Assignment #2

Saied Hosseinipoor

September 20, 2016

Problem 1

1a)

The results for **concordant** and **discordant** are **6** and **15** resepectively.

```
x = c(3, 4, 2, 1, 7, 6, 5); y = c(4, 3, 7, 6, 5, 2, 1)
# Calculate the Concordance and Discordance based on the definition
C = 0; D = 0
for(i in 1:length(x))
  for(j in 1:length(y)){
    if(x[i] < x[j]){
      if(y[i] < y[j])
        C = C + 1
      if(y[i] > y[j])
        D = D + 1
    }
  }
C; D
```

```
## [1] 6
```

```
## [1] 15
```

Problem 2

2a)

Four sets of random numbers (*a-d*) have been generated in different distributions: *Normal, Chi square, Log normal, and Exponential distributions.* The variales combined into a data frame (*df*) and then data frame melted into a new data frame (*df2*).

```
head (df, 3)
```

```
##           a         b          c         d
## 1 -0.9612687  7.592435 2.67140624 0.1818223
## 2  0.6759098 13.756017 0.07377862 1.2398098
## 3  2.4981631  3.657855 0.54546817 1.1987128
```

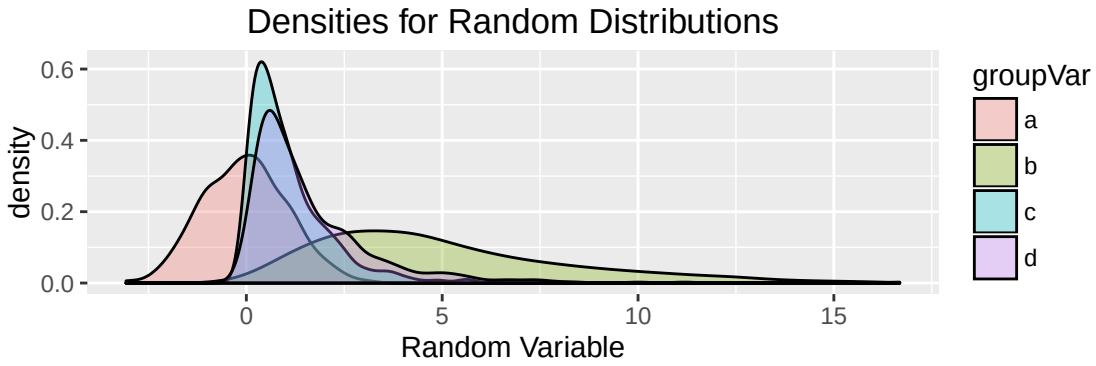


Figure 1: Density Plot for Generated Random Variables

```
head (df2, 3)
```

```
##   groupVar      value
## 1        a -0.9612687
## 2        a  0.6759098
## 3        a  2.4981631
```

2b)

The figure illustrates the density function for the random data sets. Different distributions have different behavior as shown in the

---

Problem 3

3a)

The dataset contains information ranging from early dates, when the flow of information and record of all the incidents was doubous. This data might be misleading sometimes. The early dates information might not reflect the current nature of the problem, and the volume of the information with those dates are very low. In addition, some data from very old times might be incorrect.

3b)

```
Shark.attack = read.csv("ISE 5103 GSAF.csv")          # Reads data from the file
GSAFdata = Shark.attack[Shark.attack$Year >= 2000,] # Deletes the records before 2000
GSAFdata$Date.old = GSAFdata$Date                     # Keep the old date values
GSAFdata$Date <- as.Date(GSAFdata$Case.Number ,
                         "%Y.%m.%d")                  # Converts and stores the dates as date format
```

