

다변량 데이터 기초 연산 보고서

과목	다변량데이터분석
담당교수	임태진 교수님
전공	산업정보시스템공학과
학번	20201368
이름	한채원
제출일	2022.10.04

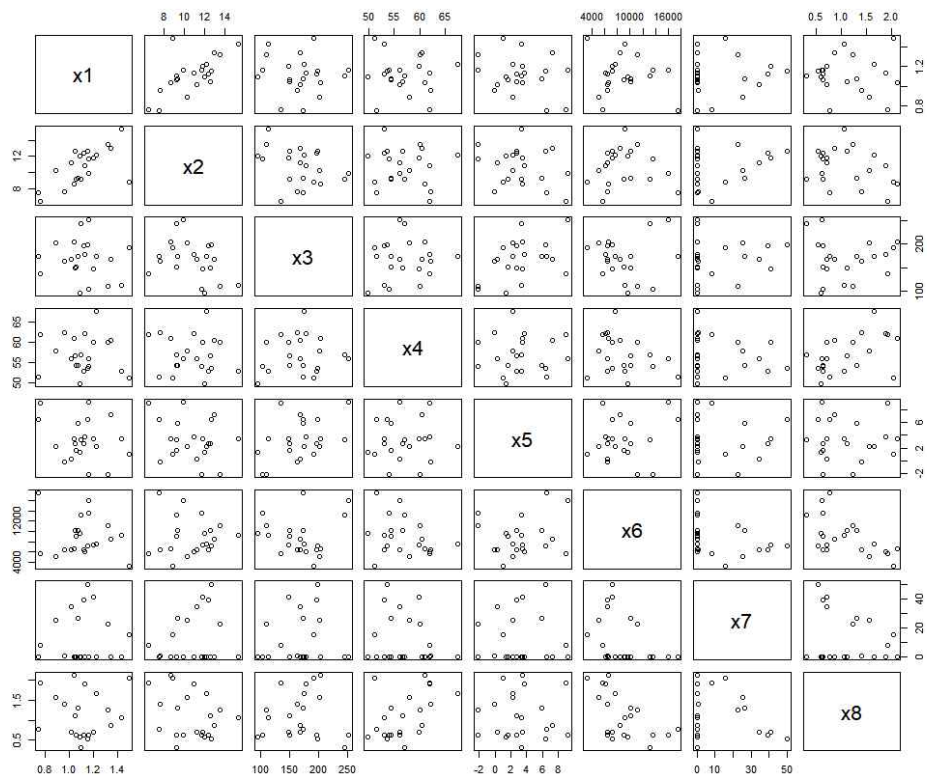
3. 22(= n)개의 회사로부터 8(= p)개의 변수 x_1, x_2, \dots, x_8 에 관해 얻어진 자료가 <표 1.1>에 주어져 있다.

(a) 변수 x_1, x_2, \dots, x_8 에 대해 산점도 행렬을 작성하여라.

```
> setwd("C:/Temp/data")
> com <- read.csv("company.csv", header = TRUE)
> str(com)
'data.frame': 22 obs. of 10 variables:
 $ ID    : int  1 2 3 4 5 6 7 8 9 10 ...
 $ x1    : num  1.06 0.89 1.43 1.02 1.49 1.32 1.22 1.1 1.34 1.12 ...
 $ x2    : num  9.2 10.3 15.4 11.2 8.8 13.5 12.2 9.2 13 12.4 ...
 $ x3    : int  151 202 113 168 192 111 175 245 168 197 ...
 $ x4    : num  54.4 57.9 53 56 51.2 60 67.6 57 60.4 53 ...
 $ x5    : num  1.6 2.2 3.4 0.3 1 -2.2 2.2 3.3 7.2 2.7 ...
 $ x6    : int  9077 5088 9212 6423 3300 11127 7642 13082 8406 6455 ...
 $ x7    : num  0 25.3 0 34.3 15.6 22.5 0 0 0 39.2 ...
 $ x8    : num  0.628 1.555 1.058 0.7 2.044 ...
 $ class: int  2 3 2 1 3 2 3 2 2 1 ...
> com.X = com[-c(1,10)]
```

company 데이터의 구조를 살펴보기 위해 str 함수를 이용하였고 문제에서는 변수 x_1, x_2, \dots, x_8 에 대해 산점도 행렬을 작성하라고 했으므로 첫 번째 변수(ID)와 마지막 변수(class)는 제거해주었고 이를 com.X에 저장해 주었다. 위의 코드 실행 결과 다음과 같은 산점 행렬도가 나왔음을 알 수 있다.

```
> plot(com.X)
```



(b) mean, cov, cor 함수를 이용하여 \bar{x}, S, R 을 얻고, (a)에서 작성된 산점도 행렬과 비교하여 이들을 해석하여라.

