**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)!)**

**Answer:**

Given below are the steps to apply principal component analysis and perform linear regression and create a model in the USCrime data.

**Step 1: Load the data into a data frame and install the required library**

**CODE:**

*library(data.table)*

*library(DAAG)*

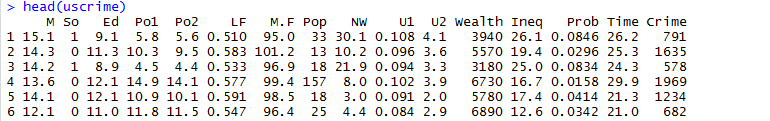
*rm(list = ls())*

*set.seed(1)*

*uscrime = read.table("C://Users/D100793/Desktop/Junk/Georgia Tech/uscrime.txt", header = TRUE,sep = '\t')*

*head(uscrime)*

**OUTPUT:**



**Step 2: Generating the principal component of the data and analyzing the principal components summary**

**CODE:**

*##Applying Principal Component Analyis*

*uscrimescaledata <- prcomp(uscrime[,-16], center=T, scale=T)*

*##Checking the principal Components*

*summary(uscrimescaledata)*

*## Plotting Standard Deviation for each principal component for analysis*

*plot(uscrimescaledata$sdev, xlab = "Principal Component", ylab = " Standard Deviation",ylim = c(0,5) , type= "b")*

*##Calculating Variance of the components*

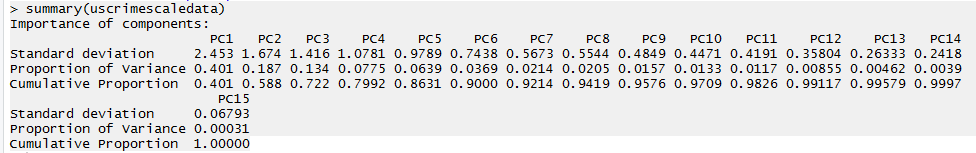
*variance = uscrimescaledata$sdev^2*

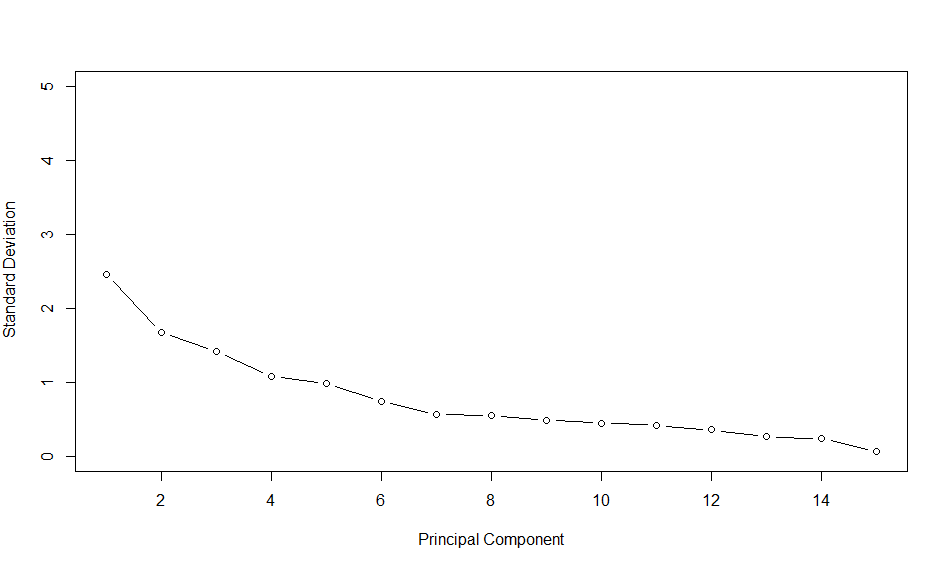
*varianceproportion = variance/sum(variance)*

