예제 데이터

1) LPGA 2008년 경기능력

LPGA2008.csv : 미국 LPGA 골프선수들의 능력 측정 변수 (비거리, 페어웨이 안착율, 샌드(벙커) 세이버, 그린 적중률, 버팅 수), 상금, 출전 라운드 수를 조사한 데이터이다.

```
LPGA<-read.csv('http://wolfpack.hnu.ac.kr/Stat_Notes/example_data/
lpga2008.csv', fileEncoding = "utf-8")
head(LPGA)
```

```
> head(LPGA)
                골퍼 평균_비거리 페어웨이_안착율 그린_적중률 평균_퍼팅수 샌드_회수 샌드_세이브   상금 참가_라운드수
1     Ahn, Shi Hyun       249.4           64.6        61.2       27.44       1.10       34.5  6063           50
2 Alfredsson, Helen       253.8           62.7        68.2       29.36       0.66       38.8 19343           74
3 Ammaccapane, Dina       246.3           70.2        64.6       30.20       0.74       40.5  1873           50
4       Bader, Beth       249.1           64.1        61.2       29.78       1.12       41.1  1212           65
5       Bae, Kyeong       244.0           62.4        60.7       28.38       1.02       43.9  2555           65
6     Baena, Marisa       254.2           64.7        60.9       29.21       1.27       33.3  2282           52
```

6개의 능력변수를 활용하여 선수들의 능력을 나타내는 지표를 만들어 선수들의 능력을 파악할 수 없나?

- 선수의 능력을 파악하기 위한 적절한 그래프는 히스토그램(개별 1개 변수)나 산점도(2개, 버블 산점도를 이용한다면 3개)를 활용하는 것이 최선이다.
- 6개 변수를 전체를 이용한다면 $6C2 = 15$ 산점도가 필요하고 동시에 활용한 능력 지표(변수)?
- 하여, 6개 변수를 축약할 수 있는 방법은 무엇일까? - 주성분 분석은 원 **데이터 변수의 차원을 축약**하는 방법이다.

2) police 데이터

Police.csv : 경찰 지원자 50명, 2개 분야(체력, 신체) 특성 변수 (지원아이디, 15개 측정변수)

```
police<-read.csv('http://wolfpack.hnu.ac.kr/Stat_Notes/example_data/poilice_body_exercise.csv', fileEncoding = "utf-8")
names(police)
```

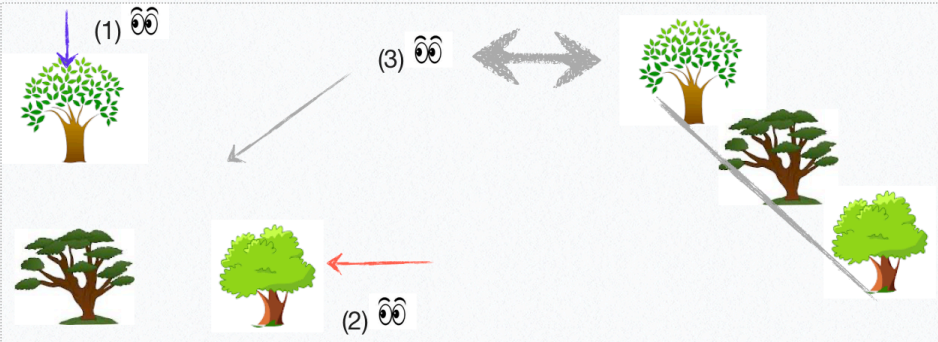
```
> names(police)
 [1] "아이디"                "시각반응"              "키"                    "몸무게"
 [5] "어깨넓이"              "골반넓이"              "가슴둘레"              "허벅지두께"
 [9] "맥박"                  "심장혈압"              "턱걸이"                "폐활량.리터."
[13] "러닝머신_5분_맥박"     "러닝머신_최대속력"     "러닝머신_달린시간.분." "비만지수"
```

지원자 50명 중 (체력과 신체) 특성 우수자 5명 선발

- 체력분야 측정변수들의 평균, 신체 특성 측정변수들의 평균을 구하여 체력분야 우수자, 신체분야 우수자를 선발한다?
- 두 분야를 합치는 방법? 평균의 평균을 구한다.
- 산술평균의 문제? (가슴둘레가 크면 허벅지도 두껍고 어깨넓이도 클 것이다) -상관관계가 매우 유의함 그러므로 3번 중복된 상태로 반영-합리적이지 못하다
- 상관관계가 높은 측정변수가 많이 존재할수록 산술평균으로 능력을 평가하는데는 중복성의 한계가 있음 - 가슴둘레가 크면 다른 신체도 커지므로 이를 고려한 새로운 지표가 필요
- 가중평균이 해결 방법이다. 가중치? 주관적으로 결정해야 하나? (상관계수가 일정 값 이상이 변수들끼리 그룹화 하여 그룹 내 변수의 개수를 가중치로 활용하면 된다)
- (시각반응~심장혈압) 9개 변수가 상관계수에 의해 (시각반응, 맥박, 심장혈압), 그리고 나머지 6개 변수로 2개 상호 배타적 그룹화 된다면- 첫 그룹 변수 3개에는 1/3, 두 번째 그룹 변수들에는 1/6의 가중치를 부여하여 가중평균을 계산하면 된다.
- 가중평균의 가중치 부여는 객관적이지 못하다. 그럼 각 변수의 상관관계(변수의 유사성 고려)를의 구조를 고려한 가중치를 계산하는 방법? 이것이 주성분 분석방법이다.
- 주성분 분석의 **가중치를 이용하여** 새롭게 만들어지는 변수를 주성분변수라 함

개념

1) 공간적 개념



- 2차원 공간 정보를 1차원(직선)으로 표시한다면 어디에서 봐야 희생되는 정보(나무의 위치와 거리)가 최소일까?
- (1)에서 본다면 가 직선 상에 나타나지 않고 (2)의 관점에서 본다면 역시 가 에 가려져 직선 상에 나타나지 않음 그러므로 위치와 공간 정보를 최소화하는 관점은 (3)이다.
- 희생되는 정보가 있다. 와 의 거리는 실제 거리보다 가까워졌음, - 이는 2차원 공간 정보를 1차원 직선으로 표현하여 잃은 정보임
- 주성분분석은 이처럼 변수의 차원(개수)를 희생 정보를 최소화 하여 축약하는 방법론임

2) 변수의 개수 - 차원

- 기성복 하의 구매 - 허리둘레와 기장(우리나라는 손쉽게 줄일 수 있으므로 허리둘레만 활용)
- 예전에는 허리둘레, 허벅지 두께, 허리에서 무릎 길기, 무릎에서 발꿈치까지 길이 등 많은 측정값이 필요하였다. - 아니면 대충 허리 둘레에 맞춰 옷을 사고 줄여 입었음
- 지금은 길이에 대한 정보는 기장, 둘레에 대한 정보는 허리 둘레에 들어가 있어 이 두 값만 알면 어디서나 쉽게 바지를 구입할 수 있음
- 그럼 이 기장, 허리둘레는 이전의 길이, 허리둘레만 있나? 아니다. 예전의 관측값들의 정보를 모두 포함하고 있는 골든 지표이다.
- (기장, 허리둘레)는 예전의 모든 측정변수의 관측값으로 만들어진 주성분 변수임.
- 많은 측정값들이 2개의 주성분변수로 축약되므로 모든 사람들의 몸에 맞는 기성복은 존재하지 않는다. 기성복이 맞는 않는 사람은 big&tall 샵, 디자이너 샵에 가야 한다.

주성분 행렬 이론

1) 원 변수 데이터 행렬

데이터 행렬 : $X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \cdots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix}_{n\times p}$

2) 변수벡터

$\underline{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix}_{p\times 1}$ (pX1) 벡터, where $\underline{x} \sim N(\underline{\mu}, \Sigma)$ 다변량 정규분포를 따른다.

평균 벡터, 공분산행렬

평균 $E(X_i) = \mu_i$, 공분산 $\sigma_{ij} = COV(X_i, X_j)$

$\underline{\mu} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_p \end{bmatrix}$ (mean matrix), $\Sigma = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1p} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2p} \\ \vdots & \vdots & \cdots & \vdots \\ \sigma_{p1} & \sigma_{p2} & \cdots & \sigma_{pp} \end{pmatrix}_{p\times p}$ (covariance matrix)

k번째 변수 데이터 벡터

$\underline{x}_k = \begin{bmatrix} x_{k1} \\ x_{k2} \\ \vdots \\ x_{kn} \end{bmatrix}_{n\times 1}$ (nX1) 벡터

3) 주성분 벡터

주성분 변수 벡터 $\underline{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_p \end{pmatrix}_{p\times 1} = \begin{pmatrix} l_{11} & l_{12} & \cdots & l_{1p} \\ l_{21} & l_{22} & \cdots & l_{2p} \\ \vdots & \vdots & \cdots & \vdots \\ l_{p1} & l_{p2} & \cdots & l_{pp} \end{pmatrix}_{p\times p} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{pmatrix}_{p\times 1} = L\underline{x}$

j번째 주성분변수 선형계수 벡터

선형 계수 = 부하값 loading $\underline{l}_j = \begin{pmatrix} l_{j1} \\ l_{j2} \\ \vdots \\ l_{jp} \end{pmatrix}_p$: (pX1) 벡터

j번째 주성분변수 $y_j = l_j' * \underline{x} = (l_{j1}\ l_{j2} \cdots\ l_{jp})_{1\times p} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{pmatrix}_{p\times 1}$: 결과 스칼라

j번째 주성분 변수 데이터 $\underline{y} = X_{n\times p} L_{p\times p}$: (nXp) 벡터

4) 주성분 변수 성질

(1)원 변수들의 선형결합이다

$$y_j = l_j' * \underline{x} = (l_{j1}\ l_{j2} \cdots\ l_{jp})_{1\times p} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{pmatrix}_{p\times 1}$$

(2) 서로 독립이다.

$\underline{l}_i' * \underline{l}_j = 0$: 선형계수의 곱은 0이다.

$$E(Y_i Y_j) = E(Y_i)E(Y_j)$$

(3) 주성분 변수는 원 변수들의 변동을 설명하고 순서대로 설명력은 줄어든다.

주성분 구하기

개체에 대한 변수 정보는 변동(분산)에 의해 정의된다. 변수의 변동은 공분산행렬(혹은 상관행렬)에 의해 측정된다.

- 공분산행렬은 변수의 측정단위를 그대로를 반영한 것이고 상관행렬은 모든 변수의 측정단위를 표준화 한 것이다.
- 어느 행렬을 사용하는 것이 적절한가? 변수의 변동을 정확하게 반영하는 것은 공분산행렬이므로 공분산행렬을 사용하는 것이 적절함
- 측정 단위의 차이가 많은 데이터에서는 단위 큰 변수의 영향도가 크므로 상관행렬을 사용하는 것이 적절하다.
- 측정단위의 차이가 큰 변수가 있다면 다른 변수들과 단위를 맞추어 공분산행렬을 사용하기를 권장한다.

1) 데이터 공분산 행렬 고유값, 고유벡터 구하기

공분산행렬 : $\Sigma = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1p} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2p} \\ \vdots & \vdots & \cdots & \vdots \\ \sigma_{p1} & \sigma_{p2} & \cdots & \sigma_{pp} \end{pmatrix}_{p\times p}$, 상관행렬 : $R = \begin{pmatrix} 1 & \rho_{12} & \cdots & \rho_{1p} \\ \sigma_{21} & 1 & \cdots & \rho_{2p} \\ \vdots & \vdots & \cdots & \vdots \\ \rho_{p1} & \rho_{p2} & \cdots & 1 \end{pmatrix}_{p\times p}$

공분산행렬은 양반정치(positive definite)행렬이므로 다음을 만족하는 0보다 큰 실수이고 행렬의 차수 p개만큼 고유값(eigenvalue)이 존재한다.

(정의) 행렬 A에 대하여 $A\underline{\mu} = \lambda\underline{\mu}$ 을 만족하는 λ_i을 고유값(eigenvalues), λ_i를 대입하여 구한 벡터 $\underline{\mu}_i$를 고유벡터(eigenvectors)라 한다.

