Homework6.R

harit

2020-09-30

rm(list = ls())  
set.seed(1)  
  
#Read the crime data  
crimedata <- read.table("uscrime.txt",stringsAsFactors = FALSE, header = TRUE)  
#Prinicipal component analysis on the crime data. The prcomp function takes  
# all predictor columns except for the response coumn (Crime) on scaled data.  
#  
principalcomponents <- prcomp(crimedata[,-16],center=TRUE,scale=TRUE)  
  
summary(principalcomponents)

## 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

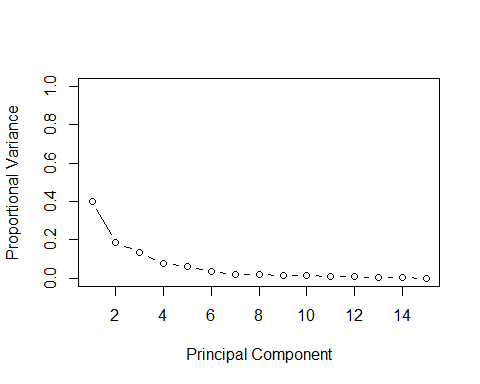
# Analysis : The first 4 PC's are more important.  
# Standard Deviation "square roots of the eigenvalues of the covariance/correlation matrix"  
# rotation - "matric of eigen valued columns  
library(DAAG)

## Loading required package: lattice

# How many PCS to choose :  
#If you divide each value by the total sum of eigenvalues  
#prior to plotting, then your plot will show the fraction  
#of total variance retained vs. number of eigenvalues.   
  
varianceofeachEigenColumn<-principalcomponents$sdev^2  
   
proportionalvariance <- varianceofeachEigenColumn/sum(varianceofeachEigenColumn)  
proportionalvariance

## [1] 0.401263510 0.186789802 0.133662956 0.077480520 0.063886598 0.036879593  
## [7] 0.021454579 0.020493418 0.015677019 0.013325395 0.011712360 0.008546007  
## [13] 0.004622779 0.003897851 0.000307611

plot(proportionalvariance,xlab="Principal Component", ylab="Proportional Variance",  
 ylim=c(0,1),type = "b")



# The plot shows that at after 5 Principle Component, there is a steady drop in the curve.  
# So let us choose the first 5 Principle Components. Hence K = 5  
  
k =5  
# Create a datadrame with first 5 principe components and the reponse column  
PCData= cbind(principalcomponents$x[,1:k],crimedata[,16])  
PCData

## PC1 PC2 PC3 PC4 PC5   
## [1,] -4.1992835 -1.09383120 -1.11907395 0.67178115 0.055283376 791  
## [2,] 1.1726630 0.67701360 -0.05244634 -0.08350709 -1.173199821 1635  
## [3,] -4.1737248 0.27677501 -0.37107658 0.37793995 0.541345246 578  
## [4,] 3.8349617 -2.57690596 0.22793998 0.38262331 -1.644746496 1969  
## [5,] 1.8392999 1.33098564 1.27882805 0.71814305 0.041590320 1234  
## [6,] 2.9072336 -0.33054213 0.53288181 1.22140635 1.374360960 682  
## [7,] 0.2457752 -0.07362562 -0.90742064 1.13685873 0.718644387 963  
## [8,] -0.1301330 -1.35985577 0.59753132 1.44045387 -0.222781388 1555  
## [9,] -3.6103169 -0.68621008 1.28372246 0.55171150 -0.324292990 856  
## [10,] 1.1672376 3.03207033 0.37984502 -0.28887026 -0.646056610 705  
## [11,] 2.5384879 -2.66771358 1.54424656 -0.87671210 -0.324083561 1674  
## [12,] 1.0065920 -0.06044849 1.18861346 -1.31261964 0.358087724 849  
## [13,] 0.5161143 0.97485189 1.83351610 -1.59117618 0.599881946 511  
## [14,] 0.4265556 1.85044812 1.02893477 -0.07789173 0.741887592 664  
## [15,] -3.3435299 0.05182823 -1.01358113 0.08840211 0.002969448 798  
## [16,] -3.0310689 -2.10295524 -1.82993161 0.52347187 -0.387454246 946  
## [17,] -0.2262961 1.44939774 -1.37565975 0.28960865 1.337784608 539  
## [18,] -0.1127499 -0.39407030 -0.38836278 3.97985093 0.410914404 929  
## [19,] 2.9195668 -1.58646124 0.97612613 0.78629766 1.356288600 750  
## [20,] 2.2998485 -1.73396487 -2.82423222 -0.23281758 -0.653038858 1225  
## [21,] 1.1501667 0.13531015 0.28506743 -2.19770548 0.084621572 742  
## [22,] -5.6594827 -1.09730404 0.10043541 -0.05245484 -0.689327990 439  
## [23,] -0.1011749 -0.57911362 0.71128354 -0.44394773 0.689939865 1216  
## [24,] 1.3836281 1.95052341 -2.98485490 -0.35942784 -0.744371276 968  
## [25,] 0.2727756 2.63013778 1.83189535 0.05207518 0.803692524 523  
## [26,] 4.0565577 1.17534729 -0.81690756 1.66990720 -2.895110075 1993  
## [27,] 0.8929694 0.79236692 1.26822542 -0.57575615 1.830793964 342  
## [28,] 0.1514495 1.44873320 0.10857670 -0.51040146 -1.023229895 1216  
## [29,] 3.5592481 -4.76202163 0.75080576 0.64692974 0.309946510 1043  
## [30,] -4.1184576 -0.38073981 1.43463965 0.63330834 -0.254715638 696  
## [31,] -0.6811731 1.66926027 -2.88645794 -1.30977099 -0.470913997 373  
## [32,] 1.7157269 -1.30836339 -0.55971313 -0.70557980 0.331277622 754  
## [33,] -1.8860627 0.59058174 1.43570145 0.18239089 0.291863659 1072  
## [34,] 1.9526349 0.52395429 -0.75642216 0.44289927 0.723474420 923  
## [35,] 1.5888864 -3.12998571 -1.73107199 -1.68604766 0.665406182 653  
## [36,] 1.0709414 -1.65628271 0.79436888 -1.85172698 0.020031154 1272  
## [37,] -4.1101715 0.15766712 2.36296974 -0.56868399 -2.469679496 831  
## [38,] -0.7254706 2.89263339 -0.36348376 -0.50612576 0.028157162 566  
## [39,] -3.3451254 -0.95045293 0.19551398 -0.27716645 0.487259213 826  
## [40,] -1.0644466 -1.05265304 0.82886286 -0.12042931 -0.645884788 1151  
## [41,] 1.4933989 1.86712106 1.81853582 -1.06112429 0.009855774 880  
## [42,] -0.6789284 1.83156328 -1.65435992 0.95121379 2.115630145 542  
## [43,] -2.4164258 -0.46701087 1.42808323 0.41149015 -0.867397522 823  
## [44,] 2.2978729 0.41865689 -0.64422929 -0.63462770 -0.703116983 1030  
## [45,] -2.9245282 -1.19488555 -3.35139309 -1.48966984 0.806659622 455  
## [46,] 1.7654525 0.95655926 0.98576138 1.05683769 0.542466034 508  
## [47,] 2.3125056 2.56161119 -1.58223354 0.59863946 -1.140712406 849