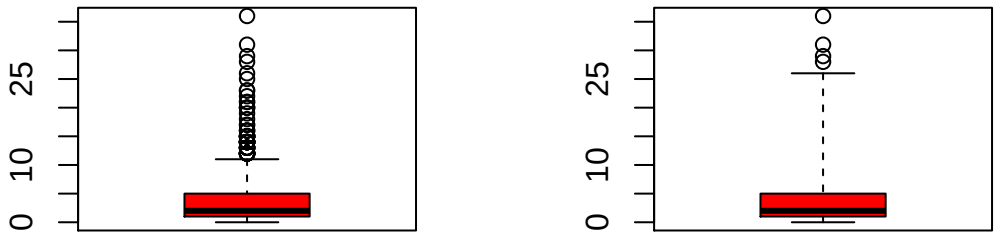


Figure 2: Box Plot and Ajusted Box Plot for Shark Attack Intervals

3c)

As mentioned above, the data from very begining which are recorded from the early acient times could be misleading. It is better to cut off the old data and use the more accurate one for better and concise results. The next step is cleaning and polishing the the date attribute in data set. To clean up the date feild, we can use **"Date"** sttribute. We will face to two types of problem here:

1. Some instances of field carry extra and unnecessary information that voilate the standard format of data as a date (e.g. *"Reported on 03-Mar-2000"*).
2. Some instances may have standard form but are not usable in our application (e.g. *"Jun-2000"*).

Fortunately, there is filed named **"Case.Number"** which is in a form that we can avoid the problem no. 1 and just convert them into a correct date with one line code. The dates with the second problem converted into ***NA***.

3d)

According to the method I used, the percentage of missing data is about 2.3%. If we choose the **Date** feild to clean up the data set, we would gain a percentage of 7.4% which more work on cleaning will result less percentage but not 2.3%.

3e)

```
GSAFdata <- GSAFdata[!is.na(GSAFdata$Date),]    # Deletes missing data respect to date
```

3f-i)

```
daysBetween <- diff(GSAFdata$Date)           # Calculates the interval between attacks
GSAFdata$DaysBetween <- c(NA, daysBetween)   # Add the new vaiable to data frame
```

3f-ii)

Box plot, as seen in figure, assumes the distribution is normal. Therefore it suggests a lot of points as outliners. In Figure, the boxplot reveals that the median is closer to first quartile than third quartile and hence data is positively skewed. The adjusted box plot does not have an assumtion on the distributuion. It shows a more reasonable outliners than regular box plot.

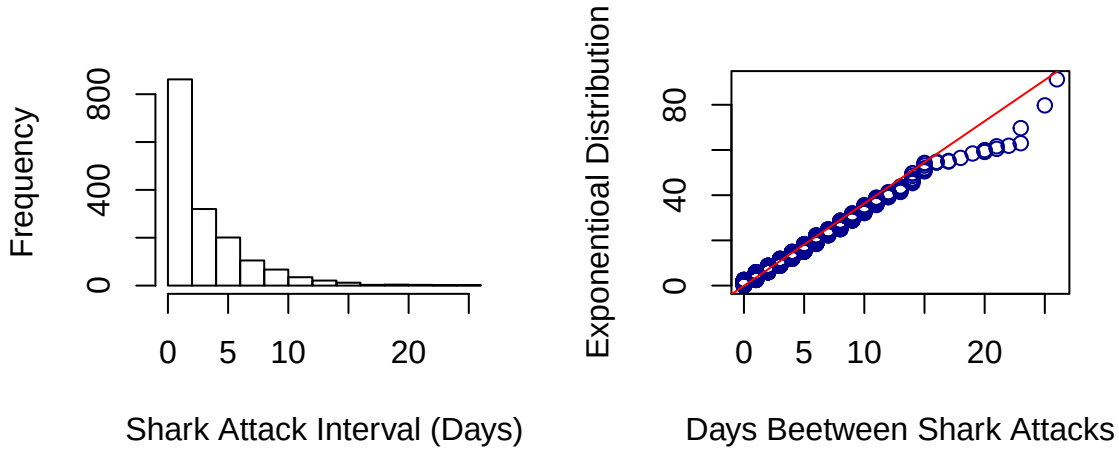


Figure 3: Histogram and Q-Q Plots for Shark Attack Intervals

3f-iii)

The result p-value in Grubb's test shows that the biggest value definitly is a ouliner. This test doesn't say anything about the more data if the next biggest number is also a ouliner or not. The Generalized ESD test shows that we have 14 outliners. Both test agreed on the most extrim value but Grubb's test didn't give any information about the next values.