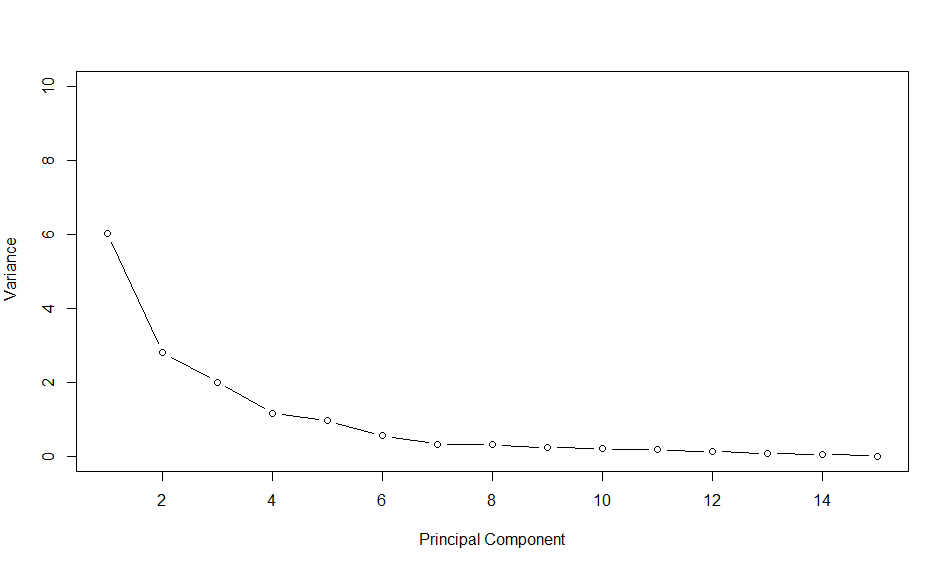
*##Plotting Variance for each principal component for analysis*

*plot(variance, xlab = "Principal Component", ylab = " Variance",ylim = c(0,10) , type= "b")*

**OUTPUT:**







**Analysis:** As we look at the summary and plots above, we see that first 4 PCs (PC1, PC2 , PC3 and PC4) have significantly higher amount of variation as compared to the other PCs. The Standard Deviation of the first 4 PCs is above 1 and the 1st three PCs are relatively more higher that the 4th one. When we calculate the variance and plot it is clearly visible in the second plot above that 1st 3 components have higher variances as compared to the other principal components.

But for our model, in order to judge which principal component is important we need to see if choose models whose variances are greater than 1 (as per Kaiser Method).

Given below are the variances of the PCs

[1] 6.01895 2.80185 2.00494 1.16221 0.95830 0.55319 0.32182 0.30740 0.23516 0.19988 0.17569 0.12819 0.06934

