HW 6

2023-09-27

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))

```
options(repos = c(CRAN = "https://cran.rstudio.com"))
install.packages("dplyr")
```

```
##
## The downloaded binary packages are in
##  /var/folders/6r/d7grt80x11v5vty52zj5cywh0000gn/T//RtmpmdZ4aa/downloaded_packages
```

```
install.packages("DAAG")
```

```
##
## The downloaded binary packages are in
##  /var/folders/6r/d7grt80x11v5vty52zj5cywh0000gn/T//RtmpmdZ4aa/downloaded_packages
```

```
install.packages("magrittr")
```

```
##
## The downloaded binary packages are in
##  /var/folders/6r/d7grt80x11v5vty52zj5cywh0000gn/T//RtmpmdZ4aa/downloaded_packages
```

```
library(dplyr)
```

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

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

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

```
library(DAAG)
library(magrittr)
# Define the URL where the data is located
url <- "http://www.statsci.org/data/general/uscrime.txt"

# Use read.table to read the data from the URL into a data frame
data <- read.table(url, header = TRUE, sep = "\t")

# Display the first few rows of the data frame
head(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
```

```r
# Extract predictor columns
predictors <- (data[, c("M","So","Ed","Po1","Po2","LF","M.F","Pop","NW","U1","U2","Wealth","Ineq","Prob

# Calculate covariance matrix
corr_matrix <- cor(predictors) # Round to 2 decimal places
corr_matrix <- round(corr_matrix, 2)
# Print covariance matrix
print(corr_matrix)
```

