Lab10_PCA(2)

Lab10 PCA(2)

```
01. 주성분 분석 배경
02. 주성분이란?
03. 주성분분석(principal component analysis)란?
04. 주성분 분석 예제1
05. 주성분 분석 실습과제1
```

01. 주성분 분석 배경

```
통계 자료를 분석 시에 주요 관심사 중의 하나는
(1) 자료가 가지는 변이의 크기와 속성을 파악하고 해석하는 것.

변이 -
변수가 하나(일변량) : 분산으로 크기 표현
변수가 두개이상(다변량) : 개별 분산 및 변수들간의 공분산(혹은 상관계수)가 포함.
```

02. 주성분이란?

주성분이란? 다변량 자료가 가지고 있는 총 변이의 주요 부분을 함유하고 있는 성분

예를 들어, n명에 대한 p과목의 성적 X1, X2, ... ,Xn를 관찰했다.

```
      X1 X2  X3  X4 ... Xn
stu1  90 100  .  80      .
stu2  80 100  .  80      .
stu3  70  90  .  80      .
..
stuN  ...
```

각 학생을 대표하는 특성을 측정하고 싶다면,
우리는 3가지 방법을 생각해 볼 수 있다.
(가) 각 과목의 평균성적
(나) 가중 평균 성적 Xa = a1X1 + a2X2 + a3X3 + .. + anXn
(다) f를 이용한 Xf = f(X1, X2, ..., Xp)

대표한다는 의미는 주성분의 의미와 연계될 수 있다.

03. 주성분분석(principal component analysis)란?

서로 연관되어 있는 변수들(X1, X2,...,Xp)을 관측시에,
이 변수들이 전체적으로 가지고 있는 정보를 최대한 확보하는 작은 수의
새로운 변수(주성분)를 생성하는 방법.

그렇지만 이 변수들은 선형 독립(무상관)을 가지는 변수들이면 좋겠다.

04. 주성분 분석 예제

(1) 자료 가져오기 및 요약 통계량

```
# install.packages("HSAUR")
library(HSAUR)
```

```
## Warning: package 'HSAUR' was built under R version 3.3.3
```

```
## Loading required package: tools
```

```
data(heptathlon)
head(heptathlon)
```

```
##                      hurdles highjump  shot run200m longjump javelin
## Joyner-Kersee (USA)    12.69     1.86 15.80   22.56     7.27   45.66
## John (GDR)             12.85     1.80 16.23   23.65     6.71   42.56
## Behmer (GDR)           13.20     1.83 14.20   23.10     6.68   44.54
## Sablovskaite (URS)     13.61     1.80 15.23   23.92     6.25   42.78
## Choubenkova (URS)      13.51     1.74 14.76   23.93     6.32   47.46
## Schulz (GDR)           13.75     1.83 13.50   24.65     6.33   42.82
##                      run800m score
## Joyner-Kersee (USA)   128.51  7291
## John (GDR)            126.12  6897
## Behmer (GDR)          124.20  6858
## Sablovskaite (URS)    132.24  6540
## Choubenkova (URS)     127.90  6540
## Schulz (GDR)          125.79  6411
```

http://cran.r-project.org/web/packages/HSAUR/HSAUR.pdf (http://cran.r-project.org/web/packages/HSAUR/HSAUR.pdf)

```
1988년 서울 올림픽 육상 여성 7종 경기에 대한 결과

hurdles(110m 허들)
highjump(높이뛰기)
shot(포환 던지기)
run200m(200m 달리기)
longjump(멀리뛰기)
javelin(창던지기)
run800m(800m 달리기)
```

(2) 데이터 변환

```
변수들 중 hurdles, run200m, run800m는 작은 값일수록 좋은 점수이기 때문에
자료를 변형시켜 준다.
즉, 높은 점수가 되도록 최대값에서 실제값을 빼준다.
```

