ISYE 6501 HW 6

Ryan Cherry

2024-02-18

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)!)

The main purpose of Principal Component Analysis (PCA) is to reduce the dimensions of the data set and to rank variables by relative importance. PCA was undertaken here in order to create a model that did not over-fit the data in hopes obtain a more reasonable prediction for crime rate compared to the one from Homework 5.

```
#read in the crime data set
crime_PCA <- read.table("C:/Users/ryanc/Downloads/uscrime.txt", header = TRUE)
head(crime_PCA)
```

```
##      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
```

PCA was undertaken but in order to obtain the best results, the data needed to be centered and scaled. When selecting principal components for a regression model, we only include those components which explain the largest proportion of variability in the data, while discarding those components that do not account for a large proportion of the variability.

Based on the summary output, it seemed like the first 5 principal components explain a good majority of the variance in the data (about 85%), which would mean that 5 principal components would be the optimal number to build a regression model with. However, a visual assessment would need to confirm this suspicion.

```r
#conducted Principal Component Analysis using the prcomp function with the data scaled
#and centered
components <- prcomp(crime_PCA[, -16], center = T, scale = T)
summary(components)
```

```
## 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
#viewed the means of each predictor after conducting PCA
components$center
```

```
##            M           So           Ed          Po1          Po2           LF
## 1.385745e+01 3.404255e-01 1.056383e+01 8.500000e+00 8.023404e+00 5.611915e-01
##          M.F          Pop           NW           U1           U2       Wealth
## 9.830213e+01 3.661702e+01 1.011277e+01 9.546809e-02 3.397872e+00 5.253830e+03
##         Ineq         Prob         Time
## 1.940000e+01 4.709138e-02 2.659792e+01
```

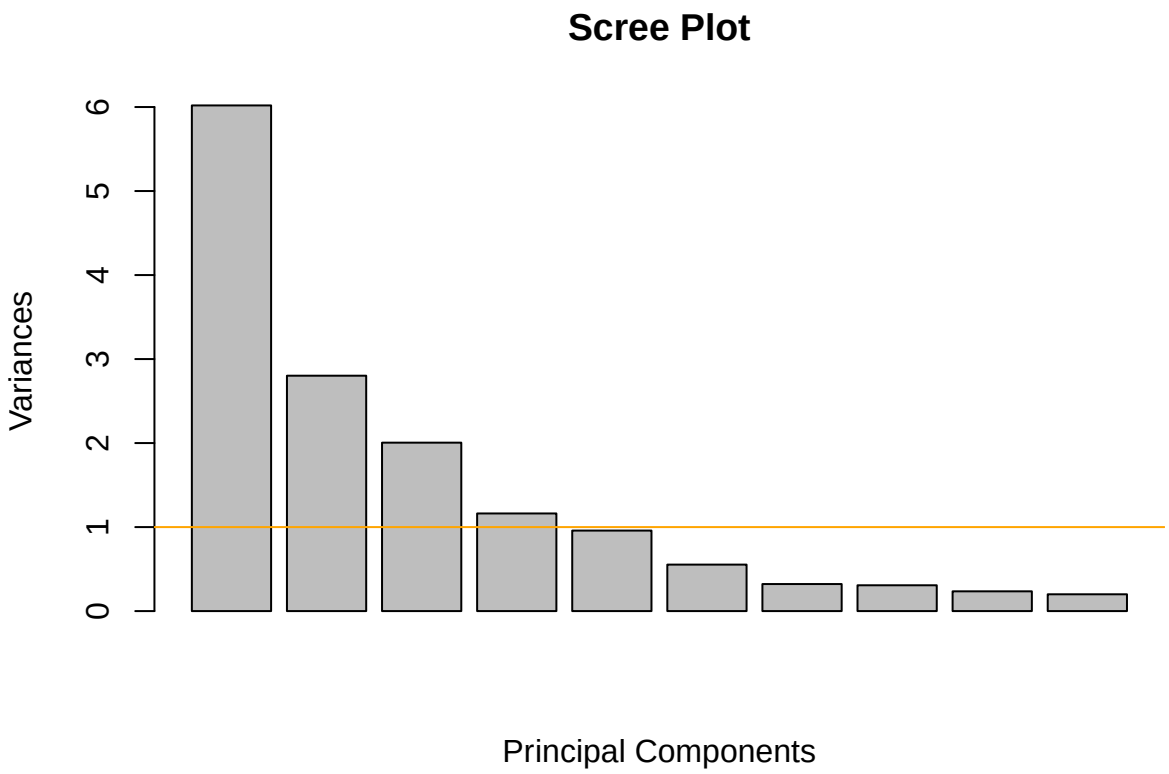
```r
#viewed the eigenvectors of the predictors in each principal component
components$rotation
```

```
##                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
```

A scree plot is meant to show the optimal number of components to include. All the principal components that exceeded the threshold of 1, which is widely used, would be the number of principal components the linear regression model would include. Based on the plot, 5 components meet or exceed the threshold of 1. As a result, the linear regression model would include the first 5 principal components as predictors.

```
#created a scree plot to determine the optimal number of principal components to include
#in the model
screeplot(components, xlab = "Principal Components", main = "Scree Plot", type = "bar")
abline(h=1, col="orange")
```



The first five principal components were used to build a linear regression model that could provide a more reasonable prediction of crime rate.