(1) \bar{x} 구하기

```
> colMeans(com.X)
```

	x1	x2	x3	x4	x5	x6
x1	1.114091	10.736364	168.181818	56.977273	3.240909	8914.045455
	x7	x8				
x7	12.000000	1.102727				

(2) S 구하기

```
> cov(com.X)
```

	x1	x2	x3	x4	x5
x1	0.034044372	0.2661299	-0.7812554	-6.752165e-02	-0.14908009
x2	0.266129870	5.0357576	-32.1259740	-8.643723e-01	-1.82012987
x3	-0.781255411	-32.1259740	1696.7272727	1.843290e+01	55.92077922

x4	-0.067521645	-0.8643723	18.4329004	1.990184e+01	0.46573593
x5	-0.149080087	-1.8201299	55.9207792	4.657359e-01	9.72348485
x6	-99.346385281	-76.6160173	4092.5151515	-4.560037e+03	1952.87424242
x7	0.138809524	7.9676190	79.3095238	-1.229762e+01	-1.00142857
x8	-0.001372165	-0.4088848	0.1195758	1.204446e+00	-0.01236926

	x6	x7	x8
x1	-9.934639e+01	1.388095e-01	-1.372165e-03
x2	-7.661602e+01	7.967619e+00	-4.088848e-01
x3	4.092515e+03	7.930952e+01	1.195758e-01
x4	-4.560037e+03	-1.229762e+01	1.204446e+00
x5	1.952874e+03	-1.001429e+00	-1.236926e-02
x6	1.260239e+07	-2.227602e+04	-1.106557e+03
x7	-2.227602e+04	2.819686e+02	-1.728324e+00
x8	-1.106557e+03	-1.728324e+00	3.092451e-01

(3) R 구하기

> cor(com.X)

	x1	x2	x3	x4	x5
x1	1.00000000	0.642744766	-0.102793192	-0.08203019	-0.259111089
x2	0.64274477	1.00000000	-0.347550467	-0.08634194	-0.260111168
x3	-0.10279319	-0.347550467	1.00000000	0.10030926	0.435367718
x4	-0.08203019	-0.086341943	0.100309264	1.00000000	0.033479746
x5	-0.25911109	-0.260111168	0.435367718	0.03347975	1.00000000
x6	-0.15167116	-0.009617468	0.027987098	-0.28793559	0.176415568
x7	0.04480188	0.211444212	0.114661857	-0.16416254	-0.019125318
x8	-0.01337310	-0.327655318	0.005220183	0.48550006	-0.007133152

	x6	x7	x8
x1	-0.151671159	0.04480188	-0.013373101
x2	-0.009617468	0.21144421	-0.327655318
x3	0.027987098	0.11466186	0.005220183
x4	-0.287935594	-0.16416254	0.485500063
x5	0.176415568	-0.01912532	-0.007133152
x6	1.000000000	-0.37368952	-0.560526327
x7	-0.373689523	1.00000000	-0.185085916
x8	-0.560526327	-0.18508592	1.000000000

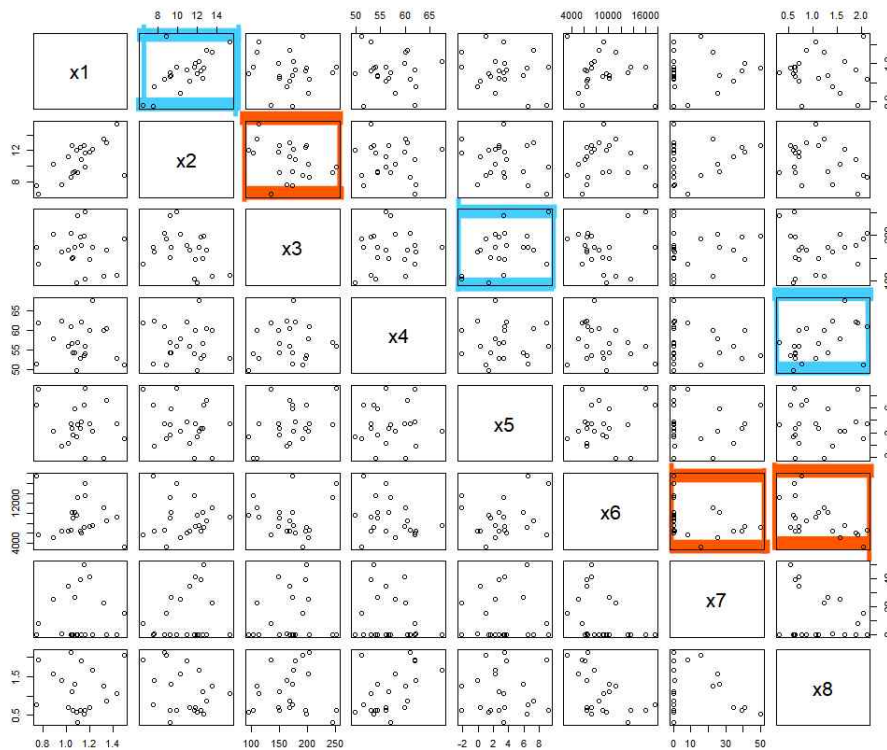
상관행렬에서 주의 깊게 봐야할 값들에 대해 다음과 같이 양의 상관은 파란색으로 음의 상관은 붉은색으로 표시해보았다.

```
> cor(com.X)
```

	x1	x2	x3	x4	x5
x1	1.00000000	0.642744766	-0.102793192	-0.08203019	-0.259111089
x2	0.64274477	1.00000000	-0.347550467	-0.08634194	-0.260111168
x3	-0.10279319	-0.347550467	1.00000000	0.10030926	0.435367718
x4	-0.08203019	-0.08634194	0.10030926	1.00000000	0.033479746
x5	-0.25911109	-0.260111168	0.435367718	0.03347975	1.00000000
x6	-0.15167116	-0.009617468	0.027987098	-0.28793559	0.176415568
x7	0.04480188	0.211444212	0.114661857	-0.16416254	-0.019125318
x8	-0.01337310	-0.327655318	0.005220183	0.48550006	-0.007133152

