Homework6

N/A

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)!)

Just removed binary variables because PCA doesn't work well with binary variables

```
crime_data = read.table("uscrime.csv", header = TRUE)
crime1 <- crime_data[-2]
head(crime1)
```

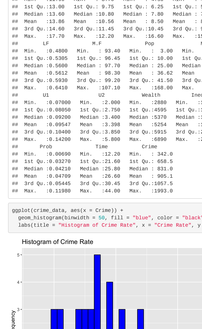
```
##       M   Ed  Po1  Po2    LF  M.F Pop   NW    U1  U2 Wealth Ineq     Prob
## 1 15.1 9.1 5.8  5.6 0.510  95.0  33 30.1 0.108 4.1  3940 26.1 0.084602
## 2 14.3 11.3 10.3  9.5 0.583 101.2  13 10.2 0.096 3.6  5570 19.4 0.029599
## 3 14.2  8.9  4.5  4.4 0.533  96.9  18 21.9 0.094 3.3  3180 25.0 0.083401
## 4 13.6 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 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 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
```

Simple Exploratory Data Analysis

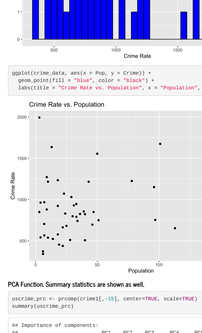
```
summary(crime1)
```

```
##        M               Ed              Po1             Po2       
##  Min.   :11.90   Min.   : 8.70   Min.   : 4.50   Min.   : 4.100  
##  1st Qu.:13.00   1st Qu.: 9.75   1st Qu.: 6.25   1st Qu.: 5.850  
##  Median :13.60   Median :10.80   Median : 7.80   Median : 7.300  
##  Mean   :13.86   Mean   :10.56   Mean   : 8.50   Mean   : 8.023  
##  3rd Qu.:14.60   3rd Qu.:11.45   3rd Qu.:10.45   3rd Qu.: 9.700  
##  Max.   :17.70   Max.   :12.20   Max.   :16.60   Max.   :15.700  
##        LF              M.F             Pop              NW       
##  Min.   :0.4800   Min.   : 93.40   Min.   :  3.00   Min.   : 0.20  
##  1st Qu.:0.5305   1st Qu.: 96.45   1st Qu.: 10.00   1st Qu.: 2.40  
##  Median :0.5600   Median : 97.70   Median : 25.00   Median : 7.60  
##  Mean   :0.5612   Mean   : 98.30   Mean   : 36.62   Mean   :10.11  
##  3rd Qu.:0.5930   3rd Qu.: 99.20   3rd Qu.: 41.50   3rd Qu.:13.25  
##  Max.   :0.6410   Max.   :107.10   Max.   :168.00   Max.   :42.30  
##        U1              U2             Wealth           Ineq      
##  Min.   :0.07000   Min.   : 2.000   Min.   : 2880   Min.   :12.60  
##  1st Qu.:0.08050   1st Qu.: 2.750   1st Qu.:4595   1st Qu.:16.55  
##  Median :0.09200   Median : 3.400   Median :5370   Median :17.60  
##  Mean   :0.09547   Mean   : 3.398   Mean   :5254   Mean   :19.40  
##  3rd Qu.:0.10400   3rd Qu.: 3.850   3rd Qu.:5915   3rd Qu.:22.75  
##  Max.   :0.14200   Max.   : 5.800   Max.   :6890   Max.   :27.60  
##       Prob            Time            Crime       
##  Min.   :0.00690   Min.   :12.20   Min.   : 342.0  
##  1st Qu.:0.03270   1st Qu.:21.60   1st Qu.: 658.5  
##  Median :0.04210   Median :25.80   Median : 831.0  
##  Mean   :0.04709   Mean   :26.60   Mean   : 905.1  
##  3rd Qu.:0.05445   3rd Qu.:30.45   3rd Qu.:1057.5  
##  Max.   :0.11980   Max.   :44.00   Max.   :1993.0
```

```
ggplot(crime_data, aes(x = Crime)) +
  geom_histogram(binwidth = 50, fill = "blue", color = "black") +
  labs(title = "Histogram of Crime Rate", x = "Crime Rate", y = "Frequency")
```




```
ggplot(crime_data, aes(x = Pop, y = Crime)) +
  geom_point(fill = "blue", color = "black") +
  labs(title = "Crime Rate vs. Population", x = "Population", y = "Crime Rate")
```



PCA Function. Summary statistics are shown as well.

```
uscrime_prc <- prcomp(crime1[,-15], center=TRUE, scale=TRUE)
summary(uscrime_prc)
```

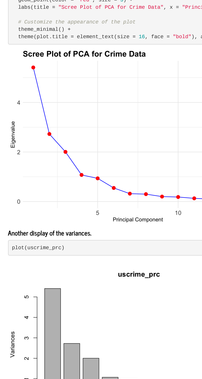
```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6     PC7
## Standard deviation     2.3252 1.6513 1.4158 1.03670 0.96745 0.74049 0.56415
## Proportion of Variance 0.3865 0.1948 0.1432 0.07677 0.06685 0.03917 0.02273
## Cumulative Proportion  0.3865 0.5813 0.7244 0.80121 0.86806 0.90723 0.92996
##                           PC8     PC9     PC10    PC11    PC12    PC13    PC14
## Standard deviation     0.54675 0.4475 0.42747 0.39445 0.31852 0.25159 0.06802
## Proportion of Variance 0.02135 0.0143 0.01305 0.00923 0.00725 0.00452 0.00033
## Cumulative Proportion  0.95132 0.9656 0.97867 0.98790 0.99515 0.99967 1.00000
```

```
eigenvalues <- uscrime_prc$sdev^2
pc_numbers <- seq_along(eigenvalues)
scree_data <- data.frame(PC = pc_numbers, Eigenvalue = eigenvalues)
```

