Homework 6

Anonymous

2024-02-21

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)!)

```
library(ggbiplot)
```

```
## Loading required package: ggplot2
```

```
library(ggplot2)

crime_data <- read.table(file= "C:\\Users\\sheya\\OneDrive\\Desktop\\uscrime.txt",
                         header = TRUE)
test <- 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)

#X is 15 predictors
X <- crime_data[,1:15]


#Calculating the principal components of the data
#Scaled TRUE so each variable in the data are scaled to have a mean
#of 0 and a SD of 1
PCA <- prcomp(X, scale = TRUE, center = TRUE)

PCA
```

```
## Standard deviations (1, .., p=15):
##  [1] 2.45335539 1.67387187 1.41596057 1.07805742 0.97892746 0.74377006
##  [7] 0.56729065 0.55443780 0.48492813 0.44708045 0.41914843 0.35803646
## [13] 0.26332811 0.24180109 0.06792764
##
## Rotation (n x k) = (15 x 15):
##                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
```

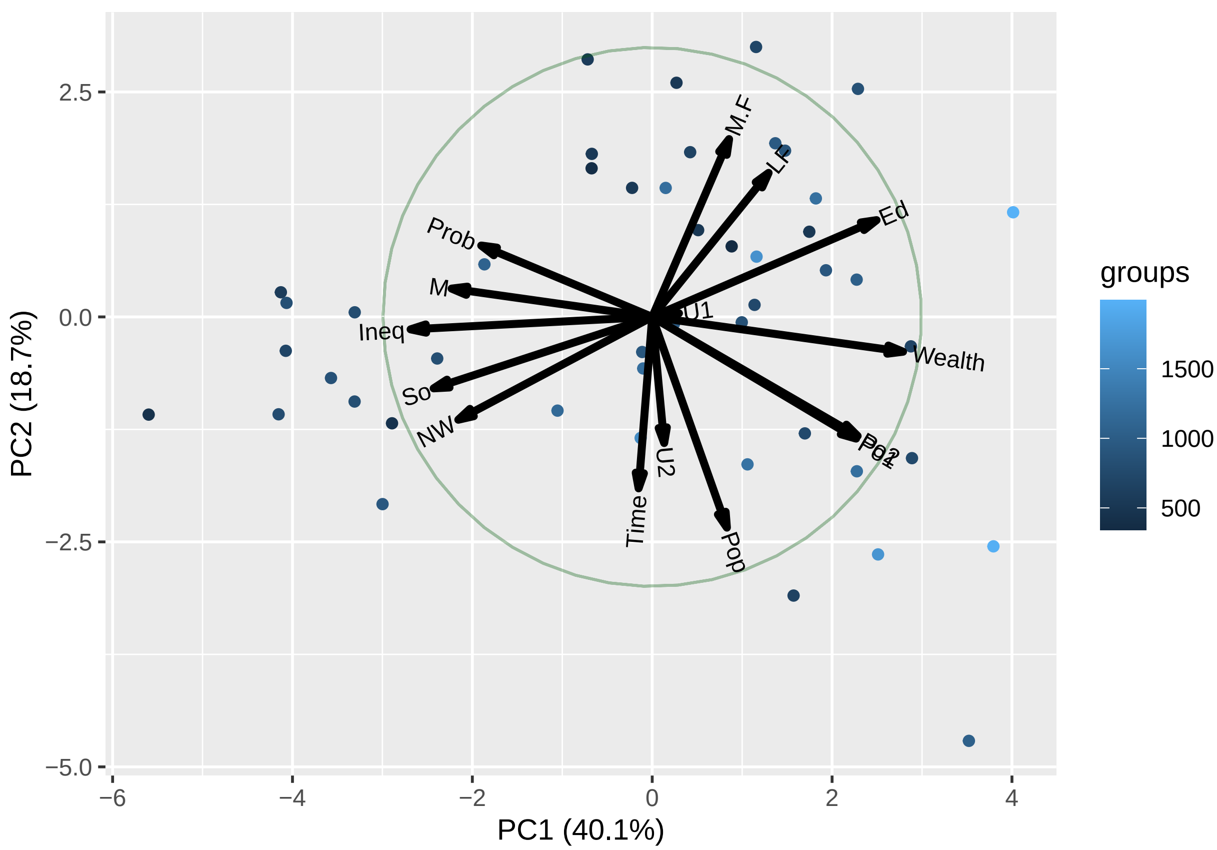
```r
summary(PCA)
```

```
## 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
```

