STAT401\_HW4

김정현

# Q1.

options(digits=3)  
library(psych)  
rmat<-matrix(  
 c(1,-0.04,0.61,0.45,0.03,-0.29,-0.3,0.45,0.3,  
 -0.04,1,-0.07,-0.12,0.49,0.43,0.3,-0.31,-0.17, 0.61,-0.07,1,0.59,0.03,-0.13,-0.24,0.59,0.32, 0.45,-0.12,0.59,1,-0.08,-0.21,-0.19,0.63,0.37, 0.03,0.49,0.03,-0.08,1,0.47,0.41,-0.14,-0.24, -0.29,0.43,-0.13,-0.21,0.47,1,0.63,-0.13,-0.15, -0.3,0.3,-0.24,-0.19,0.41,0.63,1,-0.26,-0.29, 0.45,-0.31,0.59,0.63,-0.14,-0.13,-0.26,1,0.4, 0.3,-0.17,0.32,0.37,-0.24,-0.15,-0.29,0.4,1),nrow = 9,ncol=9)

## (a)

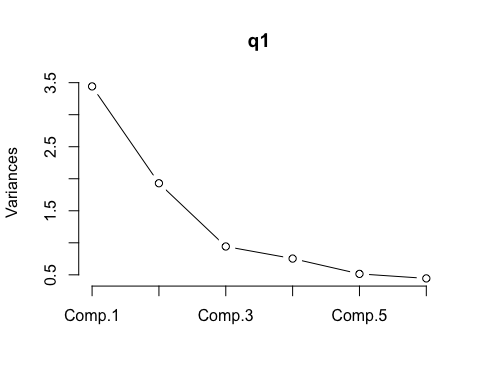
q1=princomp(covmat=rmat,cor=T)  
summary(q1)

## Importance of components:  
## Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8  
## Standard deviation 1.855 1.389 0.971 0.8682 0.7174 0.6665 0.6498 0.5710  
## Proportion of Variance 0.382 0.214 0.105 0.0838 0.0572 0.0494 0.0469 0.0362  
## Cumulative Proportion 0.382 0.597 0.702 0.7853 0.8425 0.8918 0.9388 0.9750  
## Comp.9  
## Standard deviation 0.474  
## Proportion of Variance 0.025  
## Cumulative Proportion 1.000

q1$sdev^2

## Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8 Comp.9   
## 3.442 1.930 0.942 0.754 0.515 0.444 0.422 0.326 0.225

screeplot(q1, npcs=6,type="l")



1. Choose the principal components at the total variance over 70%, then 3 PCs explain 70.2% of the total variation.
2. Choose the principal components where eigenvalues are larger than 1, then choose 2 PCs.
3. Choose the principal components based on the scree plot, then choose 3 PCs.

Therefore, we can choose 3 factors to estimate factor loadings.

## (b)

f1 = fa(rmat, nfactors=3, fm="pa", rotate="none")  
f1$communality

## [1] 0.593 0.434 0.667 0.543 0.565 0.781 0.521 0.756 0.256

## (c)

diag(f1$uniq)

## [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]  
## [1,] 0.407 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000  
## [2,] 0.000 0.566 0.000 0.000 0.000 0.000 0.000 0.000 0.000  
## [3,] 0.000 0.000 0.333 0.000 0.000 0.000 0.000 0.000 0.000  
## [4,] 0.000 0.000 0.000 0.457 0.000 0.000 0.000 0.000 0.000  
## [5,] 0.000 0.000 0.000 0.000 0.435 0.000 0.000 0.000 0.000  
## [6,] 0.000 0.000 0.000 0.000 0.000 0.219 0.000 0.000 0.000  
## [7,] 0.000 0.000 0.000 0.000 0.000 0.000 0.479 0.000 0.000  
## [8,] 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.244 0.000  
## [9,] 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.744

## (d)

f1$loadings[7,1]

## PA1   
## -0.576

Correlation between 7th statement and the first factor is -0.576.