[14] 0.05847 0.00461

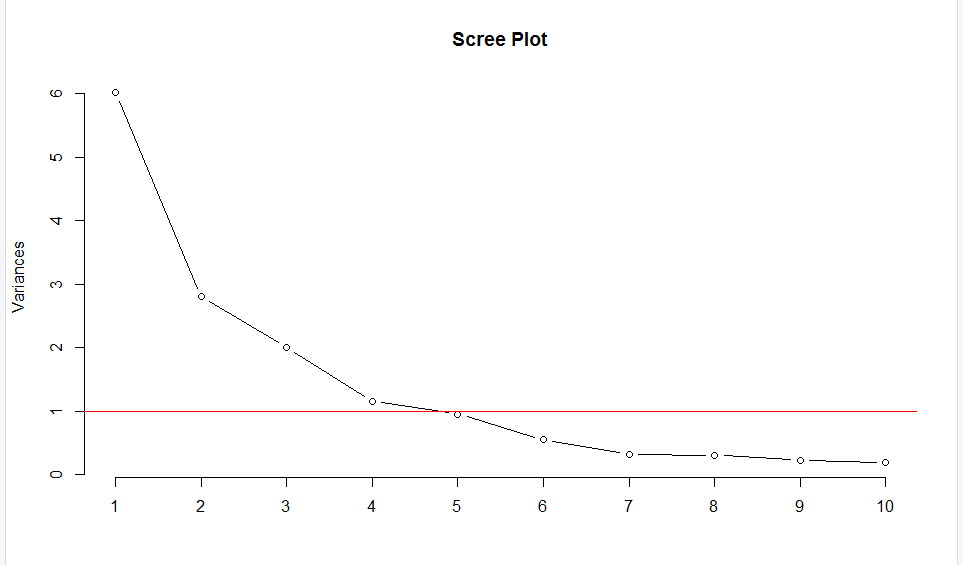
Plotting the variances using screeplot

**CODE:**

*screeplot(uscrimescaledata,main = "Scree Plot", type = "line")*

*abline(h=1, col="red")*

**OUTPUT:**



Looking at the variances, it appears that first 4 Principal Components have variances greater that and the 5th PC has a variance very near to 1. To conclude whether to choose the first 5 PCs let calculate the R-Squared of the PCA model using PC’s from 1 to 15. Using for loop we will models by adding each component at a time and save its R-Squared in a list.

**CODE:**

*for (i in 1:15){*

*PrincipalComponentList <- uscrimescaledata$x[,1:i]*

*##adding response to the PSC data*

*pcc <- cbind(uscrime[,16],PrincipalComponentList)*

*#calculate R-squared, store in list*

*PCModel <- lm(V1~., data = as.data.frame(pcc))*

*r\_squared[i] <- 1 -sum(PCModel$residuals^2)/sum((uscrime$Crime - mean(uscrime$Crime))^2)*

*#calculate cross validated R-squared, store in list*

*par(mfrow = c(3,5))*

*PCModelCV <- cv.lm(as.data.frame(pcc), PCModel, m = 5, plotit = TRUE, printit = FALSE)*

*r\_squaredCV[i] <- 1 - attr(PCModelCV,"ms")\*nrow(uscrime) / sum((uscrime$Crime - mean(uscrime$Crime))^2)*

*}*

*Rsquaretable <- data.frame(r\_squared, r\_squaredCV))*

*Rsquaretable*

**OUTPUT:**

> PCs\_Used <- c(1:length(r\_squared))

> Rsquaretable <- data.frame(r\_squared, r\_squaredCV,PCs\_Used)

> Rsquaretable

r\_squared r\_squaredCV PCs\_Used

1 0.171 0.0735 1

2 0.263 0.0910 2

3 0.272 0.0666 3

4 0.309 0.1057 4

5 0.645 0.4872 5

6 0.659 0.4628 6

7 0.688 0.4562 7

8 0.690 0.3664 8

9 0.692 0.3337 9

10 0.696 0.2954 10

11 0.697 0.1863 11

12 0.769 0.3897 12

13 0.772 0.3902 13

14 0.791 0.4736 14

15 0.803 0.4134 15

**Analysis:** Looking at the table of R-Squared and Cross Validated R-Squared above, we see that Model with PCs 1-5 has the Highest Cross Validated R-Squared value followed by model with PCs 1-14 and then PCs 1-6. All the 3 models have a high R-Squared values. So we can say based on the results that using 5 PCs will provide the most accurate and a highest quality model followed by model 14.

**Step 4: Running the Model for 5 PCs, 6 PCs and 14 PCs**

**CODE:**

## Joining Crime column from USCrime dataset to the Principal Component Dataset

PC5UScrime <- cbind(uscrimescaledata$x[,1:5],uscrime[,16])

PC6UScrime <- cbind(uscrimescaledata$x[,1:6],uscrime[,16])

PC14UScrime <- cbind(uscrimescaledata$x[,1:14],uscrime[,16])

## Running linear regression on the Principal Components data

## 5 PCs

model5PCs<- lm(V6~., data = as.data.frame(PC5UScrime))

summary(model5PCs)

## 6 PCs

model6PCs<- lm(V7~., data = as.data.frame(PC6UScrime))

summary(model6PCs)

## 14 PCs

model14PCs<- lm(V15~., data = as.data.frame(PC14UScrime))

summary(model14PCs)

**OUTPUT:**

> summary(model5PCs)

Call:

lm(formula = V6 ~ ., data = as.data.frame(PC5UScrime))

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

> summary(model6PCs)

Call:

lm(formula = V7 ~ ., data = as.data.frame(PC6UScrime))