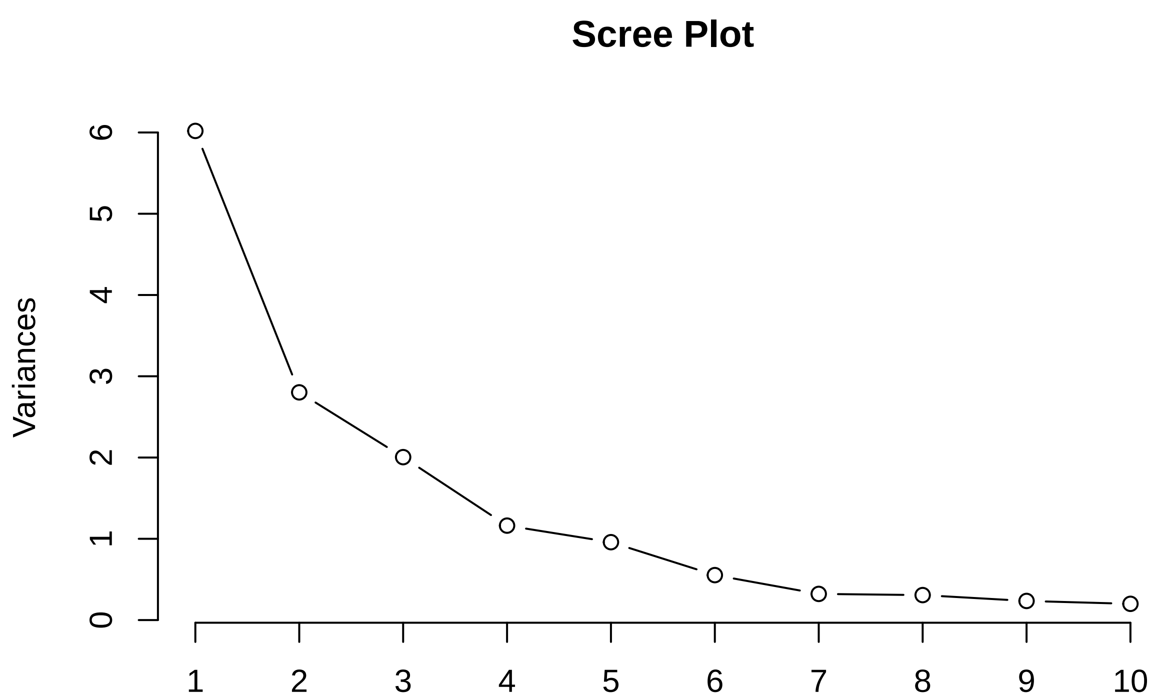
```
#From the summary of PCA, the cummulative variance PC 1 - 7 = 92.
#If we need to cover up to 90% of the variablility in the data
#Let's consider PC up to 7.

#Let's visualize the principal components using Bi-Plot.
ggbiplot(PCA,
       obs.scale = 1,
       var.axes = TRUE,
       var.scale = 1,
      groups = crime_data$Crime,
       circle = TRUE)
```




```
#The closer the vectors the closer the correlation between them.
#The plot shows NW, So, Ineq, M and Prob have positive correlation with
#PC1 as they are on the right side of 0.
#Therefore, those vectors on the PC1 axis have positive contribution on PC1
```

```r
#The scree plot visualizes the number of PC to use
screeplot(PCA, main = "Scree Plot", type = c("lines"))
```




```r
#Now will build a regression model with 7 components
new_crime_data <- as.data.frame(cbind(PCA$x[,1:7], crime_data$Crime))

#Linear regression model with 7 components
PCA_LM <- lm(V8~., data = new_crime_data)

summary(PCA_LM)
```

```
##
## Call:
## lm(formula = V8 ~ ., data = new_crime_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -475.41 -141.65   34.73  137.25  412.32
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   905.09      34.21  26.454  < 2e-16 ***
## PC1            65.22      14.10   4.626 4.04e-05 ***
## PC2           -70.08      20.66  -3.392   0.0016 **
```

```
## PC3             25.19      24.42   1.032   0.3086
## PC4             69.45      32.08   2.165   0.0366 *
## PC5           -229.04      35.33  -6.483 1.11e-07 ***
## PC6            -60.21      46.50  -1.295   0.2029
## PC7            117.26      60.96   1.923   0.0617 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 234.6 on 39 degrees of freedom
## Multiple R-squared:  0.6882, Adjusted R-squared:  0.6322
## F-statistic:  12.3 on 7 and 39 DF,  p-value: 3.513e-08
```

```r
#The new data frame shows low R2 and adjusted R2 values as comapred to the last assignment.

#Model coefficients
PCA_LM_coefficients <- PCA$rotation[,1:7]%*%PCA_LM$coefficients[-1]

#Converting standardized coefficients and intercept back into original variables
SD <- sapply(crime_data[,1:15], sd)
Mean <- sapply(crime_data[,1:15], mean)
intercept <- PCA_LM$coefficients[1]

alpha <- PCA_LM_coefficients/SD

beta <- intercept - sum(PCA_LM_coefficients*Mean/SD)

print(alpha)
```

```
##                 [,1]
## M       5.523735e+01
## So      1.397571e+02
## Ed     -6.803836e+00
## Po1     4.458638e+01
## Po2     4.642432e+01
## LF      6.733809e+02
## M.F     4.440293e+01
## Pop     9.599076e-01
## NW      5.684940e+00
## U1     -1.027735e+03
## U2      2.441589e+01
## Wealth  2.883565e-02
## Ineq    1.245113e+01
## Prob   -5.170569e+03
## Time   -2.215095e+00
```

```r
print(beta)
```

```
## (Intercept)
##   -5498.458
```

```r
#Prediction on the training data
pred_train <- as.matrix(X)%*%alpha + beta
```

```r
#Prediction on the test data from Q 8.2
pred_test <- as.matrix(test)%*%alpha + beta

pred_test
```

```
##          [,1]
## [1,] 1230.418
```

```r
#This calculates R2
R2 <- 1-sum((pred_train - crime_data$Crime)^2)/ sum((crime_data$Crime - mean(crime_data$Crime))^2)

print(R2)
```

```
## [1] 0.6881819
```

In conclusion:

Using PCA explores a data set to understand which observations in the data are most similar to each other. The main goal is to explain most of the variables in a data set with fewer variables than the original data set. In this assignment, we were asked to apply PCA and then create a regression model and compare its quality to the solution found in question 8.2. In this model, my R^2 is 69% but, in question 8.2 it was 77% and with the prediction being 1230 in this model and 1304 in the last question, it seems that this model is might be over-fitted considering my results were lower. After using 7 principal components in the model, the quality did not improve even with 92% in variance compared to my findings in question 8.2. I could continue to add more PC however, that defeats the purpose of reducing the amount of dimensions.