PCA Tutorial

R 공개강좌

서울대학교 통계연구소

National Track Records (Men)

데이터 출처: 5th ed, Applied Multivariate Statistical Analysis, R.A Johnson, D.W.Wichern(2002).

```
##Read.csv
#setwd("~/Desktop/ICloud Share/SNU/2020-2/Open_Course/R_lecture/2021.02/PCA/datasets")
raw_track <- read.csv("data/mens_track.csv", head=T)
```

```
dim(raw_track);summary(raw_track);
```

```
## [1] 55  9
```

```
##       m100            m200            m400            m800      
##  Min.   : 9.93   Min.   :19.72   Min.   :43.86   Min.   :1.700  
##  1st Qu.:10.27   1st Qu.:20.59   1st Qu.:45.56   1st Qu.:1.755  
##  Median :10.41   Median :20.81   Median :46.10   Median :1.790  
##  Mean   :10.47   Mean   :20.94   Mean   :46.44   Mean   :1.793  
##  3rd Qu.:10.59   3rd Qu.:21.29   3rd Qu.:47.30   3rd Qu.:1.815  
##  Max.   :12.18   Max.   :23.20   Max.   :52.94   Max.   :2.020  
##      m1500           m3000          mystery         marathon     
##  Min.   :3.510   Min.   :13.01   Min.   :27.38   Min.   :128.2  
##  1st Qu.:3.600   1st Qu.:13.28   1st Qu.:27.70   1st Qu.:130.7  
##  Median :3.640   Median :13.50   Median :28.19   Median :132.3  
##  Mean   :3.698   Mean   :13.85   Mean   :28.99   Mean   :136.6  
##  3rd Qu.:3.770   3rd Qu.:14.14   3rd Qu.:29.87   3rd Qu.:139.3  
##  Max.   :4.240   Max.   :16.70   Max.   :35.38   Max.   :164.7  
##    country         
##  Length:55         
##  Class :character  
##  Mode  :character  
##                    
##                    
##
```

Data Description

- 차원은 is 55 X 9, 그리고 9번째 열은 라벨링 되어 있는 데이터이다.

```
head(raw_track)
```

```
##    m100  m200  m400 m800 m1500 m3000 mystery marathon  country
## 1 10.39 20.81 46.84 1.81  3.70 14.04   29.36   137.72 argentin
## 2 10.31 20.06 44.84 1.74  3.57 13.28   27.66   128.30 australi
## 3 10.44 20.81 46.82 1.79  3.60 13.26   27.72   135.90  austria
## 4 10.34 20.68 45.04 1.73  3.60 13.22   27.45   129.95  belgium
## 5 10.28 20.58 45.91 1.80  3.75 14.68   30.55   146.62  bermuda
## 6 10.22 20.43 45.21 1.73  3.66 13.62   28.62   133.13   brazil
```

```
country <- raw_track[,9]
track <- raw_track[,1:8]
rownames(track) <- country
```

Principal Component Analysis (PCA) by R built-in function

- 데이터 행렬의 centering과 scaling을 위해서 함수 'scale'을 사용.
- PCA를 위해서 'prcomp' 함수를 사용.

```
track_scale <- scale(track, center = T, scale = T)
sum(track_scale[,1]) #verifying
```

```
## [1] -5.608014e-14
```

```
pca_track       <- prcomp(track, scale = T)
pca_track_scale <- prcomp(track_scale, scale = F)
```

- 데이터가 스케일링이 필요하다면, 'scale' 함수를 이용한 이후, prcomp를 사용하거나, prcomp를 즉각 사용한다.
- 두 가지의 결과는 동일하다.

prcomp 함수의 output

```
names(pca_track)
```

```
## [1] "sdev"     "rotation" "center"   "scale"    "x"
```

```
pca_track$sdev; # vector
pca_track$x; # matrix
pca_track$rotation; #matrix
```

- function 'procomp'의 결과는 크게 3가지이다. sdev, rotation, x가 있다.

1) sdev : Principal component의 standard deviation
2) x : PC score
3) rotation : PC loading (앞의 정의에서 ϕ들을 모은 행렬)

부수적으로, center, scale : mean and sd (center, scale이 TRUE인 경우에 주어지며, centering과 scaling에 사용된 데이터의 평균과 분산)

Standard deviations

```
pca_summary <- summary(pca_track);pca_summary
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5    PC6    PC7
## Standard deviation     2.5734 0.9368 0.39915 0.35221 0.28263 0.2607 0.2155
## Proportion of Variance 0.8278 0.1097 0.01992 0.01551 0.00999 0.0085 0.0058
## Cumulative Proportion  0.8278 0.9375 0.95739 0.97289 0.98288 0.9914 0.9972
##                           PC8
## Standard deviation     0.15033
## Proportion of Variance 0.00283
## Cumulative Proportion  1.00000
```

```
round(pca_summary$importance, 2)
```

```
##                           PC1  PC2  PC3  PC4  PC5  PC6  PC7  PC8
## Standard deviation     2.57 0.94 0.40 0.35 0.28 0.26 0.22 0.15
## Proportion of Variance 0.83 0.11 0.02 0.02 0.01 0.01 0.01 0.00
## Cumulative Proportion  0.83 0.94 0.96 0.97 0.98 0.99 1.00 1.00
```

```
pca_summary$importance[1,];pca_summary$importance[2,];pca_summary$importance[3,];
```

