AnalyticsModeling_HW6

Fall 2024

Question 9.1

Using the crime data set, 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 and compare its quality to your solution to 8.2.

```r
# load data
crime_df <- read.table("uscrime.txt",  stringsAsFactors = FALSE, header = TRUE)
head(crime_df, 5)
```

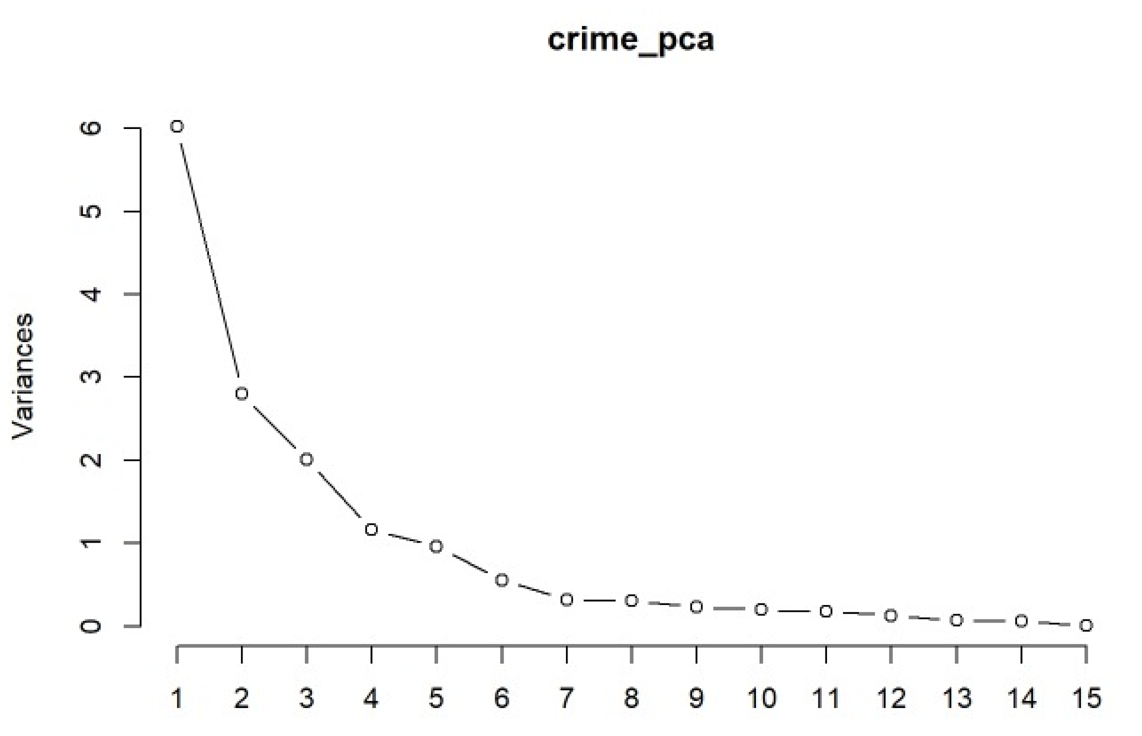
```
##      M So   Ed  Po1  Po2    LF   M.F Pop   NW    U1  U2 Wealth Ineq     Prob
## 1 15.1  1  9.1  5.8  5.6 0.510  95.0  33 30.1 0.108 4.1   3940 26.1 0.084602
## 2 14.3  0 11.3 10.3  9.5 0.583 101.2  13 10.2 0.096 3.6   5570 19.4 0.029599
## 3 14.2  1  8.9  4.5  4.4 0.533  96.9  18 21.9 0.094 3.3   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
##      Time Crime
## 1 26.2011   791
## 2 25.2999  1635
## 3 24.3006   578
## 4 29.9012  1969
## 5 21.2998  1234
```

First I did a PCA using all of the predictors from the crime data set. According to the analysis, the first six components explain 89.996% of the variance (with the first component by itself expaining 40.13%!). With less and less of the variance being explained by later components, I proceeded to do the regression analysis with the first six components.

```r
# PCA with all predictors
crime_pca <- prcomp(x = crime_df[,1:15]
                    , 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
# graph the principal components by how much of the variance they explain
screeplot(x = crime_pca
          , npcs = 15
          ,type = "lines")
```



The most positively and negatively correlated predictors for each of the six principal components are:

