ISYE6501

Homework 6

9.1]

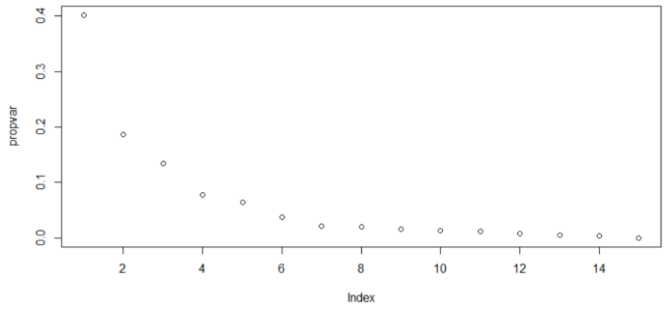
The goal for this assignment was to predict using a regression model for the key predictors that are deemed the most important based on the Principal Control Analysis method. We will then compare it to last weeks model to see which model best fits our use case.

To get started, I needed to choose the number of leading predictors to be considered in the PCA based regression model. To choose that, the principal components of each predictor were calculated using the prcomp function. The key note to be made is that the original dataset was scaled in this process, this will be critical to remember for our predictor model. The summary of our principal components for each predictor are given in Fig[1]

```
Importance of components:
                          PC1    PC2    PC3     PC4     PC5     PC6     PC7     PC8
Standard deviation     2.4534 1.6739 1.4160 1.07806 0.97893 0.74377 0.56729 0.55444
Proportion of Variance 0.4013 0.1868 0.1337 0.07748 0.06389 0.03688 0.02145 0.02049
Cumulative Proportion  0.4013 0.5880 0.7217 0.79920 0.86308 0.89996 0.92142 0.94191
                          PC9    PC10    PC11    PC12    PC13   PC14    PC15
Standard deviation     0.48493 0.44708 0.41915 0.35804 0.26333 0.2418 0.06793
Proportion of Variance 0.01568 0.01333 0.01171 0.00855 0.00462 0.0039 0.00031
Cumulative Proportion  0.95759 0.97091 0.98263 0.99117 0.99579 0.9997 1.00000
```

Fig[1]

From the summary it seems like the first 3 principal components cover roughly 72% variance in the scaled dataset. We could proceed based on the total variance we have since its covers roughly 3/4th of our scaled range, however I decided to do a visual check to see where an addition of new principal component does not result in a justifiable amount of increased proportional variance to limit the inclusion of only the key parameters. Fig[2] shows the result for the same.



Fig[2]

Based on the results from Fig[2], I concluded to include only first 5 Principal components. Using these principal components a new data frame was created with the first five principal components and their respective response values. This new data set was then used to create a regression model. Fig [3] Shows the summary of this regression model

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

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
```

Fig[3]

The model has an R squared value of 64.52% and an adjusted R squared value on 60.19%. The key to note here is that this regression model fits the scaled data from the original dataset. In order to predict the new city crime rate, we will need to convert the coefficients to fit the unscaled (original) dataset's equivalent regression model.

The standard form of this PCA regression model can be written as Fig[4]. To find the coefficients of each predictor in the original model we can then use Fig[5].

$$\begin{aligned} y_i &= b_0 + \sum_{k=1}^{L} b_k t_{ik} \\ &= b_0 + \sum_{k=1}^{L} b_k \left[\sum_{j=1}^{m} x_{ij} v_{jk}\right] \\ &= b_0 + \sum_{j=1}^{m} x_{ij} \left[\sum_{k=1}^{L} b_k v_{jk}\right] \\ &= b_0 + \sum_{j=1}^{m} x_{ij} [a_j] \end{aligned}$$

Fig[4]

Implied regression coefficient for x_j

$$a_j = \sum_{k=1}^{L} b_k v_{jk}$$

Fig[5]

I first found the eigen vector of the PCA applied dataset by means of rotation and then calculated the beta0 and sum of remaining beta values (1-5) from the PCA regression models coefficient. This information was then used to calculate the a(j) (Coefficient of regression model to original dataset) = alpha values and then transposed to shows the individual coefficients (a(j)) for each parameter as shown in Fig[6]

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

Fig[6]

We have to recollect, the coefficients we currently have are still for a scaled original dataset. While scaling each predictor column variable in the original dataset, the prcomp function subtracts the mean of all predictor values and divides by the predictor columns standard deviation. In order to unscale these coefficients to fit the unscaled original dataset we reversed the calculation and found the new coefficients and intercept as shows in Fig[7]

