ST 437/537: Applied Multivariate and Longitudinal Data Analysis

Exploratory Factor Analysis

Arnab Maity
NCSU Department of Statistics
SAS Hall 5240 919-515-1937 amaity[at]ncsu.edu

Single factor model

Suppose we observe the variables (manifest variables) $X_1, \ldots, X_p$ for each individual. Since the covariances of the manifest variables are central to factor analysis, **we can assume that the manifest variables all have zero mean**.

The single factor model is as follows.

$$\begin{aligned} X_1 &= \lambda_1 F + u_1, \\ X_2 &= \lambda_2 F + u_2, \\ \vdots &\qquad \vdots \\ X_p &= \lambda_p F + u_p, \end{aligned}$$

The main components of this models are:

- F is the latent (unobservable) variable, called the **common factor**. The common factor is shared among the observed variables.
- X_i's are observed variables (through the sample), called the **manifest variables**.
- u_i's are *measurement error* for X_i's, called the **specific variates**. These are unique to each variable.
- λ_i's are called **factor loadings**. The loadings determine the strength of the relationship betwwen the common factor and the observed variables.

While the single factor model looks like a regression model, the difficulty is that all the quantities on the right-hand side (F, λ_i, and u_i) are unknown. To estimate the loadings and interpret the factor propoerly, we use the following assumptions.

Model assumptions:

- The common factor has zero mean and unit variance:

$$E(F) = 0 \text{ and } var(F) = 1.$$

- The common factor and the specific factors are uncorrrelated:

$$cov(F, u_i) = 0 \text{ for all } i = 1 \ldots, p.$$

- The specific factors have mean zero but unknown variances:

$$E(u_i) = 0 \text{ and } var(u_i) = \psi_i \text{ for all } j.$$

- The specific factors are uncorrelated:

$$cov(u_j, u_k) = 0 \text{ when } j \neq k.$$

Thus the X's are related to each other only through the common factor, F. Conditional on F the variables are uncorrelated to one another (this property is sometimes referred to as *conditional linear independence*).

The single factor model imposes a specific variance-covariance structure among the observed variables, X's. Specifically,

$$var(X_i) = \underbrace{\lambda_i^2}_{\text{Communality}} + \underbrace{\psi_i}_{\text{Specific variance}},$$

Formally, we define *communality of* X_i as the variance of X_i that is captured by the common factor. The remaining variance is called the *specific variance of* X_i.

In addition, we can also see that, for $i \neq j$,

$$cov(X_i, X_j) = \lambda_i \lambda_j.$$

Thus the covariance among the variables are *completely* determined by the common factor; they do not involve the specific factors in any way.

Combining the results presented above, we can write

$$cov(\boldsymbol{X}) = \boldsymbol{\Sigma} = \underbrace{\begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_p \end{bmatrix} \begin{bmatrix} \lambda_1, & \cdots, & \lambda_p \end{bmatrix}}_{\Lambda\Lambda^T} + \underbrace{\begin{bmatrix} \psi_1 & & & \\ & \psi_2 & & \\ & & \ddots & \\ & & & \psi_p \end{bmatrix}}_{\Psi},$$

where $\Lambda = (\lambda_1, \ldots, \lambda_p)^T$, and $\Psi = diag(\psi_1, \ldots, \psi_p)$.

As a simple (but artificial) example, consider the covariance matrix below.

$$\underbrace{\begin{bmatrix} 1 & 0.42 & 0.48 \\ 0.42 & 1 & 0.56 \\ 0.48 & 0.56 & 1 \end{bmatrix}}_{\boldsymbol{\Sigma}} = \underbrace{\begin{bmatrix} 0.6 \\ 0.7 \\ 0.8 \end{bmatrix}}_{\Lambda} \begin{bmatrix} 0.6 & 0.7 & 0.8 \end{bmatrix} + \underbrace{\begin{bmatrix} 0.64 & 0 & 0 \\ 0 & 0.51 & 0 \\ 0 & 0 & 0.36 \end{bmatrix}}_{\Psi}$$

In other words, the matrix $\boldsymbol{\Sigma} - \boldsymbol{\Psi}$ has only rank 1 (since only one factor is needed to describe the covariance structure).

In real data, we will not be able to exactly factor out a covariance matrix like above; however, the single factor model will attempt to find the best rank one approximation. Specifically, given a dataset, we will replace $\boldsymbol{\Sigma}$ by the sample covariance matrix $\boldsymbol{S}$, and try to find the best approximation $\boldsymbol{S} \approx \hat{\Lambda}\hat{\Lambda}^T + \hat{\Psi}$.