(고유값 구하기) $|A - \lambda I| = 0$ 만족하는 λ들을 고유값이라 한다.

2) 고유값 및 고유벡터 성질

(공분산행렬) $|\Sigma - \lambda I| = 0$ 을 만족하는 고유값은

(1) $\lambda_1 \geq \lambda_2 \geq \ldots \lambda_p \geq 0$ 양의 실수이고, 원변수의 개수(차수)만큼 존재한다.

(2) 고유값 λ_k에 대응하는 고유벡터 $\underline{\mu}_k$는 무수히 많이 존재함. - 주성분 요인 회전에 활용

(3) 고유벡터 중 $\underline{\mu}'_k\underline{\mu}_k = 1$인 norm 벡터 $\underline{e}_k$를 k-번째 주성분 변수의 선형계수로 사용한다.

(4) 원 변수의 변동(분산)합은 고유값의 합과 같다.

$$\sigma_{11} + \sigma_{22} + \ldots + \sigma_{pp} = \lambda_1 + \lambda_2 + \ldots + \lambda_p$$

(5) 고유값 λ_k는 k-번째 주성분변수가 원변수의 변동을 설명하는 능력이다.

(6) k-번째 주성분 변동 기여율= $\dfrac{\lambda_k}{\lambda_1 + \lambda_2 + \ldots + \lambda_p}$ (상관행렬) = $\dfrac{\lambda_k}{p}$

(상관행렬) 원 변수의 척도가 상이하여 단위를 맞출 필요가 있을 때 사용

(1) i-번째 주성분 변수와 k-번째 원 변수 상관계수는 $\rho(Y_i, X_k) = \dfrac{e_{ik}\sqrt{\lambda_i}}{\sqrt{\sigma_{kk}}}$

(2) 서로 다른 주성분 변수는 서로 독립이다. $COV(Y_i, Y_j) = 0$ for $i \neq j$

(3) k-번째 주성분 변수의 분산은 λ_k이다. $p = \lambda_1 + \lambda_2 + \ldots + \lambda_p$

3) 주성분 구하기 절차

고유값이 제일 큰 λ_1에 의해 만들어지는 주성분변수를 제1주성분, λ_2에 의해 만들어지는 주성분변수를 제2주성분이라 함, 제3주성분, ... 그러므로 원변수가 p개이면 주성분변수도 p개만큼 구해진다.

제1주성분 변수

$\underline{l}_1'\underline{l}_1 = 1$이면서 $V(\underline{l}_1'\underline{x})$을 최대화 하는 열벡터 $\underline{l}_1$를 구하고 이를 제1주성분변수 선형계수로 하여 주성분변수를 구한다. $\underline{y}_1 = \underline{l}_1'X$

제2주성분 변수

$\underline{l}_1'\underline{l}_2 = 0$, $\underline{l}_2'\underline{l}_2 = 1$이면서 $V(\underline{l}_2'\underline{x})$을 최대화 하는 열벡터 $\underline{l}_2$를 구하고 이를 제2주성분변수 선형계수로 하여 주성분변수를 구한다. $\underline{y}_2 = \underline{l}_2'X$

제3주성분 변수

$\underline{l}_1'\underline{l}_3 = 0$, $\underline{l}_2'\underline{l}_3 = 0$, $\underline{l}_3'\underline{l}_3 = 1$이면서 $V(\underline{l}_3'\underline{x})$을 최대화 하는 열벡터 $\underline{l}_3$를 구하고 이를 제3주성분변수 선형계수로 하여 주성분변수를 구한다. $\underline{y}_3 = \underline{l}_3'X$

이렇게 계속 원변수의 개수만큼 주성분변수를 구한다.

통계소프트웨어는 $(\underline{x} - \underline{\mu})$을 이용하여 주성분변수를 구하여 주성분변수의 평균이 0이다.

4) |선형계수| |부하 loading|

공분산행렬 Σ 고유벡터 중 $\underline{e}'_k\underline{e}_k = 1$을 만족하는 Norm 고유벡터가 k-번째 주성분 변수의 선형계수

|주성분변수|

$\underline{y}_{k(n\times 1)} = X_{n\times p}\underline{e}_{k(p\times 1)}$: k-번째 주성분 변수

|k-번째 주성분 변수의 변동 기여율|

$$\frac{\lambda_k}{\lambda_1+\lambda_2+\ldots+\lambda_p}(\text{under } \Sigma), \frac{\lambda_k}{\lambda_1+\lambda_2+\ldots+\lambda_p}(\text{under } R)$$

주성분변수 개수 결정

앞에서 설명하였듯이 주성분변수는 원변수의 공분산행렬(혹은 상관행렬)을 이용하여 고유값을 구하고, 그에 대응하는 고유벡터를 선형계수로 하여 주성분변수를 구한다.

주성분변수는 서로 독립이며, 제1주성분이 원변수의 변동을 가장 많이 설명하고, 제2주성분, 제3주성분, .. 순이다. 그리고 주성분의 개수는 원변수의 개수만큼 존재한다.

(주성분변수의 주목적) 변수의 차원 축소, 즉 변수의 개수를 줄이는 것이 주된 목적임-어떻게 줄일까?

(Rule of Thumb) 80%

제1주성분변수의 원변수 변동을 설명하는 능력이 가장 크고, 제2주성분, 제3주성분, ... 순이다.

공분산행렬

k-번째 주성분변수의 변동설명 기여율 = $\dfrac{\lambda_k}{\sum_{i=1}^{p}\lambda_i}$

상관행렬

k-번째 주성분변수의 변동설명 기여율 = $\frac{\lambda_k}{p}$

상관행렬의 경우 대각원소가 1이므로 고유치의 합은 원변수의 개수와 동일하다.

원변수 변동 누적 기여율이 80%까지 주성분변수를 선택한다. 즉 20% 정보는 희생된다.

원변수의 상관관계가 높을수록 변수의 차원은 축약이 쉽게되므로 80% 규칙에 의한 주성분변수 개수는 작음(일반적으로 2~3개)

고유값 1이상

상관행렬을 이용하는 경우 원변수 변동의 합은 p이므로 평균인 1이상 고유값을 갖는 주성분변수만 선택. 대부분의 통계소프트웨어는 이 규칙을 이용하여 주성분개수를 구한다.

주성분변수 이름부여

$$\begin{pmatrix} y_{11} & y_{12} & \cdots y_{1p} \\ y_{21} & y_{22} & \cdots y_{2p} \\ \cdots & & \\ y_{n1} & y_{n2} & \cdots y_{np} \end{pmatrix} (\text{주성분}) = \begin{pmatrix} x_{11} & x_{12} & \cdots x_{1p} \\ x_{21} & x_{22} & \cdots x_{2p} \\ \cdots & & \\ x_{n1} & x_{n2} & \cdots x_{np} \end{pmatrix} (\text{원변수}) \begin{pmatrix} l_{11} & l_{12} & \cdots l_{1p} \\ l_{21} & l_{22} & \cdots l_{2p} \\ \cdots & & \\ l_{p1} & l_{p2} & \cdots l_{pp} \end{pmatrix}$$

주성분변수는 원변수들의 선형결합(선형식=고유벡터)에 의해 구해지므로 적절한 이름을 붙여주어야 활용도가 높아진다.

주성분분석의 핵심적 단계이며, 분석자의 능력(다른 단계는 소프트웨어가 계산)이 필요하다.

(허리둘레, 기장)과 같이 명확하게 이해될 수 있는 이름 부여가 중요하다. 무엇을 이용하여 이름을 붙일까? 선형계수=부하 값의 크기를 이용한다.

부하(l_{ij}) 값이 크다는 것은 주성분변수가 계산될 때 그 변수의 영향이 크다는 것을 의미한다. 즉, 부하 값이 상대적으로 큰 (절대값 기준보다는) 변수들에 의해 주성분변수의 이름이 부여된다.

상관관계가 높은 변수들은 동일 주성분 내에서 부하값이 크게나타난다. (예) 둘레에 관련된 허벅지두께, 허리둘레, 종아리두께 등은 같은 주성분에서 부하값이 다른 변수에 비해 상대적으로 클 것이다.

부하 값이 음인 경우는 다른 변수들과 음의 상관관계를 의미한다. 그러나 상관관계가 높으므로 변수의 유사성은 존재한다는 것을 의미한다.

다음은 Fisher's Iris 데이터 첫 6개 관측치

	Sepal_Length	Sepal_Width	Petal_Length	Petal_Width	group
1	50	33	14	2	Setosa
2	64	28	56	22	Virginica
3	65	28	46	15	Versicolor
4	67	31	56	24	Virginica
5	63	28	51	15	Virginica
6	46	34	14	3	Setosa

고유벡터

제1 주성분 부하 : (0.36, -0.085, 0.857, 0.358) - Petal 길이의 부하 값이 가장 크므로 제1주성분은 꽃잎 길이를 활용한 이름 부여가 가능함

```
> iris2.eigen.vectors
                [,1]           [,2]           [,3]          [,4]
[1,]  0.36138659179  0.65658877129  0.58202985131  0.3154871929
[2,] -0.08452251406  0.73016143479 -0.59791083010 -0.3197231037
[3,]  0.85667060595 -0.17337266280 -0.07623607582 -0.4798389870
[4,]  0.35828919715 -0.07548101992 -0.54583143202  0.7536574253
```

주성분 : $27.99 = 0.36 * 50 - 0.085 * 33 + 0.857 * 14 + 0.358 * 2$

```
> head(iris2.PCV)
            [,1]        [,2]        [,3]           [,4]
[1,] 27.99005350 54.34658659 7.211467246  0.01306625682
[2,] 76.61802775 51.09674998 4.230895490  0.94841352731
[3,] 65.90468390 54.01543252 9.396106124  0.78668861331
[4,] 78.16519838 55.10603856 3.091589690  2.44302064555
[5,] 69.46526375 51.83539166 7.850866042 -2.24348070747
[6,] 26.81827382 52.37491192 3.739605579 -0.81494819320
```

주성분분석 활용

- 변수의 차원을 축약하여 개체에 대한 정보를 얻기 쉽게 한다. 변수의 차원이 저차원(일반적으로 3개 정도)으로 축약되므로 산점도를 그려 개체의 특성을 쉽게 파악할 수 있다.
- 주성분변수가 상관계수가 0(독립)이라는 사실을 이용하여 회귀분석의 다중공선성 문제 해결을 위한 방법으로 사용한다.
- 주성분분석은 군집분석, 판별분석을 1차분석으로 활용된다. 개체의 유사성에 의해 분류된 군집의 특성을 분석하여 이름을 부여하는데 활용하거나 판별분석 결과 오분류 개체들의 특성 파악에 도움을 준다. - 데이터 변수가 다수(5개 이상)인 군집분석과 판별분석에는 항상 주성분분석이 필요하다.