	x6	x7	x8
x1	-0.151671159	0.04480188	-0.013373101
x2	-0.009617468	0.21144421	-0.327655318
x3	0.027987098	0.11466186	0.005220183
x4	-0.287935594	-0.16416254	0.485500063
x5	0.176415568	-0.01912532	-0.007133152
x6	1.00000000	-0.37368952	-0.560526327
x7	-0.373689523	1.00000000	-0.185085916
x8	-0.560526327	-0.18508592	1.00000000

상관계수는 측정 단위에 무관하고 r 이 1에 가까울수록 큰 양의 상관을, -1에 가까울수록 큰 음의 상관을 가지는 것을 의미하고, 0에 가까울수록 두 변수는 선형적 연관성이 작음을 의미한다. 따라서 $r_{21}=0.6427$, $r_{53}=0.4354$, $r_{84}=0.4855$ 로 양의 상관을 가지는 것을 알 수 있고, $r_{32}=-0.3476$, $r_{76}=-0.3737$, $r_{86}=-0.5605$ 로 음의 상관을 가지는 것을 알 수 있다. 이를 산점도 행렬과 다시 비교해보겠다.



파란색 테두리로 표시한 부분에서 (x_1, x_2) , (x_5, x_3) , (x_8, x_4) 은 점의 분포가 양의 선형 관계, 붉은색 테두리로 표시한 부분에서는 (x_3, x_2) , (x_7, x_6) , (x_8, x_6) 이 비교적 음의 선형 관계로 점이 분포되어 있음을 알 수 있다.

4. 위의 데이터에 대하여

(a) <예 1.2>에서와 같이 scale 함수를 이용하여 중심화된 자료행렬 C 와 표준화된 자료행렬 Z 를 구하여라.

(1) C 구하기

```
> com.C = scale(com.X, center = TRUE, scale=TRUE)
```

```
> com.C
```

	x1	x2	x3	x4	x5	x6	x7	x8
[1.]	-0.054090909	-1.5363636	-17.1818182	-2.57727273	-1.64090909	162.9545	-12.0	-0.474727273
[2.]	-0.224090909	-0.4363636	33.8181818	0.92272727	-1.04090909	-3826.0455	13.3	0.452272727
[3.]	0.315909091	4.6636364	-55.1818182	-3.97727273	0.15909091	297.9545	-12.0	-0.044727273
[4.]	-0.094090909	0.4636364	-0.1818182	-0.97727273	-2.94090909	-2491.0455	22.3	-0.402727273
[5.]	0.375909091	-1.9363636	23.8181818	-5.77727273	-2.24090909	-5614.0455	3.6	0.941272727
[6.]	0.205909091	2.7636364	-57.1818182	3.02272727	-5.44090909	2212.9545	10.5	0.138272727
[7.]	0.105909091	1.4636364	6.8181818	10.62272727	-1.04090909	-1272.0455	-12.0	0.549272727
[8.]	-0.014090909	-1.5363636	76.8181818	0.02272727	0.05909091	4167.9545	-12.0	-0.793727273
[9.]	0.225909091	2.2636364	-0.1818182	3.42272727	3.95909091	-508.0455	-12.0	-0.240727273
[10.]	0.005909091	1.6636364	28.8181818	-3.97727273	-0.54090909	-2459.0455	27.2	-0.479727273
[11.]	-0.364090909	-3.2363636	4.8181818	-5.47727273	3.25909091	8526.9545	-12.0	-0.334727273
[12.]	0.015909091	0.1636364	9.8181818	5.02272727	0.45909091	-2760.0455	-12.0	0.794272727
[13.]	0.035909091	1.9636364	30.8181818	-3.27727273	3.15909091	-1735.0455	38.2	-0.575727273
[14.]	-0.024090909	1.2636364	-72.1818182	-7.17727273	-1.84090909	758.9545	-12.0	-0.514727273
[15.]	-0.154090909	-3.1363636	-4.1818182	5.22272727	-3.34090909	-2446.0455	-11.1	0.297272727
[16.]	0.045909091	-0.8363636	83.8181818	-0.97727273	5.95909091	7076.9545	-12.0	-0.482727273
[17.]	-0.354090909	-4.3363636	-32.1818182	4.92272727	5.75909091	-3200.0455	-3.7	0.817272727
[18.]	-0.064090909	1.8636364	-18.1818182	-0.27727273	-0.54090909	1225.9545	-12.0	0.005272727
[19.]	0.045909091	0.9636364	-64.1818182	-2.97727273	-5.34090909	4592.9545	-12.0	-0.466727273
[20.]	0.085909091	1.0636364	-20.1818182	2.92272727	0.25909091	-1627.0455	29.1	-0.400727273
[21.]	-0.074090909	-2.1363636	35.8181818	4.02272727	0.25909091	-2264.0455	-12.0	1.013272727
[22.]	-0.044090909	-1.4363636	5.8181818	-2.67727273	2.65909091	1178.9545	14.6	0.203272727

```
attr(,"scaled:center")
      x1      x2      x3      x4      x5      x6      x7      x8
1.114091 10.736364 168.181818 56.977273 3.240909 8914.045455 12.000000 1.102727
```