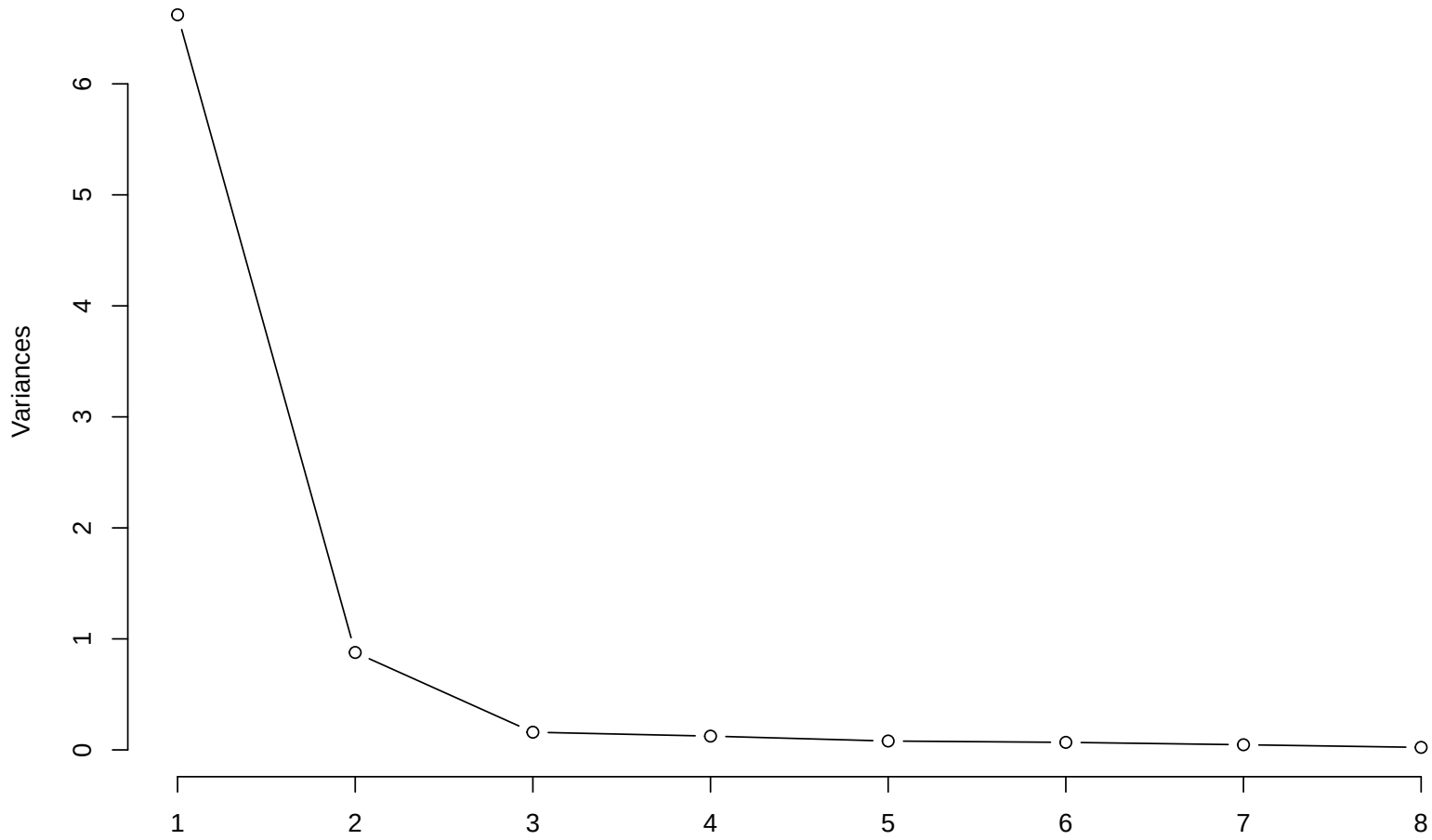
```
##       PC1       PC2       PC3       PC4       PC5       PC6       PC7       PC8
## 2.5733531 0.9368128 0.3991505 0.3522065 0.2826310 0.2607013 0.2154519 0.1503333
```

```
##     PC1     PC2     PC3     PC4     PC5     PC6     PC7     PC8
## 0.82777 0.10970 0.01992 0.01551 0.00999 0.00850 0.00580 0.00283
```

```
##     PC1     PC2     PC3     PC4     PC5     PC6     PC7     PC8
## 0.82777 0.93747 0.95739 0.97289 0.98288 0.99137 0.99717 1.00000
```