실증 데이터 확인하기

Fisher's Iris(붓꽃 데이터)

다변량 데이터셋, 영국 통계학자, 생물학자 Ronald Fisher(1936)

n=150 samples 붓꽃 3종 (Iris setosa, Iris virginica and Iris versicolor)

4개 척도 변수 Sepal (꽃받침) 넓이, 길이 / Petal 꽃잎 넓이, 길이

```
iris<-read.csv('http://wolfpack.hnu.ac.kr/Stat_Notes/example_data/iris.csv')
names(iris)
```

```
> names(iris)
[1] "Sepal_Length" "Sepal_Width"  "Petal_Length" "Petal_Width"  "group"
```

```
> table(iris$group)

    Setosa Versicolor  Virginica
        50         50         50
```

공분산 행렬 구하기

공분산 행렬을 구하는 함수 cov() 데이터는 행렬 형식이어야 함

```
iris2<-as.matrix(iris[,1:4])
cov(iris2) #covariance matrix
```

```
> cov(iris2) #covariance matrix
             Sepal_Length Sepal_Width Petal_Length Petal_Width
Sepal_Length     68.56935    -4.24340    127.43154    51.62707
Sepal_Width      -4.24340    18.99794    -32.96564   -12.16394
Petal_Length    127.43154   -32.96564    311.62779   129.56094
Petal_Width      51.62707   -12.16394    129.56094    58.10063
```

원변수 변동(분산) 합은 공분산 행렬의 대각원소의 합이다.

```
install.packages('psych')
library(psych)

tr(cov(iris2)) #sum of diagonal elements (원변수 분산합 = 변동합)
```

```
> tr(cov(iris2)) #sum of diagonal elements (원변수 변동합)
[1] 457.2957
```

고유치 구하기 eigen value

```
iris2.eigen.values=eigen(cov(iris2))$values  #eigenvalues
iris2.eigen.values

sum(iris2.eigen.values)
```

고유치 크기 순으로 422.82, 24.26, 7.82, 2.38, 원 변수가 4개 이므로 고유치도 4개임. 고유치는 원 변수 변동 설명력이므로 고유치합 457.29 = 위의 원변수 변동 합 457.29와 동일함

```
> iris2.eigen.values=eigen(cov(iris2))$values  #eigenvalues
> iris2.eigen.values
[1] 422.824171  24.267075   7.820950   2.383509
> sum(iris2.eigen.values)
[1] 457.2957
```

각 고유치를 고유치 합으로 나누면 각 주성분의 원변수 설명력임. - 제1주성분의 원변수 변동 설명 비율은 92.5%, 제2주성분은 5.3%임. 80% 설명력 기준으로는 1개 주성분변수만으로 원변수 4개의 변동을 92.5% 설명 - 4개 차원이 1개 차원으로 축소

```
> iris2.eigen.values/sum(iris2.eigen.values)
[1] 0.924618723 0.053066483 0.017102610 0.005212184
```

고유벡터 eigen vector = 부하

```
iris2.eigen.vectors=eigen(cov(iris2))$vectors  #eigenvalues
iris2.eigen.vectors
```

- 제1주성분 부하 [, 1] = 꽃잎(petal) 길이 부하(0.85) 값만 상대적으로 크므로 '꽃잎길이' 성분
- 제2주성분 부하 [, 2] = 꽃받침(sepal) 길이, 넓이 부하 값(0.66, 0.73) 상대적으로 크므로 '꽃받침 크기' 성분

```
> iris2.eigen.vectors
           [,1]        [,2]        [,3]       [,4]
[1,]  0.36138659  0.65658877  0.58202985  0.3154872
[2,] -0.08452251  0.73016143 -0.59791083 -0.3197231
[3,]  0.85667061 -0.17337266 -0.07623608 -0.4798390
[4,]  0.35828920 -0.07548102 -0.54583143  0.7536574
```

고유벡터는 서로 독립이며 norm(각 고유벡터 제곱=1)임 - 고유벡터의 제곱행렬은 대각원소(norm)=1, 비대각원소(서로 독립)는 0이다.

```
> t(iris2.eigen.vectors)%*%iris2.eigen.vectors  #product of e
ty(3)&(7)
              [,1]          [,2]          [,3]          [,4]
[1,]  1.000000e+00  3.087808e-16 -1.942890e-16  0.000000e+00
[2,]  3.087808e-16  1.000000e+00  1.804112e-16 -3.469447e-17
[3,] -1.942890e-16  1.804112e-16  1.000000e+00  5.551115e-17
[4,]  0.000000e+00 -3.469447e-17  5.551115e-17  1.000000e+00
```

주성분변수 만들기 $Y_{n\times p} = X_{n\times p}L_{p\times p}$

```
iris2.PCV=iris2%*%iris2.eigen.vectors  #주성분변수 만들기 ; DS2.PCV
head(iris2.PCV)
```

```
> head(iris2.PCV)
         [,1]     [,2]     [,3]        [,4]
[1,] 27.99005 54.34659 7.211467  0.01306626
[2,] 76.61803 51.09675 4.230895  0.94841353
[3,] 65.90468 54.01543 9.396106  0.78668861
[4,] 78.16520 55.10604 3.091590  2.44302065
[5,] 69.46526 51.83539 7.850866 -2.24348071
[6,] 26.81827 52.37491 3.739606 -0.81494819
```

주성분변수의 (공분산)분산은 고유치와 동일, 제1주성분 분산 422.8은 제일 큰 고유치 값과 동일함

주성분변수들 간은 독립이므로 (상관계수)행렬의 비대각원소는 0이다.

```
> cov(iris2.PCV) #주성분 분산 = 고유값
                 [,1]             [,2]             [,3]             [,4]
[1,]  4.228241706e+02 -1.346715431e-13 -5.043435782e-14 -2.229505613e-14
[2,] -1.346715431e-13  2.426707479e+01  1.258832252e-14 -1.580397557e-14
[3,] -5.043435782e-14  1.258832252e-14  7.820950004e+00 -6.752649618e-15
[4,] -2.229505613e-14 -1.580397557e-14 -6.752649618e-15  2.383509297e+00
> cor(iris2.PCV) #주성분변수는 서로 독립
                 [,1]             [,2]             [,3]             [,4]
[1,]  1.000000000e+00 -1.329496564e-15 -8.770348233e-16 -7.022960765e-16
[2,] -1.329496564e-15  1.000000000e+00  9.137542121e-16 -2.078017545e-15
[3,] -8.770348233e-16  9.137542121e-16  1.000000000e+00 -1.563997432e-15
[4,] -7.022960765e-16 -2.078017545e-15 -1.563997432e-15  1.000000000e+00
```

실증분석 LPGA 데이터(상관계수 행렬 이용) - 예측모형의 predictors

능력 변수 6개를 축약한 지표(주성분변수)를 만들어 선수들의 특성을 파악하자. 6개 변수는 선수들의 다음 능력을 측정한다.

- 비거리 - 장타력(힘) 측정변수 (+)
- 페어웨이 안착율 - 드라이버 정확도 (+)
- 그린 적중률 - 아이언 실정확도 (+)
- 샌드 회수 - 부정확도 측정 (-)
- 샌드 세이브 - 위기 탈출 능력 (+)
- 퍼팅 개수 - 퍼팅 능력 (-)

```
> head(lpga)
              골퍼 평균_비거리 페어웨이_안착율 그린_적중률
1     Ahn, Shi Hyun        249.4            64.6        61.2
2 Alfredsson, Helen        253.8            62.7        68.2
3 Ammaccapane, Dina        246.3            70.2        64.6
4       Bader, Beth        249.1            64.1        61.2
5       Bae, Kyeong        244.0            62.4        60.7
6     Baena, Marisa        254.2            64.7        60.9
  평균_퍼팅수 샌드_회수 샌드_세이브  상금 참가_라운드수
1       27.44       1.10        34.5  6063            50
2       29.36       0.66        38.8 19343            74
3       30.20       0.74        40.5  1873            50
4       29.78       1.12        41.1  1212            65
5       28.38       1.02        43.9  2555            65
6       29.21       1.27        33.3  2282            52
```

데이터 읽기, 기초통계량

```
lpga<-read.csv("http://wolfpack.hnu.ac.kr/Stat_Notes/example_data/lpga2008.csv",fileEncoding = "utf-8")
names(lpga)
install.packages('pastecs')
library(pastecs)
stat.desc(lpga[,2:7])
```

```
> names(lpga)
[1] "골퍼"          "평균_비거리"    "페어웨이_안착율" "그린_적중률"
[5] "평균_퍼팅수"    "샌드_회수"      "샌드_세이브"     "상금"
[9] "참가_라운드수"
```

```
> stat.desc(lpga[,2:7])
                평균_비거리 페어웨이_안착율      그린_적중률      평균_퍼팅수
nbr.val      1.570000000e+02 1.570000000e+02  157.0000000000  157.0000000000
nbr.null     0.000000000e+00 0.000000000e+00    0.0000000000    0.0000000000
nbr.na       0.000000000e+00 0.000000000e+00    0.0000000000    0.0000000000
min          2.248000000e+02 4.920000000e+01   41.9000000000   26.9500000000
max          2.693000000e+02 7.980000000e+01   71.6000000000   31.9500000000
range        4.450000000e+01 3.060000000e+01   29.7000000000    5.0000000000
sum          3.873360000e+04 1.060970000e+04 9879.6000000000 4583.5400000000
median       2.463000000e+02 6.840000000e+01   63.0000000000   29.0800000000
mean         2.467108280e+02 6.757770701e+01   62.9273885350   29.1945222930
SE.mean      7.531411599e-01 4.607921578e-01    0.3172906744    0.0826862734
CI.mean.0.95 1.487670325e+00 9.101969929e-01    0.6267403056    0.1633291629
var          8.905379226e+01 3.333571779e+01   15.8057194186    1.0734121101
std.dev      9.436831685e+00 5.773709188e+00    3.9756407557    1.0360560362
```

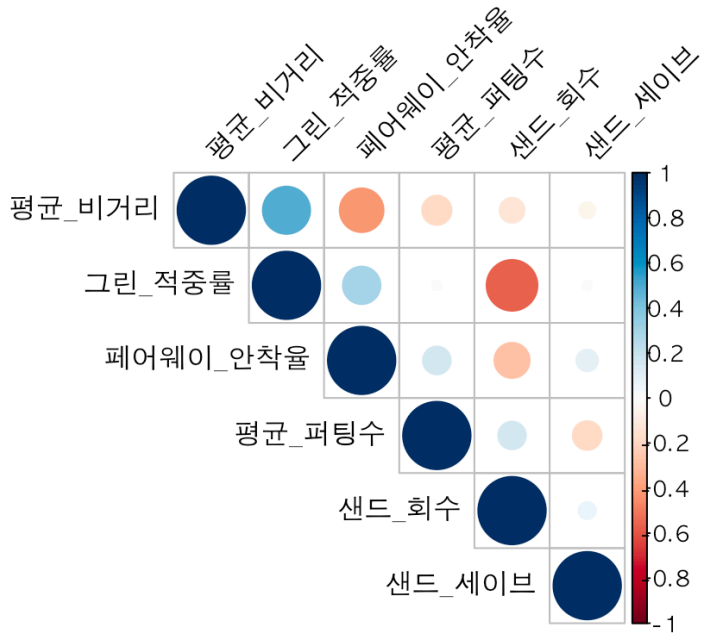
상관분석

```
library(Hmisc) #Corelation coefficeinet/p-value
lpga.cor<-rcorr(as.matrix(lpga[,2:7]), type="pearson")
lpga.cor$P  #상관계수 유의확률
```

```
> lpga.cor$P   #상관계수 유의확률
                  평균_비거리 페어웨이_안착율      그린_적중률    평균_퍼팅수
평균_비거리                    NA 4.032813550e-08 6.325961976e-11 0.0183860
페어웨이_안착율 4.032813550e-08                NA 8.140898366e-05 0.0337494
그린_적중률      6.325961976e-11 8.140898366e-05                NA 0.8189535
평균_퍼팅수      1.838605055e-02 3.374947035e-02 8.189535063e-01
샌드_회수        9.133313569e-02 5.444995673e-04 7.993605777e-15 0.0352384
샌드_세이브      4.998248784e-01 2.084720735e-01 8.358158041e-01 0.0235052
```

```
> lpga.cor$r   #상관계수 유의확률
                  평균_비거리 페어웨이_안착율      그린_적중률
평균_비거리       1.00000000000   -0.4209377595   0.49146587481
페어웨이_안착율 -0.42093775949    1.0000000000   0.30919322584
그린_적중률       0.49146587481    0.3091932258   1.00000000000
```

