HUDM6122 Homework_05

Chenguang Pan

2023-03-20

0.1 Github Address

All my latest homework can be found on Github: https://github.com/cgpan/hudm6122_homeworks . Thanks for checking if interested.

0.2 Ex 5.1

Show how the result rises from the assumptions of uncorrelated factors, independence of the specific variates, and independence of common factors and specific variances. What form does take if the factors are allowed to be correlated?

MY SOLUTION:
Based on the assumption of Exploratory Factor Analysis(EFA), a set of observed variables $\mathbf{x}$ assumed to be linked to a set of latent variables $\mathbf{f}$. Therefore, we can have a regression model in matrix form

$$\boldsymbol{x} = \boldsymbol{\Lambda f} + \boldsymbol{u}$$

, where $\boldsymbol{\Lambda}$ is a $q \times k$ matrix of factor loadings (a.k.a., the coefficients of the regression model), and the u is the vector of unexplained error of each observed variables.

Let's take the variance of the formula above

$$V(\boldsymbol{x}) = V(\boldsymbol{\Lambda f} + \boldsymbol{u})$$

. Based on the operation rule of variance, like

$$V(a + b) = V(a) + V(b) + 2Cov(ab)$$

, we combined the two formulas above, then

$$V(\boldsymbol{x}) = V(\boldsymbol{\Lambda f} + \boldsymbol{u}) = V(\boldsymbol{\Lambda f}) + V(\boldsymbol{u}) + 2Cov(\boldsymbol{\Lambda f u})$$

. Since the we assumed that the error terms are uncorrelated with the factors, therefore the $Cov(\boldsymbol{\Lambda f u}) = 0$. Then, we can continue to drive the variance formula as

$$V(\boldsymbol{x}) = V(\boldsymbol{\Lambda f}) + V(\boldsymbol{u}) = \boldsymbol{\Lambda} V(\boldsymbol{f}) \boldsymbol{\Lambda}^T + \Psi$$

. In addition, we assumed that the factors are uncorrelated with each other. The $V(\boldsymbol{f})$ is actually an identity matrix. Therefore, the formula can be written as

$$V(\boldsymbol{x}) = \boldsymbol{\Lambda} V(\boldsymbol{f}) \boldsymbol{\Lambda}^T + \Psi = \boldsymbol{\Lambda \Lambda}^T + \Psi$$

. Finally, the formula can be written as

$$\boldsymbol{\Sigma} = \boldsymbol{\Lambda \Lambda}^T + \Psi$$

.

If we allow the factors to be correlated with each other, then the $V(\boldsymbol{f})$ is not an identity matrix. Let's use the greek letter Φ to represent the variance matrix of loadings **f**. Thus, the formula should be

$$\boldsymbol{\Sigma} = \boldsymbol{\Lambda}\boldsymbol{\Phi}\boldsymbol{\Lambda}^T + \Psi$$

.

0.3 Ex 5.2

Show that the communalities in a factor analysis model are unaffected by the transformation ...

MY SOLUTION:
This question mentioned that we need to use the transformed factor loadings $\boldsymbol{\Lambda}^* = \boldsymbol{\Lambda}\boldsymbol{M}$. Let's assume that $\boldsymbol{M}$ is an $k \times k$ orthogonal matrix. We can re-write the the basic regression equation linking the observed and the factors as:

$$\boldsymbol{x} = (\boldsymbol{\Lambda}\boldsymbol{M})(\boldsymbol{M}^T\boldsymbol{f}) + \boldsymbol{u}$$

.
Using the rule of variance, we can have

$$\boldsymbol{\Sigma} = (\boldsymbol{\Lambda}\boldsymbol{M})(\boldsymbol{\Lambda}\boldsymbol{M})^T + \Psi$$

. Since the $\boldsymbol{M}$ is a orthogonal matrix and $\boldsymbol{M}\boldsymbol{M}^T = \boldsymbol{I}$. Therefore, the variance equation can be written as

$$\boldsymbol{\Sigma} = \boldsymbol{\Lambda}\boldsymbol{\Lambda}^T + \Psi$$

. That is, the transformed factor loadings $\boldsymbol{\Lambda}^* = \boldsymbol{\Lambda}\boldsymbol{M}$ will not influence the communalities (i.e., $\boldsymbol{\Lambda}\boldsymbol{\Lambda}^T$) in the a factor analysis model.