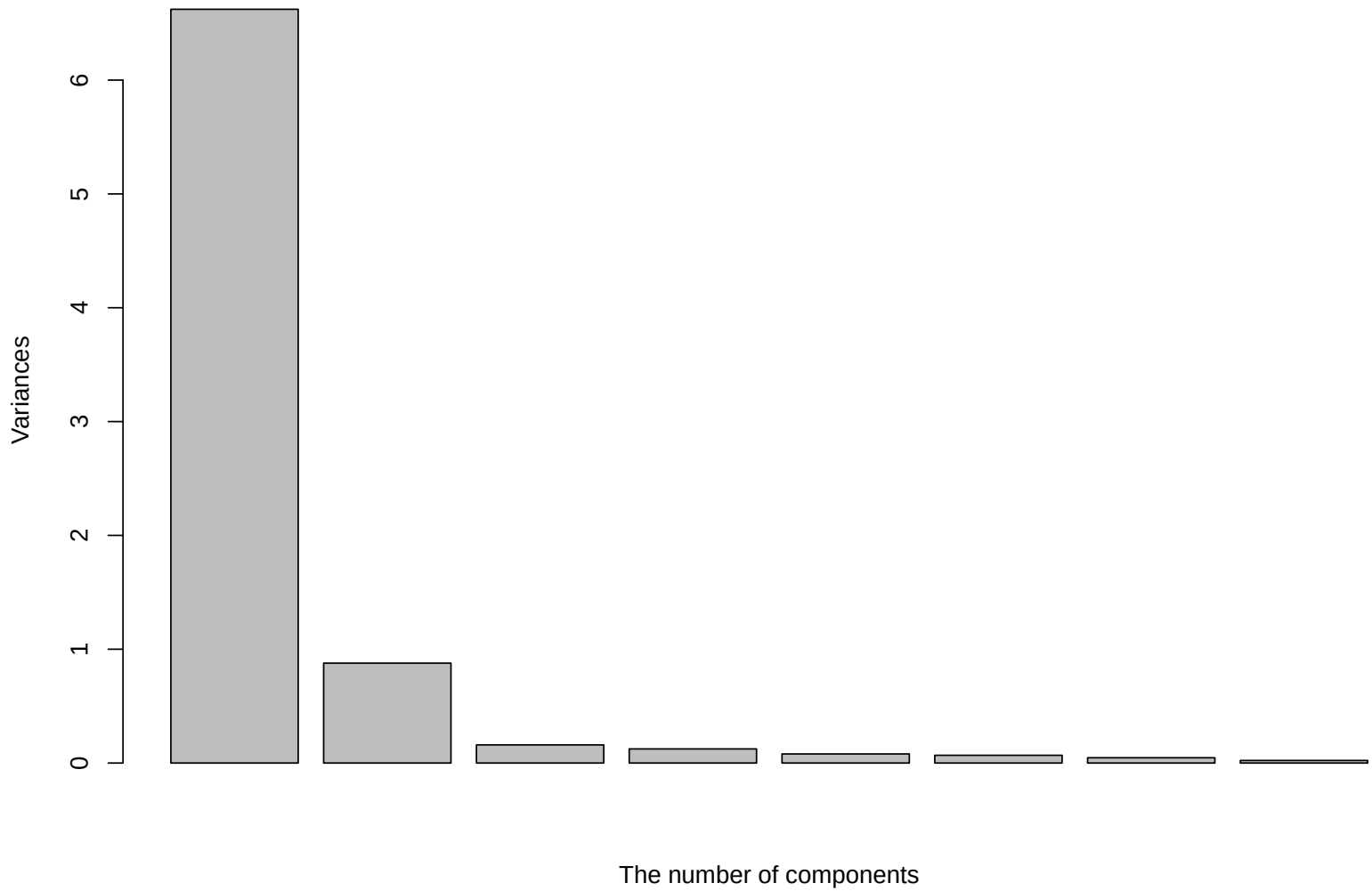
Screeplot

```
screeplot(pca_track, main = "", col = "black", type = "lines", pch =1)
```



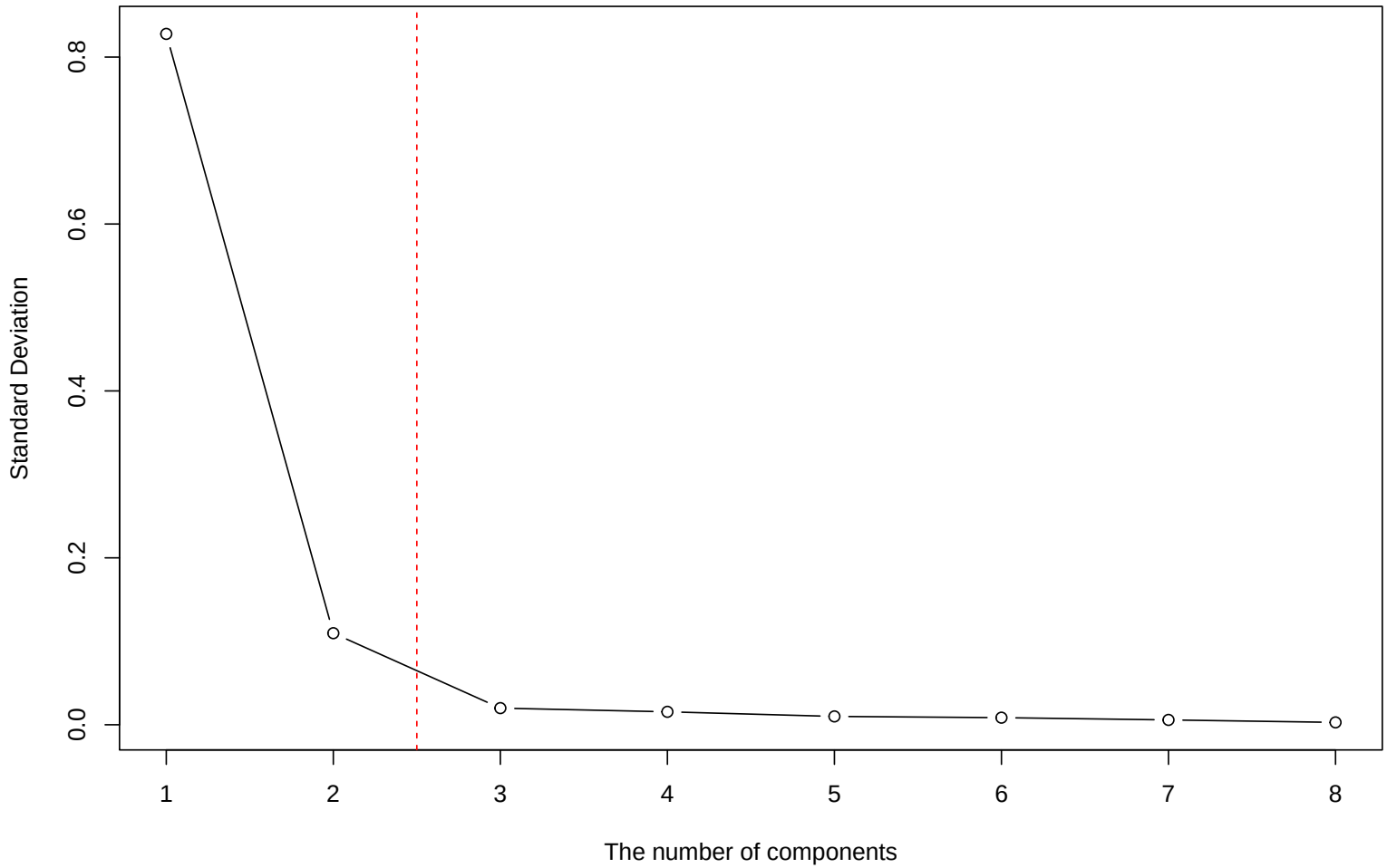
Screeplot2

```
screeplot(pca_track, main = "", xlab = "The number of components")
```