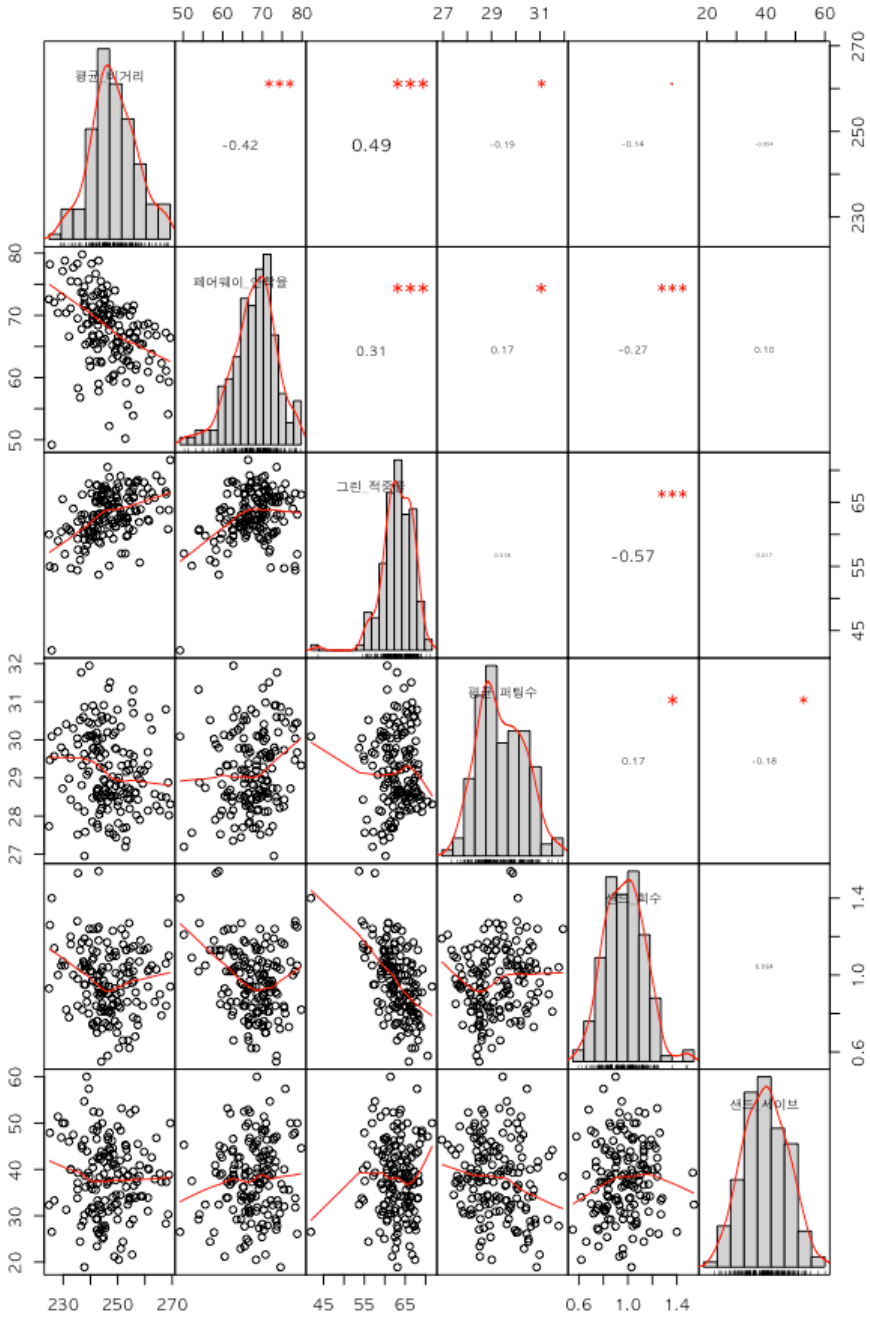
```
library(corrplot) #matrix printout
corrplot(lpga.cor$r, type = "upper", order = "hclust", tl.col = "black",
         tl.srt = 45)
```



비거리 높을수록 그린 적중율 높고 페이웨이 안착율 낮음

그린 적중율 높을수록 샌드 회수 낮아짐

```
library(PerformanceAnalytics) # Matrix of Scatter plot
chart.Correlation(lpga[,2:7], histogram=TRUE, pch=19)
```



주성분 구하기 (상관행렬 이용)

- 변수들의 측정 단위가 한자리에서 천단위까지 있어 단위의 차이로 인하여 상관행렬을 이용
- scale. = TRUE (변수 척도 단위를 표준화 하여 주성분분석) 즉, 상관계수 행렬 이용, scale.=FALSE (공분산행렬 사용)이 디폴트임.
- center=TRUE (원변수 평균이동하여 주성분분석) 디폴트

```
lpga.pca.R=prcomp(lpga[,2:7],scale.=T) #상관계수행렬 이용
names(lpga.pca.R)
```

```
> names(lpga.pca.R)
[1] "sdev"     "rotation" "center"   "scale"    "x"
```

고유값, 고유값 설명비율

```
lpga.pca.R$sdev^2  #고유값 출력
lpga.pca.R$sdev^2/sum(lpga.pca.R$sdev^2)  #고유값 비율
```

```
> lpga.pca.R$sdev^2  #고유값 출력
[1] 1.8780 1.5352 1.1624 0.8362 0.4346 0.1537
> lpga.pca.R$sdev^2/sum(lpga.pca.R$sdev^2)  #고유값 비율
[1] 0.31299 0.25587 0.19373 0.13936 0.07244 0.02562
```

```
> sum(lpga.pca.R$sdev^2)
[1] 6
```

상관계수 행렬로부터 구한 고유치의 합은 변수의 개수와 동일 - 제1주성분 변동 설명력 31.3%, 제2주성분 25.6%, 제3주성분 19.4%(누적 76.3%), 제4주성분 13.9%(누적 90.2%) -> 80% 규칙에 의하면 4개 주성분, 고유값 1이상으로 하면 3개 주성분으로 원변수 6개 차원에서 3~4개 차원으로 축소된다.

```
summary(lpga.pca.R) #변동(누적)설명기여율 출력
```

```
> summary(lpga.pca.R) #변동(누적)설명기여율 출력
Importance of components:
                          PC1    PC2    PC3    PC4    PC5    PC6
Standard deviation     1.370 1.239 1.078 0.914 0.6593 0.3921
Proportion of Variance 0.313 0.256 0.194 0.139 0.0724 0.0256
Cumulative Proportion  0.313 0.569 0.763 0.902 0.9744 1.0000
```

부하 활용 이름 부여

lpga.pca.R$rotation #부하 출력

```
> lpga.pca.R$rotation #부하 출력
                       PC1       PC2       PC3        PC4      PC5       PC6
평균_비거리      -0.39976  0.57931 -0.15475  0.321000  0.2122 -0.57670
페어웨이_안착율 -0.19523 -0.70654  0.12124 -0.009798  0.5203 -0.42097
그린_적중률      -0.66579 -0.07057 -0.10126  0.277574  0.2082  0.64892
평균_퍼팅수       0.14478 -0.35862 -0.59873  0.582617 -0.3738 -0.11317
샌드_회수         0.58124  0.16983 -0.03577  0.387920  0.6534  0.23363
샌드_세이브       0.00242 -0.05264  0.76899  0.574362 -0.2725 -0.04148
```

누적변동 기여율 80% 기준 4개, 고유값 1 이상 기준으로 주성분변수 3개 주성분 변수로 축약될 수 있다. 각 주성분의 이름은 부하값이 상대적으로 큰 원변수의 속성을 이용하면 된다.

제1주성분 : 파온 능력(-)

- PC1=-0.3998*(비거리)-0.195*(페어웨이)-0.666*(그린)+0.145*(퍼팅)+0.58*(샌드회수)+0.0024*(샌드세이브)
- **그린_적중률(-0.67), 샌드회수(반대 +0.58)**
- 그린 주위에 샌드 벙커가 많아 그린 적중률이 낮다는 것은 주변 샌드 벙커에 들어갈 가능성이 높다
- 반대로 그린 적중율이 높으면 정확도가 높아 샌드에 갈 확률이 적어진다.
- '파온 능력' 성분 : 단 - 값이 클수록 파온 능력이 높다.

제2주성분 : 장타 능력(+)

- PC1=0.579*(비거리)-0.707*(페어웨이)-0.071*(그린)-0.359*(퍼팅)+0.0.17*(샌드회수)-0.053*(샌드세이브)
- **평균비거리(0.58), 페어웨이 안착율(-0.71)**
- 드라이브 비거리 클수록 페어웨이 안착 가능성은 낮아짐
- '장타 능력' 성분 : + 값이 큰 선수의 장타 능력 높음

제3주성분 : 위기관리 능력(+)

- **샌드세이브(0.76), 퍼팅수(-0.6)**
- 샌드 세이브는 볼이 벙커에 빠졌음에도 불구하고 파 세이브를 하는 비율 - 비율이 높으면 위기 관리 능력이 높음
- 샌드세이브를 잘 하면 퍼팅 수가 줄어들게 된다. 볼이 홀 근처에 가까이 온되어야 벙커에 빠졌음에도 불구하고 파 세이브가 가능하다.
- '위기 관리 능력' 성분 : 양의 값이 큰 선수가 위기 관리 능력 높음

주성분 구하기

```
lpga.pca.R$x #주성분 변수 출력
```

PC1 = 파온 능력(-), PC2= 장타력(+), PC3=위기관리능력(+)

```
> lpga.pca.R$x
            PC1        PC2       PC3       PC4       PC5       PC6
  [1,]  0.45038  1.3125698  0.60029 -0.972896  0.922628  0.149038
  [2,] -2.00832  0.5796892 -0.30656  0.093821 -1.231113  0.353327
  [3,] -0.96275 -0.9601559 -0.27581  0.340529 -0.979161 -0.318367
  [4,]  0.87412  0.5218807 -0.14022  0.841218 -0.120286 -0.059088
```

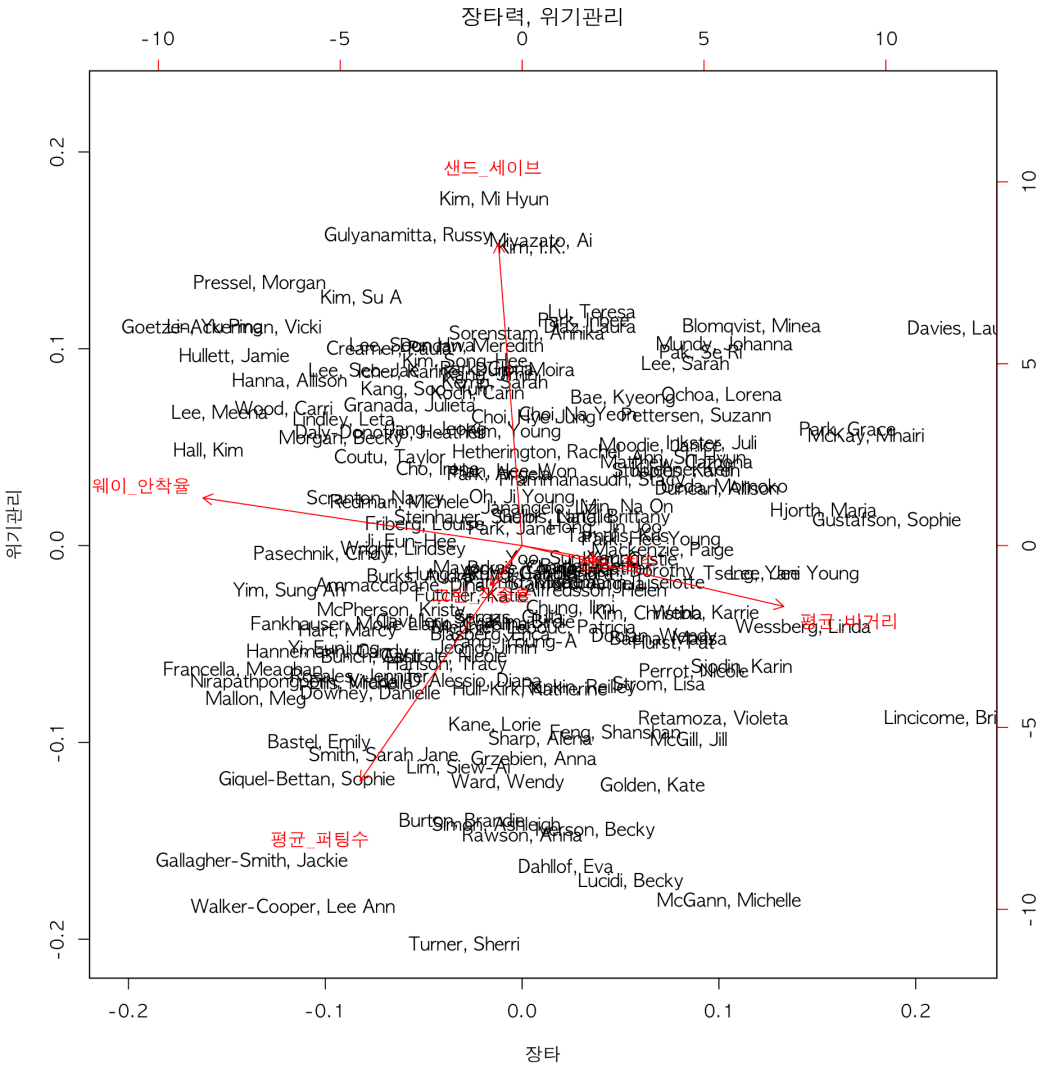
주성분 산점도 그리기

biplot() 함수이용

```
biplot(lpga.pca.R, choices = 1:2,main='파온, 장타력',xlabs=rep("·", nrow(lpga)),xlab='파온',ylab='장타')
biplot(lpga.pca.R, choices = 2:3,main='장타력, 위기관리',xlabs=lpga[,1],xlab='장타',ylab='위기관리')
```