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

(3) 주성분 분석 실행하기

```
A. stats 패키지를 불러온다.
B. score 변수 제외(8열 제외)
C. princomp() 함수 이용
  cor=T 상관계수행렬 지정, cor=F 공분산 행렬 의미
  scores = T, 주 성분의 점수를 출력 옵션

아래에서는 변수의 단위가 상이하므로 상관계수행렬을 이용하여 주성분분석을
실행하였다.

names(dat.pca)는 princomp() 결과의 개체 이름을 보여준다.
```

```
library(stats)
names(dat)
```

```
## [1] "hurdles"  "highjump" "shot"     "run200m"  "longjump" "javelin"
## [7] "run800m"  "score"
```

```
hep.data = dat[,-8]   # score 변수 제외
var(hep.data)
```

```
##            hurdles  highjump      shot   run200m  longjump    javelin
## hurdles  0.5426500 0.0465875 0.7158125 0.5526083 0.3186333  0.0202750
## highjump 0.0465875 0.0060750 0.0512550 0.0368525 0.0289200  0.0005950
## shot     0.7158125 0.0512550 2.2257190 0.9874603 0.5257018  1.4228727
## run200m  0.5526083 0.0368525 0.9874603 0.9400410 0.3757313  1.1449063
## longjump 0.3186333 0.0289200 0.5257018 0.3757313 0.2248773  0.1128357
## javelin  0.0202750 0.0005950 1.4228727 1.1449063 0.1128357 12.5716773
## run800m  4.7594000 0.3820250 5.1904192 4.9583408 2.7502933 -0.5893900
##            run800m
## hurdles   4.759400
## highjump  0.382025
## shot      5.190419
## run200m   4.958341
## longjump  2.750293
## javelin  -0.589390
## run800m  68.742142
```

- 변수와 변수가 만나는 대각선은 분산 의미.
- 다른 것은 공분산을 의미.
- 각 변수별 확인 가능.

```
dat.pca = prcomp(hep.data, scale=T)
names(dat.pca)
```

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

```
dat.pca
```

```
## Standard deviations:
## [1] 2.1119364 1.0928497 0.7218131 0.6761411 0.4952441 0.2701029 0.2213617
##
## Rotation:
##                  PC1         PC2         PC3         PC4         PC5
## hurdles  -0.4528710  0.15792058 -0.04514996  0.02653873 -0.09494792
## highjump -0.3771992  0.24807386 -0.36777902  0.67999172  0.01879888
## shot     -0.3630725 -0.28940743  0.67618919  0.12431725  0.51165201
## run200m  -0.4078950 -0.26038545  0.08359211 -0.36106580 -0.64983404
## longjump -0.4562318  0.05587394  0.13931653  0.11129249 -0.18429810
## javelin  -0.0754090 -0.84169212 -0.47156016  0.12079924  0.13510669
## run800m  -0.3749594  0.22448984 -0.39585671 -0.60341130  0.50432116
##                  PC6         PC7
## hurdles  -0.78334101  0.38024707
## highjump  0.09939981 -0.43393114
## shot     -0.05085983 -0.21762491
## run200m   0.02495639 -0.45338483
## longjump  0.59020972  0.61206388
## javelin  -0.02724076  0.17294667
## run800m   0.15555520 -0.09830963
```

```
dat.pca = princomp(hep.data, cor=T)
names(dat.pca)
```

```
## [1] "sdev"     "loadings" "center"   "scale"    "n.obs"    "scores"
## [7] "call"
```

```
dat.pca
```

```
## Call:
## princomp(x = hep.data, cor = T)
##
## Standard deviations:
##    Comp.1    Comp.2    Comp.3    Comp.4    Comp.5    Comp.6    Comp.7
## 2.1119364 1.0928497 0.7218131 0.6761411 0.4952441 0.2701029 0.2213617
##
##  7  variables and  25 observations.
```