(2) Z 구하기

```
> com.Z = scale(com.X, center = TRUE, scale=TRUE)
```

```
> com.Z
```

	x1	x2	x3	x4	x5	x6	x7
[1,]	-0.29315791	-0.68463896	-0.417122002	-0.57771516	-0.52622751	0.04590290	-0.7146294
[2,]	-1.21451134	-0.19445367	0.821002037	0.20683629	-0.33381191	-1.07776413	0.7920476
[3,]	1.71214073	2.07822360	-1.339645796	-0.89153574	0.05101929	0.08393124	-0.7146294
[4,]	-0.50994695	0.20660702	-0.004413989	-0.21906307	-0.94312798	-0.70170610	1.3280197
[5,]	2.03732429	-0.86288816	0.578232617	-1.29501935	-0.71864311	-1.58142837	0.2143888
[6,]	1.11597086	1.23153991	-1.388199680	0.67756716	-1.74485965	0.62337028	0.6253007
[7,]	0.57399826	0.65223002	0.165524604	2.38116460	-0.33381191	-0.35832428	-0.7146294
[8,]	-0.07636887	-0.68463896	1.864910540	0.00509449	0.01895002	1.17407698	-0.7146294
[9,]	1.22436538	1.00872841	-0.004413989	0.76723019	1.26965142	-0.14311204	-0.7146294
[10,]	0.03202565	0.74135462	0.699617327	-0.89153574	-0.17346558	-0.69269198	1.6198267
[11,]	-1.97327298	-1.44219805	0.116970720	-1.22777208	1.04516655	2.40196983	-0.7146294
[12,]	0.08622291	0.07292013	0.238355430	1.12588228	0.14722709	-0.77748109	-0.7146294
[13,]	0.19461744	0.87504152	0.748171211	-0.73462545	1.01309729	-0.48874740	2.2749037
[14,]	-0.13056613	0.56310542	-1.752353809	-1.60883993	-0.59036605	0.21379097	-0.7146294
[15,]	-0.83513051	-1.39763576	-0.101521757	1.17071379	-1.07140505	-0.68902999	-0.6610322
[16,]	0.24881470	-0.37270287	2.034849134	-0.21906307	1.91103676	1.99351729	-0.7146294
[17,]	-1.91907572	-1.93238335	-0.781276132	1.10346652	1.84689822	-0.90142531	-0.2203441
[18,]	-0.34735517	0.83047922	-0.441398944	-0.06215278	-0.17346558	0.34534086	-0.7146294
[19,]	0.24881470	0.42941852	-1.558138274	-0.66737818	-1.71279038	1.29379583	-0.7146294
[20,]	0.46560374	0.47398082	-0.489952828	0.65515141	0.08308855	-0.45832473	1.7329764
[21,]	-0.40155243	-0.95201276	0.869555920	0.90172472	0.08308855	-0.63776215	-0.7146294
[22,]	-0.23896065	-0.64007666	0.141247662	-0.60013092	0.85275095	0.33210137	0.8694658

x8

[1,]	-0.85367545
[2,]	0.81329670
[3,]	-0.08043055
[4,]	-0.72420189
[5,]	1.69263800
[6,]	0.24864810
[7,]	0.98772637
[8,]	-1.42731528
[9,]	-0.43288637
[10,]	-0.86266667
[11,]	-0.60192130
[12,]	1.42829614
[13,]	-1.03529809
[14,]	-0.92560521
[15,]	0.53456889

```

[16.] -0.86806140
[17.]  1.46965575
[18.]  0.00948165
[19.] -0.83928950
[20.] -0.72060540
[21.]  1.82211157
[22.]  0.36553395
attr(,"scaled:center")
      x1      x2      x3      x4      x5      x6      x7      x8
1.114091 10.736364 168.181818 56.977273  3.240909 8914.045455 12.000000  1.102727
attr(,"scaled:scale")
      x1      x2      x3      x4      x5      x6      x7
0.1845112  2.2440494 41.1913495  4.4611478  3.1182503 3549.9840305 16.7919198
      x8
0.5560981

```

(b) cov함수를 이용하여 표준화된 자료행렬 Z 로부터 공분산행렬 S_z 를 얻고, 이것이 문제 3에서 구한 R_x 와 같음을 확인하여라.

```

> cov(com.Z)
      x1      x2      x3      x4      x5      x6      x7
x1  1.00000000  0.642744766 -0.102793192 -0.08203019 -0.259111089 -0.151671159  0.04480188
x2  0.64274477  1.000000000 -0.347550467 -0.08634194 -0.260111168 -0.009617468  0.21144421
x3 -0.10279319 -0.347550467  1.000000000  0.10030926  0.435367718  0.027987098  0.11466186
x4 -0.08203019 -0.086341943  0.100309264  1.000000000  0.033479746 -0.287935594 -0.16416254
x5 -0.25911109 -0.260111168  0.435367718  0.03347975  1.000000000  0.176415568 -0.01912532
x6 -0.15167116 -0.009617468  0.027987098 -0.28793559  0.176415568  1.000000000 -0.37368952
x7  0.04480188  0.211444212  0.114661857 -0.16416254 -0.019125318 -0.373689523  1.00000000
x8 -0.01337310 -0.327655318  0.005220183  0.48550006 -0.007133152 -0.560526327 -0.18508592
      x8
x1 -0.013373101
x2 -0.327655318
x3  0.005220183
x4  0.485500063
x5 -0.007133152
x6 -0.560526327
x7 -0.185085916
x8  1.000000000

```

위의 값은 S_z 이고, 아래 값은 R_x 이다.