왼쪽 주성분 산점도는 개체(선수) 이름 없이 부하 값의 크기를 잘 보기 위함이고 오른쪽 그림은 개체 이름을 표시하였음

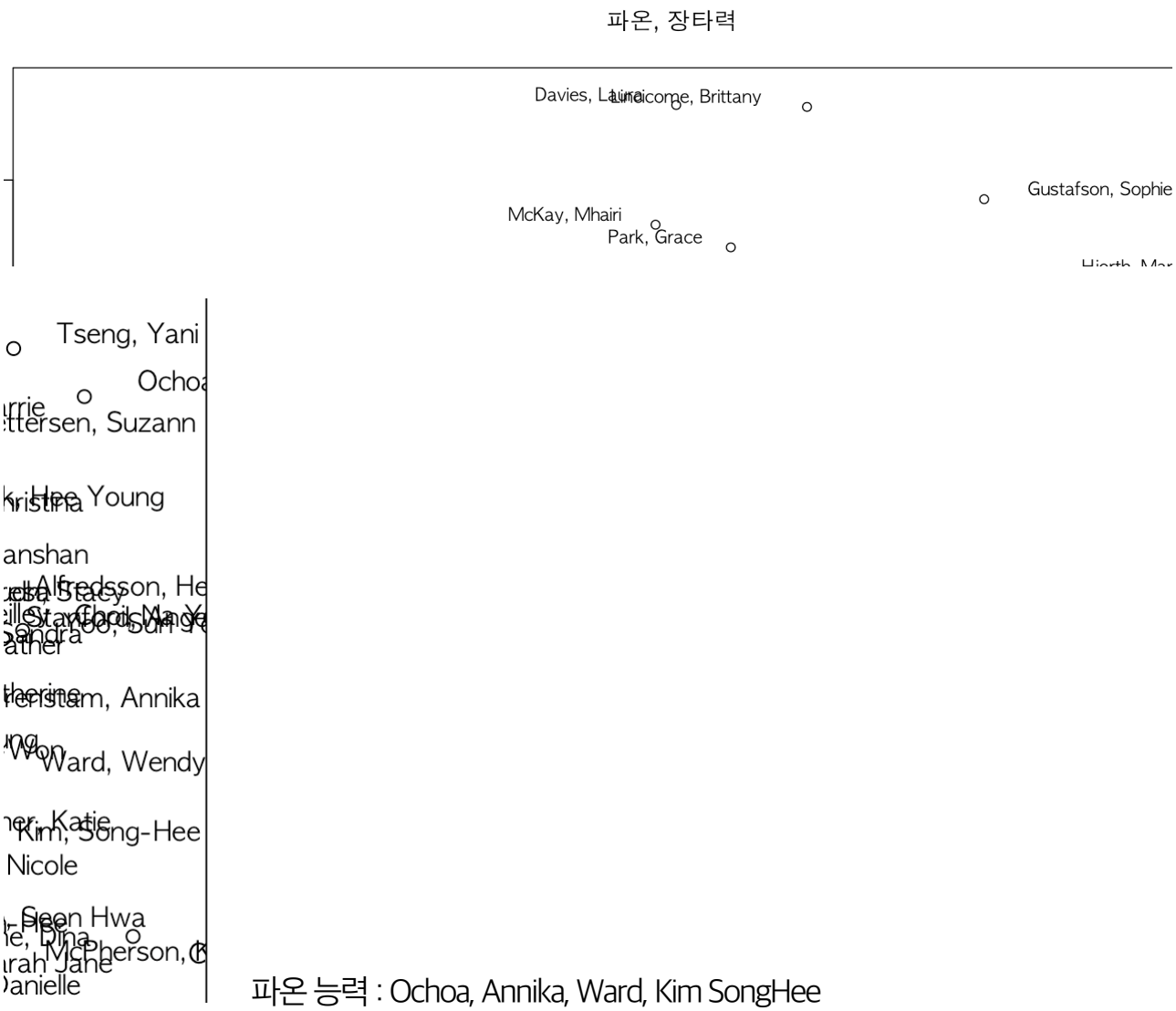
- 장타 선수 : Linchcome, Davis
- 위기관리 뛰어난 선수 : Kim Mihyun



textxy() 함수이용

```
install.packages('calibrate')
library(calibrate)
PC1=lpga.pca.R$x[,1]*(-1) #양의값이 능력 높도록변환
PC2=lpga.pca.R$x[,2]
PC3=lpga.pca.R$x[,3]
plot(PC1,PC2,main='파온, 장타력',xlab='파온',ylab='장타')
textxy(PC1,PC2,lpga$골퍼)   #선수이름 표시
```

장타 능력 : Linchcome, Davis, Gustafson, McKay



파온 능력 : Ochoa, Annika, Ward, Kim SongHee

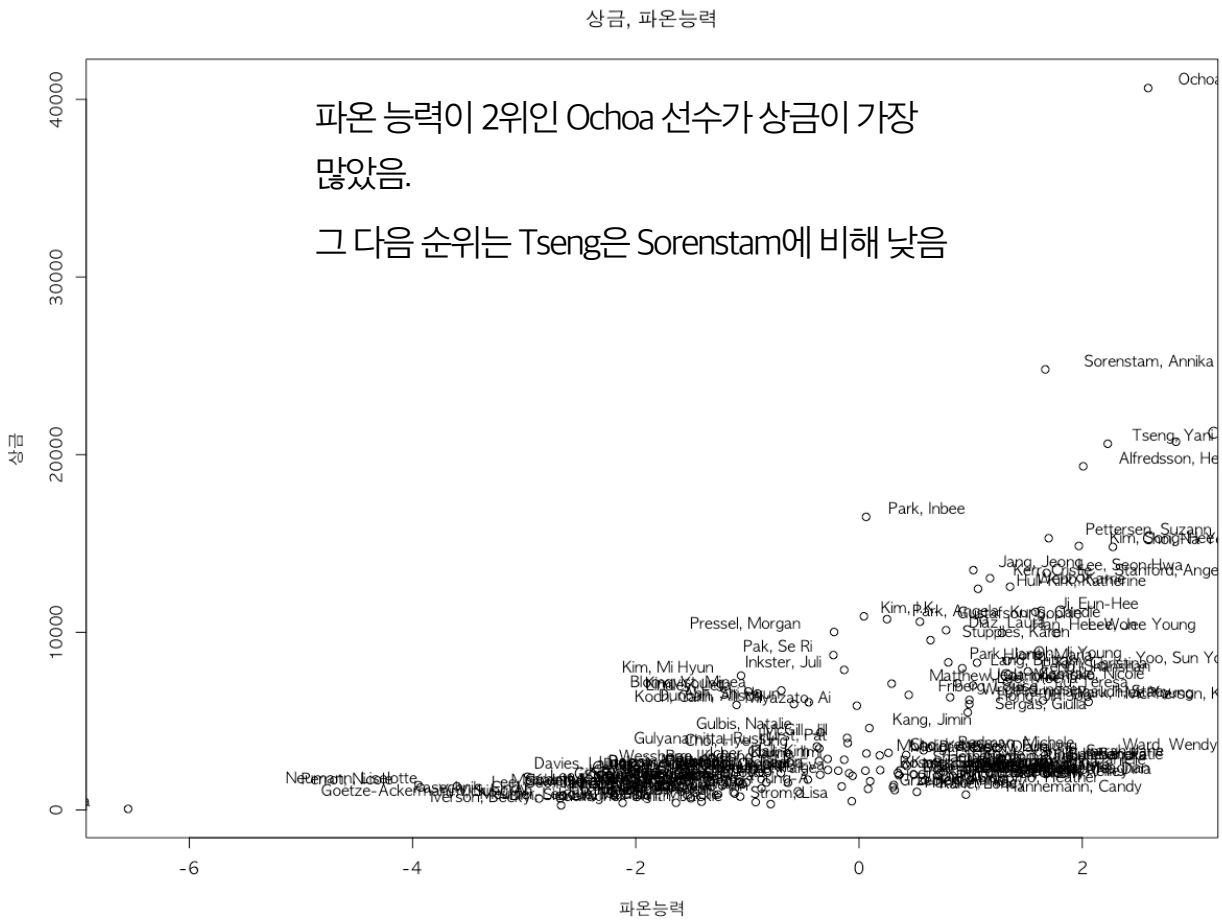
주성분결과 저장 및 활용

```
lpga.pca<-cbind(lpga,PC1,PC2,PC3)
names(lpga.pca)
rcorr(as.matrix(lpga.pca[,c(8,10,11,12)]), type="pearson")
```

```
> rcorr(as.matrix(lpga.pc
      상금   PC1   PC2   PC3
상금 1.00 0.56 0.22 0.33
PC1   0.56 1.00 0.00 0.00
PC2   0.22 0.00 1.00 0.00
PC3   0.33 0.00 0.00 1.00
```

상금과 상관관계가 가장 높은 능력은 PC1 = 파온 능력(- 변환하였음)이며, 그 다음 PC3=위기관리능력, 그리고 장타능력 순이다. 성분 모두 상금과 양의 상관관계임. 성분값이 클수록 상금이 많아진다.

```
plot(lpga.pca$PC1,lpga.pca[,8],main='상금, 파온능력',xlab='파온능력',ylab='상금')
textxy(lpga.pca$PC1,lpga.pca[,8],lpga.pca$골퍼,cex=0.9)  #선수이름 표시
```



```
lpga.pca0<-lpga.pca[,c(1,8:12)]
head(lpga.pca0[order(-PC1),])
head(lpga.pca0[order(-PC2),])
head(lpga.pca0[order(-PC3),])
```

파온 능력 우수 선수 : Creamer > Ochoa > Choi N.Y.

```
> head(lpga.pca0[order(-PC1),])
              골퍼  상금 참가_라운드수  PC1     PC2      PC3
21  Creamer, Paula 20727            88 2.840 -1.0458  1.35052
113  Ochoa, Lorena 40635            68 2.590  1.5759  1.03542
18   Choi, Na Yeon 14808            74 2.274  0.4405  0.90073
145    Tseng, Yani 20613            85 2.228  1.8096 -0.19529
157 Yoo, Sun Young  8106            85 2.174  0.3994 -0.09172
149    Ward, Wendy  3227            71 2.108 -0.1114 -1.62329
```

장타 우수 선수 : Davis > Lincicome > Gustafson

```
> head(lpga.pca0[order(-PC2),])
                  골퍼  상금 참가_라운드수     PC1   PC2     PC3
25       Davies, Laura  2130            53 -1.8234 3.480  1.4883
92  Lincicome, Brittany  2211            52 -0.8173 3.468 -1.1849
48    Gustafson, Sophie 10595            61  0.5453 2.877  0.1743
103       McKay, Mhairi  1228            66 -1.9822 2.714  0.7514
118         Park, Grace  1409            39 -1.4034 2.570  0.7944
56        Hjorth, Maria  8287            71  1.0573 2.380  0.2379
```

위기 관리 우수선수 : Kim, M.H > Gulyanamitta > Miyazato

```
> head(lpga.pca0[order(-PC3),])
                  골퍼  상금 참가_라운드수     PC1      PC2   PC3
77         Kim, Mi Hyun  7562            58 -1.05693 -0.21885 2.376
47  Gulyanamitta, Russy  3574            39 -0.37721 -0.89742 2.130
107        Miyazato, Ai  5869            70 -0.01965  0.15028 2.090
76            Kim, I.K. 10901            71  0.04384  0.08426 2.045
127     Pressel, Morgan 10018            71 -0.22358 -2.06812 1.800
79            Kim, Su A   995            56 -1.12525 -1.26613 1.703
```

실증분석 LPGA 데이터(공분산 행렬 이용)

각 척도(변수)의 단위는 상이하나 척도간 공분산의 크기가 중요하다면, 원 데이터를 평균으로 표준화 한 후 공분산 행렬을 이용하자.