(4) 주성분 분석 결과

```
총 7개의 주성분(변수의 수가 7개이므로)가능한 주성분 수도 7개이다.
표준편차를 보여준다.
```

```
summary(dat.pca)
```

```
## Importance of components:
##                           Comp.1    Comp.2     Comp.3     Comp.4
## Standard deviation     2.1119364 1.0928497 0.72181309 0.67614113
## Proportion of Variance 0.6371822 0.1706172 0.07443059 0.06530955
## Cumulative Proportion  0.6371822 0.8077994 0.88222998 0.94753952
##                           Comp.5     Comp.6      Comp.7
## Standard deviation     0.49524412 0.27010291 0.221361710
## Proportion of Variance 0.03503811 0.01042223 0.007000144
## Cumulative Proportion  0.98257763 0.99299986 1.000000000
```

(5) 각 주성분의 표준편차, 분산 비율

첫 번째 주성분이 63.72%, 두번째 주성분이 17.06% 분산 비율.
2개의 주성분이 80.8%의 정보를 갖는다.

각 주성분의 표준편차를 제곱하여 고유값을 얻을 수 있다.

제 1성분의 분산이 4.46, 제 2주성분 1.19, 제 3주성분 0.52

dat.pca 열 중에 sdev가 표준편차이다.

```
eig.val = dat.pca$sdev^2
eig.val
```

```
##     Comp.1     Comp.2     Comp.3     Comp.4     Comp.5     Comp.6
## 4.46027516 1.19432056 0.52101413 0.45716683 0.24526674 0.07295558
##     Comp.7
## 0.04900101
```

(6) PC1~PC7의 분산의 합과 상관계수의 대각 부분의 합은 같다.

```
sum(eig.val)
```

```
## [1] 7
```

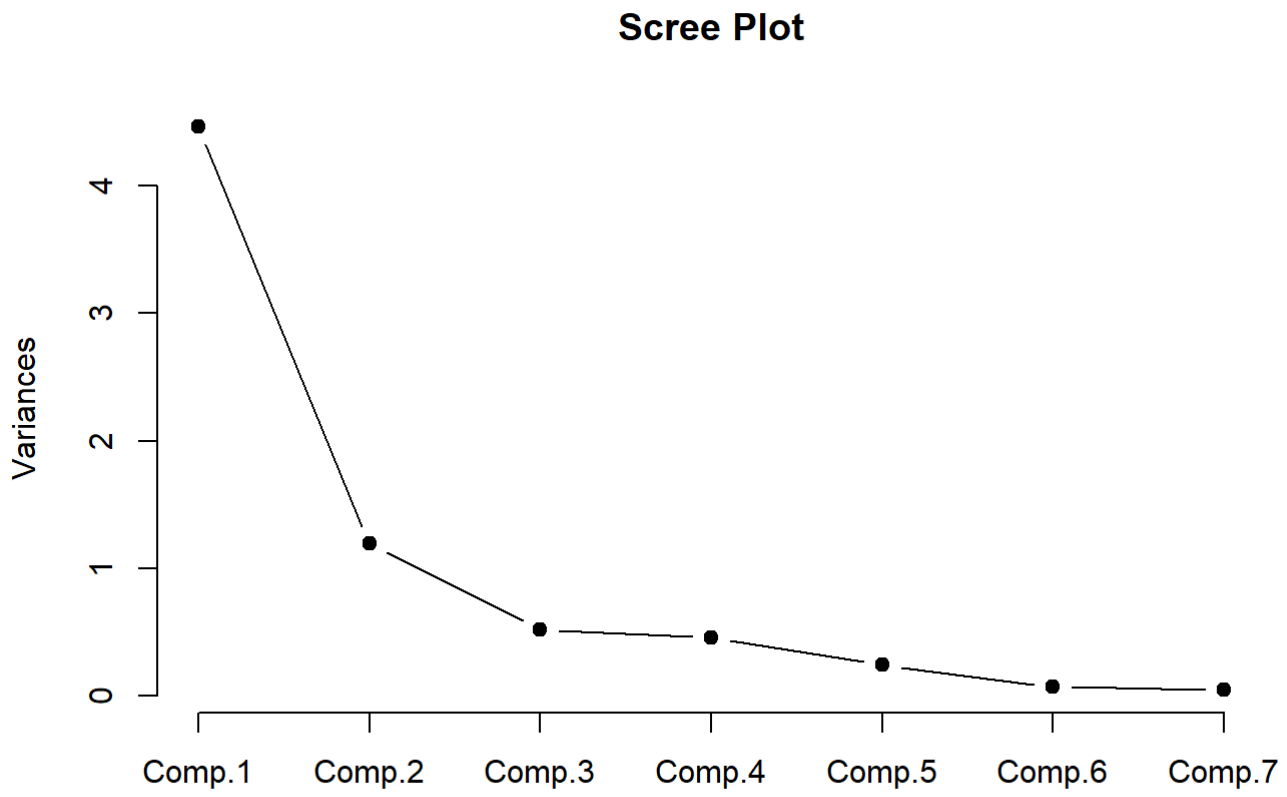
```
cor(dat[,1:7])
```

```
##              hurdles    highjump      shot   run200m   longjump
## hurdles  1.000000000 0.811402536 0.6513347 0.7737205 0.91213362
## highjump 0.811402536 1.000000000 0.4407861 0.4876637 0.78244227
## shot     0.651334688 0.440786140 1.0000000 0.6826704 0.74307300
## run200m  0.773720543 0.487663685 0.6826704 1.0000000 0.81720530
## longjump 0.912133617 0.782442273 0.7430730 0.8172053 1.00000000
## javelin  0.007762549 0.002153016 0.2689888 0.3330427 0.06710841
## run800m  0.779257110 0.591162823 0.4196196 0.6168101 0.69951116
##               javelin     run800m
## hurdles   0.007762549  0.77925711
## highjump  0.002153016  0.59116282
## shot      0.268988837  0.41961957
## run200m   0.333042722  0.61681006
## longjump  0.067108409  0.69951116
## javelin   1.000000000 -0.02004909
## run800m  -0.020049088  1.00000000
```