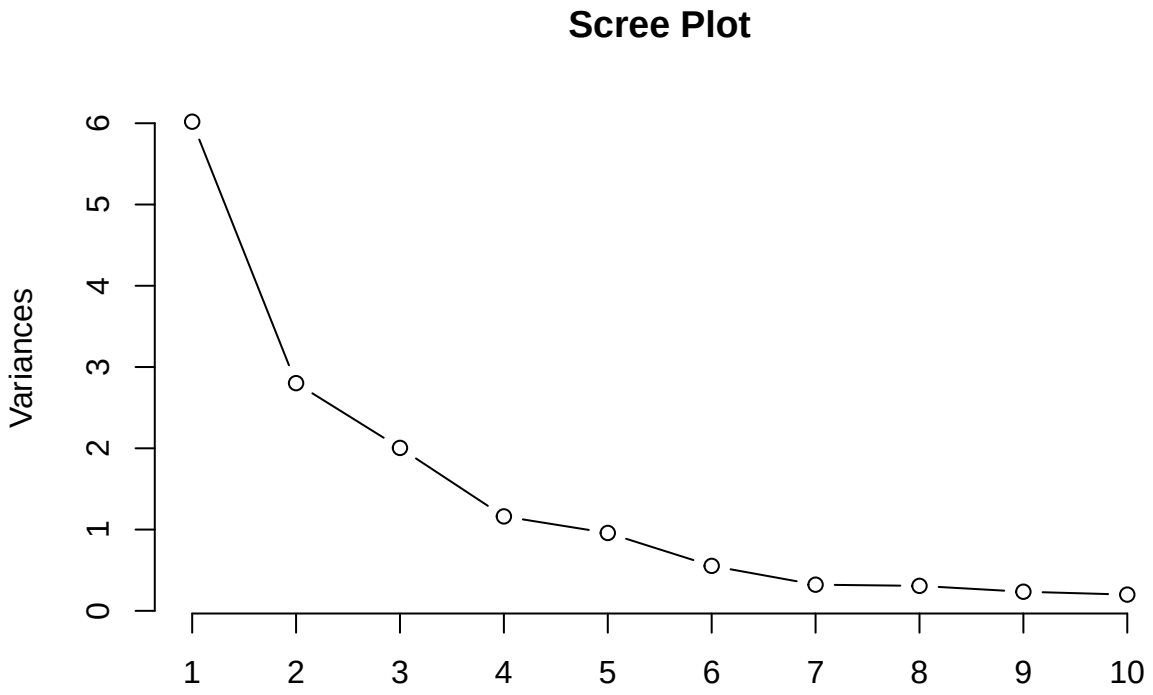
```
##              M    So    Ed   Po1   Po2    LF   M.F   Pop    NW    U1    U2 Wealth
## M         1.00  0.58 -0.53 -0.51 -0.51 -0.16 -0.03 -0.28  0.59 -0.22 -0.24  -0.67
## So        0.58  1.00 -0.70 -0.37 -0.38 -0.51 -0.31 -0.05  0.77 -0.17  0.07  -0.64
## Ed       -0.53 -0.70  1.00  0.48  0.50  0.56  0.44 -0.02 -0.66  0.02 -0.22   0.74
## Po1      -0.51 -0.37  0.48  1.00  0.99  0.12  0.03  0.53 -0.21 -0.04  0.19   0.79
## Po2      -0.51 -0.38  0.50  0.99  1.00  0.11  0.02  0.51 -0.22 -0.05  0.17   0.79
## LF       -0.16 -0.51  0.56  0.12  0.11  1.00  0.51 -0.12 -0.34 -0.23 -0.42   0.29
## M.F      -0.03 -0.31  0.44  0.03  0.02  0.51  1.00 -0.41 -0.33  0.35 -0.02   0.18
## Pop      -0.28 -0.05 -0.02  0.53  0.51 -0.12 -0.41  1.00  0.10 -0.04  0.27   0.31
## NW        0.59  0.77 -0.66 -0.21 -0.22 -0.34 -0.33  0.10  1.00 -0.16  0.08  -0.59
## U1       -0.22 -0.17  0.02 -0.04 -0.05 -0.23  0.35 -0.04 -0.16  1.00  0.75   0.04
## U2       -0.24  0.07 -0.22  0.19  0.17 -0.42 -0.02  0.27  0.08  0.75  1.00   0.09
## Wealth   -0.67 -0.64  0.74  0.79  0.79  0.29  0.18  0.31 -0.59  0.04  0.09   1.00
## Ineq      0.64  0.74 -0.77 -0.63 -0.65 -0.27 -0.17 -0.13  0.68 -0.06  0.02  -0.88
## Prob      0.36  0.53 -0.39 -0.47 -0.47 -0.25 -0.05 -0.35  0.43 -0.01 -0.06  -0.56
## Time      0.11  0.07 -0.25  0.10  0.08 -0.12 -0.43  0.46  0.23 -0.17  0.10   0.00
## Crime    -0.09 -0.09  0.32  0.69  0.67  0.19  0.21  0.34  0.03 -0.05  0.18   0.44
##            Ineq  Prob  Time Crime
## M          0.64  0.36  0.11 -0.09
## So         0.74  0.53  0.07 -0.09
## Ed        -0.77 -0.39 -0.25  0.32
## Po1       -0.63 -0.47  0.10  0.69
## Po2       -0.65 -0.47  0.08  0.67
## LF        -0.27 -0.25 -0.12  0.19
## M.F       -0.17 -0.05 -0.43  0.21
## Pop       -0.13 -0.35  0.46  0.34
## NW         0.68  0.43  0.23  0.03
## U1        -0.06 -0.01 -0.17 -0.05
## U2         0.02 -0.06  0.10  0.18
## Wealth    -0.88 -0.56  0.00  0.44
## Ineq       1.00  0.47  0.10 -0.18
```

```
## Prob    0.47  1.00 -0.44 -0.43
## Time    0.10 -0.44  1.00  0.15
## Crime  -0.18 -0.43  0.15  1.00
```

```r
# Perform PCA
crime_pca = prcomp(data[,-16], center=TRUE, scale=TRUE)
summary(crime_pca)
```

```
## 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
```

```r
# Determine number of components to retain
screeplot(crime_pca, main = "Scree Plot", type = "line")
```




```r
#we set the number of PC's we want to test, and then combine them with our original crime data.
npc = 4 # elbow suggests 4 components

uscrime_matrix <- cbind(crime_pca$x[,1:npc],data[,16])

#now that the response variable is combined to the dataset, we can create the linear regression model.
uscrime_model <- lm(V5~., data = as.data.frame(uscrime_matrix))

#Model from Question 8.2
```

```r
model_1 <- lm(Crime ~ Po1 + Ineq + M.F + Prob + M + Ed + U2, data = data)

summary(uscrime_model)
```

```
##
## Call:
## lm(formula = V5 ~ ., data = as.data.frame(uscrime_matrix))
##
## 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
```