## (e)

rmat-f1$loadings%\*%t(f1$loadings)-diag(f1$uniq)

## [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]  
## [1,] 0.000000 -0.00739 0.01826 -0.03166 0.00409 -0.00448 0.01044 0.000981  
## [2,] -0.007392 0.00000 -0.01603 0.03529 0.00625 0.03555 -0.04222 -0.041507  
## [3,] 0.018262 -0.01603 0.00000 0.01016 -0.01017 0.02068 -0.00646 -0.015184  
## [4,] -0.031659 0.03529 0.01016 0.00000 -0.00347 -0.05301 0.04567 0.014738  
## [5,] 0.004090 0.00625 -0.01017 -0.00347 0.00000 -0.02074 0.01138 0.040537  
## [6,] -0.004477 0.03555 0.02068 -0.05301 -0.02074 0.00000 0.00743 0.005962  
## [7,] 0.010440 -0.04222 -0.00646 0.04567 0.01138 0.00743 0.00000 -0.018985  
## [8,] 0.000981 -0.04151 -0.01518 0.01474 0.04054 0.00596 -0.01899 0.000000  
## [9,] 0.019803 0.04423 -0.01877 0.02269 -0.05397 0.05945 -0.05392 -0.018845  
## [,9]  
## [1,] 0.0198  
## [2,] 0.0442  
## [3,] -0.0188  
## [4,] 0.0227  
## [5,] -0.0540  
## [6,] 0.0594  
## [7,] -0.0539  
## [8,] -0.0188  
## [9,] 0.0000

We can check that our residual matrix is almost 0.

## Repeat using Varimax

f2 = fa(rmat, nfactors=3, fm="pa", rotate="varimax")  
f2$communality

## [1] 0.593 0.434 0.667 0.543 0.565 0.781 0.521 0.756 0.256

diag(f2$uniq)

## [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]  
## [1,] 0.407 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000  
## [2,] 0.000 0.566 0.000 0.000 0.000 0.000 0.000 0.000 0.000  
## [3,] 0.000 0.000 0.333 0.000 0.000 0.000 0.000 0.000 0.000  
## [4,] 0.000 0.000 0.000 0.457 0.000 0.000 0.000 0.000 0.000  
## [5,] 0.000 0.000 0.000 0.000 0.435 0.000 0.000 0.000 0.000  
## [6,] 0.000 0.000 0.000 0.000 0.000 0.219 0.000 0.000 0.000  
## [7,] 0.000 0.000 0.000 0.000 0.000 0.000 0.479 0.000 0.000  
## [8,] 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.244 0.000  
## [9,] 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.744

Note that communalities and specific variances are same with rotate = “none” method.

f2$loadings[7,1]

## PA1   
## -0.218

Correlation is -0.218, which is weaker than rotate = “none” method.

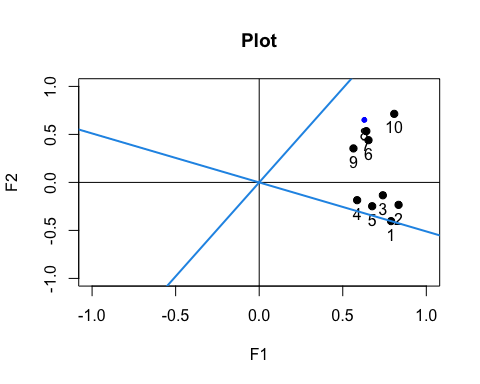
rmat-f2$loadings%\*%t(f2$loadings)-diag(f2$uniq)