0.4 Ex 5.3

Give a formula for the proportion of variance explained by the jth factor estimated by the principal factor approach.

MY SOLUTION:
The proportion of variance explained by the jth factor represents the proportion of the total variance in the observed variables that is accounted for by that factor alone. Therefore, the formula could be

$$Proportion_j = \frac{\sum_{i=1}^{q} \lambda_{ij}^2}{\boldsymbol{\Lambda}\boldsymbol{\Lambda}^T}$$

.

0.5 Ex 5.4

Apply the factor analysis model separately to the life expectancies of men and women and compare the results.

MY SOLUTION:

For this question, I present two methods to run factor analysis. The first one is similar to the method introduced in textbook. The second one is a more rigorous method including the initial dataset checking, scree plot analysis, and parallel analysis.

0.5.1 Method 1: Using the similar method introduced by textbook

The textbook does not provide the original dataset. Based on the code in the `MVA`, I create the dataset via a separated r file named "HW05 Test". This file created the `life.rdata` and `life.csv` dataset in the same file folder.

```
> load("life.rdata")
> head(life)
           m0 m25 m50 m75 w0 w25 w50 w75
Algeria    63  51  30  13 67  54  34  15
Cameroon   34  29  13   5 38  32  17   6
Madagascar 38  30  17   7 38  34  20   7
Mauritius  59  42  20   6 64  46  25   8
Reunion    56  38  18   7 62  46  25  10
Seychelles 62  44  24   7 69  50  28  14
>
> # subset the male and female dataset
> life_male <- life[,1:4]
> life_female <- life[,5:8]
>
> # test the number of factors needed for the male and female dataset separately
> sapply(1, function(f)
+   factanal(life_male, factors=f, method="mle")$PVAL)
   objective
0.0007284301
> sapply(1, function(f)
+   factanal(life_female, factors=f, method="mle")$PVAL)
   objective
4.738464e-12
```

When test the number of the factors from 1 to larger number, there is always a warning that `N factors are too many for N variables`. More details can be found on Page 143 of the textbook or here https://stats.stackexchange.com/questions/593452/efa-n-factors-are-too-many-for-n-variables

The results suggest that an one-factor solution might be adequate to account for the observed covariances in the data.

Next, I run the one-factor solution for both male and female datasets.

```
> factanal(life_male, factors = 1, method="mle")

Call:
factanal(x = life_male, factors = 1, method = "mle")

Uniquenesses:
   m0   m25   m50   m75
0.594 0.552 0.005 0.434

Loadings:
    Factor1
m0  0.638
m25 0.669
m50 0.998
m75 0.752
```

```
               Factor1
SS loadings      2.415
Proportion Var   0.604

Test of the hypothesis that 1 factor is sufficient.
The chi square statistic is 14.45 on 2 degrees of freedom.
The p-value is 0.000728
> factanal(life_female, factors = 1, method="mle")

Call:
factanal(x = life_female, factors = 1, method = "mle")

Uniquenesses:
   w0   w25   w50   w75
0.220 0.005 0.115 0.526

Loadings:
    Factor1
w0  0.883
w25 0.998
w50 0.941
w75 0.689

               Factor1
SS loadings      3.134
Proportion Var   0.784

Test of the hypothesis that 1 factor is sufficient.
The chi square statistic is 52.15 on 2 degrees of freedom.
The p-value is 4.74e-12
```

The result shows that for the one-factor solution in the male dataset, it captures the most variance of age 50 or older. But comparing to the EFA on complete dataset, this factor does not have a very clear indication for what properties it covers. The factor analysis on female dataset has the similar situation.