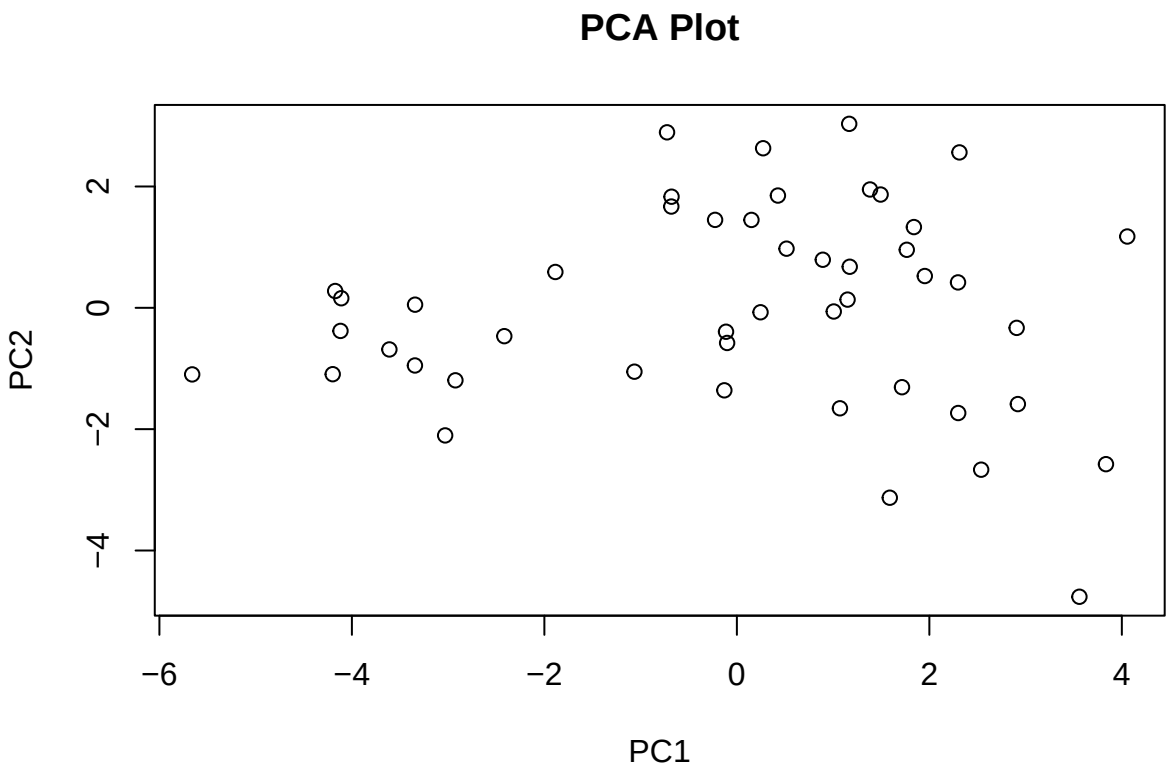
```r
summary(model_1)
```

```
##
## Call:
## lm(formula = Crime ~ Po1 + Ineq + M.F + Prob + M + Ed + U2, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -433.83 -121.23    2.28  125.26  551.58
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5666.91    1152.50  -4.917 1.63e-05 ***
## Po1           116.58      13.91   8.382 2.95e-10 ***
## Ineq           65.69      14.16   4.640 3.87e-05 ***
## M.F            10.84      12.40   0.874  0.38750
## Prob        -3858.20    1533.99  -2.515  0.01613 *
## M              97.87      34.38   2.846  0.00701 **
## Ed            170.22      54.01   3.151  0.00312 **
## U2             79.01      42.71   1.850  0.07190 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 201.3 on 39 degrees of freedom
## Multiple R-squared:  0.7704, Adjusted R-squared:  0.7291
## F-statistic: 18.69 on 7 and 39 DF,  p-value: 1.173e-10
```

```r
# Plot first two principal components
plot(crime_pca$x[,1], crime_pca$x[,2],
```

```
     xlab = "PC1", ylab = "PC2",
     main = "PCA Plot")

# Add text labels for each state
text(crime_pca$x[,1], crime_pca$x[,2], labels = rownames(crime_pca$x))
```




```
#Extracting the intercept value from a previously created linear regression model
intercept <- uscrime_model$coefficients[1]

#Extracting the coefficients of the independent variables
beta_vector <- uscrime_model$coefficients[2:(npc+1)]

#This step combines the principal components (from PCA) with their corresponding coefficients to create
#calculate the alpha_vector by multiplying the PCA rotation matrix (principal components) with the extr
directions <- crime_pca$rotation[,1:npc] %*% beta_vector

#Now, we're computing the "original alpha" and "original beta" values. we're dividing the alpha_vector
average <- sapply(data[,1:15], mean)
spread <- sapply(data[,1:15], sd)

#divide the alpha_vector by sd/spread, which means we are dividing by the standard deviation of the ind
original_alpha <- directions / spread
#This part adjusts our starting point (intercept) based on how much the variables typically vary (avera
original_beta <- intercept - sum(directions*average / spread)

#We're using the calculated original_alpha and original_beta values to make predictions. We multiply th
estimates <- as.matrix(data[,1:15]) %*% original_alpha + original_beta

#Calculating the R-squared (R2) value to assess how well our model fits the data. The SSE represents th
SSE = sum((estimates - data[,16])^2)
```

```r
SStot = sum((data[,16] - mean(data[,16]))^2)
R2 <- 1 - SSE/SStot
R2
```