## [,1] [,2] [,3] [,4] [,5] [,6] [,7]  
## [1,] -3.33e-16 -0.00739 1.83e-02 -3.17e-02 4.09e-03 -4.48e-03 1.04e-02  
## [2,] -7.39e-03 0.00000 -1.60e-02 3.53e-02 6.25e-03 3.55e-02 -4.22e-02  
## [3,] 1.83e-02 -0.01603 -4.44e-16 1.02e-02 -1.02e-02 2.07e-02 -6.46e-03  
## [4,] -3.17e-02 0.03529 1.02e-02 -5.55e-16 -3.47e-03 -5.30e-02 4.57e-02  
## [5,] 4.09e-03 0.00625 -1.02e-02 -3.47e-03 -1.11e-16 -2.07e-02 1.14e-02  
## [6,] -4.48e-03 0.03555 2.07e-02 -5.30e-02 -2.07e-02 -3.33e-16 7.43e-03  
## [7,] 1.04e-02 -0.04222 -6.46e-03 4.57e-02 1.14e-02 7.43e-03 -2.22e-16  
## [8,] 9.81e-04 -0.04151 -1.52e-02 1.47e-02 4.05e-02 5.96e-03 -1.90e-02  
## [9,] 1.98e-02 0.04423 -1.88e-02 2.27e-02 -5.40e-02 5.94e-02 -5.39e-02  
## [,8] [,9]  
## [1,] 9.81e-04 1.98e-02  
## [2,] -4.15e-02 4.42e-02  
## [3,] -1.52e-02 -1.88e-02  
## [4,] 1.47e-02 2.27e-02  
## [5,] 4.05e-02 -5.40e-02  
## [6,] 5.96e-03 5.94e-02  
## [7,] -1.90e-02 -5.39e-02  
## [8,] -5.55e-16 -1.88e-02  
## [9,] -1.88e-02 -2.22e-16

The residual matrix is close to zero compared to rotate = “none” method.

# Q2.

l1<-matrix(c(0.789,0.834,0.74,0.586,0.676,0.654,0.641,0.629,0.564,0.808,  
-0.403,-0.234,-0.134,-0.185,-0.248,0.44,0.534,0.651,0.354,0.714),ncol = 2, nrow = 10)  
  
fa.plot(l1,xlim=c(-1,1),ylim=c(-1,1))  
abline(a=0,b=l1[1,2]/l1[1,1],col=4,lwd=2)  
abline(a=0,b=-l1[1,1]/l1[1,2], col=4,lwd=2)



a1<-acos(l1[1,1]/sqrt(l1[1,1]^2+l1[1,2]^2))/(2\*pi)\*360  
cat("Angle:", a1)

## Angle: 27.1

The angle needed is 27.1.

## (b)

By this result, We can divide by (x6, x7, x8, x9, x10) and (x1, x2, x3, x4, x5) here.

# Q3.

## (a)

s.dat <- read.csv('sales.dat', header = T, sep = '')  
s.f2 <- fa(s.dat, nfactors = 2, fm = "ml", rotate = "none")  
print("Communality:")

## [1] "Communality:"

s.f2$communalities

## X1 X2 X3 X4 X5 X6 X7   
## 0.931 0.930 0.877 0.995 0.526 0.386 0.971

print("Specific Variance:")

## [1] "Specific Variance:"

diag(s.f2$uniq)

## [,1] [,2] [,3] [,4] [,5] [,6] [,7]  
## [1,] 0.0692 0.0000 0.000 0.00000 0.000 0.000 0.0000  
## [2,] 0.0000 0.0704 0.000 0.00000 0.000 0.000 0.0000  
## [3,] 0.0000 0.0000 0.123 0.00000 0.000 0.000 0.0000  
## [4,] 0.0000 0.0000 0.000 0.00499 0.000 0.000 0.0000  
## [5,] 0.0000 0.0000 0.000 0.00000 0.474 0.000 0.0000  
## [6,] 0.0000 0.0000 0.000 0.00000 0.000 0.614 0.0000  
## [7,] 0.0000 0.0000 0.000 0.00000 0.000 0.000 0.0288

print("Communality\*Communality + Specific Variance:")

## [1] "Communality\*Communality + Specific Variance:"

s.f2$loadings %\*% t(s.f2$loadings) + diag(s.f2$uniq)