Is one-factor really appropriate here? Since the P-value is still significant. I plan to try another method to search for the right number of factors.

0.5.2 Method 2: Solve this question in another way

Before running EFA, I run several tests to ensure that this dataset is good for factor analysis.

```
> # install.packages("psych")
> library(psych)
> # get the correlation matrix
> life_male_cor <- cor(life_male)
> life_female_cor <- cor(life_female)
>
> # The Kaiser-Meyer-Olkin (KMO) used to measure sampling adequacy
> # is a better measure of factorability.
> KMO(life_male_cor)
Kaiser-Meyer-Olkin factor adequacy
Call: KMO(r = life_male_cor)
```

```
Overall MSA =  0.66
MSA for each item =
  m0  m25  m50  m75
0.66 0.77 0.64 0.55
> KMO(life_female_cor)
Kaiser-Meyer-Olkin factor adequacy
Call: KMO(r = life_female_cor)
Overall MSA =  0.63
MSA for each item =
  w0  w25  w50  w75
0.54 0.59 0.64 0.82
```

According to Kaiser's (1974) guidelines, a suggested cutoff for determining the factorability of the sample data is KMO >= 60. The total KMOs are 0.66 and 0.63, indicating that, based on this test, we can probably conduct a factor analysis.

Next, Bartlett's Test of Sphericity compares an observed correlation matrix to the identity matrix. Essentially it checks to see if there is a certain redundancy between the variables that we can summarize with a few number of factors. The null hypothesis of the test is that the variables are orthogonal, i.e. not correlated.

```
> # run Bartlett's Test of Sphericity
> cortest.bartlett(life_male_cor)$p.value
[1] 8.437942e-49
> cortest.bartlett(life_female_cor)$p.value
[1] 9.75451e-127
```

Small p values (< 0.05) of the significance level indicate that a factor analysis may be useful with our data.

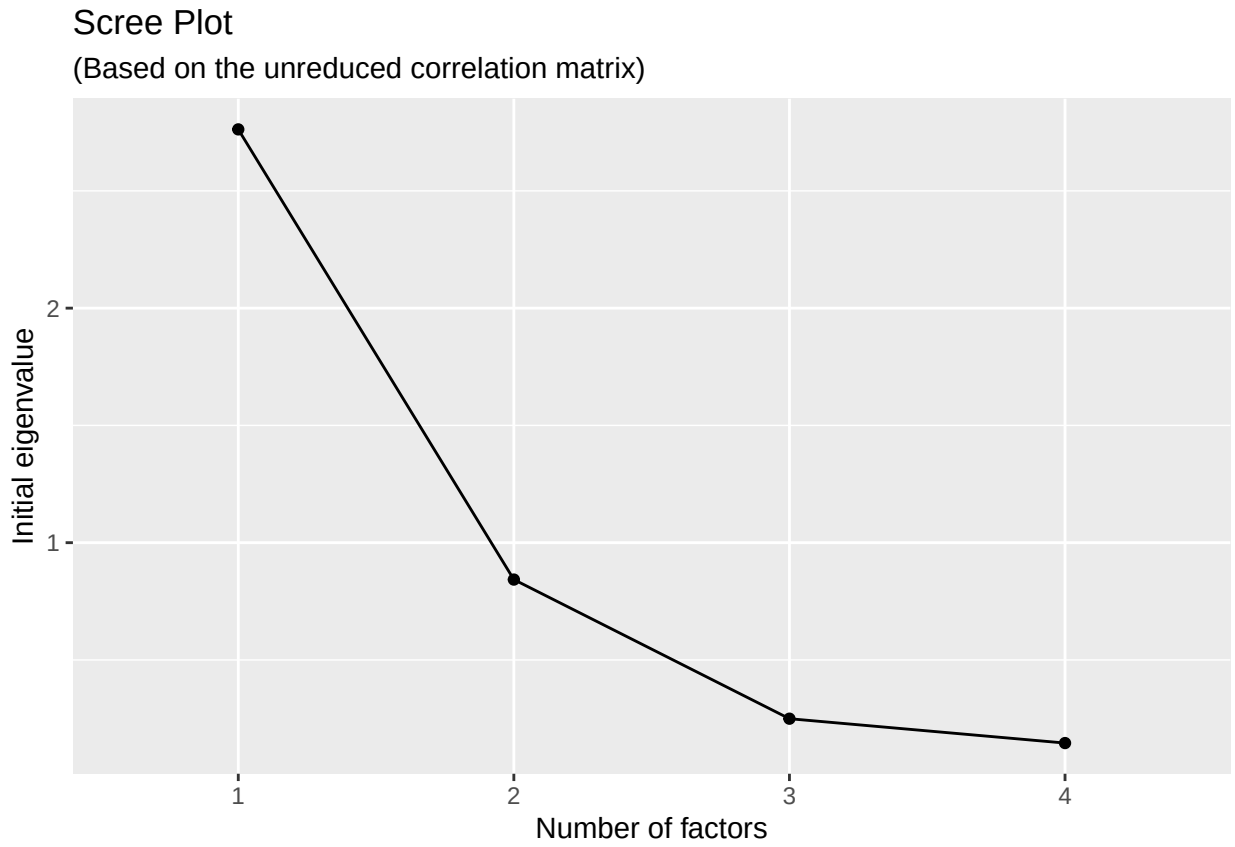
```
> # get the determinants for both correlation matrix
> det(life_male_cor)
[1] 0.08459114
> det(life_female_cor)
[1] 0.002000542
```