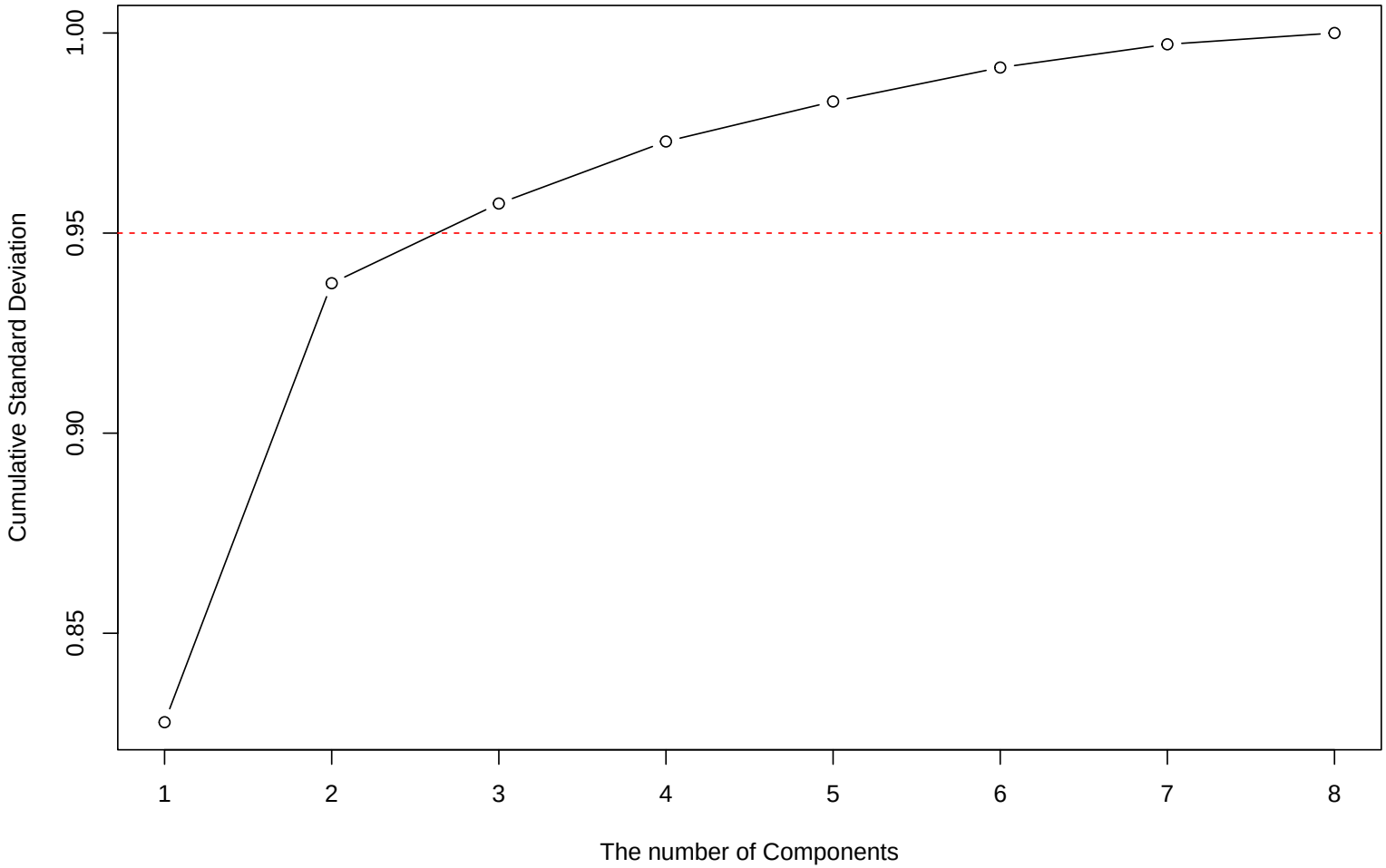
Screeplot3

```
plot(pca_track$sdev^2/sum(pca_track$sdev^2), type = "b",
     xlab = "The number of components", ylab = "Standard Deviation")
abline(v = 2.5, lty = 2, col = "red") # abline reference: https://thebook.io/006723/
```



