DEMO 6 - PRINCIPAL COMPONENT ANALYSIS AND REDUCTION OF RESULTS TO FACTOR-BASED INTERPRETATION

PART 1 OBSERVATIONS - APPLYING PRINCIPAL COMPONENT ANALYSIS

The first brief study of predictors reveal the correlation between predictors as follows -

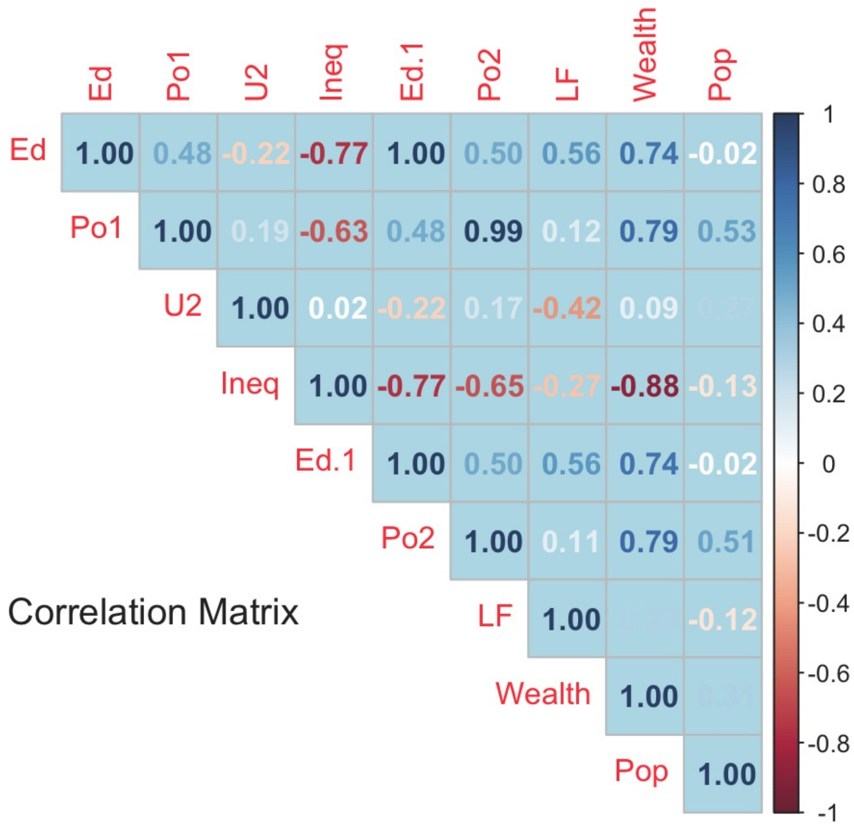


Fig1a: Correlation Matrix

From this we can observe that PO1 and PO2 had high positive correlation - as PO1 increases, so does PO2, and Ineq and Wealth have negative correlation - the inverse, as Wealth increases, Ineq decreases. And this graph is to give us a context when we evaluate our PCA algorithm.