Finally, we have positive determinants, which means the factor analysis will probably run.

Here, I begin to run EFA by using `fa()` function and make a scree plot to determine the number of factors.

```
> library(ggplot2)
> # run factor analysis using fa() function
> male_fa <- fa(life_male,
+               nfactors = ncol(life_male_cor),
+               rotate = "varimax")
> efa_model <- fa(life_male, nfactors = 2, rotate = "varimax")
> # to get the number of factors
> n_factors <- length(male_fa$e.values)
>
> # to store the data
> scree <- data.frame(Factor_n = as.factor(1:n_factors),
+                     Eigenvalue = male_fa$e.values)
> # draw scree plot using ggplot2
> ggplot(scree, aes(x = Factor_n, y = Eigenvalue, group = 1)) +
+       geom_point() + geom_line() +
+       xlab("Number of factors") +
```

```
+       ylab("Initial eigenvalue") +
+       labs( title = "Scree Plot",
+             subtitle = "(Based on the unreduced correlation matrix)")
```



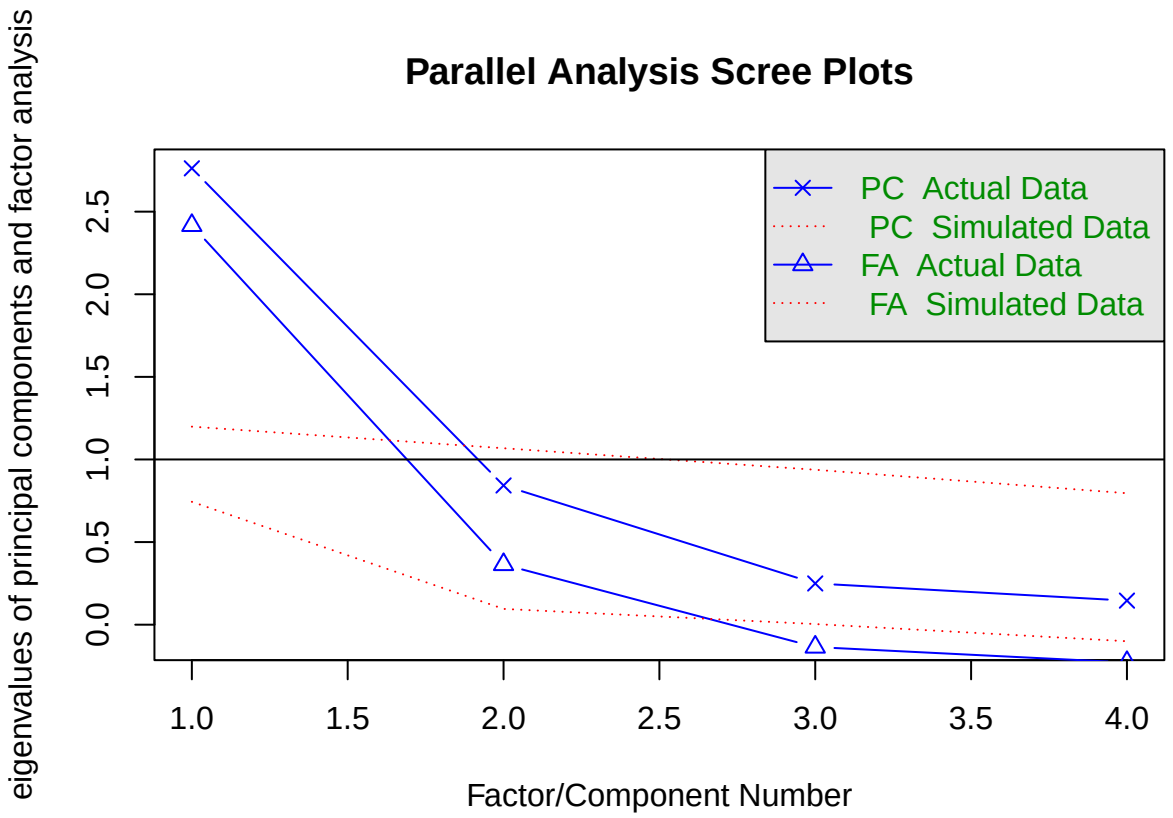
