Principal Component Analysis

Kent Ng

June 12, 2018

Leveraging the code from the Linear Regression project, let's once again build a linear model using the crime data set (9.1uscrimeSummer2018.txt). However, this time we will first apply the Principal Component Analysis and will only use the first few principal components.

First, let's set seed value as best practice. We will also load the given data into a dataframe

```
set.seed(42)
data_9.1<-read.table("9.1uscrimeSummer2018.txt",header=TRUE)
head(data_9.1)
```

```
##      M So   Ed  Po1  Po2    LF   M.F Pop   NW    U1  U2 Wealth Ineq
## 1 15.1  1  9.1  5.8  5.6 0.510  95.0  33 30.1 0.108 4.1   3940 26.1
## 2 14.3  0 11.3 10.3  9.5 0.583 101.2  13 10.2 0.096 3.6   5570 19.4
## 3 14.2  1  8.9  4.5  4.4 0.533  96.9  18 21.9 0.094 3.3   3180 25.0
## 4 13.6  0 12.1 14.9 14.1 0.577  99.4 157  8.0 0.102 3.9   6730 16.7
## 5 14.1  0 12.1 10.9 10.1 0.591  98.5  18  3.0 0.091 2.0   5780 17.4
## 6 12.1  0 11.0 11.8 11.5 0.547  96.4  25  4.4 0.084 2.9   6890 12.6
##       Prob    Time Crime
## 1 0.084602 26.2011   791
## 2 0.029599 25.2999  1635
## 3 0.083401 24.3006   578
## 4 0.015801 29.9012  1969
## 5 0.041399 21.2998  1234
## 6 0.034201 20.9995   682
```

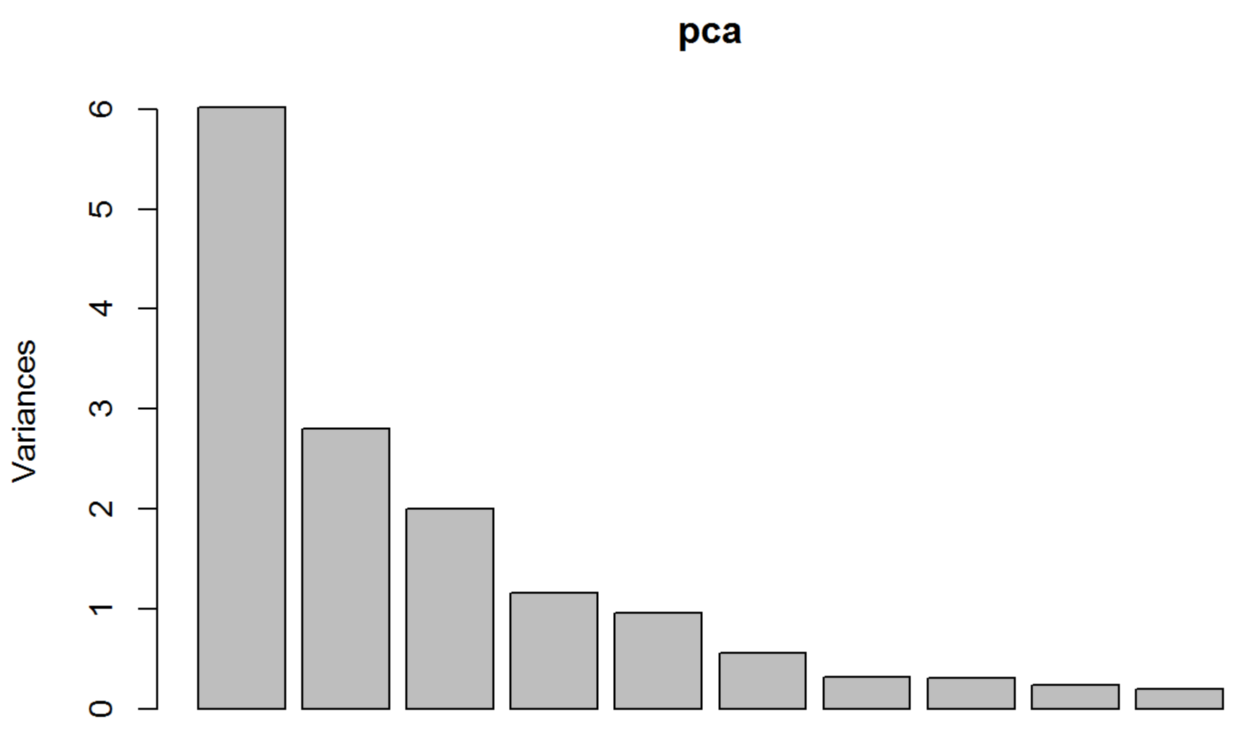
Let's leverage the prcomp() function to conduct principal component analysis (PCA) on our data

```
pca<-prcomp(data_9.1[,1:15],scale=TRUE)
summary(pca)
```