The k-factor model

We can generalize the single factor model to a multiple factor model as

$$\begin{aligned} X_1 &= \lambda_{11}f_1 + \lambda_{12}f_2 + \ldots + \lambda_{1k}f_k + u_1 \\ X_2 &= \lambda_{21}f_1 + \lambda_{22}f_2 + \ldots + \lambda_{2k}f_k + u_2 \\ \vdots & \qquad\qquad\qquad \vdots \\ X_p &= \lambda_{p1}f_1 + \lambda_{p2}f_2 + \ldots + \lambda_{pk}f_k + u_p, \end{aligned}$$

where

- We have k common factors: $f_1, \ldots, f_k$.
- The loadings λ_{ij} quantifies the streng of linear relationship between the j-th factor and the i-th manifest variable.
- The random erros $u_1, \ldots, u_p$ are specific factors.

We use the following assumptions about the factors:

- Each of the common factors have mean zero and unit variance:

$$E(f_j) = 0 \text{ and } var(f_j) = 1.$$

- The **common factors are uncorrelated**:

$$cov(f_i, f_j) = 0 \text{ if } i \neq j.$$

With this assumption, the factor model is known as the **orthogonal factor model**. There are general models where factors are allowed to be correlated, called *oblique factor models*. We will however primarily focus on the orthogonal factor model in this discussion.

- The specific factors have mean zero and unknown variances:

$$E(u_j) = 0 \text{ and } var(u_j) = \psi_j.$$

- The specific factors are uncorrelated among themselves and with the common factors:

$$cov(u_i, u_j) = 0 \text{ if } i \neq j,$$

$$cov(u_i, f_j) = 0 \text{ for all } i, j.$$

According to the orthogonal k-factor model, we can write

$$var(X_i) = \underbrace{\lambda_{i1}^2 + \ldots + \lambda_{ik}^2}_{\text{Communality}} + \underbrace{\psi_i}_{\text{Specific variance}},$$

and that for $i \neq j$,

$$cov(X_i, X_j) = \lambda_{i1}\lambda_{j1} + \ldots + \lambda_{ik}\lambda_{jk}.$$

Like the single factor model, the covariance among the variables are *completely* determined by the common factors; they do not involve the specific factors in any way.

We can still write the covariance matrix of $\boldsymbol{X}$ as

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

where the matrix Λ contains all loadings:

$$\Lambda = \begin{bmatrix} \lambda_{11} & \lambda_{12} & \cdots & \lambda_{1k} \\ \lambda_{21} & \lambda_{22} & \cdots & \lambda_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{p1} & \lambda_{p2} & \cdots & \lambda_{pk} \end{bmatrix}.$$

Notice that

- The i-th row of Λ contain loadings of all factors for the i-th variable X_i. Thus, the *sum of squares* of the i-th row gives the communality of the i-th variable.
- The j-th column of Λ contains loadings of the j-th common factor, f_j, for all the variables.

Given a dataset, we will use the sample covariance matrix $\boldsymbol{S}$, and find the best approximation of the form $\boldsymbol{S} \approx \hat{\Lambda}\hat{\Lambda}^T + \hat{\Psi}$.