From the scree plot, 1 factors maybe appropriate for the male dataset. Since only one factor's eigenvalue is greater than 1. However, the second factor is above .7. It may be appropriate too.

Tips Why I set eigenvalue = 1 as a cutoff?
Here, the eigenvalue is a measure of the amount of variance in the observed variables that is accounted for by each factor. If the eigenvalue of a factor is less than 1, it indicates that the factor explains less variance than one of the original variables and, therefore, does not contribute significantly to the explanation of the common variance among the variables.

Parallel analysis is a method for determining the number of components or factors to retain from pca or factor analysis.I also use perform parallel analysis to determine the factors. notice the results in the console will provide the suggestion.

From the returned result under the scree plot, it suggests that 2-factor solution may be good for the male dataset.

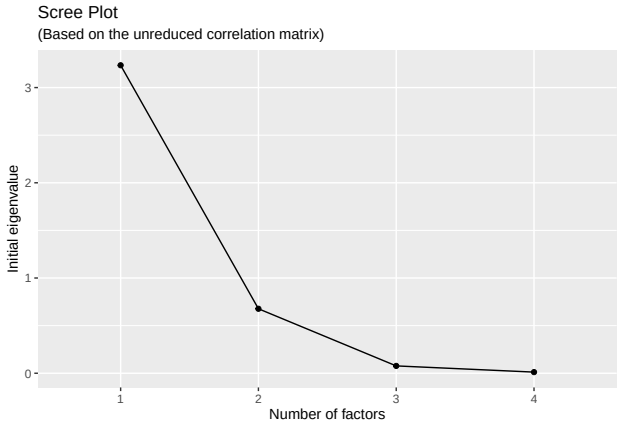
```
> parallel <- fa.parallel(life_male_cor)
```