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

> summary(model14PCs)

Call:

lm(formula = V15 ~ ., data = as.data.frame(PC14UScrime))

Residuals:

Min 1Q Median 3Q Max

-459.68 -122.31 12.01 117.70 450.93

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 905.09 30.91 29.280 < 2e-16 \*\*\*

PC1 65.22 12.74 5.121 1.4e-05 \*\*\*

PC2 -70.08 18.67 -3.754 0.000694 \*\*\*

PC3 25.19 22.07 1.142 0.262045

PC4 69.45 28.98 2.396 0.022591 \*

PC5 -229.04 31.92 -7.176 3.8e-08 \*\*\*

PC6 -60.21 42.01 -1.433 0.161470

PC7 117.26 55.08 2.129 0.041059 \*

PC8 28.72 56.36 0.510 0.613858

PC9 -37.18 64.43 -0.577 0.568006

PC10 56.32 69.89 0.806 0.426294

PC11 30.59 74.55 0.410 0.684247

PC12 289.61 87.27 3.319 0.002265 \*\*

PC13 81.79 118.66 0.689 0.495625

PC14 219.19 129.22 1.696 0.099555 .

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 211.9 on 32 degrees of freedom

Multiple R-squared: 0.7911, Adjusted R-squared: 0.6998

F-statistic: 8.658 on 14 and 32 DF, p-value: 2.639e-07

**Step 5: Unscaling the data for all the 3 models and performing Predictions**

**CODE for 5 PCs:**

*## 5PCs*

*##we first need our intercept*

*beta0 <- model5PCs$coefficients[1]*

*#Making beta vector pulling out our models coefficient*

*betas <- model$coefficients[2:6]*

*#now multply the coefficients by our rotation matrix, to create vector called alpha*

*alpha <- uscrimescaledata$rotation[,1:5] %\*% betas*

*## Calculate mean of all fields*

*mean <- sapply(uscrime[,1:15],mean)*

*stdev <- sapply(uscrime[,1:15],sd)*

*##Calculate SST*

*n = length(uscrime$Crime)*

*avgcrime = mean(uscrime$Crime)*

*SSE<-0*

*for(i in 1:n){*

*SST = SST + (uscrime$Crime[i] - avg)^2*

*}*

*##Calculate SSE*

*originalAlpha <- alpha/stdev*

*originalBeta0 <- beta0 - sum(alpha\*mean /stdev)*

*estimatedvalues <- as.matrix(uscrime[,1:15]) %\*% originalAlpha + originalBeta0*

*SSE = sum((estimatedvalues - uscrime[,16])^2)*

*R\_Squared = 1- (SSE/SST)*

*R\_Squared*

*##Using test point data from hw 5*

*test\_point1 <- 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 = 0.040, Time = 39.0)*

*#first we apply the PCA data onto the test\_point1(new city) data so we can apply our model*

*pred\_PC5 <- data.frame(predict(uscrimescaledata, test\_point1))*

*#now predict the Crime rate using PCs and test\_point1 data*

*pred1 <- predict(model5PCs, pred\_PC5)*

*pred1*

**OUTPUT for 5 PCs:**

R\_Squared

[1] 0.6451941

> R\_Squared\_adjusted <- R\_Squared - (1-R\_Squared)\*5/(nrow(uscrime)-5-1)

> R\_Squared\_adjusted

[1] 0.601925

pred1

1

1388.926

**CODE for 6 PCs:**

*## 6PCs*

*##we first need our intercept*

*beta0 <- model6PCs$coefficients[1]*

*#Making beta vector pulling out our models coefficient*

*betas <- model6PCs$coefficients[2:7]*

*#now multply the coefficients by our rotation matrix, to create vector called alpha*