The linear regression model from the last homework that was used to make the prediction had an r-squared value of 0.8031 and an adjusted r-squared value of 0.7078. These values were completely different, which indicated that the model had too many predictors (15) and was overfitting the data. There were only 4 predictors significant at the 0.05 level and several coefficient values that had large standard errors above 100, indicating that multicollinearity was a problem.

The regression model built with the first five principal components had a lower r-squared value of 0.6452 and adjusted r-squared of 0.6019, but the difference between both values was not as high. The model does not explain as much of the variability in crime rate, but the trade-off is that the model does not overfit the data and may be more accurate in its predictions. The signs of multicollinearity are not present like they were in the model from Homework 5. There are no high standard error values above 100, all of the predictors except for PC3 are significant.

Overall, the linear regression model made using PCA is a better model to use than the one from Homework 5. The model equation is as follows:

Y = 905.09 + 65.22 * PC1 -70.08 * PC2 + 25.19 * PC3 + 69.45 * PC4 - 229.04 * PC5

```
#extracted the first five principal components for use in the linear regression model
Principal_cs<- components$x[,1:5]

#combined principal components and crime response variable into a new data set for the
#regression model
adjustedcrime <-cbind(Principal_cs,crime_PCA[,16])

#created a linear regression model with the first five principal components against the
```

```r
#6th variable, crime
crime_pca_model <- lm(V6 ~., data = as.data.frame(adjustedcrime))

summary(crime_pca_model)
```

```
##
## Call:
## lm(formula = V6 ~ ., data = as.data.frame(adjustedcrime))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -420.79 -185.01   12.21  146.24  447.86
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   905.09      35.59  25.428  < 2e-16 ***
## PC1            65.22      14.67   4.447 6.51e-05 ***
## PC2           -70.08      21.49  -3.261  0.00224 **
## PC3            25.19      25.41   0.992  0.32725
## PC4            69.45      33.37   2.081  0.04374 *
## PC5          -229.04      36.75  -6.232 2.02e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 244 on 41 degrees of freedom
## Multiple R-squared:  0.6452, Adjusted R-squared:  0.6019
## F-statistic: 14.91 on 5 and 41 DF,  p-value: 2.446e-08
```

In order to complete PCA, the data had to be scaled. However, the prediction from Homework 5 was made with the original scale of the data. In order to make a reasonable prediction, the predictor and intercept coefficients all had to be unscaled back to the original scale of the data.

```r
#obtain coefficient values for each principal component in the regression model
pca_coefficients <- crime_pca_model$coefficients[2:6]

#obtain rotation values for each principal component, showing how much the coordinates of the coefficie
#had to be adjusted while still maintaining interpretability
rotation_matrix <- components$rotation[, 1:5]

#multiply rotation values by coefficients from regression model
coefficient_rotations <- rotation_matrix %*% pca_coefficients

#transpose the rotation value matrix to obtain the implied regression coefficients with
#the PCA scale
t(coefficient_rotations)
```

```
##              M       So       Ed      Po1      Po2      LF      M.F      Pop
## [1,] 60.79435 37.84824 19.94776 117.3449 111.4508 76.2549 108.1266 58.88024
##             NW       U1       U2   Wealth    Ineq      Prob     Time
## [1,] 98.07179 2.866783 32.34551 35.93336 22.1037 -34.64026 27.20502
```

```r
#unscale the PCA coefficients to get them back to the original scale of the crime data set
unscaled_predictors <- coefficient_rotations/components$scale
unscaled_predictors
```

```
##                 [,1]
## M       4.837374e+01
## So      7.901922e+01
## Ed      1.783120e+01
## Po1     3.948484e+01
## Po2     3.985892e+01
## LF      1.886946e+03
## M.F     3.669366e+01
## Pop     1.546583e+00
## NW      9.537384e+00
## U1      1.590115e+02
## U2      3.829933e+01
## Wealth  3.724014e-02
## Ineq    5.540321e+00
## Prob   -1.523521e+03
## Time    3.838779e+00
```

```r
#obtain intercept coefficient from the PCA regression model
intercept <- crime_pca_model$coefficients[1]

#unscaled the intercept to get it back to the original scale of the crime dataset
unscaled_intercept <- intercept - sum((coefficient_rotations * components$center)/
                                                    components$scale)

unscaled_intercept
```

```
## (Intercept)
##   -5933.837
```

The original base model model predicted that a city with 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, and Time = 39.0 had a crime rate of 155.This value seemed too low given the coefficient values.

The linear regression model created using principal components predicted a crime rate of 1389, which is much higher but seems more reasonable. I cannot definitively say this prediction is more accurate without more empirical data and computing accuracy measures.

But, in conclusion, the predicted crime rate of 1389 makes more sense than the previous prediction of 155 from Homework 5.

```r
#created a test data point to predict the crime rate
prediction_point <-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)

#predicted the crime rate using the PCA model
prediction <- data.frame(predict(components, prediction_point))
```

```r
pca_crime_prediction <- predict(crime_pca_model, prediction)

#final predicted crime rate
pca_crime_prediction
```

```
##        1
## 1388.926
```