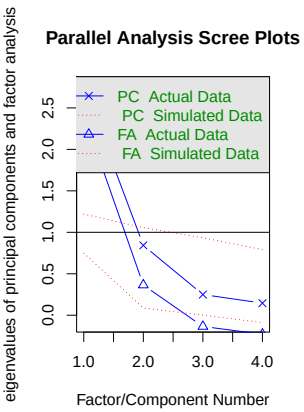



```
Parallel analysis suggests that the number of factors =  2  and the number of components =  1
```

Using the same method on the female dataset.

```
> # run factor analysis using fa() function
> female_fa <- fa(life_female,
+                 nfactors = ncol(life_female_cor),
+                 rotate = "varimax")
> # to get the number of factors
> n_factors <- length(female_fa$e.values)
>
> # to store the data
> scree <- data.frame(Factor_n = as.factor(1:n_factors),
+                     Eigenvalue = female_fa$e.values)
> par(mfrow=c(1, 2))
> # draw scree plot using ggplot2
> ggplot(scree, aes(x = Factor_n, y = Eigenvalue, group = 1)) +
+       geom_point() + geom_line() +
+       xlab("Number of factors") +
+       ylab("Initial eigenvalue") +
+       labs( title = "Scree Plot",
+             subtitle = "(Based on the unreduced correlation matrix)")
>
> parallel <- fa.parallel(life_male_cor)
Parallel analysis suggests that the number of factors =  2  and the number of components =  1
```





The results also suggest that 2-factor solution is good for female dataset.

Finally, I checked the factor loadings of both male and female datasets.

```
> male_fa$loadings

Loadings:
    MR1    MR2    MR3    MR4
m0   0.869  0.189  0.104
m25  0.803  0.310
m50  0.540  0.792  0.152
m75  0.153  0.853

                 MR1   MR2   MR3   MR4
SS loadings    1.716 1.488 0.043 0.000
Proportion Var 0.429 0.372 0.011 0.000
Cumulative Var 0.429 0.801 0.812 0.812
> female_fa$loadings

Loadings:
    MR1    MR2    MR3    MR4
w0   0.958  0.210
w25  0.793  0.605
w50  0.560  0.818
w75  0.187  0.887

                 MR1   MR2   MR3   MR4
SS loadings    1.896 1.866 0.009 0.000
Proportion Var 0.474 0.466 0.002 0.000
Cumulative Var 0.474 0.940 0.942 0.942
```

This two-factor solution looks more reasonable than the one-factor. Since in both dataset, the first factor captures the 0-25 age's life expectations and the second factor covers the 50-75 age's life expectations. We can call the first factor "Life force under middle age" and the second factor "life force of or above middle age".

0.6 Ex 5.5

The correlation matrix given below arises from the scores of 220 boys in six school subjects: (1) French, (2) English, (3) History, (4) Arithmetic, (5) Algebra, and (6) Geometry. Find the two-factor solution from a

maximum likelihood factor analysis. By plotting the derived loadings, find an orthogonal rotation that allows easier interpretation of the results.

MY SOLUTION:

```
> # import the data
> library(Matrix)
> # import the correlation matrix
> corr_lower <- matrix(c(1, 0, 0, 0, 0, 0,
+                        0.44, 1, 0,0,0,0,
+                        0.41, 0.35, 1,0,0,0,
+                        0.29, 0.35, 0.16, 1,0,0,
+                        0.33, 0.32, 0.19, 0.59, 1,0,
+                        0.25, 0.33, 0.18, 0.47, 0.46, 1),6,6, byrow = T)
> # generate a complete correlation matrix
> corr_symmetric <- forceSymmetric(corr_lower, uplo="L")
> class(corr_symmetric)
[1] "dsyMatrix"
attr(,"package")
[1] "Matrix"
```

