Assignment 3: Principle Component and factor analysis

Mohsen Pirmoradiyan, Ahmed Alhasan, Asad Enver, Ali Etminan, Mubarak Hussain

2019/12/05

Question 1: Principal components, including interpretation of them

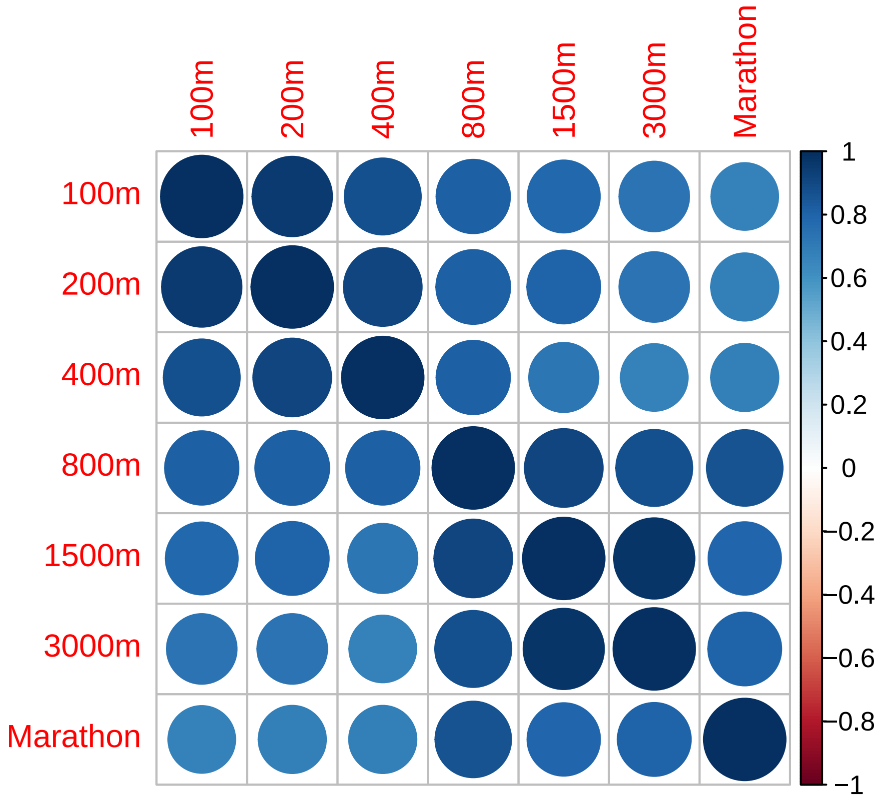
(a) Obtain the sample correlation matrix R for these data, and determine its eigenvalues and eigenvectors.

Initially, We read the T1-9 data and add column names

```
newdata <- read.delim("D:/Machine Learning/Workshop/Multivariate Statistics/Data/T1-9.dat")
colnames(newdata) <- c("Country", "100m", "200m", "400m", "800m", "1500m",
                       "3000m", "Marathon")
```

Here, we create the correlation matrix on the data with two decimal points and plot the correlation matrix for a visual comprehension. We use the *corrplot()* function from the *corrplot* package.

```
##          100m 200m 400m 800m 1500m 3000m Marathon
## 100m     1.00 0.95 0.87 0.81  0.78  0.73     0.67
## 200m     0.95 1.00 0.91 0.82  0.80  0.73     0.68
## 400m     0.87 0.91 1.00 0.81  0.72  0.67     0.68
## 800m     0.81 0.82 0.81 1.00  0.91  0.87     0.86
## 1500m    0.78 0.80 0.72 0.91  1.00  0.97     0.79
## 3000m    0.73 0.73 0.67 0.87  0.97  1.00     0.80
## Marathon 0.67 0.68 0.68 0.86  0.79  0.80     1.00
```