Screeplot4

```
plot(cumsum(pca_track$sdev^2)/sum(pca_track$sdev^2), type = "b",
     xlab = "The number of Components", ylab = "Cumulative Standard Deviation")
abline(h = 0.95, lty = 2, col = "red")
```



Score and basis

```
head(pca_track$x)
```

PC1 순서 내보기

- 첫 번째 Principal Component의 score를 통해 순서를 매겨보자.
- 'order'라는 함수 사용

```
d <- c(0.1, 0.7, 0.3, 0.5, 0.8, 0.6, 0.4, 0.2)
tmp <- order(d)
d[tmp]
```

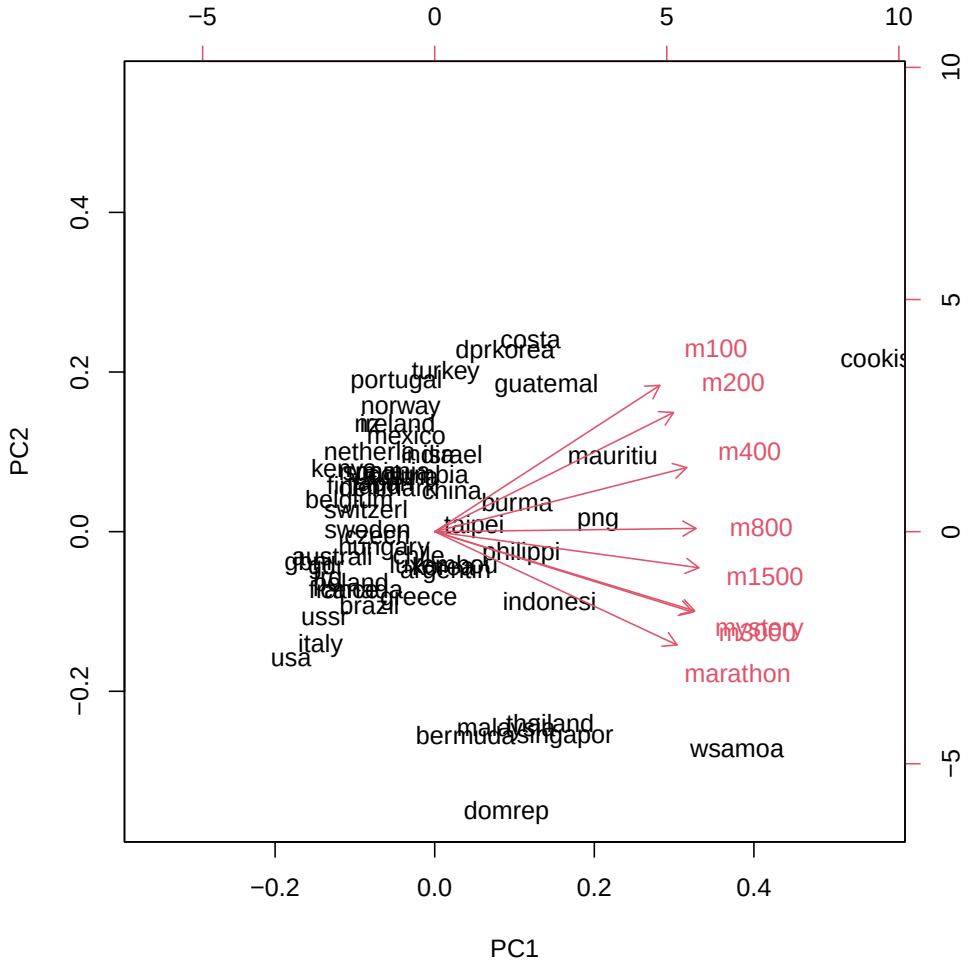
```
## [1] 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8
```

```
track_PC1<- pca_track$x[,1]
ord <- order(track_PC1)
track_PC1[ord]
```

```
##        usa       gbni      italy       ussr        gdr        frg   australi
## -3.4305560 -3.0242302 -2.7269499 -2.6268513 -2.5900916 -2.5527441 -2.4463715
##     france      kenya    belgium     poland     canada    finland   switzerl
## -2.1718986 -2.1683197 -2.0412573 -2.0006142 -1.7463509 -1.6920244 -1.6389715
##     sweden         nz     brazil   netherla      spain      czech      japan
## -1.6032283 -1.5997095 -1.5582604 -1.5554344 -1.4805855 -1.3725563 -1.2378683
##    hungary    rumania    denmark   portugal    ireland     norway    austria
## -1.2051890 -1.1964889 -1.1132385 -0.9163725 -0.8841983 -0.8114855 -0.8076439
##     mexico   columbia      chile     greece      india      korea   luxembou
## -0.6785258 -0.3900672 -0.3810838 -0.3795895 -0.1652380  0.2075449  0.2205089
##   argentin     turkey      china     israel    bermuda     taipei   dprkorea
##  0.2618959  0.2660800  0.4089696  0.4345858  0.7392569  0.9505025  1.6836873
##   malaysia     domrep      burma   philippi      costa   guatemal   indonesi
```