## X1 X2 X3 X4 X5 X6 X7  
## X1 1.000 0.930 0.883 0.572 0.664 0.554 0.931  
## X2 0.930 1.000 0.875 0.541 0.655 0.562 0.937  
## X3 0.883 0.875 1.000 0.700 0.675 0.480 0.845  
## X4 0.572 0.541 0.700 1.000 0.592 0.150 0.413  
## X5 0.664 0.655 0.675 0.592 1.000 0.341 0.619  
## X6 0.554 0.562 0.480 0.150 0.341 1.000 0.602  
## X7 0.931 0.937 0.845 0.413 0.619 0.602 1.000

s.f3 <- fa(s.dat, nfactors = 2, fm = "ml", rotate = "none")  
print("Communality:")

## [1] "Communality:"

s.f3$communalities

## X1 X2 X3 X4 X5 X6 X7   
## 0.931 0.930 0.877 0.995 0.526 0.386 0.971

print("Specific Variance:")

## [1] "Specific Variance:"

diag(s.f3$uniq)

## [,1] [,2] [,3] [,4] [,5] [,6] [,7]  
## [1,] 0.0692 0.0000 0.000 0.00000 0.000 0.000 0.0000  
## [2,] 0.0000 0.0704 0.000 0.00000 0.000 0.000 0.0000  
## [3,] 0.0000 0.0000 0.123 0.00000 0.000 0.000 0.0000  
## [4,] 0.0000 0.0000 0.000 0.00499 0.000 0.000 0.0000  
## [5,] 0.0000 0.0000 0.000 0.00000 0.474 0.000 0.0000  
## [6,] 0.0000 0.0000 0.000 0.00000 0.000 0.614 0.0000  
## [7,] 0.0000 0.0000 0.000 0.00000 0.000 0.000 0.0288

print("Communality\*Communality + Specific Variance:")

## [1] "Communality\*Communality + Specific Variance:"

s.f3$loadings %\*% t(s.f3$loadings) + diag(s.f3$uniq)

## X1 X2 X3 X4 X5 X6 X7  
## X1 1.000 0.930 0.883 0.572 0.664 0.554 0.931  
## X2 0.930 1.000 0.875 0.541 0.655 0.562 0.937  
## X3 0.883 0.875 1.000 0.700 0.675 0.480 0.845  
## X4 0.572 0.541 0.700 1.000 0.592 0.150 0.413  
## X5 0.664 0.655 0.675 0.592 1.000 0.341 0.619  
## X6 0.554 0.562 0.480 0.150 0.341 1.000 0.602  
## X7 0.931 0.937 0.845 0.413 0.619 0.602 1.000

## (c)

s.f1 <- fa(s.dat, nfactors = 1, fm = "ml", rotate = "none")  
s.f4 <- fa(s.dat, nfactors = 4, fm = "ml", rotate = "none")  
  
chi1 <- function(model, nfactors) {  
 chi.q <- model$STATISTIC  
 prob <- model$PVAL  
 cat("n =", nfactors, ": Chi Square =", chi.q, "with prob <", prob, "\n")  
}  
  
chi1(s.f1, 1)

## n = 1 : Chi Square = 163 with prob < 2.02e-27

chi1(s.f2, 2)

## n = 2 : Chi Square = 117 with prob < 1.25e-21

chi1(s.f3, 3)

## n = 3 : Chi Square = 117 with prob < 1.25e-21

Even though we use 3-factor model, our p-value is still smaller than 0.05, which is still significant. Also, 4-factor model is not appropriate since s < 0 in this case.

chi1(s.f4, 4)

## n = 4 : Chi Square = 18 with prob < NA

We check that 4 factor model does not work. Therefore, we can pick 3 factor model.

## (d)

f.ml1 = fa(s.dat, nfactors=2, n.obs=length(s.dat), fm="ml", rotate="none")  
apply(f.ml1$scores, 2, mean)

## ML1 ML2   
## -1.02e-15 -3.40e-15

apply(f.ml1$scores, 2, var)

## ML1 ML2   
## 0.996 0.978

Mean is close to 0 and Variance is close to 1.

