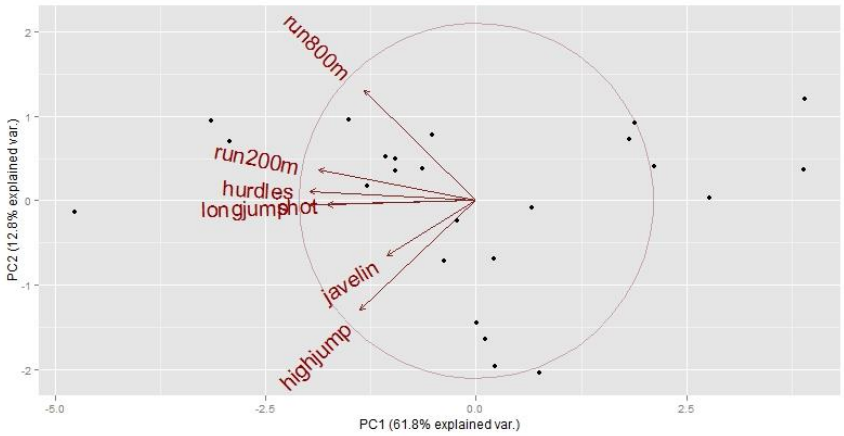
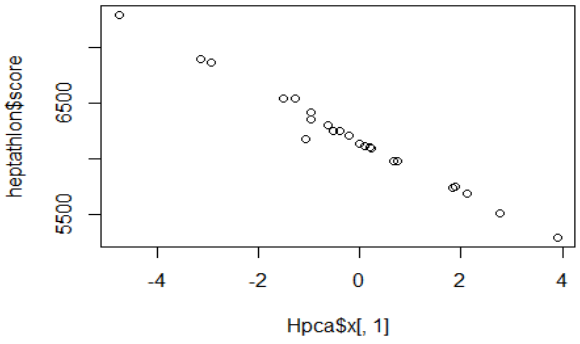
```r
#================================================================================
#                        Using R: Principal Component Analysis
#================================================================================

#--------------------------------------------------------------------------------
#(a) Mathematics of principal components
#i. Using the data mtcars, create the correlation ...
data(mtcars)
str(mtcars)
corMat <- matrix(0, nrow=length(mtcars), ncol=length(mtcars))
#Other method of correlation can be used. Here we choose the linear method for using
in PCA later.
for(i in 1:(length(mtcars))){
  for(j in 1:(length(mtcars))){
    corMat[i,j] <- cor(mtcars[,i],mtcars[,j])
  }
}

#ii. Compute the eigenvalues and eigenvectors of corMat.
Eigen.corMat <- eigen(corMat)

#iii. Use prcomp to compute the principal components of the ...
Prc.mtcars <- prcomp(mtcars, scale=T)

#iv. Compare the results from (ii) and (iii) - Are they the same? Different? Why?
#Constructing the covariance matrix of the decomposition, gives us the correlation.
#By multiplying of the definition of equivalent eigenvector to their transpose results
#in the covariance matrix too. So the eigenvectors of both of them should be the same
#if they have the same length. So if we normalize them they should be the same.
#But as long as they do not have the same size so have different value.

#v. Using R demonstrate that principal components 1 and 2 ...
#It can be said that two vector are orthogonal to each other provided they inner
product between them be equal to zero. Like here.
innerproduct <- Prc.mtcars$rotation[,1] %*% Prc.mtcars$rotation[,2]

#--------------------------------------------------------------------------------
#(b) The HSAUR2 package contains the data heptathlon ...
library(HSAUR2)

#i. Look at histograms of each numerical variable ...
data(heptathlon)
apply(heptathlon[,1:8],2,hist)
#Some of them look like to have the normal distribution. But it appears that there are
some outliers.

#ii. Examine the event results using the Grubb's test. ...
library(outliers)
for(i in 1:8){
  print(grubbs.test(heptathlon[,i]))
}
#Based on the Grubb's test, There is an outlier in each of hurdles, highjump,
longjump, run800m and score. The score is the result of other variables so outlier in
variables create an outlier in score and it should not be evaluated. It seems that the
test shows an identical outlier for each variable and it is "Launa (PNG)".
heptathlon <- heptathlon[-25,]

#iii. As is, some event results are "good" if the values ...
```