평균 표준화

```
lpga.center<-scale(lpga[,2:7],scale=F)
stat.desc(lpga.center)
```

평균은 0으로 되었으나 분산은 1이 아니다.

```
> stat.desc(lpga.center)
              평균_비거리 페어웨이_안착율 그린_적중률 평균_퍼팅수   샌드_회수 샌드_세이브
nbr.val       1.570e+02     1.570e+02   1.570e+02   1.570e+02  1.570e+02   1.570e+02
nbr.null      0.000e+00     0.000e+00   0.000e+00   0.000e+00  0.000e+00   0.000e+00
nbr.na        0.000e+00     0.000e+00   0.000e+00   0.000e+00  0.000e+00   0.000e+00
min          -2.191e+01    -1.838e+01  -2.103e+01  -2.245e+00 -4.210e-01  -1.905e+01
max           2.259e+01     1.222e+01   8.673e+00   2.755e+00  5.690e-01   2.205e+01
range         4.450e+01     3.060e+01   2.970e+01   5.000e+00  9.900e-01   4.110e+01
sum           1.847e-12     6.679e-13   3.553e-13  -2.416e-13 -4.885e-15  -1.457e-13
median       -4.108e-01     8.223e-01   7.261e-02  -1.145e-01 -9.554e-04  -5.350e-02
mean          1.177e-14     4.254e-15   2.263e-15  -1.539e-15 -3.111e-17  -9.268e-16
SE.mean       7.531e-01     4.608e-01   3.173e-01   8.269e-02  1.423e-02   6.519e-01
CI.mean.0.95  1.488e+00     9.102e-01   6.267e-01   1.633e-01  2.812e-02   1.288e+00
var           8.905e+01     3.334e+01   1.581e+01   1.073e+00  3.181e-02   6.672e+01
```

평균 표준화 하였으므로 상관계수는 동일하다.

```
> lpga.cor<-rcorr(as.matrix(lpga.center), type="pearson")
> lpga.cor$r
              평균_비거리 페어웨이_안착율 그린_적중률 평균_퍼팅수 샌드_회수 샌드_세이브
평균_비거리       1.00000      -0.4209     0.49147    -0.18799  -0.13521    -0.05425
페어웨이_안착율  -0.42094       1.0000     0.30919     0.16956  -0.27289     0.10093
그린_적중률       0.49147       0.3092     1.00000     0.01841  -0.56845     0.01667
평균_퍼팅수      -0.18799       0.1696     0.01841     1.00000   0.16819    -0.18073
샌드_회수        -0.13521      -0.2729    -0.56845     0.16819   1.00000     0.06436
샌드_세이브      -0.05425       0.1009     0.01667    -0.18073   0.06436     1.00000
```

주성분 구하기

```
lpga.pca.S=prcomp(lpga.center,scale.=F) #공분산행렬 이용
summary(lpga.pca.S) #변동(누적)설명기여율 출력
lpga.pca.S$rotation #부하 출력
```

주성분 2개이면 충분 (누적 기여율 80% 규칙 기준)

```
> summary(lpga.pca.S) #변동(누적)설명기여율 출력
Importance of components:
                          PC1    PC2    PC3     PC4     PC5     PC6
Standard deviation      10.04 8.144 5.724 2.2526 0.98700 0.13590
Proportion of Variance   0.49 0.322 0.159 0.0246 0.00473 0.00009
Cumulative Proportion    0.49 0.811 0.971 0.9952 0.99991 1.00000
```

제 1주성분 : 장타능력

평균비거리 (-) dominant

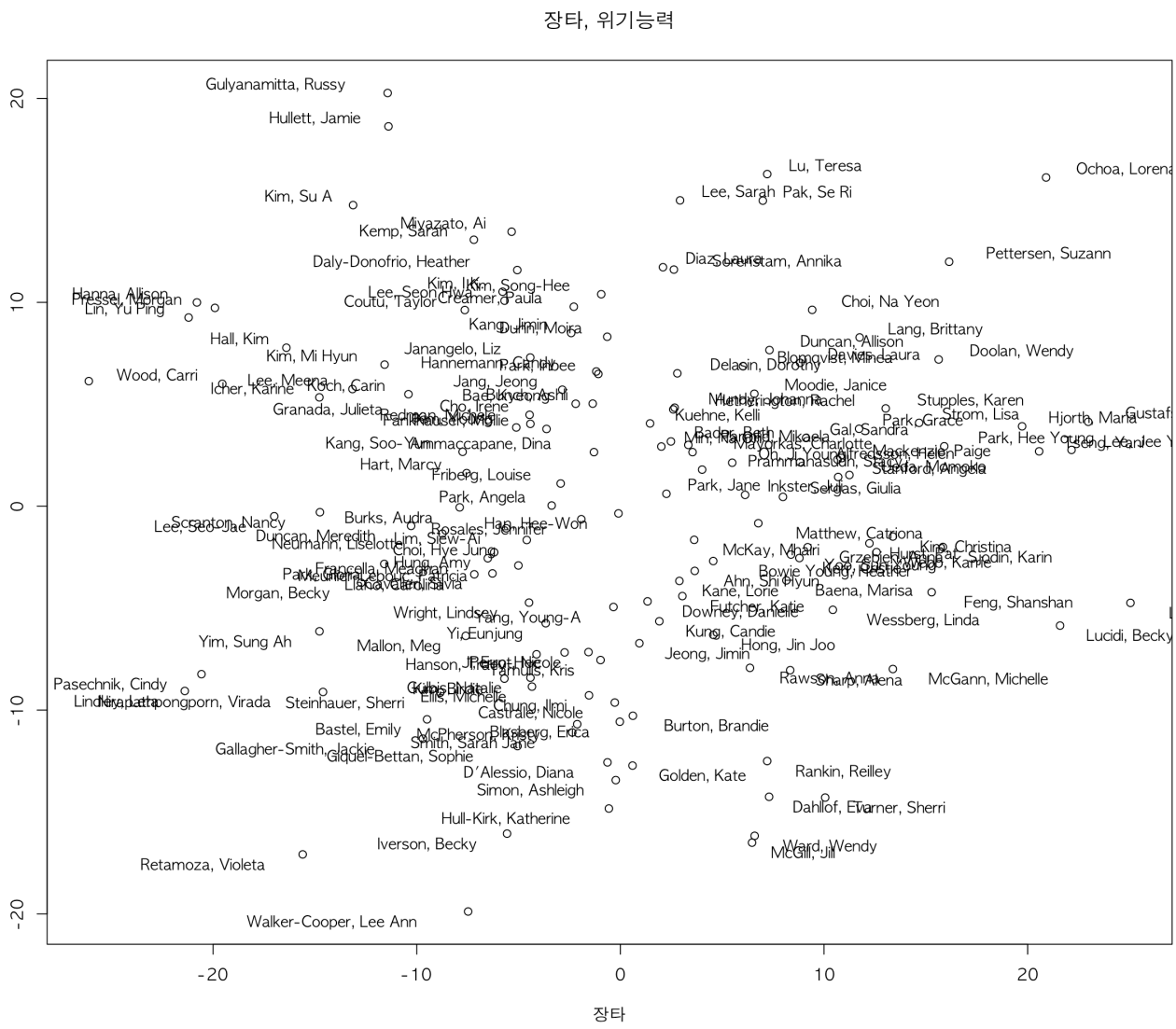
제2주성분 : 위기 능력

샌드세이브(-) dominant

```
> lpga.pca.S$rotation #부하 출력
                       PC1         PC2        PC3        PC4         PC5          PC6
평균_비거리      -0.923306 -0.1620436 -0.16133 -0.30727   0.027940   0.0051465
페어웨이_안착율   0.306363 -0.0446009 -0.83157 -0.46108 -0.006903   0.0001135
그린_적중률      -0.173627 -0.0758908 -0.52047  0.83119 -0.036214 -0.0318738
평균_퍼팅수       0.017656  0.0268101 -0.02350  0.03660  0.997526  0.0449250
샌드_회수         0.002071 -0.0001703  0.01494 -0.02650  0.046260 -0.9984639
샌드_세이브       0.152269 -0.9824838  0.10392  0.00841  0.025715  0.0030071
```

주성분 점수가 모두 음인 값을 클수록 좋은 능력으로 - 변환하였음

```
PC1=lpga.pca.S$x[,1]*(-1) #양의값이 능력 높도록변환
PC2=lpga.pca.S$x[,2]*(-1) #양의값이 능력 높도록변환
plot(PC1,PC2,main='장타, 위기능력',xlab='장타',ylab='위기능력')
textxy(PC1,PC2,lpga$골퍼,cex=0.9)   #선수이름 표시
```



상금과 상관계수 (장타능력, 위기관리능력)과 양의 관계가 있으며 0.34로 비슷함 - 상관계수 행렬의 경우 파온 능력은 0.56으로 상금과 상관관계가 매우 높았음. 성분 구한 방법은 다르지만 위기관리 능력 성분은 상금과 상관계수가 동일함

```
lpga.pca00<-cbind(lpga,PC1,PC2)
rcorr(as.matrix(lpga.pca00[,c(8,10,11)]), type="pearson")
```

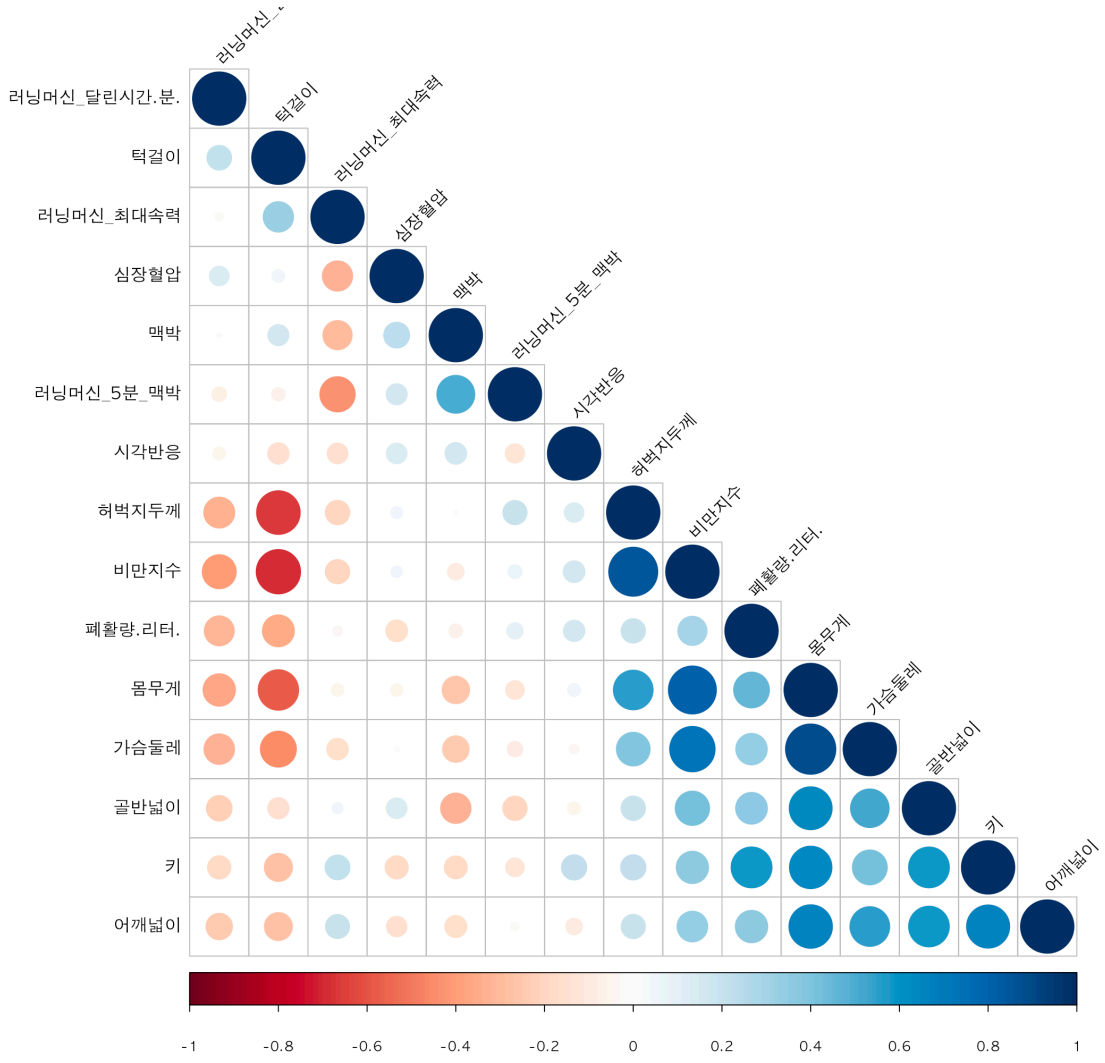
```
> rcorr(as.matrix(lpga
     상금  PC1  PC2
상금 1.00 0.34 0.33
PC1  0.34 1.00 0.00
PC2  0.33 0.00 1.00
```