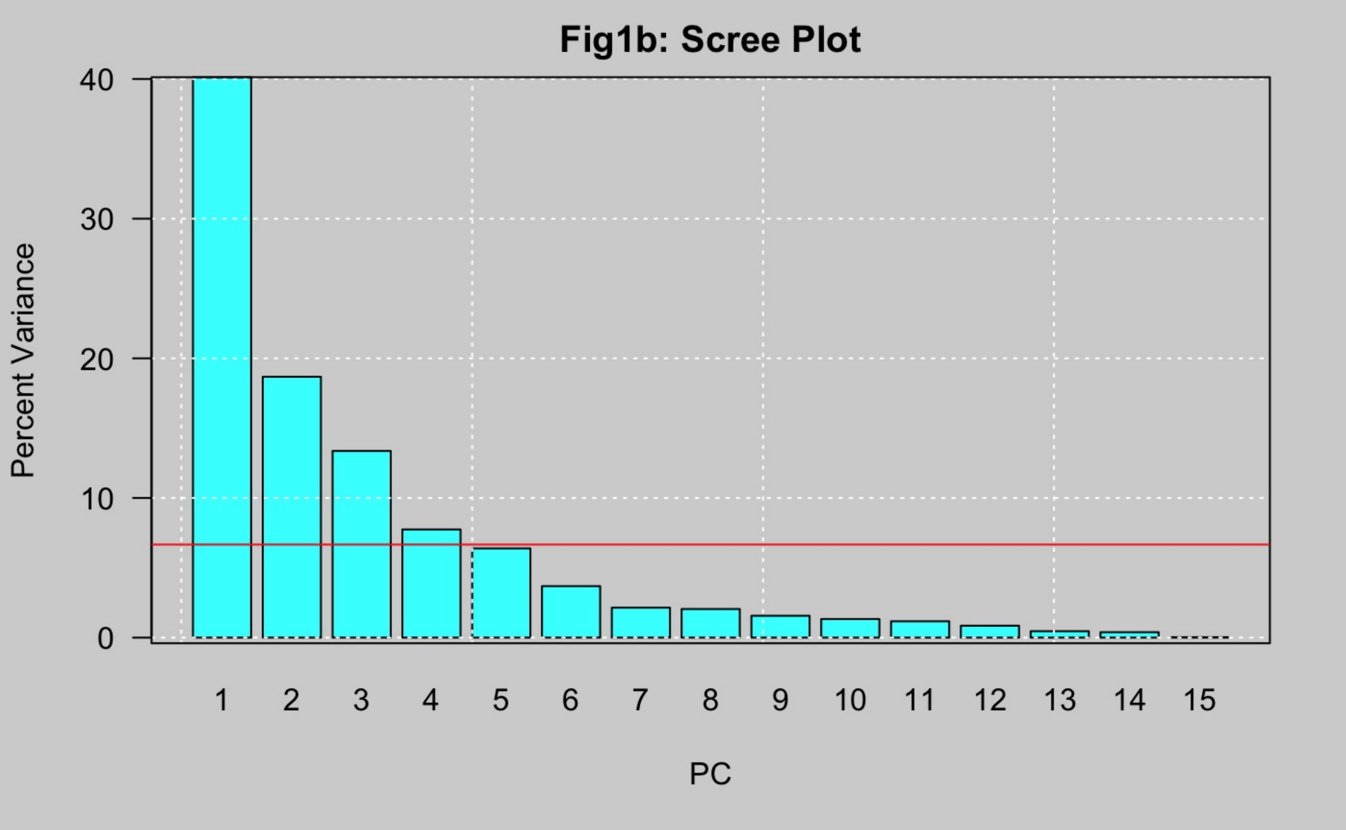
After centering and scaling the results of the PCA algorithm, we get -

Importance of components:

```
                          PC1    PC2    PC3     PC4     PC5     PC6     PC7     PC8     PC9
PC10
Standard deviation     2.4534 1.6739 1.4160 1.07806 0.97893 0.74377 0.56729 0.55444 0.48493
0.44708
Proportion of Variance 0.4013 0.1868 0.1337 0.07748 0.06389 0.03688 0.02145 0.02049 0.01568
0.01333
Cumulative Proportion  0.4013 0.5880 0.7217 0.79920 0.86308 0.89996 0.92142 0.94191 0.95759
0.97091

                          PC11    PC12    PC13   PC14    PC15
Standard deviation     0.41915 0.35804 0.26333 0.2418 0.06793
Proportion of Variance 0.01171 0.00855 0.00462 0.0039 0.00031
Cumulative Proportion  0.98263 0.99117 0.99579 0.9997 1.00000
```

We then use a scree plot to determine 'n', the number of largest variance covering principal components to be taken. By rule of thumb, we usually take upto 80% variance coverage. And the following graph tells us that we ought to take the first five PCs.



PART 2 OBSERVATIONS - MAPPING TO ORIGINAL VARIABLES

Now we need to interpret this in terms of original predictors -

```
         PC1   PC2   PC3   PC4   PC5
M       0.30 -0.06 -0.17  0.02  0.36
So      0.33  0.16 -0.02 -0.29  0.12
Ed     -0.34 -0.21 -0.07 -0.08  0.02
Po1    -0.31  0.27 -0.05 -0.33  0.24
Po2    -0.31  0.26 -0.05 -0.35  0.20
LF     -0.18 -0.32 -0.27  0.14  0.39
M.F    -0.12 -0.39  0.20 -0.01  0.58
Pop    -0.11  0.47 -0.08  0.03  0.08
NW      0.29  0.23 -0.08 -0.24  0.36
U1     -0.04 -0.01  0.66  0.18  0.13
U2     -0.02  0.28  0.58  0.07  0.13
Wealth -0.38  0.08 -0.01 -0.12 -0.01
Ineq    0.37  0.03  0.00  0.08  0.22
Prob    0.26 -0.16  0.12 -0.49 -0.17
Time    0.02  0.38 -0.22  0.54  0.15
```

As you can see, Ineq and Wealth contribute the maximum effect to PC1 (the most dominant PC of 40% variance coverage) and PO1 and PO2 follow next in line. By evaluating the signs, you can see that this is consistent with the correlation plot we gauged earlier.