Although the question has told that we can use two-factor solution to conduct factor analysis, I still checked the right number of factors

```
> # test for the right number of factors
> sapply(1:2, function(f)
+   factanal(covmat=as.matrix(corr_symmetric), factors = f,
+            method = "mle", n.obs = 220)$PVAL)
   objective    objective
5.369962e-08 7.026721e-01
```

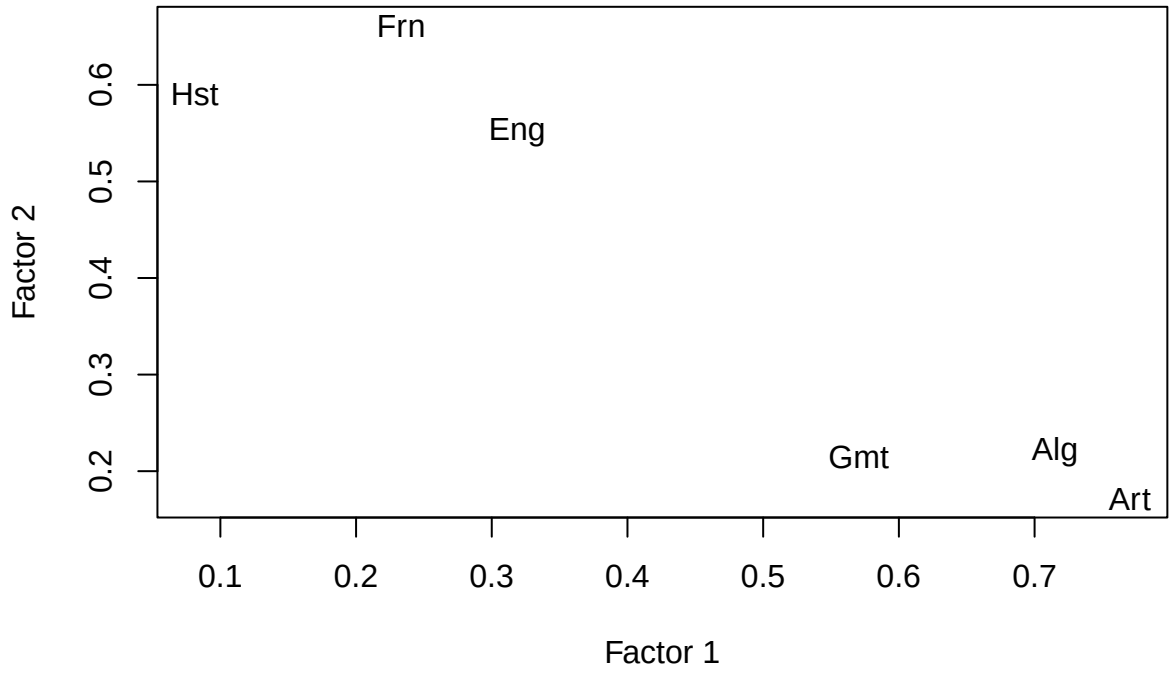
The result shows that the two-factor solution is adequate here. The result from the two-factor varimax solution are obtained from

```
> fa_ <- factanal(covmat = as.matrix(corr_symmetric), factors = 2,
+           method="mle", n.obs = 220)
> fa_$loadings

Loadings:
     Factor1 Factor2
[1,] 0.233   0.661
[2,] 0.319   0.551
[3,]         0.591
[4,] 0.770   0.172
[5,] 0.715   0.220
[6,] 0.570   0.215

               Factor1 Factor2
SS loadings      1.593   1.215
Proportion Var   0.265   0.202
Cumulative Var   0.265   0.468
>
> # plot the derived loadings
> loadings <- fa_$loadings[,1:2]
```

```
> plot(loadings[,1], loadings[,2],
+      type="n",xlab="Factor 1",ylab="Factor 2")
> text(loadings[,1], loadings[,2],
+      abbreviate(c("French","English","History",
+                   "Arithmetic","Algebra","Geometry"),3),cex=1)
```



The R built-in function `factanal()` uses the *Varimax* orthogonal rotation by default. The graph clearly shows that the first factor captures the variance of math-related abilities and the second one covers the most of the variance of social-science related abilities. Therefore, we can call the first factor "Quantitative Ability" and the second "Verbal Ability". The result is intuitively reasonable.