Now using R's *eigen()* function we can generate the eigenvalues and vectors

```
eigens <- eigen(cor.mat)
eigens
```

```
## eigen() decomposition
## $values
## [1] 5.81458751 0.63289555 0.27996672 0.12621720 0.08357546 0.04794771
## [7] 0.01480986
##
## $vectors
##            [,1]       [,2]       [,3]        [,4]          [,5]
## [1,] -0.3780137 -0.4080580 -0.1529867  0.56748241 -0.1664930114
## [2,] -0.3832178 -0.4191811 -0.1068760  0.20859774  0.0725741168
## [3,] -0.3678770 -0.4543498  0.2538770 -0.64269274  0.3435756128
## [4,] -0.3955707  0.1665768  0.1489084 -0.30075411 -0.8045954554
## [5,] -0.3886790  0.3081361 -0.4192825 -0.10311769  0.0005056435
## [6,] -0.3754736  0.4236816 -0.4039770 -0.05517137  0.3844333911
## [7,] -0.3555112  0.3861870  0.7345408  0.34244207  0.2319606395
##             [,6]        [,7]
## [1,]  0.53618577  0.17302608
## [2,] -0.70807761 -0.34037467
## [3,]  0.21041433  0.13567193
## [4,]  0.03425513 -0.23367066
## [5,] -0.27605326  0.70095940
## [6,]  0.29119235 -0.52980202
## [7,] -0.06844651  0.09572857
```

(b) Determine the first two principal components for the standardized variables. Prepare a table showing the correlations of the standardized variables with the components, and the cumulative percentage of the total (standardized) sample variance explained by the two components.

From part *(a)* we have the eigenvectors of the correlation matrix. Each eigenvector represents the principle component for a variable. Now we can generate a table showing the correlation between variables and different components.

```
eigen.vectors <- eigens$vectors
row.names(eigen.vectors) <- c("100m", "200m", "400m", "800m", "1500m",
                              "3000m", "Marathon")
colnames(eigen.vectors) <- c("COMP1", "COMP2", "COMP3", "COMP4", "COMP5",
                             "COMP6", "COMP7")
eigen.vectors
```

```
##               COMP1      COMP2      COMP3       COMP4         COMP5
## 100m     -0.3780137 -0.4080580 -0.1529867  0.56748241 -0.1664930114
## 200m     -0.3832178 -0.4191811 -0.1068760  0.20859774  0.0725741168
## 400m     -0.3678770 -0.4543498  0.2538770 -0.64269274  0.3435756128
## 800m     -0.3955707  0.1665768  0.1489084 -0.30075411 -0.8045954554
## 1500m    -0.3886790  0.3081361 -0.4192825 -0.10311769  0.0005056435
## 3000m    -0.3754736  0.4236816 -0.4039770 -0.05517137  0.3844333911
## Marathon -0.3555112  0.3861870  0.7345408  0.34244207  0.2319606395
##                COMP6       COMP7
## 100m      0.53618577  0.17302608
## 200m     -0.70807761 -0.34037467
## 400m      0.21041433  0.13567193
## 800m      0.03425513 -0.23367066
## 1500m    -0.27605326  0.70095940
## 3000m     0.29119235 -0.52980202
## Marathon -0.06844651  0.09572857
```

Alternatively, we can use the *prcomp()* function to determine the components as well as their standard deviations.

```
alt.pca <- prcomp(newdata[,-1], center = TRUE, scale. = TRUE)
alt.pca
```

```
## Standard deviations (1, .., p=7):
## [1] 2.4112080 0.7932983 0.5275500 0.3533452 0.2992749 0.2238216 0.1180274
##
## Rotation (n x k) = (7 x 7):
##                PC1        PC2        PC3         PC4         PC5
## 100m     0.3780517 -0.4064791 -0.1352023  0.59017966  0.12707449
## 200m     0.3835141 -0.4149006 -0.1106178  0.18841481 -0.03892977
## 400m     0.3678282 -0.4589479  0.2374657 -0.64601856 -0.34221913
## 800m     0.3946267  0.1624079  0.1538720 -0.29094675  0.82015012
## 1500m    0.3890193  0.3099075 -0.4219711 -0.06779742 -0.01673003
## 3000m    0.3758435  0.4235226 -0.4070261 -0.08149891 -0.35953606
## Marathon 0.3554880  0.3875676  0.7387067  0.32087268 -0.25105840
##                 PC6        PC7
## 100m     -0.54516504  0.10911971
```

```
## 200m       0.73929156 -0.29149692
## 400m      -0.21448460  0.13123133
## 800m      -0.04669980 -0.18638610
## 1500m      0.21110164  0.72465261
## 3000m     -0.23953857 -0.56605043
## Marathon  0.07820443  0.07500991
```