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

#iv. Perform a principal component analysis on the 7 event ...
Hpca <- prcomp(heptathlon[,1:7], scale=T)

#v. Use ggbiplot to visualize the first two principal components. ...
library(devtools)
install_github("vqv/ggbiplot")
library(ggbiplot)
ggbiplot(Hpca, circle=T, choices=c(1,2), obs.scale=1, varname.size=7)
```

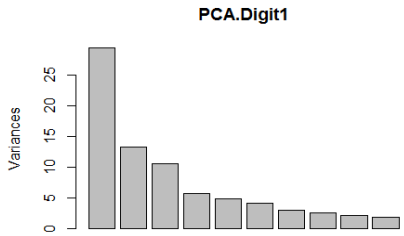


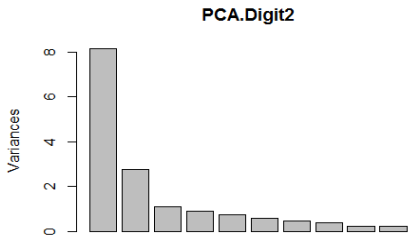

```r
#By the projection of all variables on both PCs, longjump, hurdles, run200m and shot
have more effect in first PC rather than second PC (Actually have small effect on
PC2). highjump, run800m have almost same effect on PC1 and PC2; Their effect on PC1 is
less than the first 4 variables described. javelin has less effect than other
#variables on PC1 and have moderate effect on PC2. So, PC1 distinguishes competitors
with high longjump, hurdles, run200m, shot, also run800m, highjump and javelin
#and competitors with low high longjump, hurdles, run200m, shot, run800m, highjump and
javelin. By ignoring the small effect of longjump, hurdles, run200m and shot on the
PC2, PC2 distinguishes the compeitors with high run800m, low highjump, low javelin and
competitors with low run800m, high highjump, high javelin.

#vi. The PCA projections onto principal components 1; 2; 3; : : ...
par(mfrow=c(1,1))
plot(Hpca$x[,1],heptathlon$score)
```

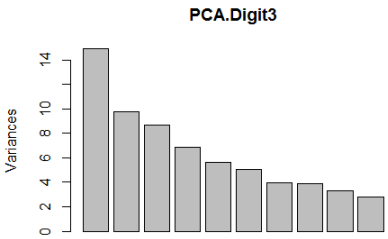



```r
#Based on the plot, there is strong negative linear correlation between scores and
PC1. It means that competitors with high score have high negative value in the
direction of PC1. As can be seen before all the variables have a same effect on PC1
(All of them on the same side). It means that strong relationship or even linear
correlation can be expected.

#-------------------------------------------------------------------------------
#(c) Handwriting Analysis
```

```r
#i. Choose three different digit data sets to download and ...
Digit1 <- read.table("train.0", sep = ",")
Digit2 <- read.table("train.1", sep = ",")
Digit3 <- read.table("train.2", sep = ",")
PCA.Digit1 <- prcomp(Digit1)
PCA.Digit2 <- prcomp(Digit2)
PCA.Digit3 <- prcomp(Digit3)
plot(PCA.Digit1)
plot(PCA.Digit2)
plot(PCA.Digit3)
```








```r
summary(PCA.Digit1)
summary(PCA.Digit2)
summary(PCA.Digit3)
#The number of PCs depends on the precision we are seeking. One can go for highest precision and use all the dimensions. Actually I believe we should do the sensitivity analysis to find out our required precision. Here without such knowledge, I choose where I have 70 percent of the original data. Based on the plots, the 2 or 3 PCs have more effect on digits 1 and 2. And for digit 3 more PCs needed. Considering 70%, we can choose 8, 4 and 18 PCs for digits 1 to 3 respectively.

#ii. PCA in general is very useful in image analysis ...
#Images have lots of correlated variables and difficult to categorize. PCA is used to transform these correlated variables into uncorrelated smaller variables in terms of reducing number of colors from 3 to 1 and also reduce the number of effective variable.
```