Factor Analysis Tutorial

R 공개강좌

서울대학교 통계연구소

오늘 다룰 내용

- ‘psych‘ 패키지를 이용한 인자 분석
- ‘stats‘ 패키지를 이용한 인자 분석

Data Description

```
data_SP <- read.table("data/SalesPeople.txt",header=T)
head(data_SP)
```

```
##      SG    SP    NAS CT MRT ART MT
## 1  93.0  96.0  97.8  9  12   9 20
## 2  88.8  91.8  96.8  7  10  10 15
## 3  95.0 100.3  99.0  8  12   9 26
## 4 101.3 103.8 106.8 13  14  12 29
## 5 102.0 107.8 103.0 10  15  12 32
## 6  95.8  97.5  99.3 10  14  11 21
```

```
dim(data_SP)
```

```
## [1] 50  7
```

- SG : Index of sales growth
- SP : Index of sales profitability
- NAS : Index of new-account sales
- CT : Score on creativity test
- MRT : Score on mechanical reasoning test
- ART : Score on abstract reasoning test
- MT : Score on mathematics test

Contents

factanal 함수

```
factanal_fit_SP <-  factanal(x=data_SP,factors=3,
                             rotation="varimax", # varimax : orthogonal rotation opt
                             scores="regression")
```

- factanal 함수의 사용법은 fa 함수와 비슷하나, 최대가능도(maximum likelihood) 추정만 제공한다는 점이 다르다.

factanal 함수의 결과물

```
factanal_fit_SP$loadings
```

```
##
## Loadings:
##     Factor1 Factor2 Factor3
## SG  0.793   0.374   0.438
## SP  0.911   0.317   0.185
## NAS 0.651   0.544   0.438
## CT  0.255   0.964
## MRT 0.542   0.465   0.207
## ART 0.299           0.950
## MT  0.917   0.180   0.298
##
##                Factor1 Factor2 Factor3
## SS loadings      3.175   1.718   1.453
## Proportion Var   0.454   0.245   0.208
## Cumulative Var   0.454   0.699   0.906
```

```
round(head(factanal_fit_SP$scores),3)
```

```
##      Factor1 Factor2 Factor3
## [1,]  -0.787  -0.364  -0.492
## [2,]  -1.416  -0.738   0.205
## [3,]  -0.099  -0.800  -0.679
## [4,]  -0.460   0.579   0.829
## [5,]   0.145  -0.369   0.683
## [6,]  -0.999  -0.069   0.531
```

- 앞 분석과 마찬가지의 결과물들을 추출할 수 있다.
- 3개의 인자가 각 변수를 설명하는 정도를 나타내는 로딩(loading) 값
- 각 표본마다 갖는 인자 값을 regression 방법을 통해 계산한 결과의 일부

Contents

fa 함수

```
library(psych)


fa_fit_SP <- fa(r=data_SP,
                nfactors=3,
                fm="ml", # Maximum likelihood estimation
                max.iter=100,
                rotate="varimax", # orthogonal rotation option
                scores="regression")
```

- 'psych' 패키지의 'fa' 함수를 이용하여 7개의 변수에 대한 인자 분석을 수행해보도록 한다.

fa 함수의 결과물 I

```
fa_fit_SP$loadings
```

```
##
## Loadings:
##     ML3   ML1   ML2
## SG  0.793 0.374 0.438
## SP  0.911 0.317 0.185
## NAS 0.651 0.544 0.438
## CT  0.255 0.964
## MRT 0.542 0.465 0.207
## ART 0.299       0.950
## MT  0.917 0.180 0.298
##
##                  ML3   ML1   ML2
## SS loadings    3.175 1.718 1.453
## Proportion Var 0.454 0.245 0.208
## Cumulative Var 0.454 0.699 0.906
```

```
# loading 들의 빈칸은 loading 추정값이 0에 가까워서 빈칸으로 표기됨.
# SS loadings : the sum of squared of loadings
# Proportion var는 각 요인이 설명하는 총 분산의 비율을 말하는 것
```