(3) R 구하기

```
> cor(com.X)
```

	x1	x2	x3	x4	x5
x1	1.00000000	0.642744766	-0.102793192	-0.08203019	-0.259111089
x2	0.64274477	1.000000000	-0.347550467	-0.08634194	-0.260111168
x3	-0.10279319	-0.347550467	1.000000000	0.10030926	0.435367718
x4	-0.08203019	-0.086341943	0.100309264	1.00000000	0.033479746
x5	-0.25911109	-0.260111168	0.435367718	0.03347975	1.000000000
x6	-0.15167116	-0.009617468	0.027987098	-0.28793559	0.176415568
x7	0.04480188	0.211444212	0.114661857	-0.16416254	-0.019125318
x8	-0.01337310	-0.327655318	0.005220183	0.48550006	-0.007133152

	x6	x7	x8
x1	-0.151671159	0.04480188	-0.013373101
x2	-0.009617468	0.21144421	-0.327655318
x3	0.027987098	0.11466186	0.005220183
x4	-0.287935594	-0.16416254	0.485500063
x5	0.176415568	-0.01912532	-0.007133152
x6	1.000000000	-0.37368952	-0.560526327
x7	-0.373689523	1.00000000	-0.185085916
x8	-0.560526327	-0.18508592	1.000000000

둘의 값은 정확히 일치함을 알 수 있다. 따라서 상관계수는 표준화된 공분산임을 알 수 있으며, 결국 자료 분석에서 상관 행렬을 분석 대상으로 한다는 것은 모든 변수를 표준화한 후 분석한다는 것과 같은 의미를 가진다.

(c) 각 집단(Class)별로 평균벡터와 표준편차를 구하고 이를 비교하여라(aggregate 함수 또는 tapply 함수를 이용하여라).

(1) 평균 구하기

```
> aggregate(~class, data=com[-1], mean)
```

	class	x1	x2	x3	x4	x5	x6	x7	x8
1	1	1.122500	12.025000	178.0000	55.65000	3.225000	6836.000	41.200000	0.6380000
2	2	1.139091	11.209091	157.9091	55.19091	3.354545	11613.545	4.463636	0.8294545
3	3	1.070000	9.257143	178.7143	60.54286	3.071429	5859.429	7.157143	1.7977143

(2) 표준편차 구하기

```
> aggregate(~class, data=com[-1], sd)
```

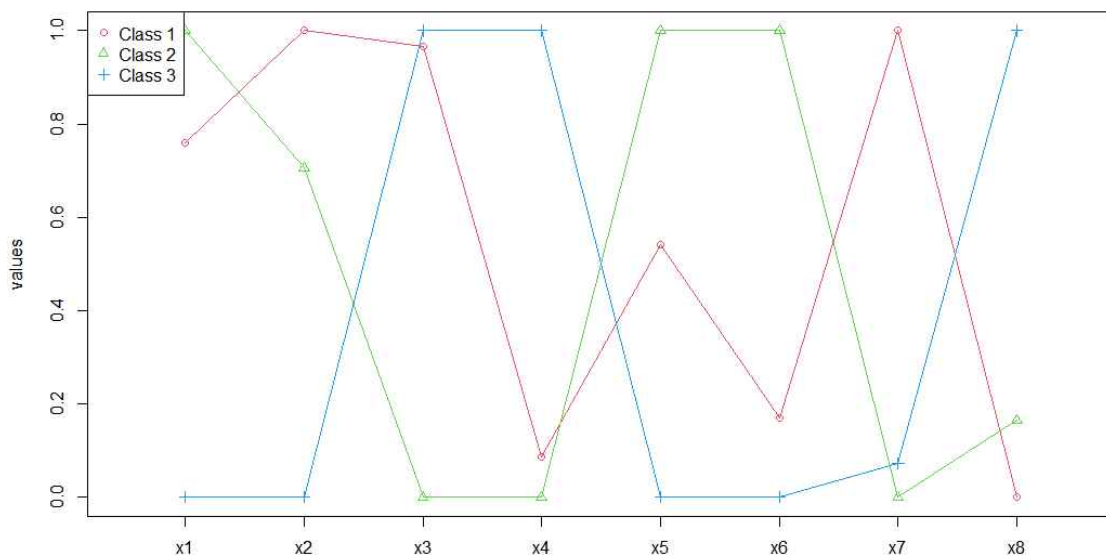
	class	x1	x2	x3	x4	x5	x6	x7	x8
1	1	0.07588368	0.6652067	24.50850	3.109662	2.515784	460.7168	6.648809	0.08263575
2	2	0.18245298	2.3666240	53.00274	3.267860	3.650579	2997.3077	9.973191	0.31339667
3	3	0.23944380	1.9998809	23.85172	5.016592	2.932413	1380.0134	9.954707	0.26617458