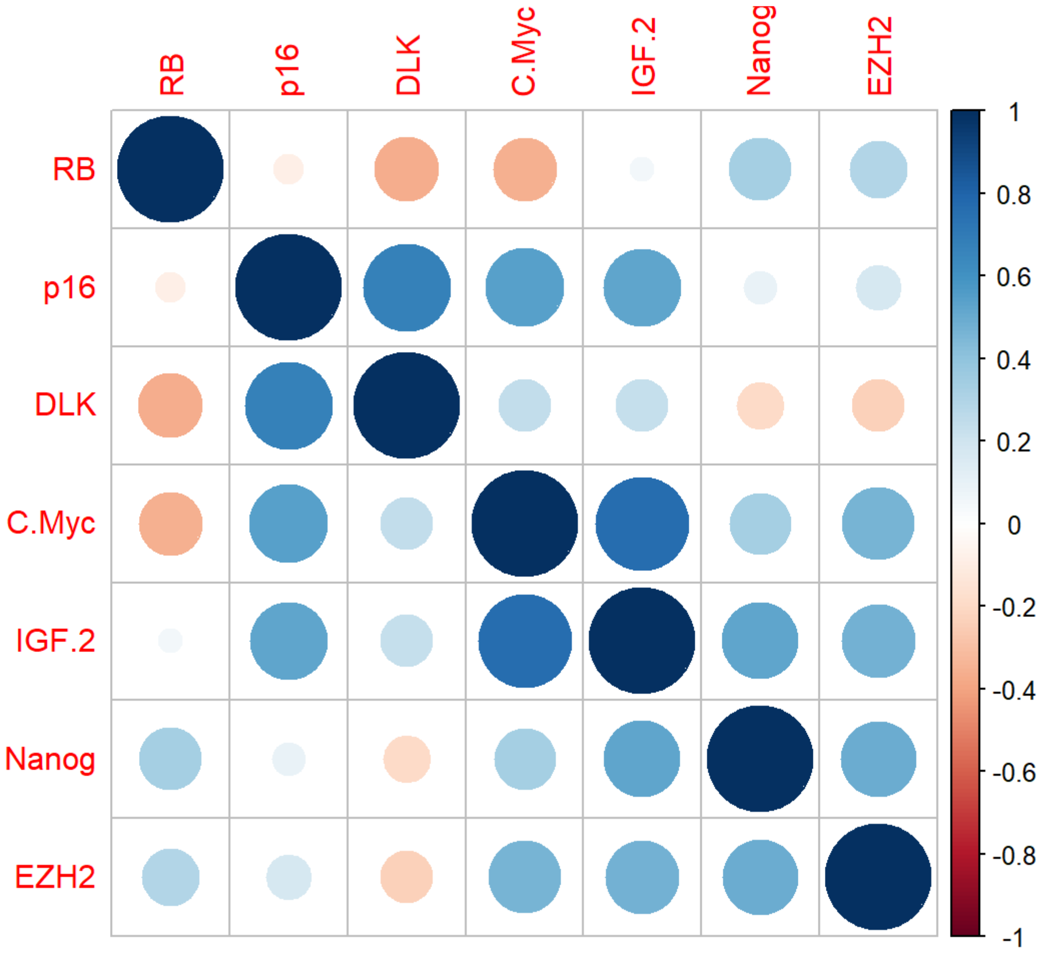
As a quick example, consider the Hemangioma data [Table 8.2 of Applied Multivariate Statistics with R by Daniel Zelterman. New York: Springer]. The dataset contains age (in days) at the time of surgery and expression of seven genetic markers for infants who were surgically treated for hemangioma. The hypothesis of interest is that exposure to the tumor might influence the expression measurements of some of these genes over time.

```
# read the data
dat <- read.table("data/hemangioma.txt", header = T)

# snapshot
head(dat)
```

```
##   Age       RB      p16       DLK     Nanog     C.Myc     EZH2     IGF.2
## 1  81 2.046149 3.067127  308974.7  94.17336  6.489601 2.764101 11175.689
## 2  95 6.540000 1.900000   70988.3 381.83000  1.000000 7.090000  5340.170
## 3  95 3.610000 3.820000  153060.6 237.28000  0.000000 5.570000  6310.240
## 4 165 1.912267 3.735868  596991.6  88.23737  0.000000 2.469633  7008.523
## 5 286 2.625436 5.168293  369600.6 282.29727 12.225828 1.628923  7104.238
## 6 299 2.870760 5.755246 1119257.5 176.75143  8.764235 3.511469  9342.126
```

```
# visual of the correlation matrix
# arrange the variables according their correlation
library(corrplot)
corrplot( cor(dat[, -1]), order = "hclust" )
```



Based on the grouping seen in the correlation matrix, it is conceivable that one factor might be controlling `C.Myc`, `IGF.2`, `Nanog` and `EZH2`. Perhaps, there is another factor controlling the other

For now let us not worry about the exact methods of obtaining Λ, and just inspect the result by using the function `factanal()` in `R`. For this demonstration, we will fit a three factor model ($k = 3$). We do not include `Age` in the analysis. To make all the variables on the same scale, we first standardize the variables.

```
Z <- scale(dat[, -1], scale = TRUE)
out <- factanal(Z, factors = 3)
out
```

```
##
## Call:
## factanal(x = Z, factors = 3)
##
## Uniquenesses:
##    RB   p16   DLK Nanog C.Myc  EZH2 IGF.2
## 0.050 0.293 0.005 0.609 0.005 0.490 0.249
##
## Loadings:
##       Factor1 Factor2 Factor3
## RB     0.141  -0.144   0.954
## p16    0.366   0.757
## DLK   -0.163   0.961  -0.211
## Nanog  0.559           0.275
## C.Myc  0.841   0.295  -0.448
## EZH2   0.682           0.193
## IGF.2  0.780   0.377
##
##                Factor1 Factor2 Factor3
## SS loadings      2.274   1.757   1.269
## Proportion Var   0.325   0.251   0.181
## Cumulative Var   0.325   0.576   0.757
##
## Test of the hypothesis that 3 factors are sufficient.
## The chi square statistic is 1.86 on 3 degrees of freedom.
## The p-value is 0.603
```

The part of the output marked `Uniquenesses` gives the specific variances for each variable. This is the part of $var(X_i)$ not captured by the common factors. Ideally, we would like these values to be small.