[] 옵션을 추가하면 모든 loading을 확인할 수 있다.

```
fa_fit_SP$loadings[]
```

```
##             ML3       ML1         ML2
## SG  0.7934765 0.3738864 0.43821537
## SP  0.9114843 0.3170547 0.18490791
## NAS 0.6513180 0.5439311 0.43794924
## CT  0.2550448 0.9641641 0.01957359
## MRT 0.5420313 0.4654264 0.20726995
## ART 0.2991398 0.0539954 0.95006923
## MT  0.9174081 0.1796415 0.29762827
```

- 3개의 인자가 각 변수를 설명하는 정도를 나타내는 로딩(loading) 값이다.

fa 함수의 결과물 II

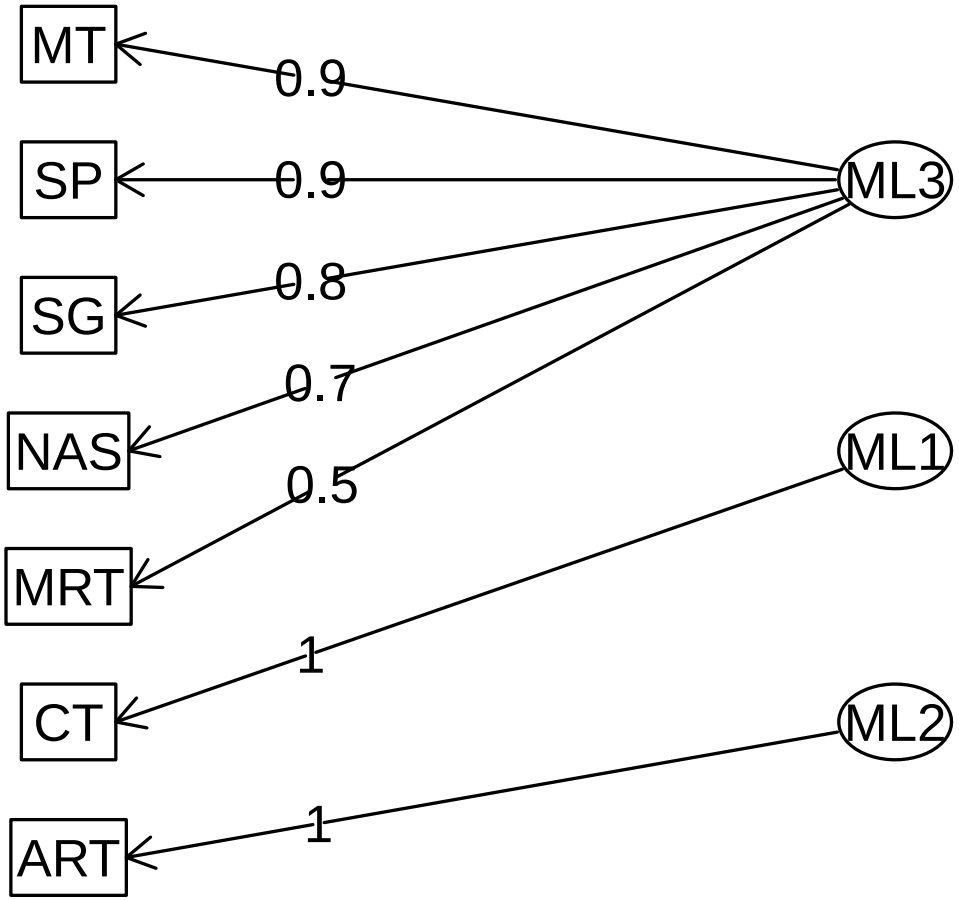
```
round(head(fa_fit_SP$scores),3) # 각 관측값들에 대한 인자별 점수
```

```
##          ML3    ML1    ML2
## [1,] -0.787 -0.364 -0.492
## [2,] -1.416 -0.738  0.205
## [3,] -0.099 -0.800 -0.679
## [4,] -0.460  0.579  0.829
## [5,]  0.145 -0.369  0.683
## [6,] -0.999 -0.069  0.531
```

- 각 표본마다 갖는 인자 값을 regression 방법을 통해 계산한 결과이다.