실증분석 경찰 지원자 데이터(상관계수 행렬 이용) - 지원자 선발

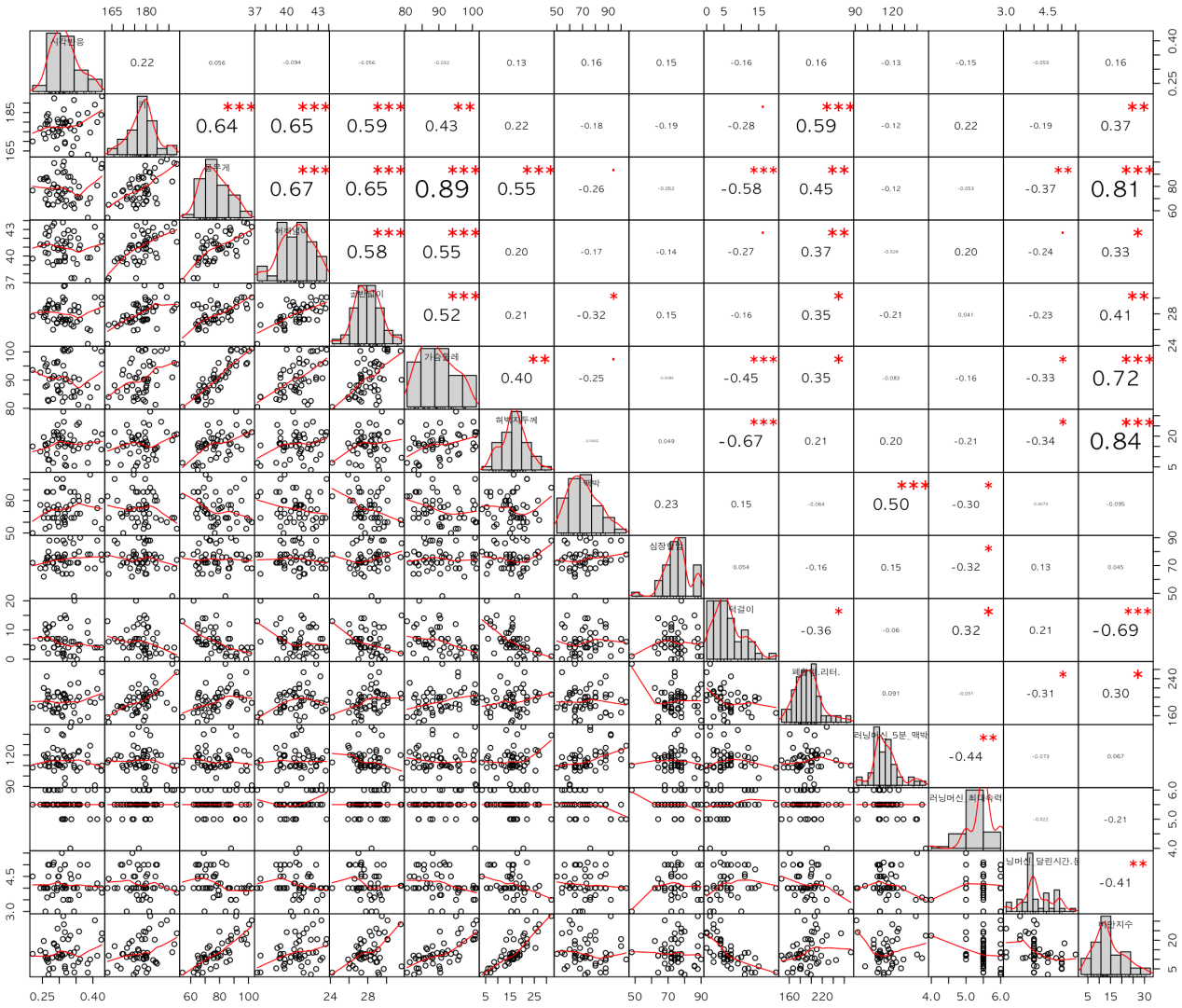
```
police<-read.csv('http://wolfpack.hnu.ac.kr/Stat_Notes/example_data/
poilice_body_exercise.csv',fileEncoding = "utf-8")
names(police)
```

```
> names(police)
 [1] "아이디"              "시각반응"            "키"                  "몸무게"
 [5] "어깨넓이"            "골반넓이"            "가슴둘레"            "허벅지두께"
 [9] "맥박"                "심장혈압"            "턱걸이"              "폐활량.리터."
[13] "러닝머신_5분_맥박"   "러닝머신_최대속력"   "러닝머신_달린시간.분." "비만지수"
```

```
library(Hmisc) #Corelation coefficeinet/p-value
police.cor<-rcorr(as.matrix(police[,2:16]), type="pearson")
library(corrplot) #matrix printout
corrplot(police.cor$r, type = “lower", order = "hclust", tl.col = "black",
         tl.srt = 45)
```



```
library(PerformanceAnalytics) # Matrix of Scatter plot
chart.Correlation(police[,2:16], histogram=TRUE, pch=19)
```



```
police.pca.R=prcomp(police[,3:12,14,15,16],scale.=T) #상관계수행렬 이용
police.pca.R$sdev^2  #고유값 출력
summary(police.pca.R) #변동(누적)설명기여율 출력
police.pca.R$rotation #부하 출력
```

원변수 중 (러닝머신 맥박, 시각반응 변수) 제외 후 주성분분석 : 고유치 1이상 4개 주성분 필요, 누적 기여율 80% 규칙 기준 5개 주성분 필요

```
> police.pca.R$sdev^2  #고유값 출력
 [1] 5.20529785 2.06329250 1.23394882 1.06563429 0.85532452 0
[11] 0.17983546 0.04563330 0.04268967
> summary(police.pca.R) #변동(누적)설명기여율 출력
Importance of components:
                          PC1    PC2     PC3     PC4     PC5
Standard deviation     2.2815 1.4364 1.11083 1.03230 0.92484
Proportion of Variance 0.4004 0.1587 0.09492 0.08197 0.06579
Cumulative Proportion  0.4004 0.5591 0.65404 0.73601 0.80181
```

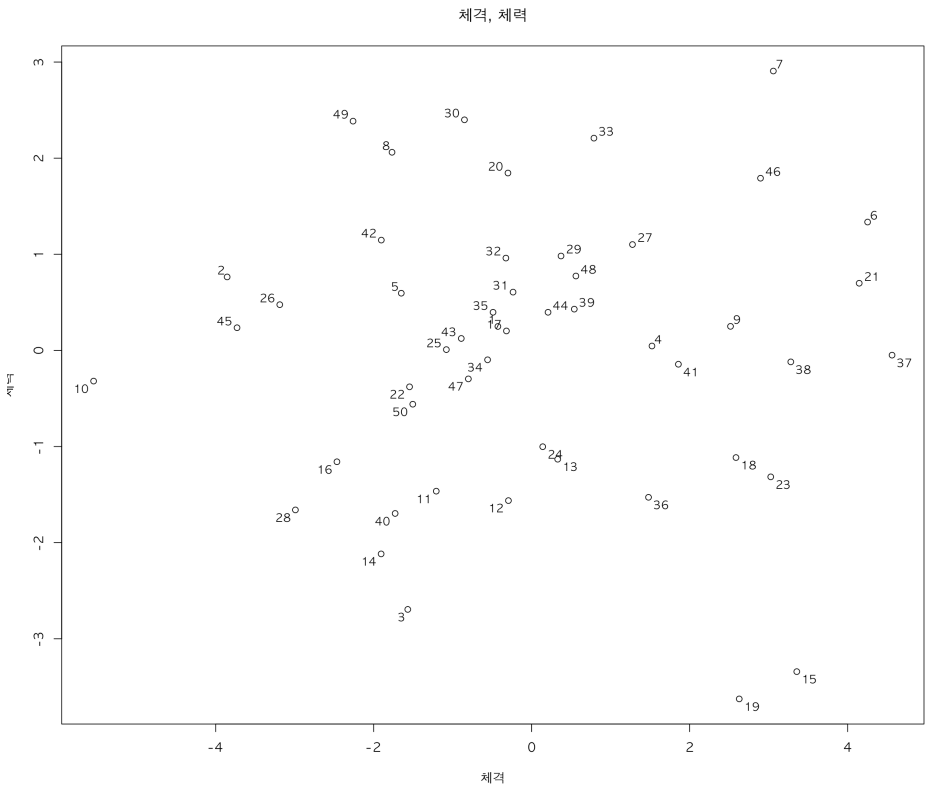
- 제1주성분 : 체격 우수성 (-)
- 2주성분 : 체력 우수성 (+)
- 3주성분 : 체력 우수성2, 골반, 혈압 (-)
- 4주성분 : 러닝 능력 (+)

```
> police.pca.R$rotation #부하 출력
                                PC1          PC2         PC3         PC4
키                       -0.30373811  0.311858525  0.11302395 -0.24892586
몸무게                   -0.41729531  0.009936957  0.07617391  0.09677683 -
어깨넓이                 -0.30225486  0.286673479  0.17594275 -0.10519502 -0
골반넓이                 -0.29453269  0.200366911  0.44564305  0.09270891 -0
가슴둘레                 -0.36230656 -0.039282494  0.12325722  0.15780795 -0
허벅지두께               -0.28507663 -0.354850399 -0.24575344  0.09076689 -0
맥박                      0.11878211 -0.308987436  0.14453678 -0.67740563 -
심장혈압                  0.03660970 -0.321385754  0.67217284  0.07137570 -0
턱걸이                    0.29185670  0.292290063  0.26766526 -0.10071384 -
폐활량.리터.             -0.25206050  0.128445794 -0.05698113 -0.51283091  0
러닝머신_최대속력         0.03036164  0.522962864 -0.20588801  0.12107311 -0.3
러닝머신_달린시간.분.     0.20517031  0.029295122  0.27134414  0.32595189  0.6
비만지수                 -0.36848959 -0.278949478 -0.11254413  0.13361216 -0
```

```
PC1=police.pca.R$x[,1]*(-1) #양의값이 능력 높도록변환
PC2=police.pca.R$x[,2]
PC3=police.pca.R$x[,3]*(-1) #양의값이 능력 높도록변환
PC4=police.pca.R$x[,4]
police.pca<-cbind(police,PC1,PC2,PC3,PC4)
plot(police.pca$PC1,police.pca$PC2,main='체격, 체력',xlab='체격',ylab='체력')
textxy(police.pca$PC1,police.pca$PC2,police.pca$아이디,cex=0.9)  #지원자 아이디 표시
```