결과 값을 보면 유독 집단 2의 평균과 표준편차가 변수 x_6 에서 집단 1과 집단 3보다 더 큰 값을 알 수 있다. 따라서 변수 간 분포 상태가 고르지 않으므로 정확한 분석을 위

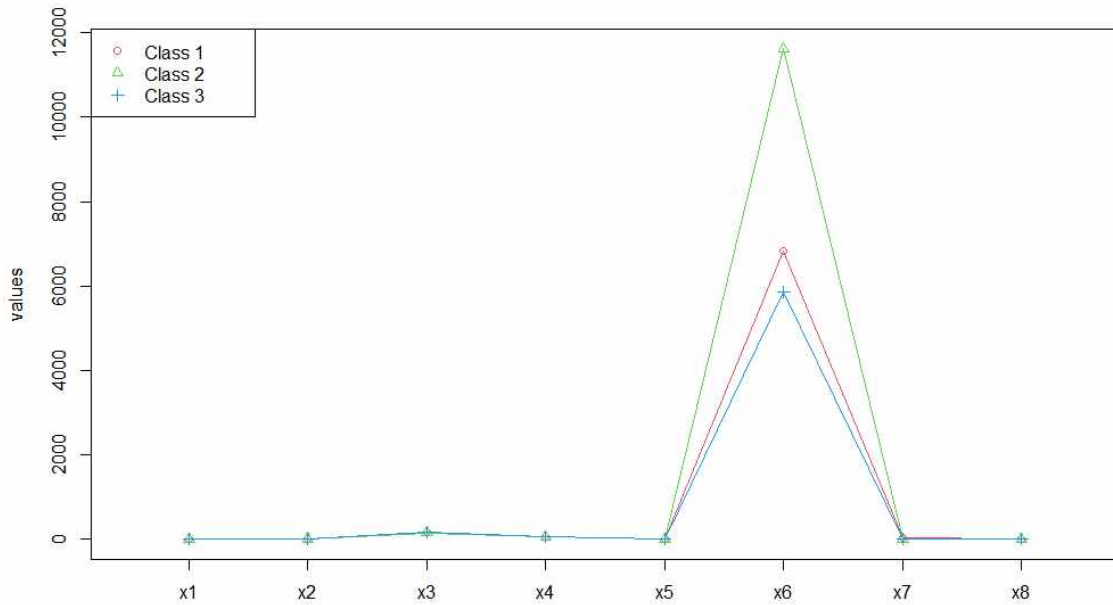
해 척도화가 필요하다.

(d) (c)에서 구한 각 집단별 평균벡터에 대하여 프로파일도표를 작성하고 이를 해석하여라(예 1.10을 참조하여라).

```
> com.M = aggregate(~class, data=com[-1], mean)
> library(plotrix)
> ladderplot(com.M[-1],scale=TRUE,lty=1,pch=1:3,col=2:4,vertical = TRUE)
> legend("topleft",pch=1:3,col=2:4,c("Class 1","Class 2","Class 3"))
```



집단 2와 집단 3을 보면 그래프가 서로 반대 양상을 보이고 있음을 알 수 있다. 집단 2 같은 경우 x_5 에서 큰 수치값을 가지지만 집단 3의 경우는 가장 작은 수치값을 가짐을 파악할 수 있다. 만약 x_5 가 마케팅 역량지수라고 보면 집단 2는 집단 3보다 우월한 실적을 보인다고 판단할 수 있다. 집단 1의 경우는 비교적 집단 2와 3보다 모든 변수에서 골고루 어느 정도의 수치값을 유지하고 있다. 만약 `scale = FALSE`로 설정하면 아래와 같이 x_6 에서의 값이 튀므로 `scale = TRUE`로 설정해준다.



-> scale=FALSE로 설정했을 시의 그래프

(e) 각 자료값 x_{ij} 에 대해 다음과 같은 변환(0-1 standardization)을 취하여 얻은 새로운 자료행렬 V 를 계산하여라(아래 예 1.12를 참조하여라). 즉,

$$V = \{v_{ij}\} = \left\{ \frac{x_{ij} - \min\{x_j\}}{\max\{x_j\} - \min\{x_j\}} \right\}$$

> com.X.scale = as.data.frame(lapply(com.X,rescale))