PCA with Scale

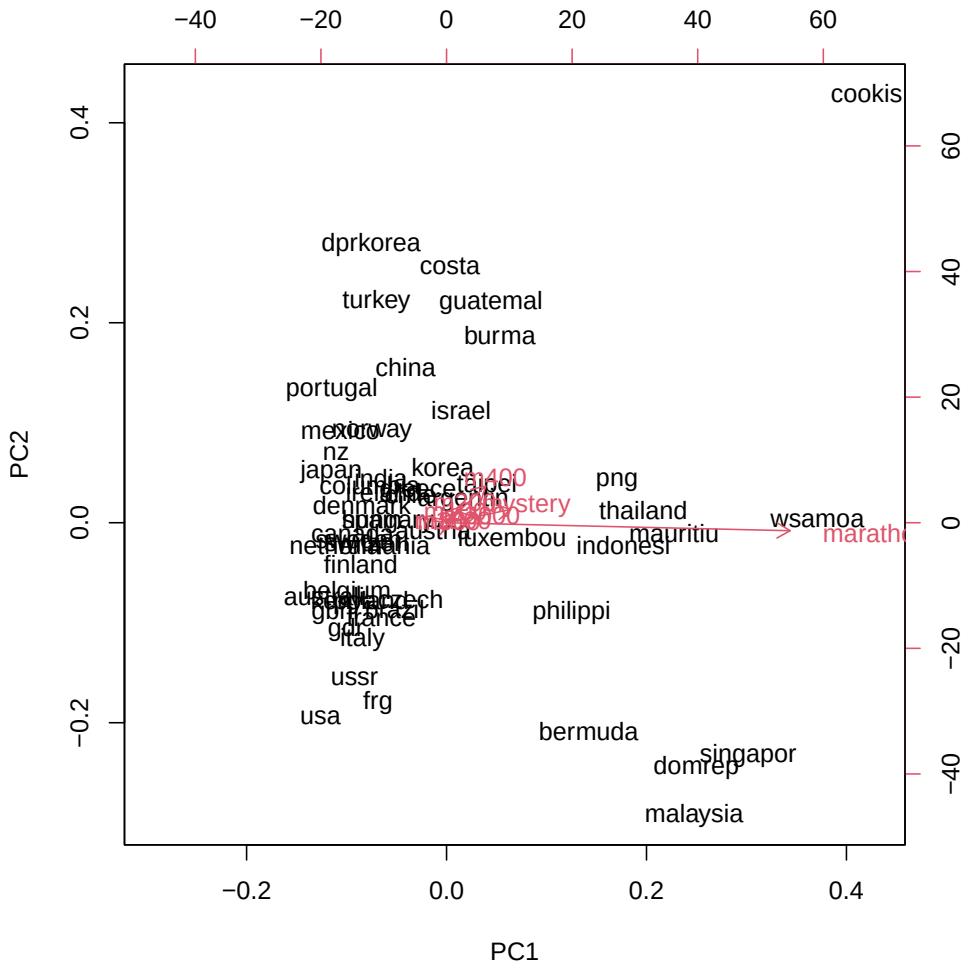
```
#PCA with scale
pca_track_ws <- prcomp(track, scale = T)
biplot(pca_track_ws)
```



PCA without scale

- 어느 것이 더 선호될까요?

```
pca_track_wos <- prcomp(track, scale = F)
biplot(pca_track_wos)
```



Connection between PCA and SVD (Singular Value Decomposition)

- Y : centered and scaled data matrix.
- Y의 Singular Value Decomposition에 의한 분해.

$$Y = UDV^t$$

```
track_scale <- scale(track, center = T, scale = T)
Y = track_scale; svd_Y = svd(Y)
track_U = svd_Y$u; track_V = svd_Y$v; track_D = diag(svd_Y$d)
sum(sqrt((track_U %*% track_D %*% t(track_V) - Y)^2))
```

```
## [1] 4.036355e-13
```

- 함수 prcomp는 크게 standard deviation(sdev), score(x), basis(rotation)를 제공합니다.
- prcomp의 output 중 'x'는 SVD의 결과물 중 U와 D의 곱으로 표현됩니다. score $= UD$.
- prcomp의 output 중 'rotation'는 V와 같습니다.