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

```r
R2_adjust <- R2 - (1-R2)*npc/(nrow(data)-npc-1)
R2_adjust
```

```
## [1] 0.2433132
```

```r
aic_model_1 <- AIC(model_1)
bic_model_1 <- BIC(model_1)

cat("AIC_Model_1:", aic_model_1, "\n")
```

```
## AIC_Model_1: 641.2546
```

```r
cat("BIC_Model_1:", bic_model_1, "\n")
```

```
## BIC_Model_1: 657.9059
```

```r
aic_uscrime_model <- AIC(uscrime_model)
bic_uscrime_model <- BIC(uscrime_model)

cat("AIC_uscrime_model:", aic_uscrime_model, "\n")
```

```
## AIC_uscrime_model: 687.0241
```

```r
cat("BIC_uscrime_model:", bic_uscrime_model, "\n")
```

```
## BIC_uscrime_model: 698.125
```

```r
#we can now move on to using last weeks data on our new model to compare.
new_city <- 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 = 0.040,
                       Time = 39.0)

pred_newcity <- data.frame(predict(crime_pca, new_city))

pred_newcity_model <- predict(uscrime_model, pred_newcity)

pred_newcity_model
```

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

```r
#we can now move on to using last weeks data on our new model to compare.
new_city <- 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 = 0.040,
                       Time = 39.0)

pred_model_1 <- predict(model_1, new_city)

pred_model_1
```

```
##        1
```

```
## 1273.781
```