> com.X.scale

	x1	x2	x3	x4	x5	x6	x7
1	0.41891892	0.31111111	0.35256410	0.25842697	0.33333333	0.4085284	0.00000000
2	0.18918919	0.43333333	0.67948718	0.45505618	0.38596491	0.1264408	0.50398406
3	0.91891892	1.00000000	0.10897436	0.17977528	0.49122807	0.4180751	0.00000000
4	0.36486486	0.53333333	0.46153846	0.34831461	0.21929825	0.2208472	0.68326693
5	1.00000000	0.26666667	0.61538462	0.07865169	0.28070175	0.00000000	0.31075697
6	0.77027027	0.7888889	0.09615385	0.57303371	0.00000000	0.5534969	0.44820717
7	0.63513514	0.6444444	0.50641026	1.00000000	0.38596491	0.3070504	0.00000000
8	0.47297297	0.31111111	0.95512821	0.40449438	0.48245614	0.6917474	0.00000000
9	0.79729730	0.73333333	0.46153846	0.59550562	0.82456140	0.3610777	0.00000000
10	0.50000000	0.66666667	0.64743590	0.17977528	0.42982456	0.2231101	0.78087649
11	0.00000000	0.1222222	0.49358974	0.09550562	0.76315789	1.0000000	0.00000000
12	0.51351351	0.5000000	0.52564103	0.68539326	0.51754386	0.2018245	0.00000000
13	0.54054054	0.7000000	0.66025641	0.21910112	0.75438596	0.2743087	1.00000000
14	0.45945946	0.6222222	0.00000000	0.00000000	0.31578947	0.4506753	0.00000000
15	0.28378378	0.13333333	0.43589744	0.69662921	0.18421053	0.2240294	0.01792829

```

16 0.55405405 0.3888889 1.00000000 0.34831461 1.00000000 0.8974613 0.00000000
17 0.01351351 0.0000000 0.25641026 0.67977528 0.98245614 0.1707093 0.16533865
18 0.40540541 0.6888889 0.34615385 0.38764045 0.42982456 0.4836999 0.00000000
19 0.55405405 0.5888889 0.05128205 0.23595506 0.00877193 0.7218019 0.00000000
20 0.60810811 0.6000000 0.33333333 0.56741573 0.50000000 0.2819461 0.81872510
21 0.39189189 0.2444444 0.69230769 0.62921348 0.50000000 0.2368998 0.00000000
22 0.43243243 0.3222222 0.50000000 0.25280899 0.71052632 0.4803762 0.52988048

```

x8

```

1 0.1765357
2 0.6895407
3 0.4144992
4 0.2163807
5 0.9601550
6 0.5157720
7 0.7432208
8 0.0000000
9 0.3060321
10 0.1737687
11 0.2540122
12 0.8788046
13 0.1206419
14 0.1543996
15 0.6037631
16 0.1721085
17 0.8915329
18 0.4421693
19 0.1809629
20 0.2174875
21 1.0000000
22 0.5517432

```

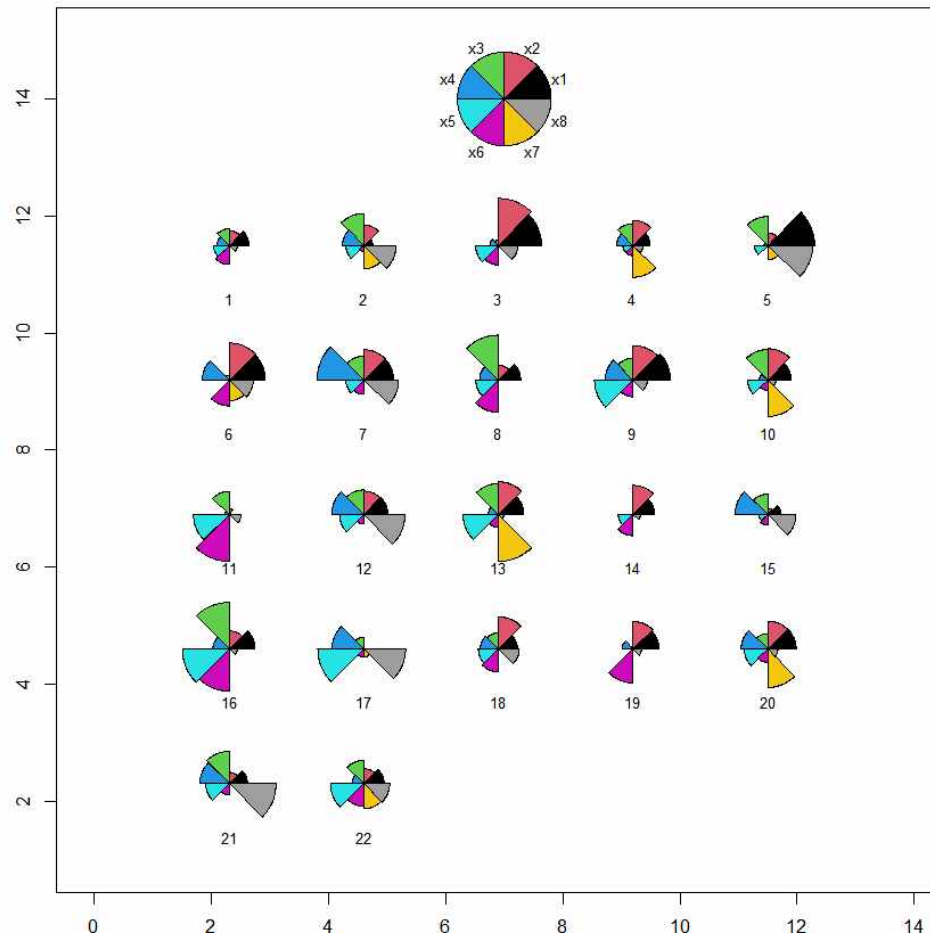
(f) 자료행렬 V 에 대하여 별도표와 레이더도표를 각각 작성하고 이들을 해석하여라 (예 1.7과 예 1.9를 참조하여라).