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

We can further visualize our results by plotting our variances

```
plot(pca)
```



From the table and graph above, it appears that the first 4 components accounts for 86% of the variance in our data. Let's extract the first 4 components to be used in our linear model below.

```
PCA_data<-data.frame(cbind(pca$x[,1:4],data_9.1$Crime))
names(PCA_data)<-c('P1','P2','P3','P4','Crime')
```

Now let's use our PCA components to build the linear model.

```
lm<-lm(Crime~.,PCA_data)
summary(lm)
```

```
##
## Call:
## lm(formula = Crime ~ ., data = PCA_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -557.76 -210.91  -29.08  197.26  810.35
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   905.09      49.07  18.443  < 2e-16 ***
## P1             65.22      20.22   3.225  0.00244 **
## P2            -70.08      29.63  -2.365  0.02273 *
## P3             25.19      35.03   0.719  0.47602
## P4             69.45      46.01   1.509  0.13872
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 336.4 on 42 degrees of freedom
## Multiple R-squared:  0.3091, Adjusted R-squared:  0.2433
## F-statistic: 4.698 on 4 and 42 DF,  p-value: 0.003178
```

We now need to convert our model's coefficients

```
coefficients_conv<-(pca$rotation[,1:4] %*%lm$coefficients[2:5])/pca$scale
```

We will also have to adjust our intercept based on the PCA center

```
int<-lm$coefficient[1]-sum(coefficients_conv*pca$center)
```

Let's now use the data point for the new city to once again predict the crime rate.

```
point<-data.frame(M=14.0,So = 0,Ed = 10.0,Po1 = 12.0,Po2 = 15.5, LF = 0.640, M.F = 94.0, Pop = 1
50, NW = 1.1, U1 = 0.120, U2 = 3.6, Wealth = 3200, Ineq = 20.1, Prob = 0.04, Time = 39.0)
```

We can now calculate the crime rate based on the converted intercept and coefficients.

```
crime <- sum(coefficients_conv[1,1] %*% point$M,coefficients_conv[2,1] %*% point$So,coefficients
_conv[3,1] %*% point$Ed,coefficients_conv[4,1] %*% point$Po1,coefficients_conv[5,1] %*% point$Po
2,coefficients_conv[6,1] %*% point$LF,coefficients_conv[7,1] %*% point$M.F,coefficients_conv[8,1
] %*% point$Pop,coefficients_conv[9,1] %*% point$NW,coefficients_conv[10,1] %*% point$U1,coeffic
ients_conv[11,1] %*% point$U2,coefficients_conv[12,1] %*% point$Wealth,coefficients_conv[13,1] %
*% point$Ineq,coefficients_conv[14,1] %*% point$Prob,coefficients_conv[15,1] %*% point$Time,int)

crime
```

```
## [1] 1112.678
```

My linear regression model from the Linear Regression project predicted 1304 as the crime rate for the new city. Here, we can see that our linear model using PCA predicts 1112.678 as the city's crime rate. Looking at the summary of the model (below), the R squared value (0.3091) for this model is actually much worse compared to that of the model in 8.2 (which was 0.69).

```
summary(lm)
```

```
##
## Call:
## lm(formula = Crime ~ ., data = PCA_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -557.76 -210.91  -29.08  197.26  810.35
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   905.09      49.07  18.443  < 2e-16 ***
## P1             65.22      20.22   3.225  0.00244 **
## P2            -70.08      29.63  -2.365  0.02273 *
## P3             25.19      35.03   0.719  0.47602
## P4             69.45      46.01   1.509  0.13872
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 336.4 on 42 degrees of freedom
## Multiple R-squared:  0.3091, Adjusted R-squared:  0.2433
## F-statistic: 4.698 on 4 and 42 DF,  p-value: 0.003178
```