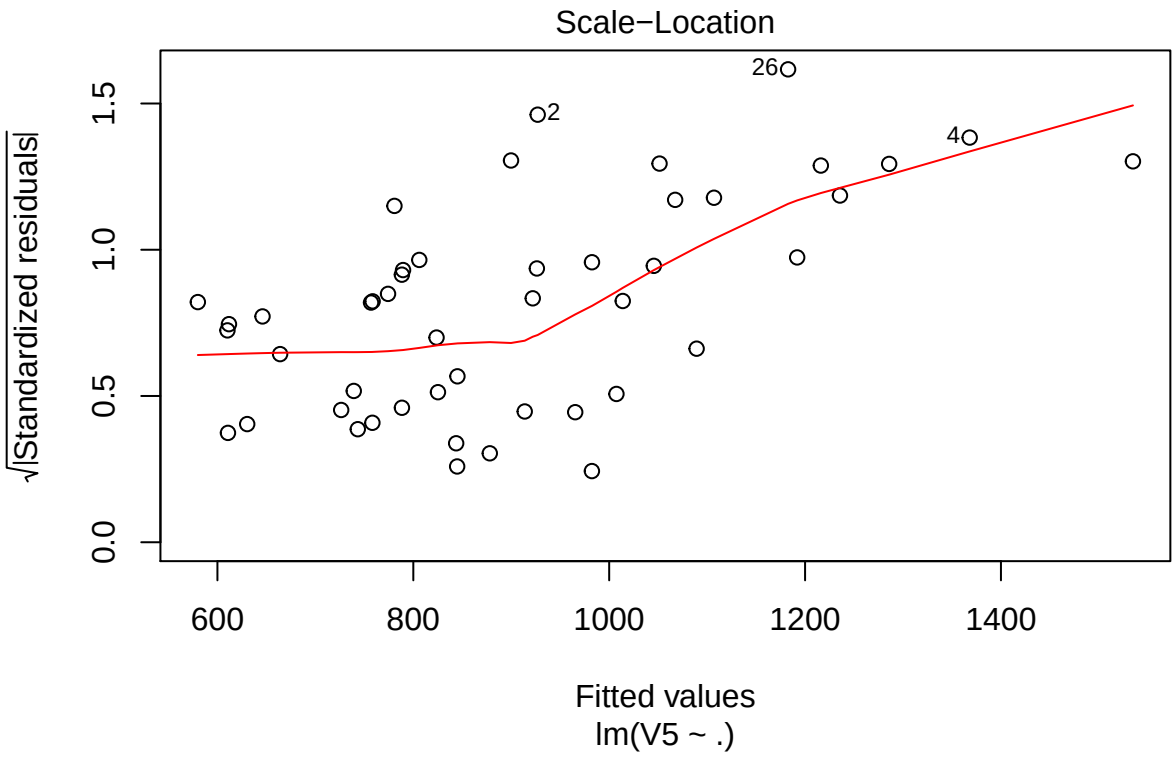
The model using only the 4 principal compnents has a R^2 of 30.9%. The adjusted R^2 is relatively lower (24.3%) which might be a small sign of overfitting, but it is better than the model with all variables.