PART 3 OBSERVATIONS - COMPARISON WITH PREVIOUS REGRESSION MODELS

Let us keep the results of the crime data -

1. Base Model (Without Any Change)
2. Model With Statistically Significant Predictor Set
3. Model Using First Five PCs

Results of Base Model (Without Any Change)

```
Call:
lm(formula = Crime ~ ., data = df)

Residuals:

    Min      1Q  Median      3Q     Max 
-395.74  -98.09   -6.69  112.99  512.67 
```

```
Coefficients:

              Estimate Std. Error t value Pr(>|t|)
(Intercept) -5.984e+03  1.628e+03  -3.675 0.000893
M            8.783e+01  4.171e+01   2.106 0.043443
So          -3.803e+00  1.488e+02  -0.026 0.979765
Ed           1.883e+02  6.209e+01   3.033 0.004861
Po1          1.928e+02  1.061e+02   1.817 0.078892
Po2         -1.094e+02  1.175e+02  -0.931 0.358830
LF          -6.638e+02  1.470e+03  -0.452 0.654654
M.F          1.741e+01  2.035e+01   0.855 0.398995
Pop         -7.330e-01  1.290e+00  -0.568 0.573845
NW           4.204e+00  6.481e+00   0.649 0.521279
U1          -5.827e+03  4.210e+03  -1.384 0.176238
U2           1.678e+02  8.234e+01   2.038 0.050161
Wealth       9.617e-02  1.037e-01   0.928 0.360754
Ineq         7.067e+01  2.272e+01   3.111 0.003983
Prob        -4.855e+03  2.272e+03  -2.137 0.040627

Time        -3.479e+00  7.165e+00  -0.486 0.630708

Residual standard error: 209.1 on 31 degrees of freedom
Multiple R-squared:  0.8031,    Adjusted R-squared:  0.7078
F-statistic: 8.429 on 15 and 31 DF,  p-value: 3.539e-07
"RMSE of Base Model = 209.06"
```

Results of Model With Statistically Significant Predictor Set

```
Call:
lm(formula = Crime ~ Ed + Po1 + Ineq, data = testData)

Residuals:

         Min       1Q   Median       3Q      Max
    -155.608  -89.161    8.519   93.773  158.841

Coefficients:

                Estimate Std. Error t value Pr(>|t|)
    (Intercept)   897.96      46.88  19.155 1.31e-06
    Ed            363.09      91.11   3.985  0.00724
    Po1           670.37     125.05   5.361  0.00173
    Ineq          592.25     101.69   5.824  0.00113

Residual standard error: 135.1 on 6 degrees of freedom
Multiple R-squared:  0.8602,    Adjusted R-squared:  0.7903
F-statistic: 12.31 on 3 and 6 DF,  p-value: 0.005655
"RMSE of Test Model = 135.07"
```

Results of Model Using First Five PCs

```
Call:
lm(formula = V6 ~ ., data = as.data.frame(uscrimePC))

Residuals:

       Min     1Q Median     3Q    Max
    -420.8 -185.0   12.2  146.2  447.9

Coefficients:

            Estimate Std. Error t value Pr(>|t|)
(Intercept)    905.1       35.6   25.43  < 2e-16
PC1             65.2       14.7    4.45  6.5e-05
PC2            -70.1       21.5   -3.26   0.0022
PC3             25.2       25.4    0.99   0.3272
PC4             69.4       33.4    2.08   0.0437
PC5           -229.0       36.8   -6.23  2.0e-07

Residual standard error: 244 on 41 degrees of freedom
Multiple R-squared:  0.645, Adjusted R-squared:  0.602
F-statistic: 14.9 on 5 and 41 DF,  p-value: 2.45e-08
```