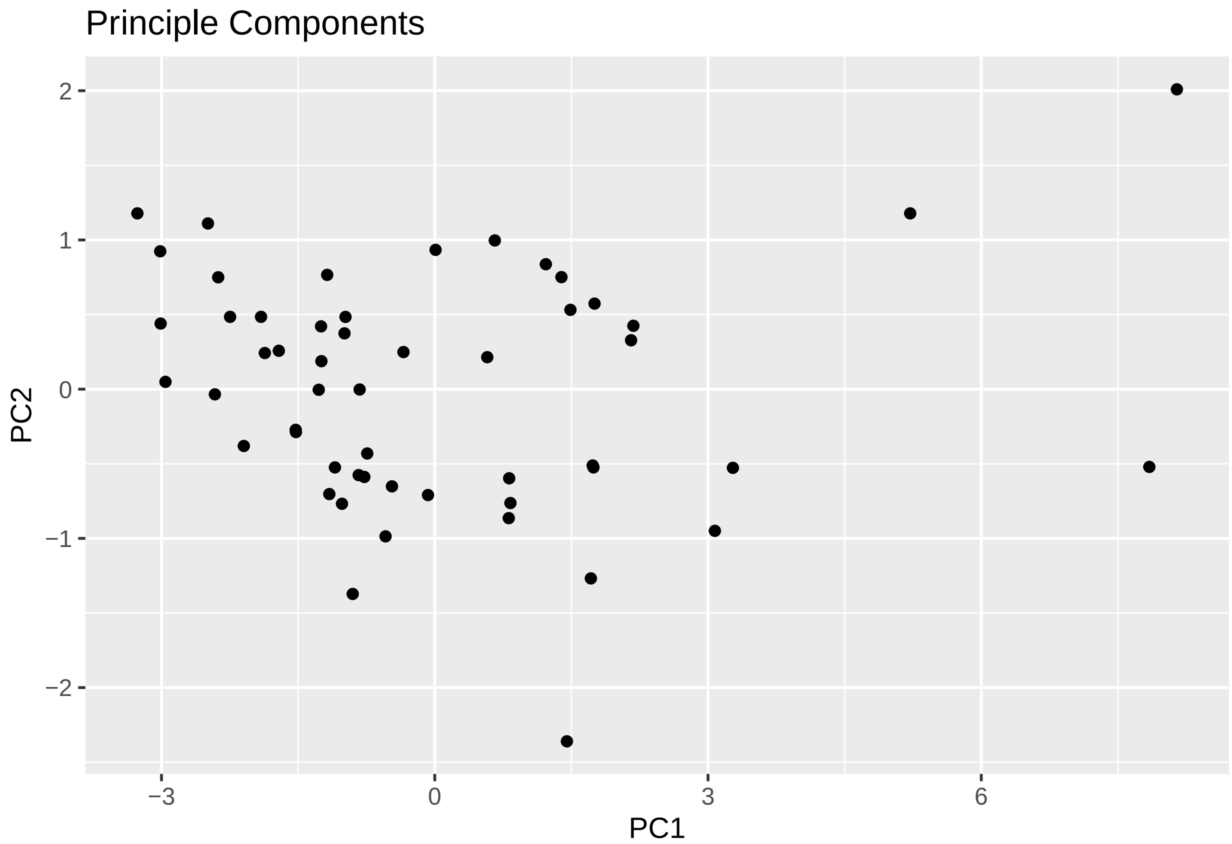
However, we are going to stick to the values obtained from part (a) for the rest of the solution.

The first two principle components:

```
eigens$vectors[,1:2]
```

```
##            [,1]       [,2]
## [1,] -0.3780137 -0.4080580
## [2,] -0.3832178 -0.4191811
## [3,] -0.3678770 -0.4543498
## [4,] -0.3955707  0.1665768
## [5,] -0.3886790  0.3081361
## [6,] -0.3754736  0.4236816
## [7,] -0.3555112  0.3861870
```

As the next step, we draw a plot using the first principle component (PC1) for the x axis and the second principle component (PC2) for the y axis:



For the next part, we calculate the contribution (accounted sample variance) of the first principle component in percentage

```r
PC1 <- (eigens$values[1]/sum(eigens$values))*100
PC1
```

```
## [1] 83.06554
```

We do the same calculations for the second principle component

```r
PC2 <- (eigens$values[2]/sum(eigens$values))*100
PC2
```

```
## [1] 9.041365
```

Finally, we calculate the cumulative proportion of the total sample variance for PC1 and PC2 in percentage

```r
total.var <- PC1 + PC2
total.var
```

```
## [1] 92.1069
```

From the results we can see that the first two principle components have a 92.1% contribution to the total variance. Therefore, the cumulative proportion of the total sample variance for PC1 and PC2 is 0.92

(c) Interpret the two principal components obtained in Part b. (Note that the first component is essentially a normalized unit vector and might measure the athletic excellence of a given nation. The second component might measure the relative strength of a nation at the various running distances.)