0.7 Ex 5.6

The matrix below shows the correlations between ratings on nine statements about pain made by 123 people suffering from extreme pain. Each statement was scored on a scale from 1 to 6, ranging from agreement to disagreement. The nine pain statements were as follows:

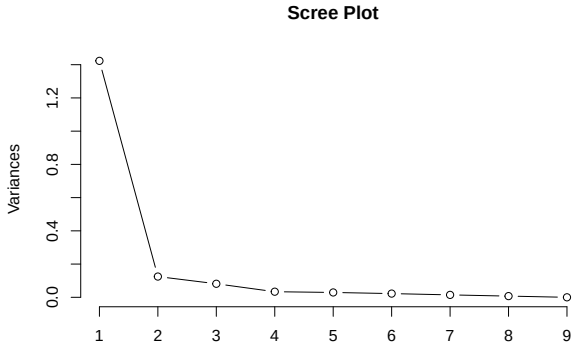
MY SOLUTION:
First, to change the lower triangular matrix into the complete correlation matrix.

```
> library(Matrix)
> # import the correlation matrix
> corr_lower <- matrix(c(1, 0, 0, 0, 0, 0, 0, 0, 0,
+                        -0.04, 1, 0,0,0,0,0,0,0,
+                        0.61, -0.07, 1,0,0,0,0,0,0,
+                        0.45, -0.12, 0.59, 1,0,0,0,0,0,
```

```
+                        0.03, 0.49, 0.03, -0.08, 1,0,0,0,0,
+                        -0.29, 0.43, -0.13, -0.21, 0.47, 1,0,0,0,
+                        -0.30, 0.30, -0.24, -0.19, 0.41, 0.63,1,0,0,
+                        0.45, -0.31,0.59,0.63,-0.14,-0.13,-0.26,1,0,
+                        0.30,-0.17,0.32,0.37,-0.24,-0.15,-0.29,0.40,1),9,9, byrow = T)
> # generate a complete correlation matrix
> corr_symmetric <- forceSymmetric(corr_lower, uplo="L")
```

The correlation matrix looks good. Next, I run the PCA first.

```
> # run the PCA first
> # use prcomp to calculate the principal components
> pca <- prcomp(corr_symmetric, scale. = FALSE)
> # get the PCA results
> summary(pca)
Importance of components:
                          PC1     PC2     PC3     PC4     PC5    PC6     PC7
Standard deviation     1.1929 0.35339 0.28547 0.18364 0.17072 0.1496 0.12066
Proportion of Variance 0.8195 0.07192 0.04693 0.01942 0.01678 0.0129 0.00838
Cumulative Proportion  0.8195 0.89142 0.93835 0.95777 0.97456 0.9875 0.99584
                           PC8       PC9
Standard deviation     0.08504 3.367e-17
Proportion of Variance 0.00416 0.000e+00
Cumulative Proportion  1.00000 1.000e+00
> # draw the scree plot
> plot(pca, type = "l",
+      main = "Scree Plot")
```



The scree-plot shows that 3 principle components may be adequate. However, based on the results from the PCA analysis, the first two components can explain 88.63% variance of the total. Therefore, I choose the first two to represent the data.

Next, I run maximum likelihood factor analysis.

```
> # explore the number of factors
> sapply(1:5, function(f)
+   factanal(covmat=as.matrix(corr_symmetric),
+            factors=f, method="mle", n.obs = 123)$PVAL)
   objective    objective    objective    objective    objective
3.582705e-23 3.330276e-06 8.377428e-02 1.370264e-01 3.166230e-01
```

The result shows that three-factor solution might be adequate to account for the observed covariances in the data.