(7) 스크리 그림과 주성분 계수

- type : 선으로 보겠다.
- pch=19 : 점의 표시 형식
- main : 그래프 제목

```
screeplot(dat.pca, type="lines", pch=19, main="Scree Plot")
```




```
dat.pca$loadings[,1:2]   # 주성분 1,2만 보기
```

```
##              Comp.1      Comp.2
## hurdles  -0.4528710  0.15792058
## highjump -0.3771992  0.24807386
## shot     -0.3630725 -0.28940743
## run200m  -0.4078950 -0.26038545
## longjump -0.4562318  0.05587394
## javelin  -0.0754090 -0.84169212
## run800m  -0.3749594  0.22448984
```

```
dat.pca$loadings
```

```
##
## Loadings:
##          Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7
## hurdles  -0.453  0.158                       0.783 -0.380
## highjump -0.377  0.248 -0.368 -0.680               0.434
## shot     -0.363 -0.289  0.676 -0.124 -0.512        0.218
## run200m  -0.408 -0.260         0.361  0.650        0.453
## longjump -0.456         0.139 -0.111  0.184 -0.590 -0.612
## javelin         -0.842 -0.472 -0.121 -0.135       -0.173
## run800m  -0.375  0.224 -0.396  0.603 -0.504 -0.156
##
##                Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7
## SS loadings     1.000  1.000  1.000  1.000  1.000  1.000  1.000
## Proportion Var  0.143  0.143  0.143  0.143  0.143  0.143  0.143
## Cumulative Var  0.143  0.286  0.429  0.571  0.714  0.857  1.000
```

```
hurdles  -0.4528710  0.15792058
highjump -0.3771992  0.24807386
shot     -0.3630725 -0.28940743
run200m  -0.4078950 -0.26038545
longjump -0.4562318  0.05587394
javelin  -0.0754090 -0.84169212
run800m  -0.3749594  0.22448984
```