체격 우수 지원자 : 37, 6, 21, 38, 15, 7

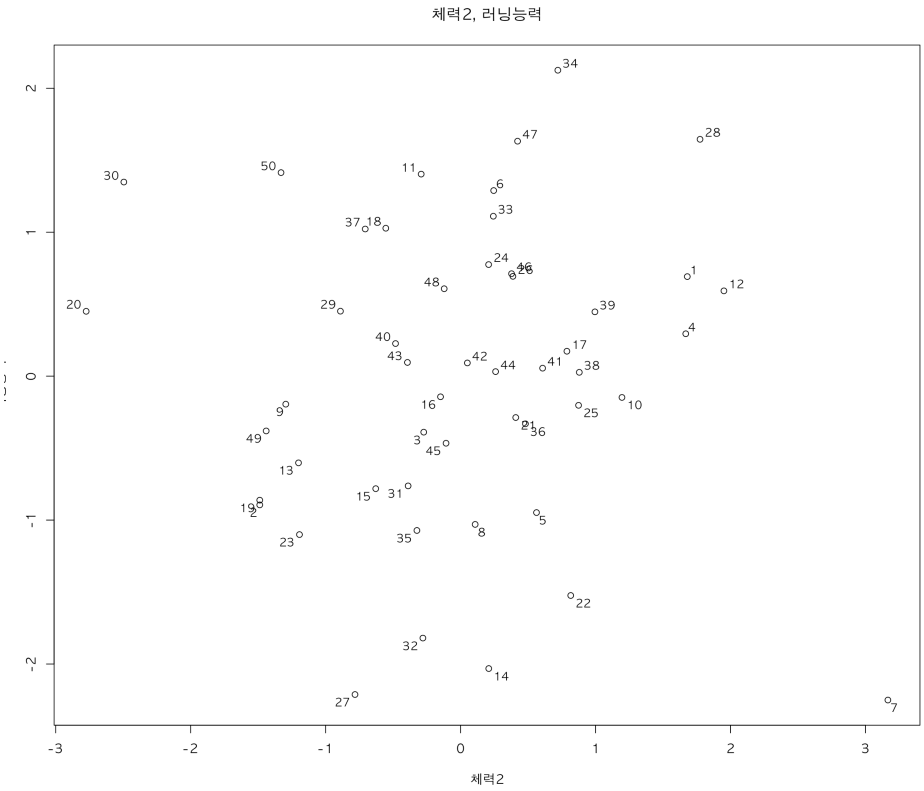
체력 우수 지원자 : 7, 30, 49, 8, 33, 20




```
plot(police.pca$PC3,police.pca$PC4,main='체력2, 러닝능력',xlab='체력2',ylab='러닝능력')
textxy(police.pca$PC3,police.pca$PC4,police.pca$아이디,cex=0.9)  #지원자 아이디 표시
```

체력2 우수 지원자 : 7, 12, 28, 1, 4

체력 우수 지원자 : 34, 28,47



주성분은 서로 독립이므로 4개 주성분의 합으로 지원자의 능력 척도로 이용할 수 있음

4개 주성분의 합 = 지원자 능력 척도

지원자 6, 7, 46, 21, 37 5명을 선발

```
police.pca$PC0=PC1+PC2+PC3+PC4
police.pca0<-police.pca[,c(1,17:21)]
head(police.pca0[order(-police.pca0$PC0),])
```

	아이디	PC1	PC2	PC3	PC4	PC0
6	6	4.2528887	1.33541093	0.2451924	1.2892673	7.122759
7	7	3.0594689	2.90658636	3.1660754	-2.2501429	6.881988
46	46	2.8973530	1.79155858	0.3776063	0.7116877	5.778206
21	21	4.1465710	0.69881792	0.4087024	-0.2875213	4.966570
37	37	4.5622994	-0.04934484	-0.7064664	1.0221878	4.828676
33	33	0.7881872	2.20834345	0.2425902	1.1110003	4.350121

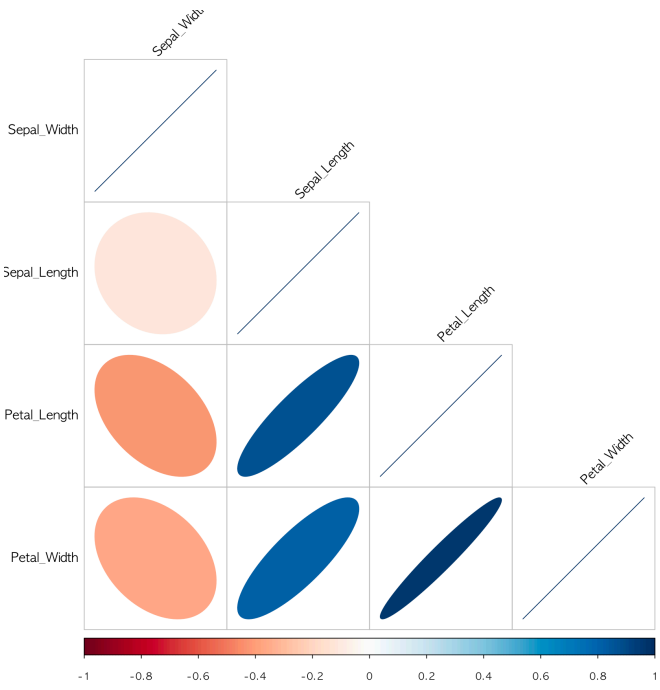
실증분석 IRIS 데이터(공분산 행렬 이용) - 개체 군집/판별

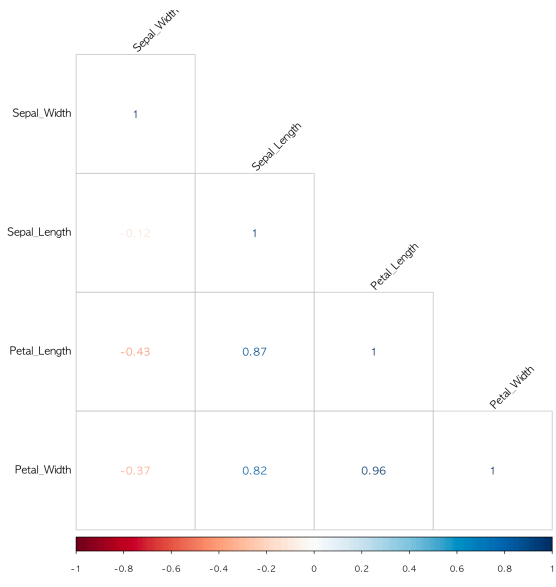
```
iris<-read.csv('http://wolfpack.hnu.ac.kr/Stat_Notes/example_data/iris.csv')
names(iris)
```

```
> names(iris)
[1] "Sepal_Length" "Sepal_Width"  "Petal_Length" "Petal_Width"  "group"
```

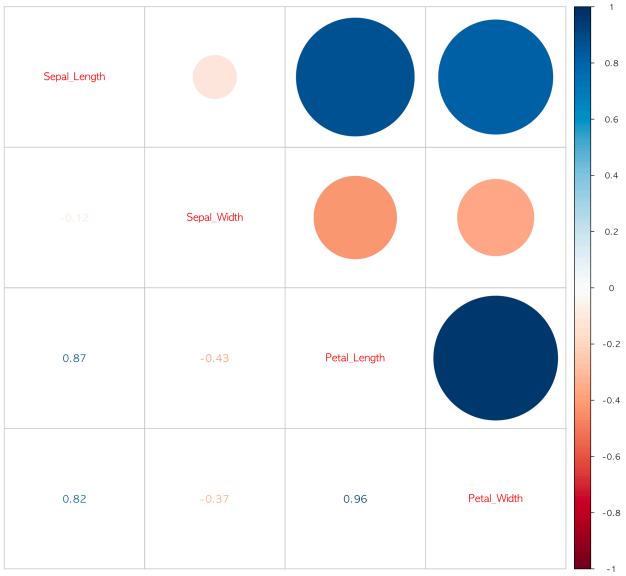
```
library(Hmisc) #Corelation coefficeinet/p-value
iris.cor<-rcorr(as.matrix(iris[,1:4]), type="pearson")
library(corrplot) #matrix printout
corrplot(iris.cor$r, type = "lower", order = "hclust", tl.col = "black",
         tl.srt = 45,method="ellipse")
corrplot(iris.cor$r, type = "lower", order = "hclust", tl.col = "black",
         tl.srt = 45,method="number")
```

Method = 'square', 'pie', 'shade', 'circle' (디폴트)

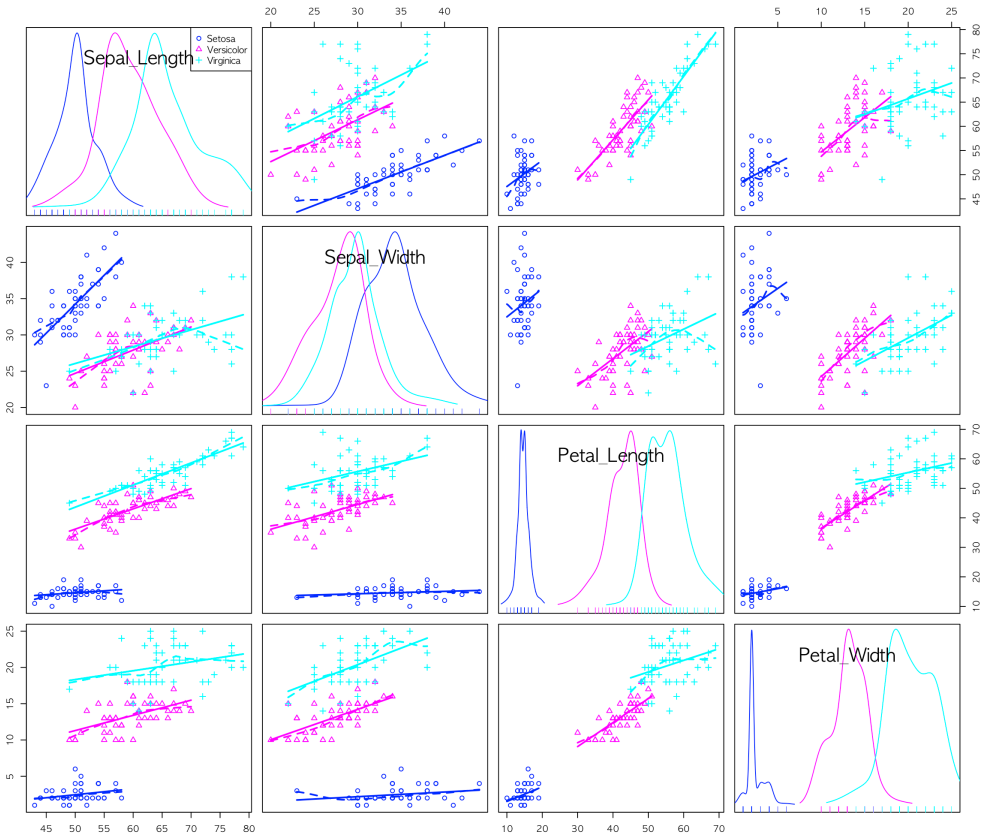




```
corrplot.mixed(iris.cor$r,lower='number',upper='circle')
```




```
library(car) # Matrix of Scatter plot
scatterplotMatrix(~Sepal_Length+Sepal_Width+Petal_Length+Petal_Width|
group, data=iris)
```



```
iris.pca.R=prcomp(iris[,1:4],scale.=T) #상관계수행렬 이용
iris.pca.R$sdev^2  #고유값 출력
iris.pca.R$rotation #부하 출력
```

주성분 2개, 제1주성분 = 꽃크기, 제2주성분 = 꽃받침넓이

```
> iris.pca.R$sdev^2  #고유값 출력
[1] 2.91849782 0.91403047 0.14675688 0.02071484
> iris.pca.R$rotation #부하 출력
                    PC1         PC2        PC3         PC4
Sepal_Length  0.5210659 -0.37741762  0.7195664 -0.2612863
Sepal_Width  -0.2693474 -0.92329566 -0.2443818  0.1235096
Petal_Length  0.5804131 -0.02449161 -0.1421264  0.8014492
Petal_Width   0.5648565 -0.06694199 -0.6342727 -0.5235971
```

```
iris$꽃크기=iris.pca.R$x[,1]
iris$꽃받침넓이=iris.pca.R$x[,2]*(-1) #양의값이 능력 높도록변환
scatterplotMatrix(~꽃크기+꽃받침넓이|group, data=iris)
```

꽃 크기 성분이 분꽃 종을 잘 분류함. Virginia 종의 꽃 크기가 가장 크고 Setosa가 가장 작음