The part of the output marked `Loadings` gives the Λ matrix (in this case a 7×3 matrix, since we have 7 variables and 3 common factors). The blanks correspond to zero loadings. It seems the first common factor mostly contribute to `Nanog`, `C.Myc`, `EZH2` and `IGF.2`. The second factor has strong relationship with `p16` and `DLK`. Finally, the third factor essentially relates to `RB`.

The communalities of the variables can be computed by taking the sum of squares of the rows of the loading matrix. In this example, we compute the communalities as below.

```
# Loading matrix Lambda
L <- out$loadings

# communalities are row sum of squares
rowSums(L^2)
```

```
##        RB       p16       DLK     Nanog     C.Myc      EZH2     IGF.2 
## 0.9501293 0.7074504 0.9950027 0.3907492 0.9950049 0.5099920 0.7506007
```

Note that we standardized each variable prior to performing the factor analysis. Thus each variable has variance one. If this three factor model is a good fit to the data, we should expect communalities for each variable to be close to 1.

Better visual of the loadings

We can get a better visual of the loadings by using the `print()` command with various options:

- reducing the printed decimals (`digits`)
- hiding small loadings (`cutoff`)
- sorting the loadings (`sort`)

In our example, we print only 2 decimals, hide loadings less that 0.2, and sort the loadings.

```
print(out, digits = 2, cutoff = .2, sort = TRUE)
```

```
## 
## Call:
## factanal(x = Z, factors = 3)
## 
## Uniquenesses:
##    RB   p16   DLK Nanog C.Myc  EZH2 IGF.2 
##  0.05  0.29  0.00  0.61  0.00  0.49  0.25 
## 
## Loadings:
##       Factor1 Factor2 Factor3
## Nanog  0.56            0.27  
## C.Myc  0.84    0.29   -0.45  
## EZH2   0.68                  
## IGF.2  0.78    0.38          
## p16    0.37    0.76          
## DLK            0.96   -0.21  
## RB                     0.95  
## 
##                Factor1 Factor2 Factor3
## SS loadings       2.27    1.76    1.27
## Proportion Var    0.32    0.25    0.18
## Cumulative Var    0.32    0.58    0.76
## 
## Test of the hypothesis that 3 factors are sufficient.
## The chi square statistic is 1.86 on 3 degrees of freedom.
## The p-value is 0.603
```

Estimation methods

Recall that we want the loadings matrix Λ so that the approximation $\boldsymbol{S} = \Lambda\Lambda^T + \Psi$ is as precise as possible. Two methods to estimate the loadings matrix, Λ, are discussed below.

The principal factor analysis

The procedure of estimating Λ typically involves the following steps:

1. Obtain an initial estimator of communalities and uniquenesses, $\hat{\Psi}$
2. Determine the reduced covariance matrix $\boldsymbol{S}^* = \boldsymbol{S} - \hat{\Psi}$. Obtain estimate of the loading matrix $\hat{\Lambda}$ using PCA on S^*.
3. Update the estimates of communalities and $\hat{\Psi}$ using the estimated loadings
4. Repeat steps (2) - (3) until convergence: We can use some convergence criterion such that the max absolute difference between the new $\hat{\Psi}$ and the old $\hat{\Psi}$ is smaller than a pre-specified tolerance.

Some researchers also prefer to not iterate the steps at all; they stop after step (3) and take the results as the final estimates.

A real life computational difficulty is that sometimes the communality estimates may *exceed* the observed sample variance of some variables, resulting in negative estimate of the specific variances. This is clearly an unacceptable solution. This phenomenon is called a *Heywood case*, attributed to [Heywood, H.(1931), “On finite sequences of real numbers,” Proceedings of the Royal Society of London, Series A, Containing Papers of a Mathematical and Physical Character, 134, 486 – 501].

Maximum likelihood factor analysis

The maximum likelihood factor analysis assumes that the data being analyzed have a multivariate normal distribution.