The main problem associated with these methods is they are assuming the data are distributed normally. The adjusted box plot showed us that data may not follow the normal distribution. It is possible to have different distribution like ***exponential*** or ***log-normal*** ditribution. Therefore it is better to use adjusted box plot to omit the outliers.

3g)

The plots visually impliy that the distribution could be an exponential distribution.

3h)

The plots show that the data has a good fit to the exponential distribution and also the p-value of tests are greater than 5%. Thus H_0 is not rejected and days between attack follows the exponential distribution.

3i)

The problem states that if the shark attcks occur in **Poisson process**, the time between attacks follows the **exponential distribution**. As we showed in the previous section, the time between attacks is a exponential process, thus ***"the shark attack is a Poisson Pocess"***.

Problem 4

4a)

The investigation on data set reveals that the variable ***tariff*** has the most missing values and ***fiveop*** and ***intresmi*** are in the next places. More details are available in R code.

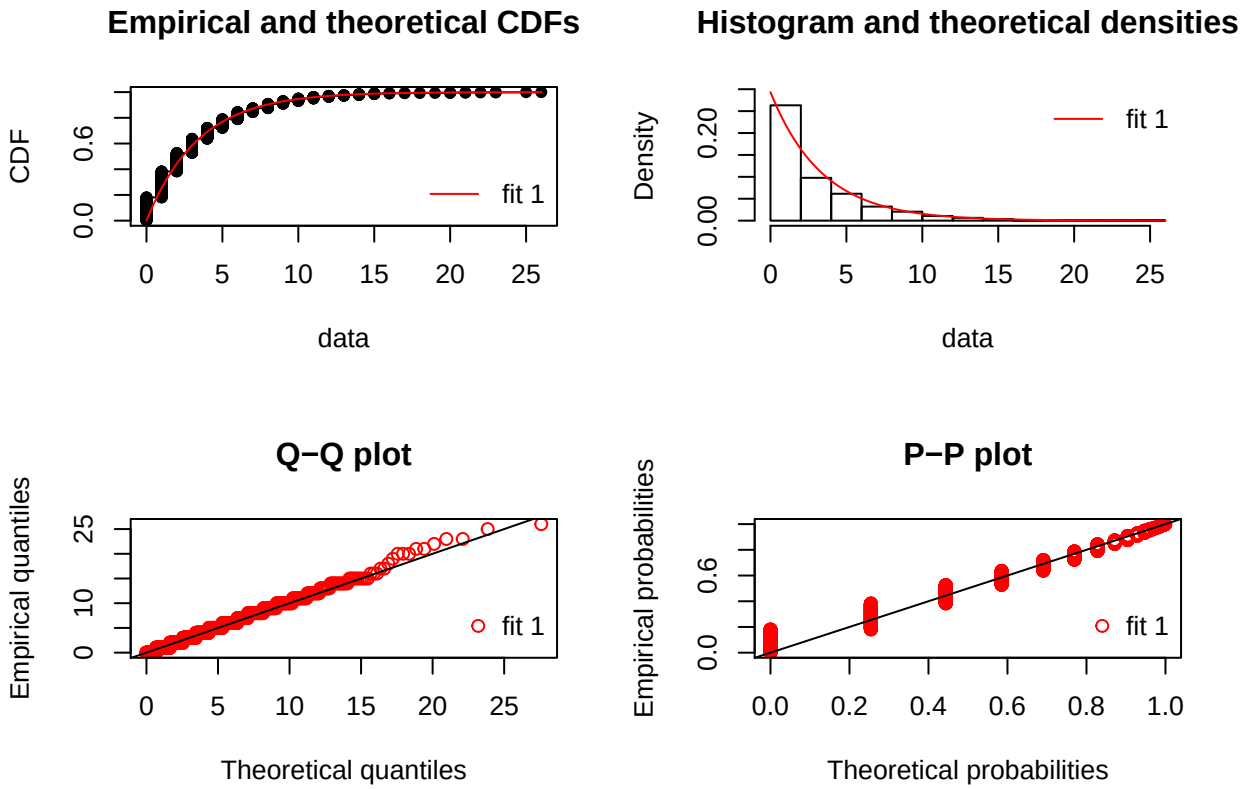


Figure 4: Fitting Exponential Model

4b)

Chi square tests have been performed on the original data set, data without Nepal, and data without Philipines. The p-values were almost same for all the tests, Therefore tehre is no significant change in different cases. It seems that the missing value for **tariff** variable is not related to **Country** variable. Details are available in R code.

Problem 5

5a)

```
data("mtcars")
corMat <- cor(mtcars)

mtcars.eigen <- eigen(corMat)

mtcars.PC <- prcomp(mtcars, scale. = TRUE)    # Compute the principal component of mtcars

round(sum(abs(mtcars.PC$rotation) - abs(mtcars.eigen$vectors)), 2) # Compare with difference
```

```
## [1] 0
```

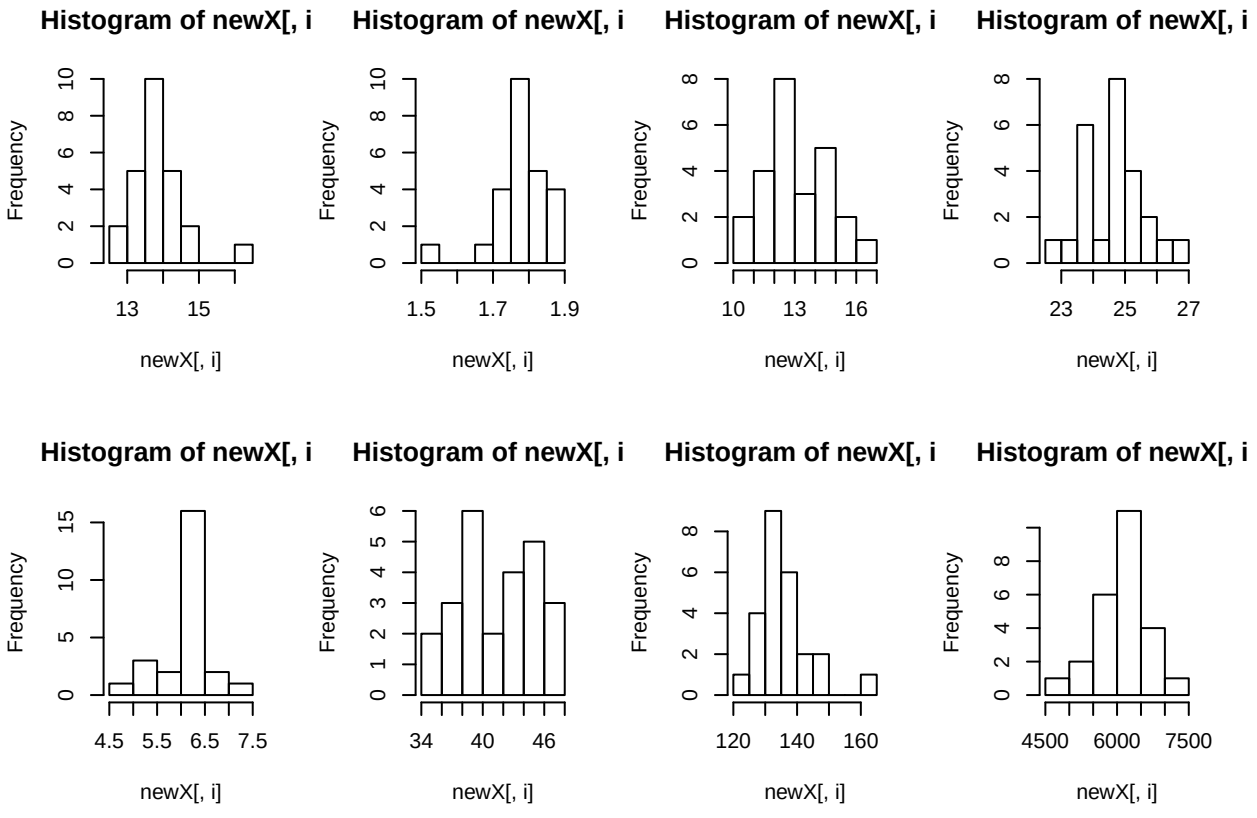
```
round(mtcars.PC$rotation[,1] %*% mtcars.PC$rotation[,2],2)    # Checks orthogonality
```

```
##      [,1]
## [1,]    0
```