From the results we can see that all variables contribute almost equally to the first component. However, for the second component we can see a larger range in the values. This could be an indication of a nation's strength (running time) for various distances. We could consider the second component as the **"Distance Component"**

(d) Rank the nations based on their score on the first principal component. Does this ranking correspond with your inituitive notion of athletic excellence for the various countries?

```r
records <- newdata[,-1]
rownames(records) <- newdata[,1]
adj <- records * alt.pca$rotation[,1]
rank <-apply(adj, 1, mean)
rank[order(rank)]
```

```
##      GBR      USA      GER      KEN      JPN      POL      RUS      POR
## 12.19416 12.38759 12.73875 12.74530 12.93473 12.95111 12.96582 12.98061
##      MEX      CZE      SUI      IRL      CHN      NOR      BEL      ITA
## 13.02063 13.04203 13.07281 13.07311 13.10980 13.11383 13.12450 13.15005
##      ESP      CAN      HUN      FRA      AUS      ROM      BRA      SWE
## 13.20500 13.24278 13.25396 13.33627 13.34653 13.35122 13.36300 13.48556
##      NZL      NED      DEN      ISR      GRE      TUR      CHI      FIN
## 13.51719 13.51748 13.53058 13.56087 13.66123 13.66511 13.67349 13.70068
```

```
##      SIN     KORN      LUX      AUT     KORS      COL      MYA      CRC
## 13.79291 13.84790 13.86628 13.86851 13.91380 13.93757 13.94283 14.09425
##      IND      MAS      DOM      INA      THA      BER      TPE      PHI
## 14.14224 14.26923 14.44488 14.49047 14.55582 14.55783 14.57365 14.67134
##      MRI      GUA      SAM      PNG      COK
## 14.76272 15.04259 16.54804 17.21139 17.84879
```

- The countries ranking according to the first PC corresponds with their overall performance across all races.

```r
test  <- function(records, m= 3){
  p   <- dim(records)[2]
  n   <- dim(records)[1]


  num <- tcrossprod(loadings) + diag(pc$uniquenesses)
  T   <- (n-1-(2*p+4*m+5)/6) * log(det(num)) / det((n-1)/n*S)
  T0  <- qchisq(0.95, df = (((p-m)**2-p-m)/2))

  if(T>T0){
    print("Reject H0")
    }
  else{
    print("Can not reject H0")
    }
}


S  <- cov(records)
pc <- principal(S, nfactors=3, rotate="varimax")
loadings <- pc$loadings[,1:3]
test(records, 3)
```

```
## [1] "Can not reject H0"
```

```r
s_scores  <-as.matrix(records)%*%loadings
```