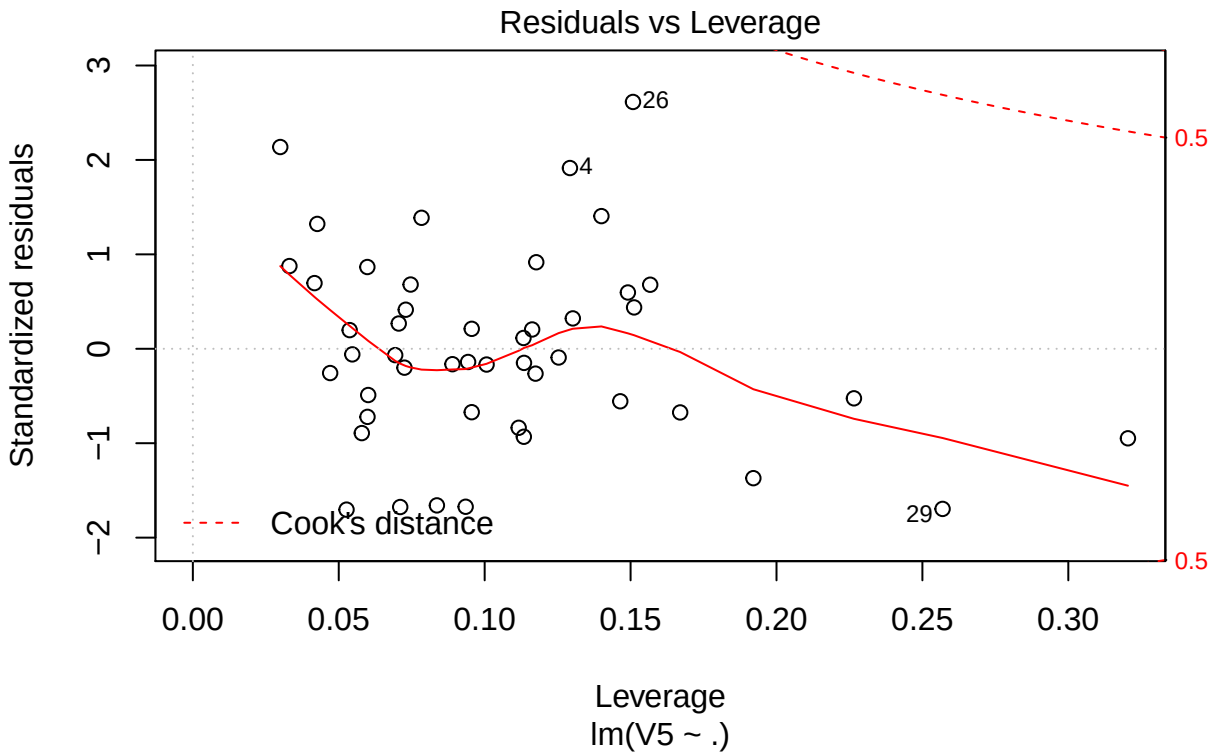
To annalyze the goodness-of-the fit of the model it is important to plot the results of the model and understand if the assumptions for linear regression are being met or not.

```
# Plots of the results
plot(linear_reg)
```




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(V5 ~ .)
26
2
4
−2
−1
0
1
2
3


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
26
2
4
600
800
1000
1200
1400
Fitted values
lm(V5 ~ .)



Looking at the residual vs fitted plot, we can see assumption of linearity and constant variance does not seem to hold as most of the data points are concentrated on the left side of the plot. The QQplot reveals a distribution close to normal, with some outliers on both sides of the distribution.

Now that we have the linear regression model, we need to transform the components in terms of the the original variables. First we find the intercept, then the model coefficients to create the β vector.

```
beta0 <- linear_reg$coefficients[1]
betas <- linear_reg$coefficients[2:(k+1)]
```

Now we multiply the coefficients by the rotated matrix, to create the α vector. Then, we will be able to recover the original α values by dividing the α vector by σ. To recover the original β we subtract it frmo the intercept of the sum of $\frac{\alpha * \mu}{\sigma}$.

```
alpha <- crime_pca$rotation[,1:k] %*% betas
mu <- sapply(crime_data[,1:15],mean)
sigma <- sapply(crime_data[,1:15],sd)
origAlpha <- alpha/sigma
origBeta0 <- beta0 - sum(alpha*mu /sigma)
```

Here, the estimates gives us the model $Y = ax + b$ where a is the scaled α and b is the original intercept. Then we use the estimates to calculate the R^2 values to observe the accuracy of the regression model.

```
estimates <- as.matrix(crime_data[,1:15]) %*% origAlpha + origBeta0
SSE = sum((estimates - crime_data[,16])^2)
SStot = sum((crime_data[,16] - mean(crime_data[,16]))^2)
```

```
R2 <- 1 - SSE/SStot
R2
```

```
## [1] 0.3091121
```

As the final step of the analysis, we use the newdata (same as last homework) to see how the new model predicts the crime rate. For that, we apply the PCA data onto the newdata so we can use the model, and then predict the crime rate using principal components and the newdata.

```
newdata = 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 = 39.0)
pred_df <- data.frame(predict(crime_pca, newdata))
pred <- predict(linear_reg, pred_df)
pred
```

```
##        1
## 1112.678
```

4. Conclusion

Compared to last week's prediction of 1304.245 using the leap function and 1038.413 using the step function, and R^2 of 78% and 74% respectively, this model seems slightly less sufficient at prescribing values. Even though this was a small test to compare both methods, we observed that with a R^2 of 30.9% and a prediction of 1112.678, the PCA model can deliver a prediction with almost the same accuracy, and significantly less predictors.