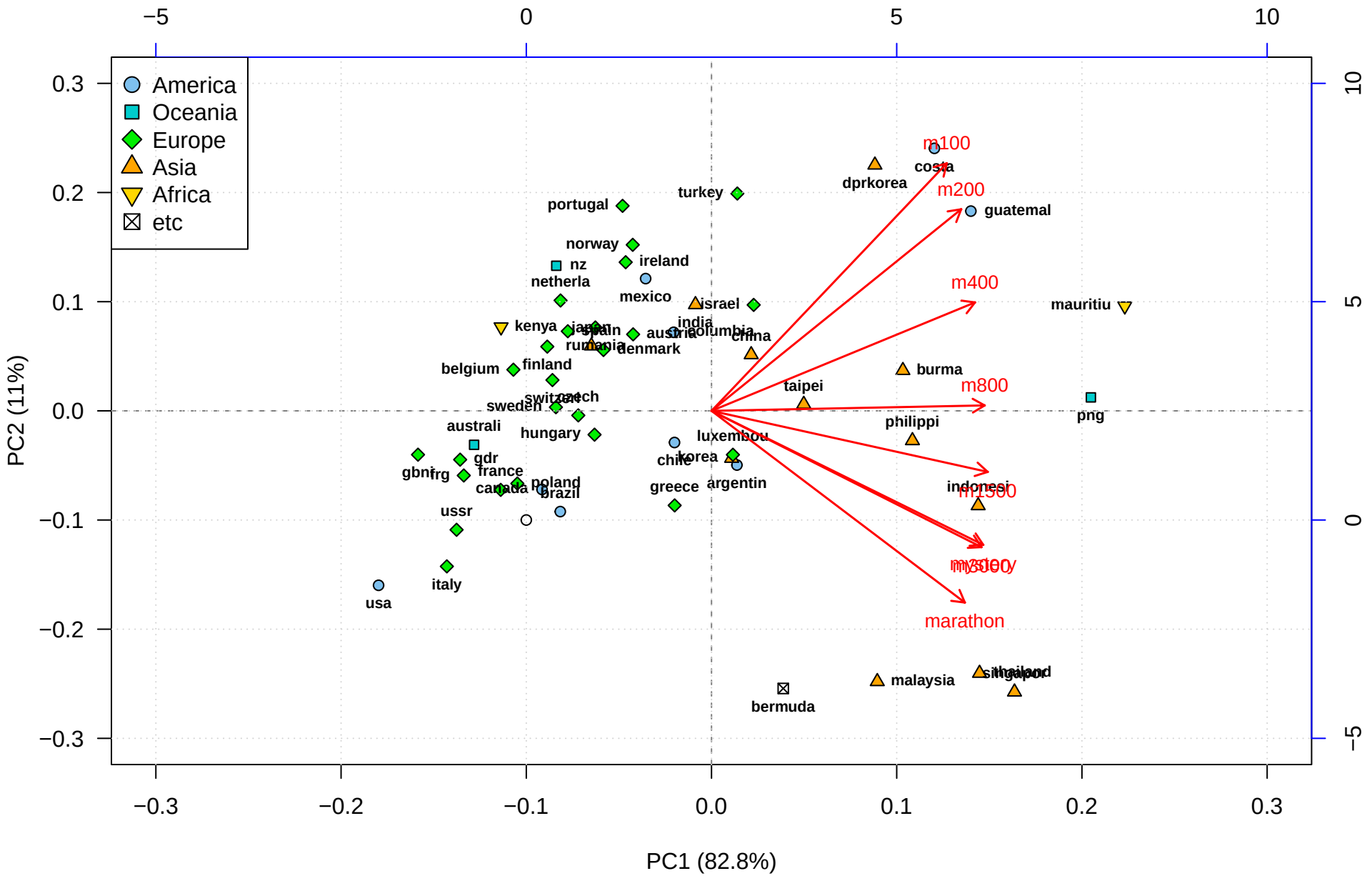
```
pca_track <- prcomp(track, scale = T)
sum(sqrt((pca_track$x - track_U %*% track_D)^2))
```

```
## [1] 2.26985e-13
```

```
sum(sqrt((pca_track$rotation - track_V )^2))
```

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

Biplot Manually



Biplot Manually Cont'd

```
raw_track_conti <- read.csv("data//mens_track_conti.csv", head=T)
dim(raw_track_conti); n = dim(raw_track_conti)[1]
```

```
## [1] 55 10
```

```
country <- raw_track_conti[9]
conti <- raw_track_conti[10]
score <- pca_track$x
```

```
color_names <- c("skyblue2", "cyan3", "green2", "orange1", "gold", "darkgrey")
#ref: https://www.r-graph-gallery.com/42-colors-names.html
col.group <- c()
col.group[conti=="America"] <- color_names[1]
col.group[conti=="Oceania"] <- color_names[2]
col.group[conti=="EU"]      <- color_names[3]
col.group[conti=="Asia"]    <- color_names[4]
col.group[conti=="Africa"]  <- color_names[5]
col.group[conti=="etc"]     <- color_names[6]
```

```
pch_names <- c(21:25, 7)
#ref: http://www.sthda.com/english/wiki/r-plot-pch-symbols-the-different-point-shapes-available-in-r
pch.group <- c()
pch.group[conti=="America"] <- pch_names[1]
pch.group[conti=="Oceania"] <- pch_names[2]
pch.group[conti=="EU"]      <- pch_names[3]
pch.group[conti=="Asia"]    <- pch_names[4]
pch.group[conti=="Africa"]  <- pch_names[5]
pch.group[conti=="etc"]     <- pch_names[6]
```