주성분의 고유값의 크기 순으로 그린 것으로 screeplot() 함수를 이용한다.
고유값이 1보다 큰 주성분이 2개가 됨을 알 수 있다.
Var(PC1), Var(PC2)

상관계수행렬을 이용한 경우 제1, 제2주성분은 다음과 같다.
PC1 = -0.453x hurdles - 0.377xhighjump - 0.363shot ... -0.075xjavalin - 0.375 x run800m
PC2 = 0.158xhurdles + 0.248 x highjump - 0.289 x shot - 0.2603 x run200m + 0.056 x longjump -0.84xjavalin + 0.22xrun800m

제 2주성분은 javalin의 계수가 다른 변수에 비해 상대적으로 절대값이 큰 것으로 볼때,
창던지기와 밀접한 관련이 있는 성분으로 파악이 가능하다.

(8) 주성분 점수 및 행렬도(biplot)

각 개체에 대한 첫 번째, 두 번째 주성분점수 및 행렬도이다.
행렬도(biplot)는 각 개체의 관찰값은 주성분 점수로 한다.
각 변수와 주성분과의 관계를 나타낸다.

(가) Gabriel(1971)에 의해 제안된 방법.
다변량 자료가 가지는 정보를 기하하적으로 탐색하는 방법.

첫 번째 주성분을 X축,
두 번째 주성분을 Y축으로 하여,
각 변수(육상 7종 경기)와
각 객체(운동 선수)의 산점도를 나타내었다.
화살표는 벡터를 의미.

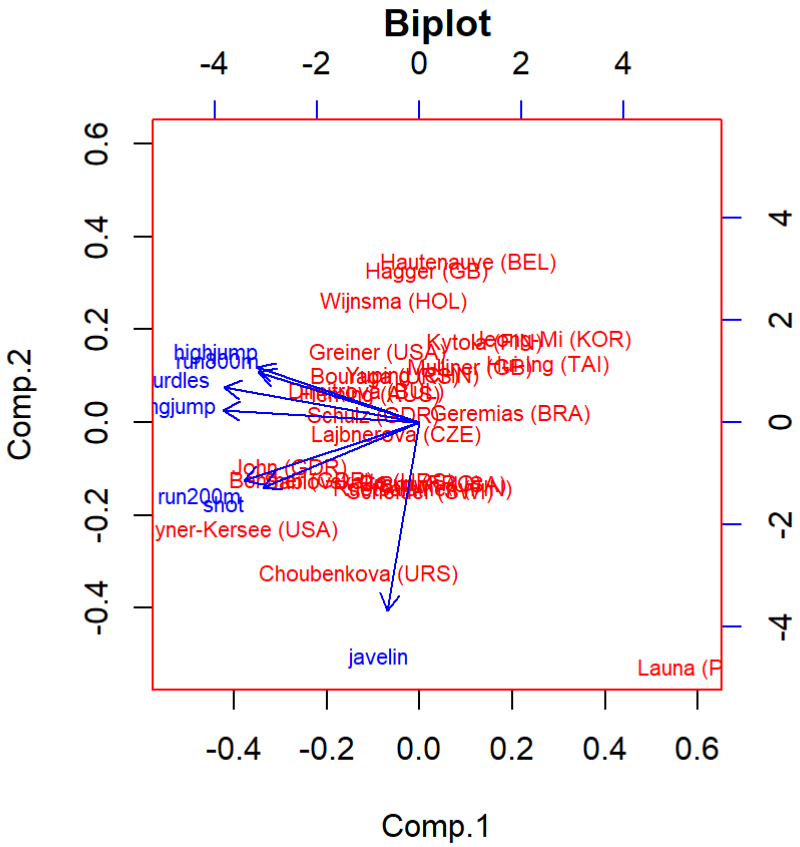
(나) 행렬도에서 (highjump, run800m, hurdles, longjump)가 서로 가까운 곳에 위치하고,
벡터 방향 또한 비슷하다.
(run200m, shot)이 가깝게 위치하고, javalin은 다른 변수들과 다른 방향에 위치한다.