인자 분석의 도식화 (로딩 값)f

```
fa.diagram(fa_fit_SP,main="")
```



fa 함수 (quartimax rotation)

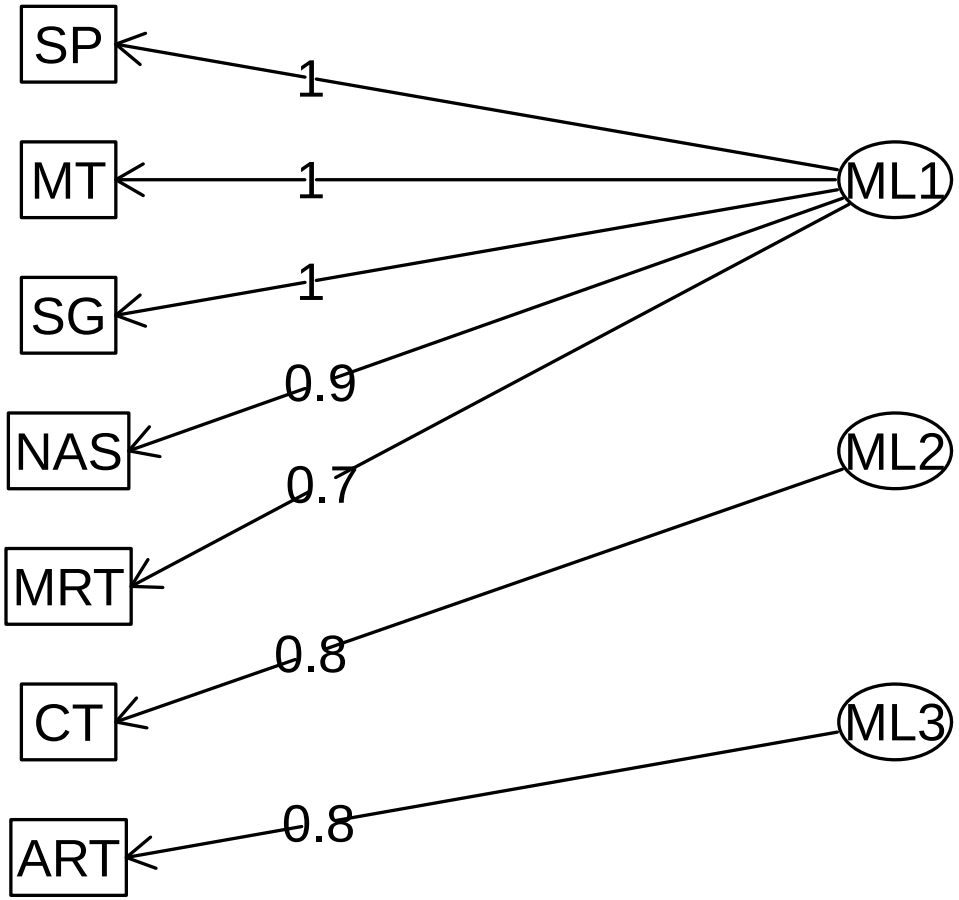
```
#install.packages("GPArotation")
library(GPArotation)

library(psych)


fa_fit_SP_q <- fa(r=data_SP,
                  nfactors=3,
                  fm="ml", # Maximum likelihood estimation
                  max.iter=100,
                  rotate="quartimax", # orthogonal rotation option
                  scores="regression")
```

인자 분석의 도식화 (로딩 값)f

```
fa.diagram(fa_fit_SP_q,main="")
```



인자분석 + 변수 군집화 (로딩 값)

```
# loading 의 추정값의 절댓값이 cutoff보다 작으면 0으로 지정.
thres <- function(x,cutoff){
  x[abs(x) <= cutoff] <- 0
  return(x)
}
set.seed(200813)
km_SP_loadings <- kmeans(thres(fa_fit_SP$loadings[],0.1),
                         centers=3,iter.max=100,nstart=25)
km_SP_loadings$cluster
```

```
##  SG  SP NAS  CT MRT ART  MT
##   1   1   1   2   1   3   1
```

- 각 변수 마다 주어진 로딩 값을 통해 군집 분석을 수행한 결과이다.