The results show that the rotation matrix of principal component is the eigen vecotor of the same data. In addition, the PC1 and PC2 are orthogonal.

5b)

```
data("heptathlon")
par(mfrow=c(2,4))                # Divides the screen into four sections
temp <- apply(heptathlon[,1:8], 2, hist)# Draws histograms
```




```
par(mfrow=c(1,1))                # Resets the screen in normal
```

Since the number of records in this example is not much, it is difficult to say if the distribution is normal. But I guess the distribution is normal.

Applying the Grubb's test and looking into the results shows that **Launa (PNG)** definitly is the outliener. She is an ouliner at "*hurdles*", "*highjump*", "*run800m*", and "*score*". We can omit her record to have a cleaner data set. Details are avialable in R code.

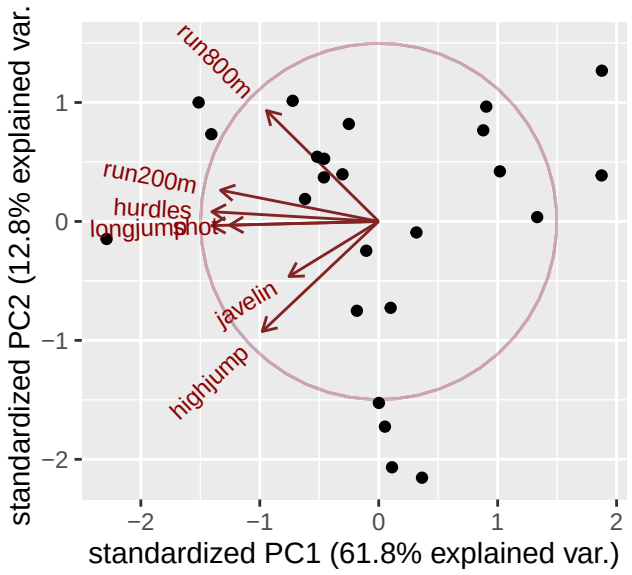


Figure 5: Biplot of Principle Component Analysis

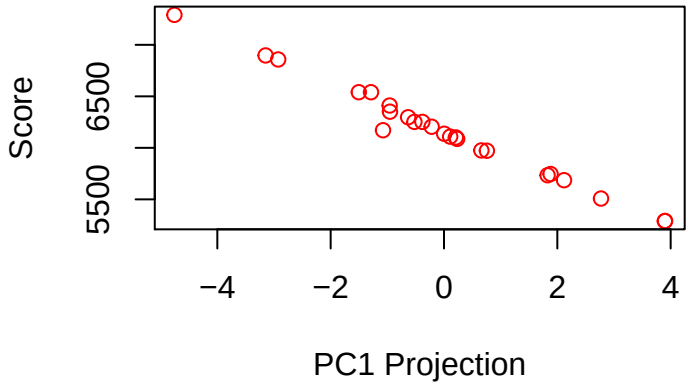


Figure 6: Biplot of Principle Component Analysis

```
heptathlon$hurdles <- max(heptathlon$hurdles) - heptathlon$hurdles
heptathlon$run200m <- max(heptathlon$run200m) - heptathlon$run200m
heptathlon$run800m <- max(heptathlon$run800m) - heptathlon$run800m
```

The Biplot here shows that *Javelin* is higly related to **PC1** in negative way. Other variables shows more relation to **PC2**; *runnings* and *hurdles* are in positive direction where *jumpings* and *shot* are in negative direction.