One of the benefits of PCA is multi collinearity reduction. Multi collinaearity is particularly problematic as the marginal effect of each coefficient is no longer interpretable relative to the response varible. As can be seen from the correlation matrix, some of our variables are highly correlated. For instance, including per capita expenditure on police protection in both 1959 and 1960 does not improve the model, these two variables are highly correlated with a value of 0.99... neither does including the two unemployment variables, which are highly correlated and could indicate that unemployment is uniform across the population. PCA produces orthogonal (uncorrelated) principal components, reducing multicollinearity in the model, and making the model simpler which helps to reduce overfitting.

```
       M    So    Ed   Po1   Po2    LF   M.F   Pop    NW    U1    U2 Wealth  Ineq  Prob  Time Crime
```

M 1.00 0.58 -0.53 -0.51 -0.51 -0.16 -0.03 -0.28 0.59 -0.22 -0.24 -0.67 0.64 0.36 0.11 -0.09 So 0.58 1.00 -0.70 -0.37 -0.38 -0.51 -0.31 -0.05 0.77 -0.17 0.07 -0.64 0.74 0.53 0.07 -0.09 Ed -0.53 -0.70 1.00 0.48 0.50 0.56 0.44 -0.02 -0.66 0.02 -0.22 0.74 -0.77 -0.39 -0.25 0.32 Po1 -0.51 -0.37 0.48 1.00 0.99 0.12 0.03 0.53 -0.21 -0.04 0.19 0.79 -0.63 -0.47 0.10 0.69 Po2 -0.51 -0.38 0.50 0.99 1.00 0.11 0.02 0.51 -0.22 -0.05 0.17 0.79 -0.65 -0.47 0.08 0.67 LF -0.16 -0.51 0.56 0.12 0.11 1.00 0.51 -0.12 -0.34 -0.23 -0.42 0.29 -0.27 -0.25 -0.12 0.19 M.F -0.03 -0.31 0.44 0.03 0.02 0.51 1.00 -0.41 -0.33 0.35 -0.02 0.18 -0.17 -0.05 -0.43 0.21 Pop -0.28 -0.05 -0.02 0.53 0.51 -0.12 -0.41 1.00 0.10 -0.04 0.27 0.31 -0.13 -0.35 0.46 0.34 NW 0.59 0.77 -0.66 -0.21 -0.22 -0.34 -0.33 0.10 1.00 -0.16 0.08 -0.59 0.68 0.43 0.23 0.03 U1 -0.22 -0.17 0.02 -0.04 -0.05 -0.23 0.35 -0.04 -0.16 1.00 0.75 0.04 -0.06 -0.01 -0.17 -0.05 U2 -0.24 0.07 -0.22 0.19 0.17 -0.42 -0.02 0.27 0.08 0.75 1.00 0.09 0.02 -0.06 0.10 0.18 Wealth -0.67 -0.64 0.74 0.79 0.79 0.29 0.18 0.31 -0.59 0.04 0.09 1.00 -0.88 -0.56 0.00 0.44 Ineq 0.64 0.74 -0.77 -0.63 -0.65 -0.27 -0.17 -0.13 0.68 -0.06 0.02 -0.88 1.00 0.47 0.10 -0.18 Prob 0.36 0.53 -0.39 -0.47 -0.47 -0.25 -0.05 -0.35 0.43 -0.01 -0.06 -0.56 0.47 1.00 -0.44 -0.43 Time 0.11 0.07 -0.25 0.10 0.08 -0.12 -0.43 0.46 0.23 -0.17 0.10 0.00 0.10 -0.44 1.00 0.15 Crime -0.09 -0.09 0.32 0.69 0.67 0.19 0.21 0.34 0.03 -0.05 0.18 0.44 -0.18 -0.43 0.15 1.00