--> 가까운 거리와 방향일수록 변수들의 상관성이 높아진다.
--> 각 개체가 특정 변수에 가깝게 위치할수록 그 개체는 해당 변수와 관련이 높다고 할 수 있다.

```
dat.pca$scores[,1:2]
```

```
##                          Comp.1      Comp.2
## Joyner-Kersee (USA) -4.20643487 -1.26802363
## John (GDR)          -2.94161870 -0.53452561
## Behmer (GDR)        -2.70427114 -0.69275901
## Sablovskaite (URS)  -1.37105209 -0.70655862
## Choubenkova (URS)   -1.38704979 -1.78931718
## Schulz (GDR)        -1.06537236  0.08104469
## Fleming (AUS)       -1.12307639  0.33042906
## Greiner (USA)       -0.94221015  0.82345074
## Lajbnerova (CZE)    -0.54118484 -0.14933917
## Bouraga (URS)       -0.77548704  0.53686251
## Wijnsma (HOL)       -0.56773896  1.42507414
## Dimitrova (BUL)     -1.21091937  0.36106077
## Scheider (SWI)       0.01578005 -0.82307249
## Braun (FRG)          0.00385205 -0.72953750
## Ruotsalainen (FIN)   0.09261899 -0.77877955
## Yuping (CHN)        -0.14005513  0.54831883
## Hagger (GB)          0.17465745  1.77914066
## Brown (USA)          0.52996001 -0.74195530
## Mulliner (GB)        1.14869009  0.64788023
## Hautenauve (BEL)     1.10808552  1.88531477
## Kytola (FIN)         1.47689483  0.94353198
## Geremias (BRA)       2.05556037  0.09495979
## Hui-Ing (TAI)        2.93969248  0.67514662
## Jeong-Mi (KOR)       3.03136461  0.97939889
## Launa (PNG)          6.39931438 -2.89774561
```

```
biplot(dat.pca, cex=0.7, col=c("red", "blue"), main="Biplot")
```