Under the multivariate normality assumotion, one can write a “likelihood function” for the loading matrix Λ and uniquenesses Ψ, see section 5.5.2 of [Everitt and Hothorn (2011), An Introduction to Applied Multivariate Analysis with R] for more details. We can think of the negative of likelihood function as a criterion that measures the discrepency between $\boldsymbol{S}$ and the covariance structure posited by the factor model, $\Lambda\Lambda^T + \Psi$. Thus we minimize the negative likelihood function (or, equivalently maximize the likelihood) to obtain etimates $\hat{\Lambda}$. This optimization is done iteratively. Such estimates are called the *maximum likelihood estimates* (MLE).

The function `factanal()` fits MLE by default.

There are many other methods to estimate the loadings. Some of them are mentioned below.

- Minimum Residual Factor Analysis: [Harman, Harry and Jones, Wayne (1966) Factor analysis by minimizing residuals (minres), Psychometrika, 31, 3, 351-368.]
- Alpha Factor Analysis: [Kaiser, Henry F. and Caffrey, John. Alpha factor analysis, Psychometrika, (30) 1-14.]
- Weighted/Generalized Least Squares Factor Analysis
- Minimum rank factor analysis: [Shapiro, A. and ten Berge, Jos M. F, (2002) Statistical inference of minimum rank factor analysis. Psychometika, (67) 79-84.]

Many of these options are available in the `psych` package; the command for factor analysis is `fa()`.

How to choose the number of factors?

We can formally determine the number of factors using the MLE approach. For any k (number of factors), we can test the hypothesis

$$H_0 : k \text{ common factors are sufficient.}$$

If we reject this hypothesis (i.e., obtain a small p-value), we need to add more factors. In EFA, we typically can not pre-specify k. Thus we need to perform sequential tests, starting from $k = 1$, and gradually add more factors until we can not reject the hypothesis anymore.

The test statistic follows a χ^2_ν distribution (thus the test is called a chi-squared test) for large sample sizes, where $\nu = (p - k)^2/2 - (p + k)/2$.

If at some point the degrees of freedom, ν, of the test become zero, then it might be that no non-trivial solution is appropriate or the factor model itself is questionable.

In the Hemangeoma data, let us perform these tests for $k = 1, 2$ and 3.

```
# k = 1
out.1 <- factanal(Z, factors = 1)
out.1
```

```
##
## Call:
## factanal(x = Z, factors = 1)
##
## Uniquenesses:
##    RB   p16   DLK Nanog C.Myc  EZH2 IGF.2
## 0.999 0.661 0.931 0.748 0.303 0.738 0.139
##
## Loadings:
##       Factor1
## RB
## p16    0.582
## DLK    0.263
## Nanog  0.502
## C.Myc  0.835
## EZH2   0.512
## IGF.2  0.928
##
##                Factor1
## SS loadings      2.482
## Proportion Var   0.355
##
## Test of the hypothesis that 1 factor is sufficient.
## The chi square statistic is 32.05 on 14 degrees of freedom.
## The p-value is 0.00395
```

```
# k = 2
out.2 <- factanal(Z, factors = 2)
out.2
```

```
##
## Call:
## factanal(x = Z, factors = 2)
##
## Uniquenesses:
##    RB   p16   DLK Nanog C.Myc  EZH2 IGF.2
## 0.846 0.335 0.060 0.578 0.316 0.527 0.166
##
## Loadings:
##       Factor1 Factor2
## RB     0.107  -0.377
## p16    0.427   0.695
## DLK            0.970
## Nanog  0.615  -0.209
## C.Myc  0.780   0.273
## EZH2   0.647  -0.234
## IGF.2  0.878   0.252
##
##                Factor1 Factor2
## SS loadings      2.370   1.802
## Proportion Var   0.339   0.257
## Cumulative Var   0.339   0.596
##
## Test of the hypothesis that 2 factors are sufficient.
## The chi square statistic is 14.92 on 8 degrees of freedom.
## The p-value is 0.0607
```

```
# k = 3
out.3 <- factanal(Z, factors = 3)
out.3
```

```
## 
## Call:
## factanal(x = Z, factors = 3)
## 
## Uniquenesses:
##    RB   p16   DLK Nanog C.Myc  EZH2 IGF.2 
## 0.050 0.293 0.005 0.609 0.005 0.490 0.249 
## 
## Loadings:
##       Factor1 Factor2 Factor3
## RB     0.141  -0.144   0.954 
## p16    0.366   0.757         
## DLK   -0.163   0.961  -0.211 
## Nanog  0.559           0.275 
## C.Myc  0.841   0.295  -0.448 
## EZH2   0.682           0.193 
## IGF.2  0.780   0.377         
## 
##                Factor1 Factor2 Factor3
## SS loadings      2.274   1.757   1.269
## Proportion Var   0.325   0.251   0.181
## Cumulative Var   0.325   0.576   0.757
## 
## Test of the hypothesis that 3 factors are sufficient.
## The chi square statistic is 1.86 on 3 degrees of freedom.
## The p-value is 0.603
```