The next figure shows that PC1 is a very desriptive variable and we can condense all the original attributes into one variable and still keep the variance for analyzing purposes. The PC1 variable which is the reduced dimension of all the original variables, has a negative relationship with the score variable. In the other words, we can reduce 7 dimensions into a single dimension without loosing much information. Since the original variables are correlated, as could be expected, it is possible to condense all of them into a simple variable.

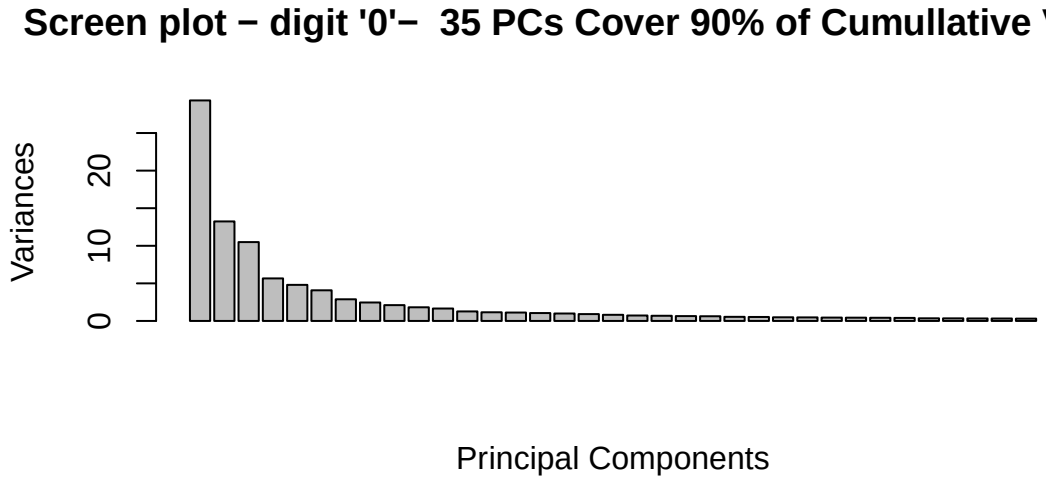


Figure 7: Dimension Reduction Using PCA Technique

5c)

The main goal in using ***Principal Component Analysis*** is to reduce the dimension of the problem. In the image processing problems like this problem, each pixel of picture is considered as a feature. Therefore we will have a very huge amount of features. In this case which contains very small images, we have 256 variables. If we decide to keep 90% of the variance which is an acceptable number, the following results would be achievable. The worst case has 48 dimensions that comparing to the original dimensions 256 is a very good improvement.

```
hw.0 <-read.csv("train.0")
hw.0.pc <-prcomp(hw.0)
hw.0.pc.summary <- summary(hw.0.pc)
i = 1
while (hw.0.pc.summary$importance[3,i] < 0.90) {
  i = i + 1
}
hw.0.pc.cut = i
screeplot(hw.0.pc,
          xlab = "Principal Components",
          main = paste ("Screen plot - digit '0'- ",
                        hw.0.pc.cut,"PCs Cover 90% of Cumullative Variation"),
          npcs = hw.0.pc.cut)
```

Problem 6

The data contains information about shape, texture, and margin of leaves. The given data are in form of csv files which describe the image of the leaf. Data is available on ***https://www.kaggle.com/c/leaf-classification***. No data is missing. Some data manupolation is available in R code.