(9) 변수의 상관계수 확인

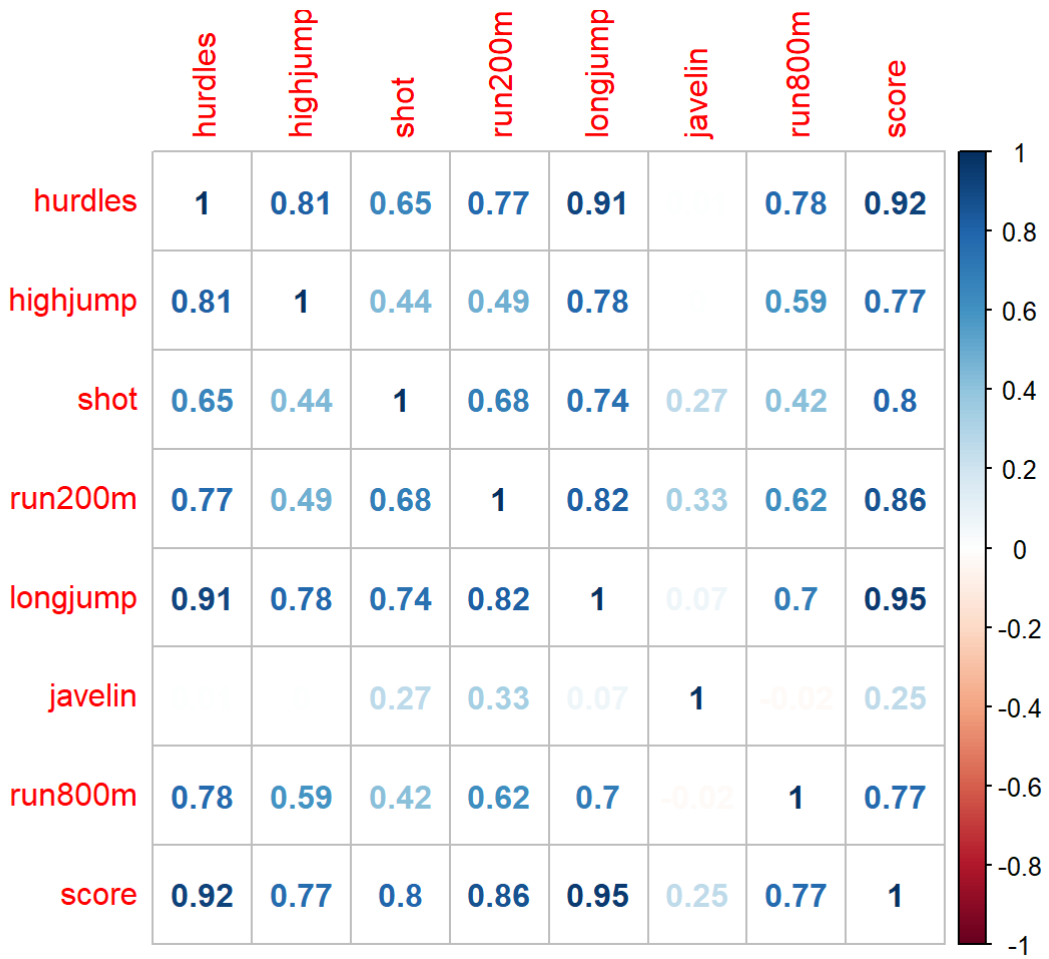
행렬도에서 (highjump, run800m, hurdles, longjump)가 서로 가까운 곳에 위치하고,
벡터 방향 또한 비슷하다.
(run200m, shot)이 가깝게 위치하고, javalin은 다른 변수들과 다른 방향에 위치한다.

```
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 3.3.3
```

```
## corrplot 0.84 loaded
```

```
CorrEu <- cor(dat)
corrplot(CorrEu, method="number")
```



05. 실습과제 1

http://data-mining-tutorials.blogspot.kr/2013/01/new-features-for-pca-in-tanagra.html
99명의 소비자가 맥주 구매에 중요하게 생각하는 요인에 대해 점수를 준 엑셀 자료이다.
독립변수는 7개, 케이스 수가 99개인 자료이다.
변수는 가격, 크기, 알코올, 평판, 색, 향기, 맛이 있으며, 점수는 0~100으로 되어 있다.
이 자료를 이용하여 주성분 분석을 해 보자.

[예제 분석]
(가) 데이터 자료 읽어오기
(나) 주성분 분석 실행하기
(다) 주성분 분석 결과
     몇개의 주성분이 80%이상이 되는가?
       summary를 이용하여 정보를 확인해 보자.

(라) 스트리 그림과 주성분 계수를 확인해 보자.
(마) 주성분 점수 및 행렬도를 확인해 보자.
(바) corrplot를 이용하여 상관계수를 확인해 보자.

주성분 점수 및 행렬도(biplot)

```
#install.packages("readxl")
library(readxl)
```

```
## Warning: package 'readxl' was built under R version 3.3.3
```

```
my_data <- read_excel("beer_pca.xls")
my_data
```

```
## # A tibble: 99 x 7
##     cost  size alcohol reputat color aroma taste
##    <dbl> <dbl>   <dbl>   <dbl> <dbl> <dbl> <dbl>
##  1    10    15      20      85    40    30    50
##  2   100    70      50      30    75    60    80
##  3    65    30      35      80    80    60    90
##  4     0     0      20      30    80    90   100
##  5    10    25      10     100    50    40    60
##  6    25    35      30      40    45    30    65
##  7     5    10      15      65    50    65    85
##  8    20     5      10      40    60    50    95
##  9    15    10      25      30    95    80   100
## 10    10    15      20      85    40    30    50
## # ... with 89 more rows
```