인자분석 + 표본 군집화 (스코어 값) I

```
set.seed(200813)
km_SP_scores <- kmeans(fa_fit_SP$scores,centers=3,iter.max=100,nstart=25)
km_SP_scores$cluster
```

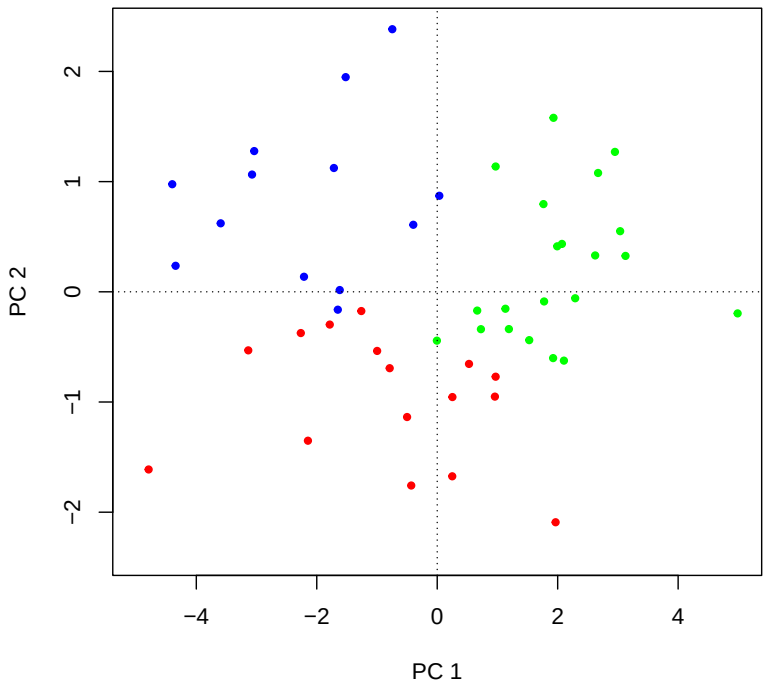
```
##  [1] 2 1 2 3 1 1 2 3 1 3 3 2 3 1 2 2 3 3 1 3 2 1 2 3 3 2 3 3 1 3 3 2 1 1 3 1 2 1
## [39] 3 3 1 2 3 1 1 3 1 2 3 3
```

- 일반적으로 인자 분석은 변수의 공통된 특성을 보고자 하지만 각 표본마다 계산된 스코어 값을 통해 군집 분석을 수행할 수도 있다.

인자분석 + 표본 군집화 (스코어 값) II

- 시각화를 위해 첫 두 개의 주성분 축을 이용한다.
- 군집마다 다른 색으로 강조하여 표본들을 구분해본다.

```
pr_SP <- prcomp(data_SP,scale=TRUE)
pc1 <- pr_SP$x[,1]
pc2 <- pr_SP$x[,2]
Mx <- max(abs(pc1))
My <- max(abs(pc2))
col_vec <- c("red","blue","green")
plot(pc1,pc2,xlab="PC 1",ylab="PC 2",pch=20,
     xlim=c(-Mx,Mx),ylim=c(-My,My),col=col_vec[km_SP_scores$cluster])
abline(h=0,lty="dotted")
abline(v=0,lty="dotted")
```



새로운 데이터의 score 값

```
newdata <- matrix(c(110,98,105,15,18,12,35),nrow=1)
colnames(newdata) <- colnames(data_SP)
score_newdata <- predict(fa_fit_SP,newdata,data_SP)
score_newdata
```

```
##              ML3       ML1       ML2
## [1,] -0.3285779 1.063189 0.7927602
```

Cereal data

- 여러 시리얼 브랜드에 대한 소비자들의 인식을 조사하기 위한 연구가 진행되었다.
- 각 응답자들은 25개의 항목에서 12개의 시리얼 중 가장 선호하는 3가지 시리얼을 선택하였다.
- 연구 목적 중 한 가지는 시리얼 브랜드의 특징을 잠재된 요인의 함수로서 분석하고자 하는 것이다.