(1) 별도표 작성

```

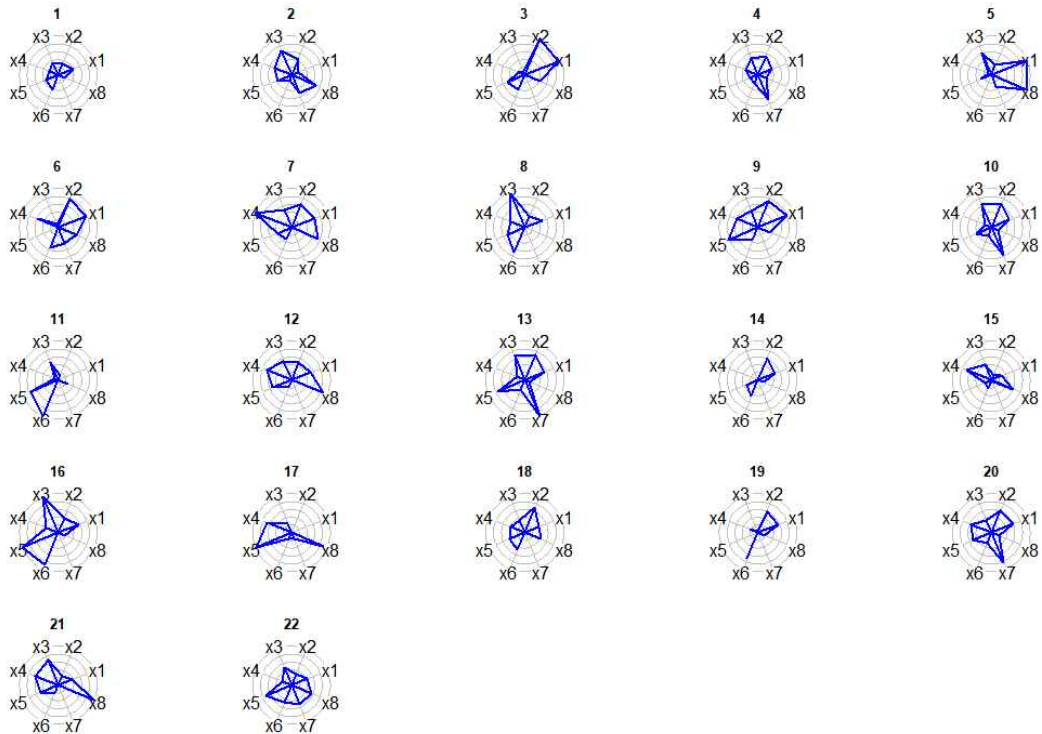
> stars(com.X.scale,full = TRUE,scale = TRUE,radius = TRUE,draw.segments
= TRUE, frame.plot = TRUE, labels = rownames(com.X.scale), nrow = 5, ncol
= 5, cex = 0.8, len = 0.8, lwd = 1, axes = TRUE, ylim = c(1,15), key.loc =
c(7,14))

```



(2) 레이더표 작성

```
> par(mfrow=c(6,5))
> for (i in 1:nrow(com.X.scale)) {radial.plot(com.X.scale[i,],rp.type =
"rp",radial.lim = c(0,1), lty = 1, lwd = 2, labels = names(com.X.scale),
show.grid = TRUE, show.radial.grid = TRUE, show.grid.labels = F, rad.col =
"gray", line.col = "blue", grid.col = "gray", main = rownames(com.X.scale)[i])}
```



별도표와 레이더표를 보면 상당히 비슷한 양상을 보이고 있음을 알 수 있다. 22개의 항목에 대해 공통적으로 나타나는 점은 변수간(x_1, x_2, \dots, x_8)에 균형이 없음을 알 수 있다. 균형이 고르지 않고 치우쳐져 있음을 알 수 있다. 위에서 구한 자료행렬 V값의 수치를 자세히 보면

	<u>x1</u>	<u>x2</u>	<u>x3</u>	<u>x4</u>	<u>x5</u>	<u>x6</u>	<u>x7</u>
1	0.41891892	0.31111111	0.35256410	0.25842697	0.33333333	0.4085284	0.00000000
2	0.18918919	0.43333333	0.67948718	0.45505618	0.38596491	0.1264408	0.50398406
3	0.91891892	1.00000000	0.10897436	0.17977528	0.49122807	0.4180751	0.00000000

ID가 1번인 사람은 x_7 에 해당하는 값이 0인 반면에 x_1 에 해당하는 값은 0.4189로 비교적 차이가 나고 있음을 알 수 있다. 이는 ID가 3인 사람에서도 마찬가지로 상당한 차이가 나고 있음을 알 수 있다. 위의 그림에서도 높은 값을 가질수록 별도표에서는 큰 체적을 가지고 있고 레이더표에서도 같은 양상을 보이고 있다. 그리고 문제 3번에서 변수간의 상관관계를 구했었는데 (x_1, x_2) , (x_5, x_3) , (x_8, x_4) 은 양의 상관관계, (x_3, x_2) , (x_7, x_6) , (x_8, x_6) 는 음의 상관관계를 가졌었다. 이를 별도표에서 나타나는 특징으로 해석을 해보겠다. x_1 의 체적이 클수록 x_2 의 체적도 대부분 큰 양상을 확인할 수 있고 반대로 x_8 의 체적이 클수록 x_6 의 체적은 대부분 작은 양상을 확인할 수 있었다.