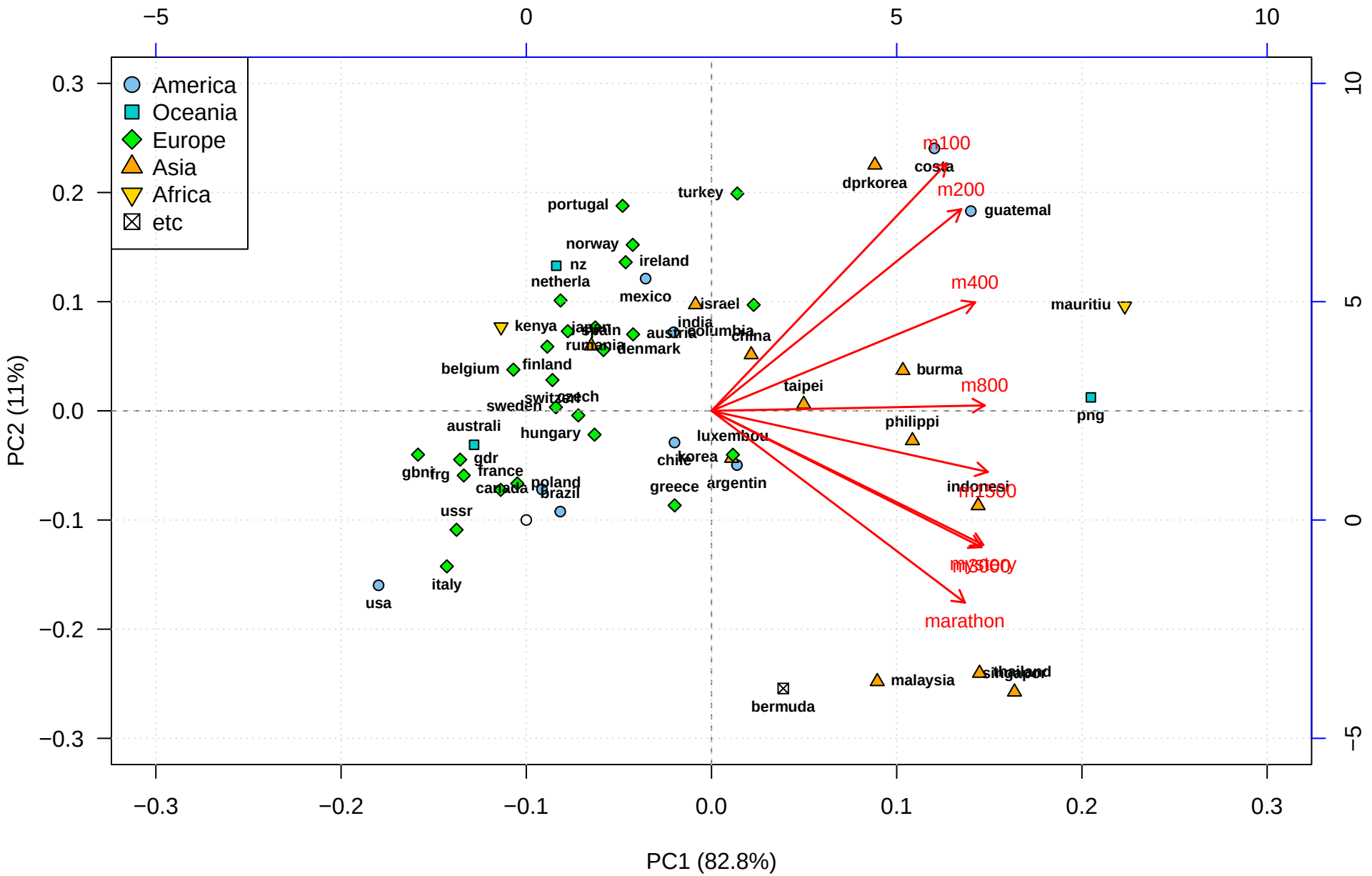
Biplot Manually (Cont'd)

```
plt_xlim = c(-0.3, 0.3)
plt_ylim = c(-0.3, 0.3)
sdev<- pca_track$sdev
dev <- round(sdev^2/sum(sdev^2) , 3) * 100

# Label position
l.pos <- c() # Create a vector of y axis coordinates
lo <- which(track_basis[,2] < 0) # Get the variables on the bottom half of the plot
hi <- which(track_basis[,2] > 0) # Get variables on the top half

# Replace values in the vector
l.pos <- replace(l.pos, lo, "1")
l.pos <- replace(l.pos, hi, "3")
```

Biplot Manually (Cont'd)

```
plot(score[,1]/(sdev[1] * sqrt(n)), score[,2]/(sdev[2] * sqrt(n)), xlab=paste0("PC1 (", dev[1] , "%)"), ylab = p
     pch = pch.group,col="black", bg=col.group, cex=1, las=1, xlim = plt_xlim, ylim = plt_ylim)
abline(v=0, lty=2, col="grey50")
abline(h=0, lty=2, col="grey50")

# Add labels
text(score[,1]/(sdev[1] * sqrt(n)), score[,2]/(sdev[2] * sqrt(n)),
     labels=row.names(score), pos=c(1,3,4,2), font=2, cex = .7)
#pos: Values of 1,2,3 and 4, respectively indicate positions below, to the left of, above and to the right of the
#specified coordinate in fractions of a character with.
#Basis
track_basis <- as.matrix(pca_track$rotation[,1:2])/2.5 #matrix
points(track_basis[,1], track_basis[,2], cex = .1, col="red")
grid()
#Add arrows
arrows(x0=0, x1=track_basis[,1], y0=0, y1=track_basis[,2], code = 2, col="red", length=0.1, lwd=1.5)

# Variable labels
text(track_basis[,1], track_basis[,2], labels=row.names(track_basis), col="red", pos=l.pos, cex = .8)

## Allow a second plot on the same graph
par(new=TRUE)
plot(0,0, axes = FALSE, xlab = "", ylab = "", xlim = c(-5,10), ylim = c(-5, 10))
axis(3,xlim= c(-5,10), col="blue",col.axis="black", at = seq(-5,10, by =5))
axis(4,ylim= c(-5,10), col="blue",col.axis="black", at = seq(-5,10, by =5))

legend("topleft", legend=c("America", "Oceania", "Europe", "Asia", "Africa", "etc"), col="black",
      pt.bg=color_names, pch=pch_names, pt.cex=1.5)
```

Biplot Manually (Cont'd)



해보기

- National Track Records (Women) 으로 PCA 분석해보기.

1. 데이터 행렬을 scaling 해보기.
2. Screeplot을 통해 선택되어야 할 Principal Components 갯수 생각해보기.
3. 두 개의 Principal Components 해석하기.
4. 첫 번째 Principal Components를 Ranking 매겨보기.
5. 이 자료를 바탕으로 직접 Biplot을 그려보고 직접 자료를 해석해보기.