Based on the output from our screeplot: PC1 is the most important component, explaining 40.1% of the total variance in the data. This component has the largest standard deviation and proportion of variance. The first 5 PCs (PC1-PC5) explain the bulk of the variance, around 86%. So these top 5 components capture most of the meaningful information in the data. PC1-PC4 have standard deviations > 1, so they account for a substantial amount of variance each. PC5-PC15, gradually explain smaller and smaller amounts of variance. By PC10, each component is explaining only around 1% of variance. PC13-PC15 explain less than 0.5% variance each. 100% of the total variance is explained by the full set of 15 PCs. But over 99% is covered just with PC1-PC10. In summary, PC1 is the most important and PCs 1-5 contain most of the useful information. The later components explain diminishing amounts of variance and are likely less informative. Analyzing just the top 5-10 PCs would likely be sufficient for most purposes.

I decided to select 4 as the ideal number of components using the scree plot. This seemed reasonable since by increasing the number of components after 4, the amount of variance captured seemed to level off. One could argue that 6 seems to be where the elbow is most defined, but as can be seen from the scree plot, there is an immaterial difference between the variance explained by 4 and 5, and adding an additional component to make it 6, doesn't explain much additional variance. Typically its best to retain the components that come before or at the elbow point because they explain the majority of the variance while keeping the model simpler.

Following this step we created our new model, using the components created and our dependent variable (Crime). To see how well the PC model did, I compared the summary results with my model from Q. 8.2.

It could be argued that my model from the previous homework boast some superior explanatory power, as it was rigorously validated using backward elimination and forward selection. I will call my model from Q 8.2, model_1. So from the results, The PC model demonstrates a higher Residual Standard Error (RSE) of 336, suggesting that its predictions exhibit a larger average error compared to the actual data. Conversely, model_1 exhibits a lower RSE, indicating superior prediction accuracy. Furthermore, the PC model presents a significantly lower Multiple R-squared (R^2) value of 0.309, implying that it explains only about 30.9% of the variance in the dependent variable. In contrast, model_1 has a higher R^2 of 0.77, signifying its capacity to capture a larger portion of the data's variability. The adjusted R-squared of model_1 is 0.729, indicating

robust explanatory power while accommodating additional predictors. Additionally, model_1 has a more substantial F-statistic and an extremely low p-value, highlighting its statistical significance.

Next, we compute the "original alpha" and "original beta" values for our model. We divide the alpha vector, which reflects the directions of the principal components, by the standard deviation (spread) of the independent variables. This step is crucial for unscaling the coefficients, as it reverses the standardization applied during PCA. The original beta is adjusted based on the average and spread of the independent variables. With the calculated original alpha and beta values, we proceed to make predictions. We apply these coefficients to the independent variables in our dataset (columns 1 to 15) to estimate the dependent variable (V6) using the linear regression model. Finally, we calculate the R-squared (R^2) value, which is 0.309 and the adjusted R-Square was slightly lower 0.225. This confirmed that the results from our estimates are consistent with the results from the model summary discussed earlier.

One additional step that I took was to evaluate the AIC and BIC of both models to get a better idea of their fit. The AIC and BIC values provide evidence that Model_1 fits the data better than the uscrime_model. Model_1 has a substantially lower AIC of 641 compared to 687 for the uscrime_model. Similarly, the BIC is 658 for Model_1 versus 698 for uscrime_model. The difference in AIC values of over 10 and BIC values of over 10 between the two models indicates overwhelmingly strong evidence that Model_1 is superior. The lower AIC and BIC values for Model_1 show that it provides a better fit to the data while penalizing model complexity. The uscrime_model, which uses principal components from PCA analysis as predictors, is much more complex with more parameters. The higher AIC and BIC values reflect this increased model complexity and suggest the uscrime_model is overfitting the data compared to Model_1.

Finally, while model_1 predicted a crime rate of 1274 for a new data, the PC model was lower at 1113. Both seemed reasonable as they fall within the dataset but as with my conclusion on model_1, the model performance will only be as good as the quality fo data that we are working with.