The p-value for $k = 1$ is very small and clearly suggests that the one factor model is not sufficient. While for $k = 2$ produces p-value lerger than 0.05, the value is stillquite close to 0.05. In contrast, $k = 3$ provides a p-value of 0.6, giving the best fit. If we attemp to increase the numer of factors further, say $k = 4$, the follwoing happens.

```
factanal(Z, factors = 4)
```

```
## Error in factanal(Z, factors = 4): 4 factors are too many for 7 variables
```

This is a case where 4 factors are too many for the dataset.

Fit indices

The `factanal()` function by default produces test results of a chi-squared test. This test is based on comparing the sample and fitted covariance matrices. There are several other indices or measures we can use to assess how well the factor model fits the observed data. A few of these indices are shown below.

- **Akaike Information Criterion (AIC):** Assesses model fit accounting for number of variables present. Small values of AIC are preffered.

- **Bayesian Information Criterion (BIC):** larger penalty for larger sample size. Small values of BIC are preffered.
- **Tucker-Lewis Index:** compares a k-factor model to a model with no constraint. A model with index > 0.95 is considered good.
- **Root Mean Square Error of Approximation (RMSEA):** small values (e.g., < 0.05) is considered good.

Many of these options are available in the `psych` package; the command for factor analysis is `fa()`.

We present the output of the MLE approach using the `fa()` below using a three-factor model.

```
library(psych)
# fa takes a correlation matrix (not the full data) in argument r,
# and the sample size in the argument
n <- nrow(Z)
fa.mle <- fa(r = cor(Z), n.obs = n,
             nfactors = 3, # number of factors to extract
             rotate = "varimax", # how to rotate the factor loadigs
             fm = "ml"  # fm = "ml" produces the MLE solution
             )
fa.mle
```

```
## Factor Analysis using method =  ml
## Call: fa(r = cor(Z), nfactors = 3, n.obs = n, rotate = "varimax", fm = "ml")
## Standardized loadings (pattern matrix) based upon correlation matrix
##         ML2   ML1   ML3   h2    u2 com
## RB     0.14 -0.14  0.95 0.95 0.050 1.1
## p16    0.37  0.76 -0.03 0.71 0.293 1.4
## DLK   -0.16  0.96 -0.21 1.00 0.005 1.2
## Nanog  0.56 -0.05  0.27 0.39 0.609 1.5
## C.Myc  0.84  0.29 -0.45 1.00 0.005 1.8
## EZH2   0.68 -0.09  0.19 0.51 0.490 1.2
## IGF.2  0.78  0.38  0.00 0.75 0.249 1.4
## 
##                        ML2  ML1  ML3
## SS loadings           2.27 1.76 1.27
## Proportion Var        0.32 0.25 0.18
## Cumulative Var        0.32 0.58 0.76
## Proportion Explained  0.43 0.33 0.24
## Cumulative Proportion 0.43 0.76 1.00
## 
## Mean item complexity =  1.4
## Test of the hypothesis that 3 factors are sufficient.
## 
## The degrees of freedom for the null model are  21  and the objective function was 
4.13 with Chi Square of  61.32
## The degrees of freedom for the model are 3  and the objective function was  0.14 
## 
## The root mean square of the residuals (RMSR) is  0.03 
## The df corrected root mean square of the residuals is  0.08 
## 
## The harmonic number of observations is  19 with the empirical chi square  0.78  wi
th prob <  0.85 
## The total number of observations was  19  with Likelihood Chi Square =  1.86  with 
prob <  0.6 
## 
## Tucker Lewis Index of factoring reliability =  1.25
## RMSEA index =  0  and the 90 % confidence intervals are  0 0.331
## BIC =  -6.98
## Fit based upon off diagonal values = 0.99
## Measures of factor score adequacy             
##                                                    ML2  ML1  ML3
## Correlation of (regression) scores with factors   0.99 1.00 0.98
## Multiple R square of scores with factors          0.99 0.99 0.95
## Minimum correlation of possible factor scores     0.97 0.98 0.91
```