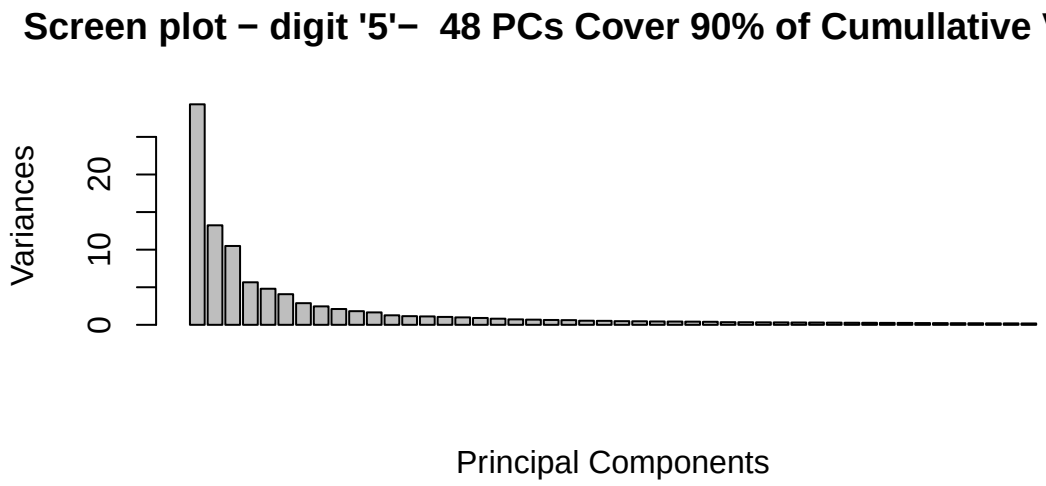


Figure 8: Dimension Reduction Using PCA Technique

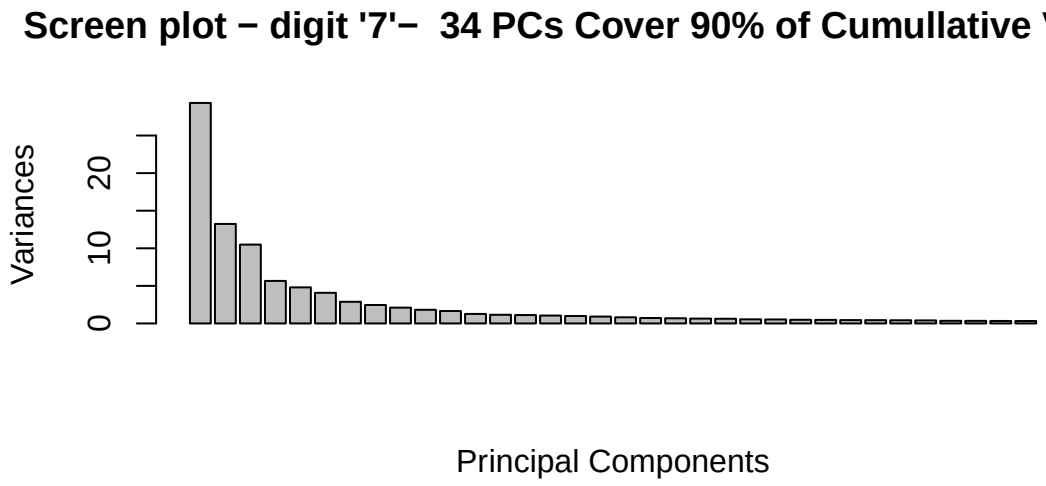


Figure 9: Dimension Reduction Using PCA Technique