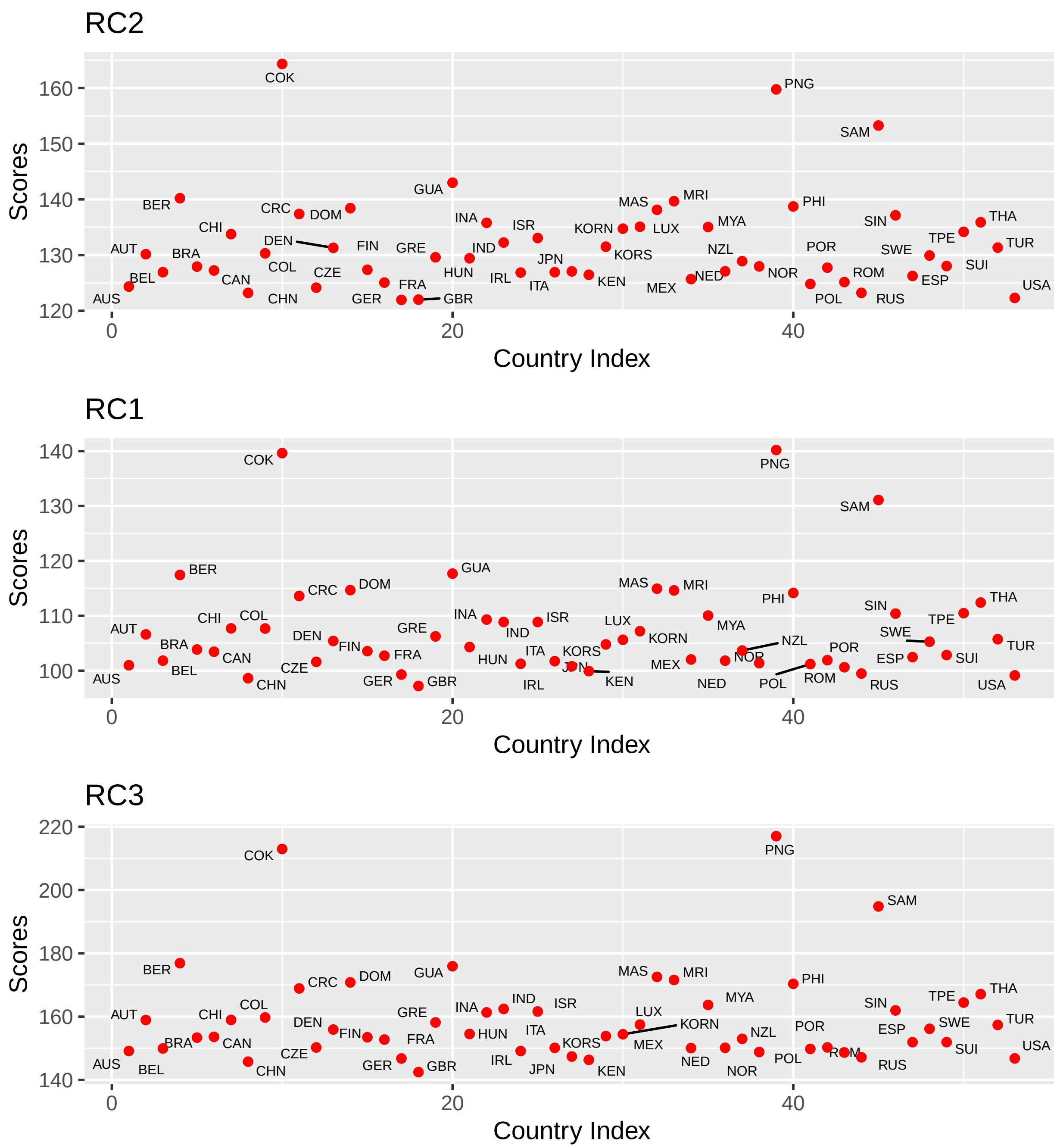
```r
p1 <- ggplot(data = as.data.frame(s_scores), aes(y = s_scores[,1], x = seq(1, length(s_scores[,1]))))+
        geom_text_repel(label = rownames(s_scores), size = 2)+
        xlab("Country Index")+
        ylab("Scores")+
        ggtitle("RC2")+
        geom_point(color = 'red')

p2 <- ggplot(data = as.data.frame(s_scores), aes(y = s_scores[,2], x = seq(1, length(s_scores[,2]))))+
        geom_text_repel(label = rownames(s_scores), size = 2)+
        xlab("Country Index")+
        ylab("Scores")+
        ggtitle("RC1")+
        geom_point(color = 'red')

p3 <- ggplot(data = as.data.frame(s_scores), aes(y = s_scores[,3], x = seq(1, length(s_scores[,3]))))+
```

```
        geom_text_repel(label = rownames(s_scores), size = 2)+
        xlab("Country Index")+
        ylab("Scores")+
        ggtitle("RC3")+
        geom_point(color = 'red')


grid.arrange(p1, p2, p3, nrow = 3)
```



Appendix

```
library(corrplot)
library(ggplot2)

newdata <- read.delim("D:/Machine Learning/Workshop/Multivariate Statistics/Data/T1-9.dat")
colnames(newdata) <- c("Country", "100m", "200m", "400m", "800m", "1500m",
                       "3000m", "Marathon")

cor.mat <- round(cor(newdata[,-1]),2)
cor.mat

corrplot(cor.mat)

eigens <- eigen(cor.mat)
eigens


eigen.vectors <- eigens$vectors
row.names(eigen.vectors) <- c("100m", "200m", "400m", "800m", "1500m",
                              "3000m", "Marathon")
colnames(eigen.vectors) <- c("COMP1", "COMP2", "COMP3", "COMP4", "COMP5",
                             "COMP6", "COMP7")
eigen.vectors


alt.pca <- prcomp(newdata[,-1], center = TRUE, scale. = TRUE)
alt.pca

eigens$vectors[,1:2]
prcomp(cor.mat)


ggplot() +
  geom_point(aes(alt.pca$x[,1],alt.pca$x[,2])) +
  xlab("PC1") + ylab("PC2") + ggtitle("Principle Components")

PC1 <- (eigens$values[1]/sum(eigens$values))*100
PC1

PC2 <- (eigens$values[2]/sum(eigens$values))*100
PC2

total.var <- PC1 + PC2
total.var
```