After unscaling the data and using the coefficients and intercepts to evaluate the linear regression model in terms of the predictors, we get the following RMSE value for the PCA model -

```
"RMSE of PCA linear regression = 227.91"
```

PART 4 OBSERVATIONS - EVALUATION OF DATA INSTANCE

Given the test points as mentioned in the assignment, we can evaluate the instance -

```
"Crime Prediction using PCA linear regression = 1389"
```

ANSWER: 1398

Final Inferences -

The two major observations we can draw is that -

1. The PCA model did worse than the base model but that can be excused due to overfitting of the base model. The PCA model performed at almost the same level as the predictor pruning model in homework 8.2 in terms of RMSE.
2. It accounted for 80% of the variance in the data and was consistent with the multicollinearities known to be present. And then we unscaled and transformed back to the predictor form to compare the two models.

CODE FOR ALL FOUR STAGES

In [ ]:

```
require(ggthemes)
library(tidyverse)
library(magrittr)
library(TTR)
library(tidyr)
library(dplyr)
library(lubridate)
library(ggplot2)
library(plotly)
library(fpp2)
library(forecast)
library(caTools)
library(reshape2)
library(psych)
require(graphics)
require(Matrix)
library(corrplot)
library(mltools)
library(fBasics)
library(kableExtra)
library(DAAG)
library(caret)

#Correlation Plot
df <- read.table("uscrime.txt",stringsAsFactors = F, header=T)
crimepca <- prcomp(df[,c(1:15)], center = TRUE,scale. = TRUE)
headers = c("Ed","Po1","U2","Ineq","Ed","Po2","LF","Wealth","Pop")
newdata <- df[headers]
correlation = cor(newdata )
corrplot(correlation, method = 'number',type='upper',bg="lightblue")
mtext("Fig1a: Correlation Matrix", at=.95, line=-15, cex=1.2)

#Summary check and scaling
summary(crimepca)
crimepca$center
crimepca$scale
crimepca$rotation <- -crimepca$rotation #removing the default direction
crimepca$rotation

#Scree plot preparation
variance <- (crimepca$sdev)^2
loadings <- crimepca$rotation
rownames(loadings) <- colnames(df[,1:15])
scores <- crimepca$x
varPercent <- variance/sum(variance) * 100
par(bg = 'lightgrey')
barplot(varPercent, xlab='PC', ylab='Percent Variance',names.arg=1:length(varPercent), las=1, col='cyan', main="F
grid (lty = 3, col = "white")
box( col = 'black')
abline(h=1/ncol(df[,1:15])*100, col='red')
```

```
#Loadings
round(loadings, 2)[ , 1:5]

#Unscaling and Transforming Back
PC<-crimepca$x[,1:5]
uscrimePC<-cbind(PC,df[,16])
pca.model<- lm( V6 ~., data = as.data.frame(uscrimePC))
summary(pca.model)
print(sprintf("RMSE of PCA Model = %0.2f", sigma(pca.model) ))
Scaled.intercept <- pca.model$coefficients[1]
Scaled.intercept
Scaled.Coefficients <- pca.model$coefficients[2:6]
Scaled.Coefficients
a<-crimepca$rotation[,1:5]%*%Scaled.Coefficients
t(a)
Intercept.unscaled<- Scaled.intercept-sum(a*crimepca$center/crimepca$scale)
Intercept.unscaled
A.unscaled<- a/crimepca$scale
A.unscaled
y.pred<-Intercept.unscaled+as.matrix(df[,1:15])%*%A.unscaled

#Calculating Accuracy
rss <- sum((y.pred - df[,16]) ^ 2)  ## residual sum of squares
tss <- sum((df[,16] - mean(df[,16])) ^ 2)  ## total sum of squares
rsq <- 1 - rss/tss
print(sprintf("R-squared of PCA linear regression = %0.2f", rsq ))

#Calculate RMSE of New Model
rmse.pca<-sqrt(mean((df[,16]-y.pred)^2))
print(sprintf("RMSE of PCA linear regression = %0.2f", rmse.pca))


#Evaluating Test Instance
testpts <-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
pred.pca <- Intercept.unscaled+as.matrix(testpts)%*%A.unscaled
print(sprintf("Crime Prediction using PCA linear regression = %0.0f", pred.pca))
```

THE END---

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js