- PC1: `Wealth` | `Ineq`
- PC2: `M.F` | `Pop`
- PC3: `LF` | `U1`
- PC4: `Prob` | `Time`
- PC5: `Prob` | `M.F`
- PC6: `LF` | `M`

```r
print(crime_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
# combine the principal components with the crime data set
pca_crime_df <- as.data.frame(
                              cbind(crime_pca$x[,1:6]
                                    , crime_df[,16]
                                    )
                              )
head(pca_crime_df)
```

```
##         PC1        PC2         PC3         PC4         PC5        PC6   V7
## 1 -4.199284 -1.0938312 -1.11907395  0.67178115  0.05528338  0.3073383  791
## 2  1.172663  0.6770136 -0.05244634 -0.08350709 -1.17319982 -0.5832373 1635
## 3 -4.173725  0.2767750 -0.37107658  0.37793995  0.54134525  0.7187223  578
## 4  3.834962 -2.5769060  0.22793998  0.38262331 -1.64474650  0.7294884 1969
## 5  1.839300  1.3309856  1.27882805  0.71814305  0.04159032 -0.3940902 1234
## 6  2.907234 -0.3305421  0.53288181  1.22140635  1.37436096 -0.6922513  682
```

```r
# linear regression model with principal components
lm_pca <- lm(formula = V7~.
             , data = as.data.frame(pca_crime_df)
               )

summary(lm_pca)
```

```
##
## Call:
## lm(formula = V7 ~ ., data = as.data.frame(pca_crime_df))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -377.15 -172.23   25.81  132.10  480.38
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   905.09      35.35  25.604  < 2e-16 ***
## PC1            65.22      14.56   4.478 6.14e-05 ***
## PC2           -70.08      21.35  -3.283  0.00214 **
## PC3            25.19      25.23   0.998  0.32409
## PC4            69.45      33.14   2.095  0.04252 *
## PC5          -229.04      36.50  -6.275 1.94e-07 ***
## PC6           -60.21      48.04  -1.253  0.21734
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 242.3 on 40 degrees of freedom
## Multiple R-squared:  0.6586, Adjusted R-squared:  0.6074
## F-statistic: 12.86 on 6 and 40 DF,  p-value: 4.869e-08
```

PCA found the new dimension factors and regression found the coefficients of those factors, so I can interpret the new model in terms of the original factors by calculating the implied regression coefficient for the original factors. This is the sum of the coefficients multiplied by the eigenvectors of the transformed matrix.

```r
# calculate the implied regression coefficient
intercept <- lm_pca$coefficients[1]
b_vector <- lm_pca$coefficients[2:6]

# matrix multiply the coefficients and the eigenvectors of the transformed matrix of data
a_vector <- crime_pca$rotation[,1:5]%*%b_vector

# get the original data set's alpha vector and coefficient vector
mean <- sapply(crime_df[,1:15], mean)
sdv <- sapply(crime_df[,1:15], sd)
orig_b_vector <- intercept - sum(a_vector*mean/sdv)
orig_a_vector <- a_vector/sdv

# calculate the implied regression coefficient for the original predictors
implied_coefficients <- as.matrix(crime_df[,1:15]) %*% orig_a_vector+orig_b_vector

# calculate evaluation metrics
sse = sum((implied_coefficients- crime_df[,16])^2)
total_sse = sum((crime_df[,16] - mean(crime_df[,16]))^2)
rsquared <- 1 - sse/total_sse
adj_rsquared <- rsquared-(1-rsquared)*6/(nrow(crime_df)-6-1)
adj_rsquared
```

```
## [1] 0.5919732
```

The model created with Principal Components ended up with an Adjusted R-squared value of 0.592. I compared that model performance to last week's model, which had an Adjusted R-Squared value of 0.7307. It seems like PCA didn't work quite as well as just limiting the data set only to significant predictors, but that may be due to the small size of the data set. I expect PCA would work comparably well with a larger data set.

```r
# test city
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 = 0.04
                   , Time = 39.0
                   )

# train  new model with only significant attributes
new_lm_model <-  lm(formula = Crime ~ M+Ed+Po1+U2+Ineq+Prob
               , data = crime_df
               )
summary(new_lm_model)
```

```
##
## Call:
## lm(formula = Crime ~ M + Ed + Po1 + U2 + Ineq + Prob, data = crime_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -470.68  -78.41  -19.68  133.12  556.23
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5040.50     899.84  -5.602 1.72e-06 ***
## M             105.02      33.30   3.154  0.00305 **
## Ed            196.47      44.75   4.390 8.07e-05 ***
## Po1           115.02      13.75   8.363 2.56e-10 ***
## U2             89.37      40.91   2.185  0.03483 *
## Ineq           67.65      13.94   4.855 1.88e-05 ***
## Prob        -3801.84    1528.10  -2.488  0.01711 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 200.7 on 40 degrees of freedom
## Multiple R-squared:  0.7659, Adjusted R-squared:  0.7307
## F-statistic: 21.81 on 6 and 40 DF,  p-value: 3.418e-11
```

```r
# predict the crime rate for the city using the model trained on significant attributes
crime_preds_new_lm_model <- predict(new_lm_model, city)
crime_preds_new_lm_model
```

```
##        1
## 1304.245
```