Cereal data

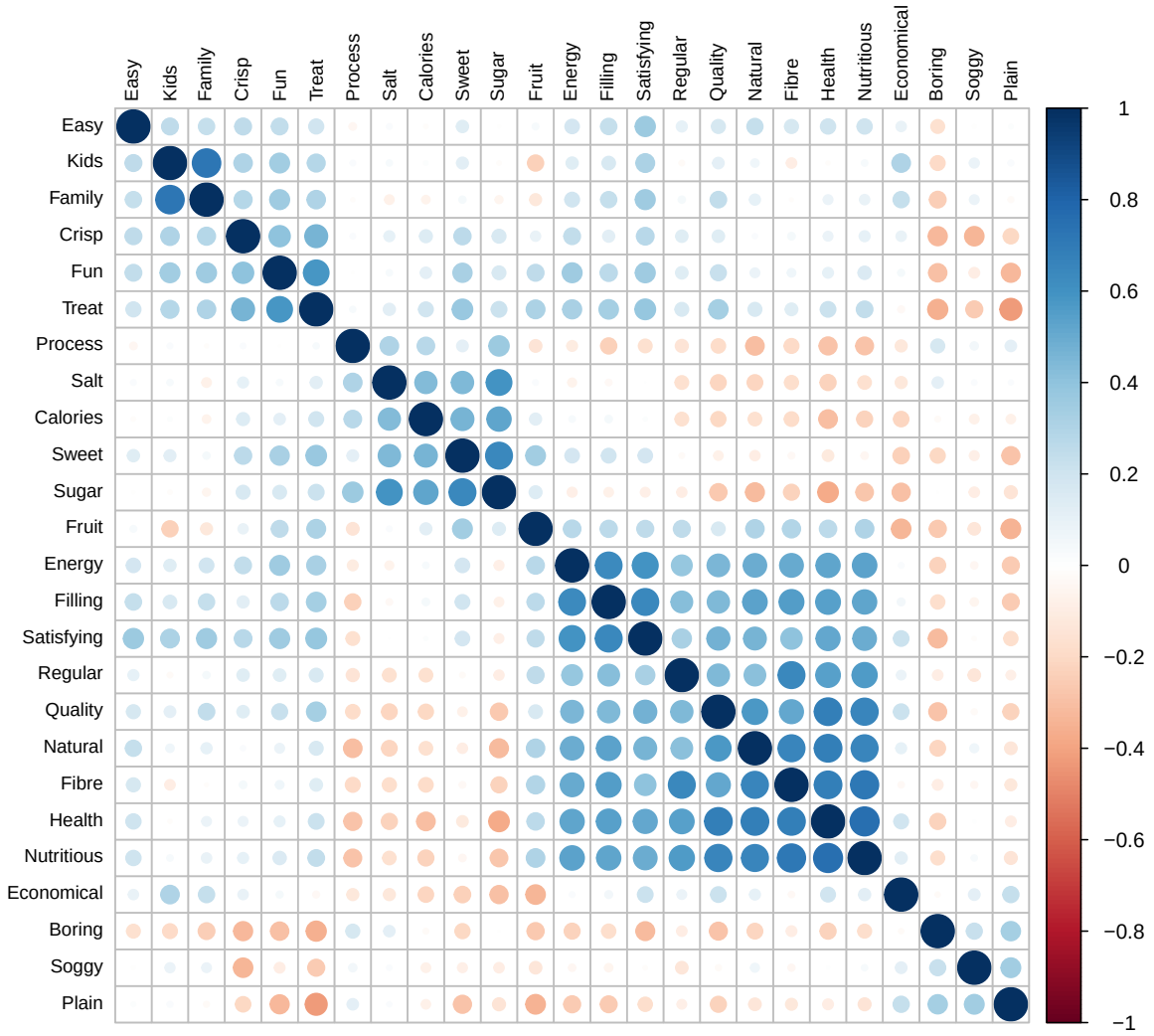
```
cereal <- read.csv("data/cereal.csv",header=T)

library(corrplot)
cortest.bartlett(cor(cereal[,-1]))
```

```
## $chisq
## [1] 1153.931
##
## $p.value
## [1] 2.437858e-100
##
## $df
## [1] 300
```