*alpha <- uscrimescaledata$rotation[,1:6] %\*% betas*

*## Calculate mean of all fields*

*mean <- sapply(uscrime[,1:15],mean)*

*stdev <- sapply(uscrime[,1:15],sd)*

*##Calculate SST*

*n = length(uscrime$Crime)*

*avgcrime = mean(uscrime$Crime)*

*SST<-0*

*for(i in 1:n){*

*SST = SST + (uscrime$Crime[i] - avgcrime)^2*

*}*

*##Calculate SSE*

*originalAlpha <- alpha/stdev*

*originalBeta0 <- beta0 - sum(alpha\*mean /stdev)*

*estimatedvalues <- as.matrix(uscrime[,1:15]) %\*% originalAlpha + originalBeta0*

*SSE = sum((estimatedvalues - uscrime[,16])^2)*

*R\_Squared = 1- (SSE/SST)*

*R\_Squared*

*R\_Squared\_adjusted <- R\_Squared - (1-R\_Squared)\*5/(nrow(uscrime)-5-1)*

*R\_Squared\_adjusted*

*##Using test point data from hw 5*

*test\_point1 <- 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 = 0.040, Time = 39.0)*

*#first we apply the PCA data onto the test\_point1(new city) data so we can apply our model*

*pred\_PC6 <- data.frame(predict(uscrimescaledata, test\_point1))*

*#now predict the Crime rate using PCs and test\_point1 data*

*pred2 <- predict(model6PCs, pred\_PC6)*

*pred2*

**OUTPUT for 6 PCs:**

> R\_Squared = 1- (SSE/SST)

> R\_Squared

[1] 0.6586023

> R\_Squared\_adjusted <- R\_Squared - (1-R\_Squared)\*5/(nrow(uscrime)-5-1)

> R\_Squared\_adjusted

[1] 0.6169685

pred2

1

1248.427

**CODE for 14 PCs:**

## 14PCs

##we first need our intercept

beta0 <- model14PCs$coefficients[1]

#Making beta vector pulling out our models coefficient

betas <- model14PCs$coefficients[2:15]

#now multply the coefficients by our rotation matrix, to create vector called alpha

alpha <- uscrimescaledata$rotation[,1:14] %\*% betas

## Calculate mean of all fields

mean <- sapply(uscrime[,1:15],mean)

stdev <- sapply(uscrime[,1:15],sd)

##Calculate SST

n = length(uscrime$Crime)

avgcrime = mean(uscrime$Crime)

SST<-0

for(i in 1:n){

SST = SST + (uscrime$Crime[i] - avgcrime)^2

}

##Calculate SSE

originalAlpha <- alpha/stdev

originalBeta0 <- beta0 - sum(alpha\*mean /stdev)

estimatedvalues <- as.matrix(uscrime[,1:15]) %\*% originalAlpha + originalBeta0

SSE = sum((estimatedvalues - uscrime[,16])^2)

R\_Squared = 1- (SSE/SST)

R\_Squared

R\_Squared\_adjusted <- R\_Squared - (1-R\_Squared)\*5/(nrow(uscrime)-5-1)

R\_Squared\_adjusted

##Using test point data from hw 5

test\_point1 <- 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 = 0.040, Time = 39.0)

#first we apply the PCA data onto the test\_point1(new city) data so we can apply our model

pred\_PC14 <- data.frame(predict(uscrimescaledata, test\_point1))

#now predict the Crime rate using PCs and test\_point1 data

pred3 <- predict(model14PCs, pred\_PC14)

pred3

**OUTPUT for 14 PCs:**

R\_Squared

[1] 0.7911447

R\_Squared\_adjusted

[1] 0.7656745

> pred3

1

957.264

**Analysis:** The table below summarizes the analysis performed in model used in HW 5 and the 3 PCA models we used here.

|  |  |  |
| --- | --- | --- |
| **Model** | **R-Squared** | **Prediction** |
| NO PCA (HW 5) | 0.719 | 1304 |
| 5 PCs | 0.6451941 | 1389 |
| 6 PCs | 0.6586023 | 1248 |
| 14 PCs | 0.7911447 | 957 |

All the 3 models have a prediction between Minimum of 352 and maximum 1960 of in the crime data values. However the accuracy with 5 and 6 Principal components is slightly less as compared to the model with no PCA. The model with 14 PCs has a higher accuracy and lesser variability in data.