```
      M       So      Ed      Po1      Po2       LF      M.F      Pop       NW       U1       U2     Wealth     Ineq      Prob     Time
[1,] 48.37374 79.01922 17.8312 39.48484 39.85892 1886.946 36.69366 1.546583 9.537384 159.0115 38.29933 0.03724014 5.540321 -1523.521 3.838779
> og_beta_0
(Intercept)
  -5933.837
```

Fig[7]

A regression model was then created rearranging the found intercept and coefficient in the standard form of the regression model. This was used to predict the values for the original unscaled dataset and the corresponding R squared and adjusted R squared calculations were made to compare with the older model. As expected they were 64.52% and 60.19% as shows in Fig[8] which confirmed that the regression model was successfully transformed back to fit the original unscaled dataset.

```
> 1-SSE/SStot
[1] 0.6451941
>
> R2=1-SSE/SStot
> R2-(1-R2)*5/(nrow(crime_data)-5-1)
[1] 0.601925
```

Fig[8]

This new model was then used to predict the same dataset from last weeks homework which predicted a value of 1388.926, as shown in Fig[9],compared to last weeks model predicting 1304.245 as shown in Fig[10]

```
> predicted_model_fit
         [,1]
[1,] 1388.926
```

Fig[9]

When comparing the adjusted R squared values of this weeks model of 60.19% to last weeks value of 73.07%. It seems that a PCA adjusted approach is not as good as performing a standard linear regression on all 15 predictors. Fig[9] shows details from last week's homework

```
Call:
lm(formula = Crime ~ M + Ed + Po1 + U2 + Ineq + Prob, data = crime_table)

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
```

```
> model_3_fit
       1
1304.245
```

Fig[8]

R Code for Reference

```r
rm(list=ls())
set.seed(100)

#Read Crime data to R
crime_data=read.table("M:/OMSA/ISYE6501/HW5/crime.txt",header=TRUE)
head(crime_data)

#Computing Principal Components
crime_data.pca=prcomp(crime_data[,-16],center=T,scale=T)
summary(crime_data.pca)

#Finding optimal number of N for PCA
#calculate proportional variance
var=crime_data.pca$sdev^2
propvar=var/sum(var)
plot(propvar)

#Using 5 as N for PCA (Choosing firt 5 PC values) we will run a PCA model
#store new dataframe
transformed_crime=cbind(crime_data.pca$x[,1:5],crime_data[,16])
head(transformed_crime)

#build regression model
model=lm(V6~.,data=as.data.frame(transformed_crime))
summary(model)


#Converting PCA model coefficients to original data
beta_0=model$coefficients[1]
sum_betas=model$coefficients[2:6]
sum_betas

   #Find eigen vector of PCA
crime_data.pca$rotation[,1:5]

alphas=crime_data.pca$rotation[,1:5] %*% sum_betas
t(alphas)

#Converting from scaled to unscaled data coefficients
og_alpha=alphas/sapply(crime_data[,1:15],sd)
og_beta_0=beta_0-sum(alphas*sapply(crime_data[,1:15],mean)/sapply(crime_data[,1:15],sd))
t(og_alpha)
og_beta_0
```

```r

#Regression model for original coefficients is as follows
predict_model_og=as.matrix(crime_data[,1:15])%*%og_alpha+og_beta_0
predict_model_og

#Calculate R2 and adjusted R2 value to compare if model predicts with same efficiency post conversion
SSE=sum((predict_model_og-crime_data[,16])^2)
SStot=sum((crime_data[,16]-mean(crime_data[,16]))^2)
1-SSE/SStot

R2=1-SSE/SStot
R2-(1-R2)*5/(nrow(crime_data)-5-1)

#using new model to predct last weeks new city data
new_city_data=data.frame(M = 14.00,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)
predicted_model_fit=as.matrix(new_city_data[,1:15])%*%og_alpha+og_beta_0
predicted_model_fit
```