LinearRegressionPCmodel <- lm(V6~.,data = as.data.frame(PCData))  
summary(LinearRegressionPCmodel)

##   
## Call:  
## lm(formula = V6 ~ ., data = as.data.frame(PCData))  
##   
## 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

# We need to go back to the original coordinate system  
# 1. We need to scale back the data  
# xs = x-meanx/sigmax => x = xs\*sigmax + meanx  
#ys = y-meany/sigmay => y = ys\*sigmay + meany  
# Intercept= Intercept(sc) - ScaledCoefficients (Meanx/sigmax) -ScaledCoefficients(Meany/Sigmay)  
  
#Intercept  
Interceptscaled <- LinearRegressionPCmodel$coefficients[1]  
Interceptscaled

## (Intercept)   
## 905.0851

#Scaled Coefficents  
scaledcoefs <-LinearRegressionPCmodel$coefficients[2:(k+1)]  
  
# Rotate tge scaled data back. OriginalRotatationScaledData = RotationMatrix \* Scaled data  
OriginalRotatationScaledData <- principalcomponents$rotation[,1:k]%\*%scaledcoefs  
  
summary(OriginalRotatationScaledData)

## V1   
## Min. :-34.64   
## 1st Qu.: 24.65   
## Median : 37.85   
## Mean : 51.64   
## 3rd Qu.: 87.16   
## Max. :117.34

mu <- sapply(crimedata[,1:15],mean)  
sigma <-sapply(crimedata[,1:15],sd)  
  
originalCoeff = OriginalRotatationScaledData/sigma  
OriginalIntercept = Interceptscaled - sum(OriginalRotatationScaledData\*mu/sigma)  
  
  
original = as.matrix(crimedata[,1:15]) %\*% originalCoeff+OriginalIntercept  
  
# Find the accuracy   
sse = sum((original - crimedata[,16])^2)  
totalSumofSquares = sum((crimedata[,16]-mean(crimedata[,16]))^2)  
RSquared = 1- (sse/totalSumofSquares)  
AdjustedRSqaured = RSquared - (1-RSquared)\*k/(nrow(crimedata)-k-1)  
AdjustedRSqaured

## [1] 0.601925

RSquared

## [1] 0.6451941

AdjustedRSqaured

## [1] 0.601925

testpt <- 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)  
#Predict the crime rate for the data point  
# Replace PCA data into the test point  
PCATestPoint <- data.frame(predict(principalcomponents,testpt))  
predict\_model <-predict(LinearRegressionPCmodel,PCATestPoint)  
predict\_model

## 1   
## 1388.926