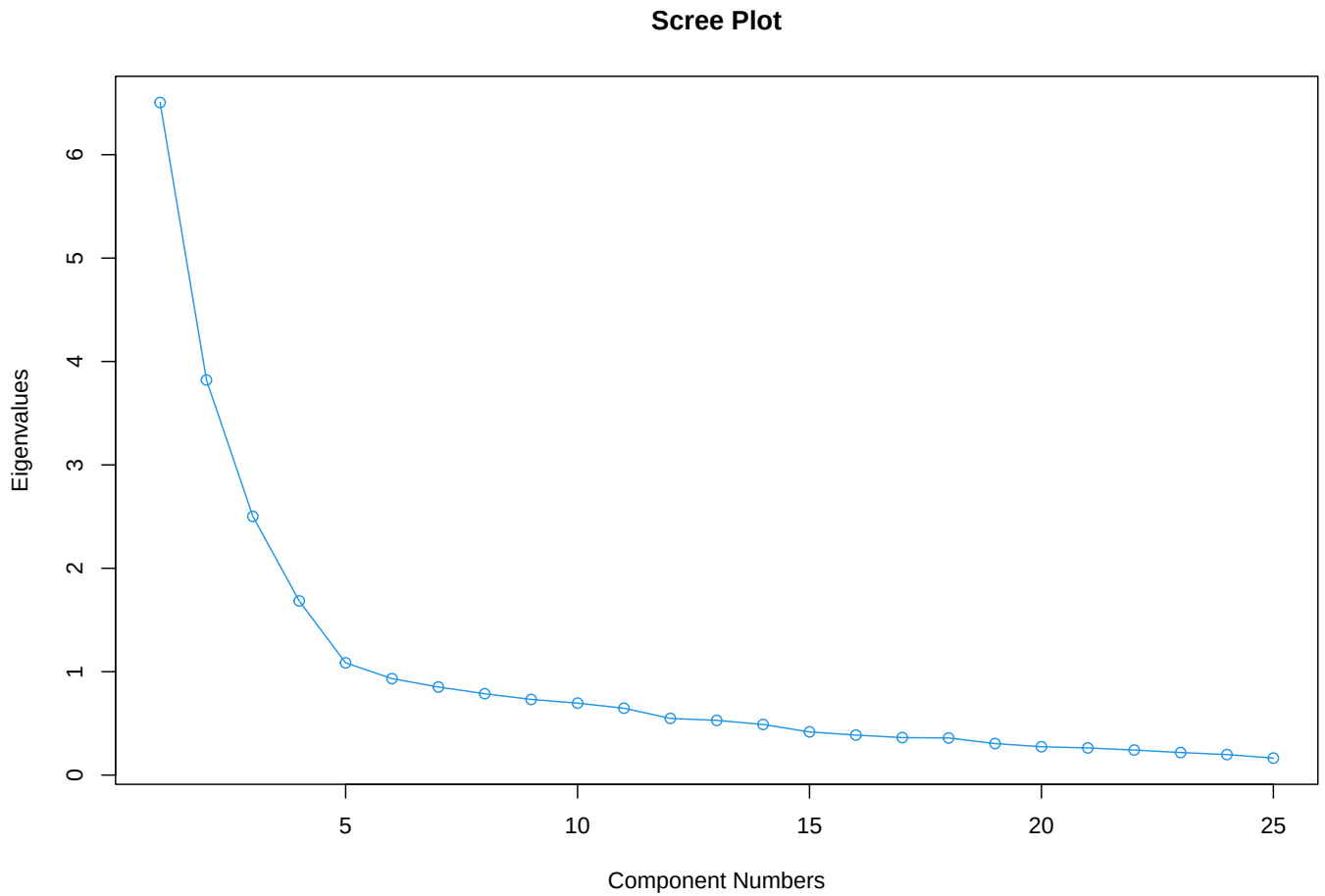
Cereal data

```
# Correlation plot
corrplot(cor(cereal[,-1]), order = "hclust", tl.col='black', tl.cex=.75)
```



Cereal data

```
## Scree plot
Eigenvalues=eigen(cor(cereal[,-1]))$values
plot(Eigenvalues,main="Scree Plot",xlab="Component Numbers",type="o",col=4)
```



Cereal data

```
fa1 <- fa(
  r = cereal[,-1], nfactors = 9, fm = "ml",
  max.iter = 100, rotate = "varimax", scores = "regression"
)
fa.diagram(fa1)
```