Create a scree plot using ggplot2

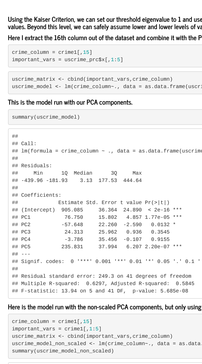
```
ggplot(data = scree_data, aes(x = PC, y = Eigenvalue)) +
  geom_line(color = "blue") +
  geom_point(color = "red", size = 3) +
  labs(title = "Scree Plot of PCA for Crime Data", x = "Principal Component", y = "Eigenvalue") +

  # Customize the appearance of the plot
  theme_minimal() +
  theme(plot.title = element_text(size = 16, face = "bold"), axis.text = element_text(size = 12))
```



Another display of the variances.

```
plot(uscrime_prc)
```



Using the Kaiser Criterion, we can set our threshold eigenvalue to 1 and use all of the points above that. Therefore we have 5 values. Beyond this level, we can safely assume lower and lower levels of variance are being explained.

Here I extract the 16th column out of the dataset and combine it with the PCA components.

```
crime_column = crime1[,15]
important_vars = uscrime_prc$x[,1:5]
```

```
uscrime_matrix <- cbind(important_vars,crime_column)
uscrime_model <- lm(crime_column~., data = as.data.frame(uscrime_matrix))
```

This is the model run with our PCA components.

```
summary(uscrime_model)
```

```
##
## Call:
## lm(formula = crime_column ~ ., data = as.data.frame(uscrime_matrix))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -439.96 -149.91    3.13  177.53  444.64
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  905.085     36.364  24.890  < 2e-16 ***
## PC1           76.750     15.802   4.857 1.77e-05 ***
## PC2          -57.648     22.260  -2.590   0.0132 *
## PC3           24.313     25.902   0.938   0.3545
## PC4           -3.786     35.456  -0.107   0.9155
## PC5          235.831     37.994   6.207 2.20e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 249.3 on 41 degrees of freedom
## Multiple R-squared:  0.6297,    Adjusted R-squared:  0.5845
## F-statistic: 13.94 on 5 and 41 DF,  p-value: 5.685e-08
```

Here is the model run with the non-scaled PCA components, but only using columns 1-5(original values).

```
crime_column = crime1[,15]
important_vars = crime1[,1:5]
uscrime_matrix <- cbind(important_vars,crime_column)
uscrime_model_non_scaled <- lm(crime_column~., data = as.data.frame(uscrime_matrix))
summary(uscrime_model_non_scaled)
```

```
##
## Call:
## lm(formula = crime_column ~ ., data = as.data.frame(uscrime_matrix))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -535.24 -185.81   18.17  153.44  576.49
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2426.93     883.90  -2.746  0.00892 **
## M             118.11      38.61   3.059  0.00390 **
## Ed            37.63       54.78   0.687  0.49596
## Po1          242.59      120.05   2.021  0.04987 *
## Po2         -145.34      129.95  -1.118  0.26988
## LF           716.15     1241.25   0.577  0.56712
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 260.4 on 41 degrees of freedom
## Multiple R-squared:  0.5961,    Adjusted R-squared:  0.5499
## F-statistic:  12.1 on 5 and 41 DF,  p-value: 3.12e-07
```

With the non-scaled data, I end up with an r-squared of around 0.59.

Now I run some predictions with our new PCA data from last weeks data.

```
new_data <- data.frame(
  M = 14.0,
  So = 0,
  Ed = 10.0,
  Po1 = 12.0,
  Po2 = 10.5,
  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
)
```

Apply the PCA data to the test data and then use the model to predict the new crime rate

```
predict1 <- data.frame(predict(uscrime_prc, new_data))
predict2 <- predict(uscrime_model, predict1)
predict2
```

```
##        1
## 1443.039
```

Last week my crime rate prediction was 89723, with using linear regression. With PCA, the predicted crime rate comes out to be 1443.039. That means that there was a sizeable difference and applying PCA to this small dataset probably caused the data to be overfitted. With so few data points, linear regression may be the best bet.

Now we can compare our R^2 values for some further analysis.

```
coeffs <- coef(uscrime_model)
intercept <- coeffs[1]
beta_vector <- coeffs[2:6]

# Calculate alpha vector
alpha_vector <- uscrime_prc$rotation[, 1:5] %*% beta_vector
# Standardize the data
crime_data <- scale(crime1[, 1:14])
# Calculate predictions using matrix multiplication
estimates <- cbind(1, crime_data) %*% c(intercept, alpha_vector)
# Calculate R-squared
SSE <- sum((estimates - crime1[, 15])^2)
SStot <- sum((crime1[, 15] - mean(crime1[, 15]))^2)
R2 <- 1 - SSE / SStot
R2
```

```
## [1] 0.6296769
```

Based off some simple data manipulations and calculations, I obtained an R^2 value of 0.63 for this model. Last week my model was around the same, sitting at 0.64. Based strictly off the R-squared values, both models did not perform very well, most likely due to the small sample size of the data.