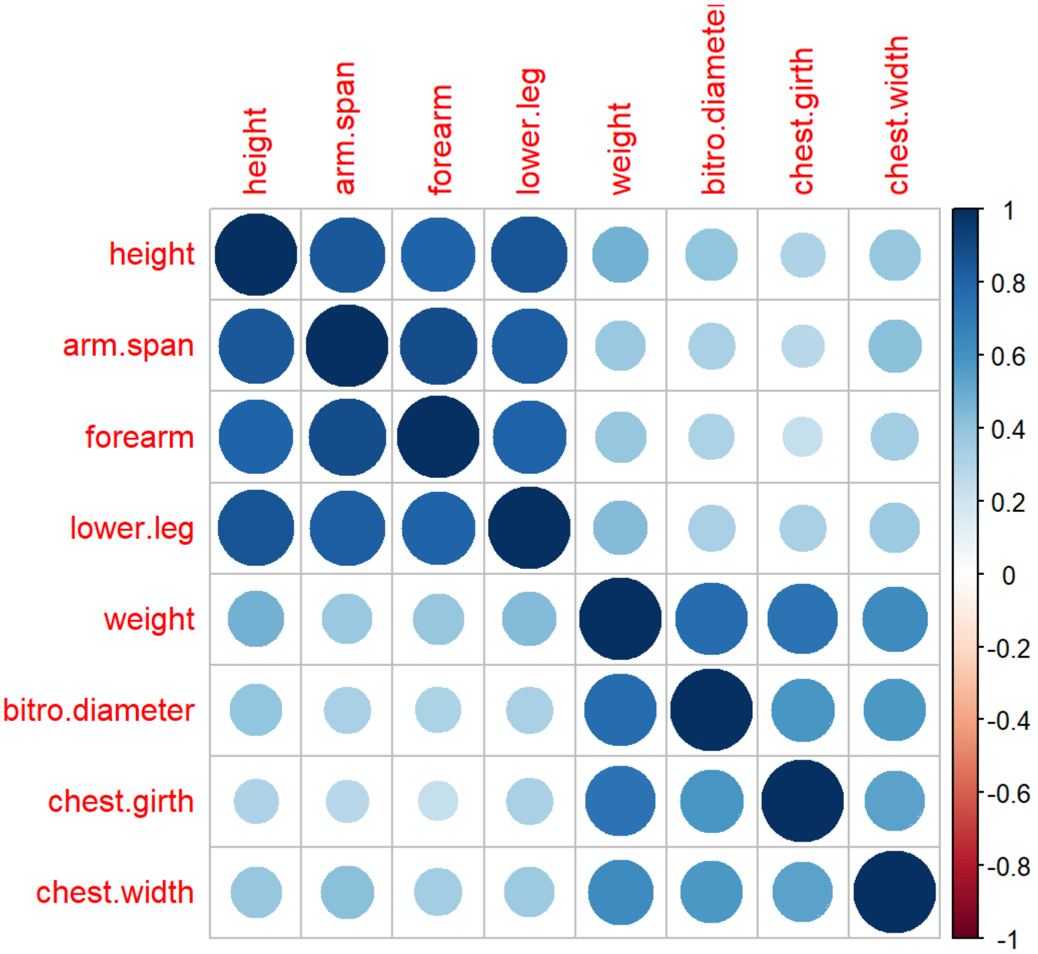
The columns names `ML1` – `ML3` present the loadings for the three factors; the column `h2` gives the communalities, and `u2` shows the uniquenesses. Below the loadings, various indices are reported.

Factor Rotation

An interesting aspect of factor analysis is that the loadings matrix is not unique. In other words, **two or more different loadings matrices can produce the exact same fit.** We show such an example below.

Consider the `Harman23.cor` dataset in the `datasets` package. The list contains a correlation matrix of eight physical measurements on $n = 305$ girls between ages seven and seventeen.

```
library(corrplot)
corrplot(Harman23.cor$cov)
```