## (e)

f.ml1

## Factor Analysis using method = ml  
## Call: fa(r = s.dat, nfactors = 2, n.obs = length(s.dat), rotate = "none",   
## fm = "ml")  
## Standardized loadings (pattern matrix) based upon correlation matrix  
## ML1 ML2 h2 u2 com  
## X1 0.70 0.67 0.93 0.069 2.0  
## X2 0.67 0.69 0.93 0.070 2.0  
## X3 0.80 0.49 0.88 0.123 1.7  
## X4 0.98 -0.17 1.00 0.005 1.1  
## X5 0.65 0.31 0.53 0.474 1.4  
## X6 0.25 0.57 0.39 0.614 1.4  
## X7 0.56 0.81 0.97 0.029 1.8  
##   
## ML1 ML2  
## SS loadings 3.33 2.28  
## Proportion Var 0.48 0.33  
## Cumulative Var 0.48 0.80  
## Proportion Explained 0.59 0.41  
## Cumulative Proportion 0.59 1.00  
##   
## Mean item complexity = 1.6  
## Test of the hypothesis that 2 factors are sufficient.  
##   
## df null model = 21 with the objective function = 10.9 with Chi Square = 500  
## df of the model are 8 and the objective function was 2.63   
##   
## The root mean square of the residuals (RMSR) is 0.06   
## The df corrected root mean square of the residuals is 0.09   
##   
## The harmonic n.obs is 50 with the empirical chi square 6.79 with prob < 0.56   
## The total n.obs was 50 with Likelihood Chi Square = 117 with prob < 1.3e-21   
##   
## Tucker Lewis Index of factoring reliability = 0.382  
## RMSEA index = 0.522 and the 90 % confidence intervals are 0.446 0.614  
## BIC = 85.9  
## Fit based upon off diagonal values = 0.99  
## Measures of factor score adequacy   
## ML1 ML2  
## Correlation of (regression) scores with factors 1.00 0.99  
## Multiple R square of scores with factors 1.00 0.98  
## Minimum correlation of possible factor scores 0.99 0.96

x1, x3, x4, x5 is allocated in factor 1 and x2, x6, x7 is allocated in factor2.

simpsum <- cbind(round(as.matrix(s.dat[,c(1,3,4,5)]),1) %\*% c(1,1,1,1),  
 round(as.matrix(s.dat[,c(2,6,7)]),1) %\*% c(1,1,1))  
simpsum

## [,1] [,2]  
## [1,] 212 125  
## [2,] 203 117  
## [3,] 214 135  
## [4,] 235 145  
## [5,] 230 152  
## [6,] 219 130  
## [7,] 216 134  
## [8,] 264 188  
## [9,] 234 152  
## [10,] 241 170  
## [11,] 236 154  
## [12,] 228 151  
## [13,] 244 158  
## [14,] 220 150  
## [15,] 226 149  
## [16,] 192 114  
## [17,] 227 148  
## [18,] 232 157  
## [19,] 217 147  
## [20,] 240 143  
## [21,] 206 117  
## [22,] 220 168  
## [23,] 201 116  
## [24,] 238 163  
## [25,] 251 172  
## [26,] 216 132  
## [27,] 244 169  
## [28,] 238 180  
## [29,] 210 121  
## [30,] 237 176  
## [31,] 250 172  
## [32,] 209 111  
## [33,] 218 141  
## [34,] 215 124  
## [35,] 252 174  
## [36,] 234 176  
## [37,] 230 126  
## [38,] 230 142  
## [39,] 251 175  
## [40,] 248 168  
## [41,] 232 154  
## [42,] 220 132  
## [43,] 247 156  
## [44,] 186 111  
## [45,] 218 137  
## [46,] 245 161  
## [47,] 209 121  
## [48,] 195 107  
## [49,] 237 158  
## [50,] 239 168

One can check that factor scores in (d) and (e) are different.