Let us fit a two-factor model to this correlation matrix. Below I show two different loadings matrices, both of which produce the same communalities and the same uniquenesses.

```
# Model fit-1
fa.out.none <- fa(r = Harman23.cor$cov, nfactors = 2,
   n.obs = Harman23.cor$n.obs, fm = "ml",
   rotate = "none")

# Model-fit-2
fa.out.varimax <- fa(r = Harman23.cor$cov, nfactors = 2,
                n.obs = Harman23.cor$n.obs, fm = "ml",
                rotate = "varimax")

fa.out.none$loadings
```

```
##
## Loadings:
##                 ML1    ML2
## height          0.880 -0.237
## arm.span        0.874 -0.360
## forearm         0.846 -0.344
## lower.leg       0.855 -0.263
## weight          0.705  0.644
## bitro.diameter  0.589  0.538
## chest.girth     0.527  0.554
## chest.width     0.574  0.365
##
##                   ML1   ML2
## SS loadings    4.434 1.518
## Proportion Var 0.554 0.190
## Cumulative Var 0.554 0.744
```

```
fa.out.varimax$loadings
```

```
##
## Loadings:
##                 ML1   ML2
## height         0.865 0.287
## arm.span       0.927 0.181
## forearm        0.895 0.179
## lower.leg      0.859 0.252
## weight         0.233 0.925
## bitro.diameter 0.194 0.774
## chest.girth    0.134 0.752
## chest.width    0.278 0.621
##
##                   ML1   ML2
## SS loadings    3.335 2.617
## Proportion Var 0.417 0.327
## Cumulative Var 0.417 0.744
```

These are two different loading matrices. However, the produce the same communalities, as we see below.

```
# communalities for model-1 and model-2
com.1 <- rowSums(fa.out.none$loadings^2)
com.2 <- rowSums(fa.out.varimax$loadings^2)
comboth <- cbind(com.1, com.2)
colnames(comboth) <- c("Model-1", "Model-2")
round(comboth, 3)
```

```
##                Model-1 Model-2
## height           0.830   0.830
## arm.span         0.893   0.893
## forearm          0.834   0.834
## lower.leg        0.801   0.801
## weight           0.911   0.911
## bitro.diameter   0.636   0.636
## chest.girth      0.584   0.584
## chest.width      0.463   0.463
```

It is evident the communalities are identical. Similarly, the uniquenesses are identical as well, as seen below.

```
# Uniquenesses for the two models
u2 <- cbind(fa.out.none$uniquenesses, fa.out.varimax$uniquenesses)
colnames(u2) <- c("Model-1", "Model-2")
round(u2, 3)
```

```
##                Model-1 Model-2
## height           0.170   0.170
## arm.span         0.107   0.107
## forearm          0.166   0.166
## lower.leg        0.199   0.199
## weight           0.089   0.089
## bitro.diameter   0.364   0.364
## chest.girth      0.416   0.416
## chest.width      0.537   0.537
```

It can be shown that there are infinite number of such loadings. Given a particular loadings matrix, we can obtain other loadings matrices by applying a method called **roration.** Often, after performing EFA, we find the resulting loadings matrix difficult to interpret. Then we might apply rotation to obtain other loadings, which will provide identical communalities and uniquenesses, in hope of getting one that is easy to interpret. Specifically, **loadings are easy to interpret if each variable loads highly on at most one factor; and that all factor loadings are either large or close to zero.**

There are two types of rotation:

- *orthogonal rotation*: the factors remain uncorrelated after rotation
- *oblique rotation*: the rotated factors can be correlated.

We will mostly focus on orthogonal rotation. For more details on both types of rotations, please Chapter 5.7 of Everitt and Hothorn (2011) textbook. Among many available techinques for orthogonal rotations, the follwoing two are popular:

- *Varimax rotation:* tries to have factors with only a few large loadings and as many near-zero loadings as possible. This is parhaps one of the most popular rotation methods.
- *Quartimax rotation:* tries to force each variable to correlate highly with at most one factor, and have zero of small correlation with the rest of the factors.

We show examples of a estimted EFA model before and after a `varimax` rotation. We use the `Herman23.cor` data. To simplyfy our presentation, we regard any loadings with absolute value below 0.2 to be insignificant. In the `fa()` (in `psych` library) The argument `rotate` controls the rotation we want to apply. In the function `factanal()` the argument is `rotation`. We show the results from the `fa()` function.

Loadings with no rotation

```
# No rotation
fa.out.none <- fa(r = Harman23.cor$cov, nfactors = 2,
   n.obs = Harman23.cor$n.obs, fm = "ml",
   rotate = "none")
print(fa.out.none$loadings, digits = 2, cutoff = 0.3)
```

```
##
## Loadings:
##                ML1   ML2
## height          0.88
## arm.span        0.87 -0.36
## forearm         0.85 -0.34
## lower.leg       0.86
## weight          0.70  0.64
## bitro.diameter  0.59  0.54
## chest.girth     0.53  0.55
## chest.width     0.57  0.37
##
##                 ML1  ML2
## SS loadings    4.43 1.52
## Proportion Var 0.55 0.19
## Cumulative Var 0.55 0.74
```

Loadings with varimax rotation

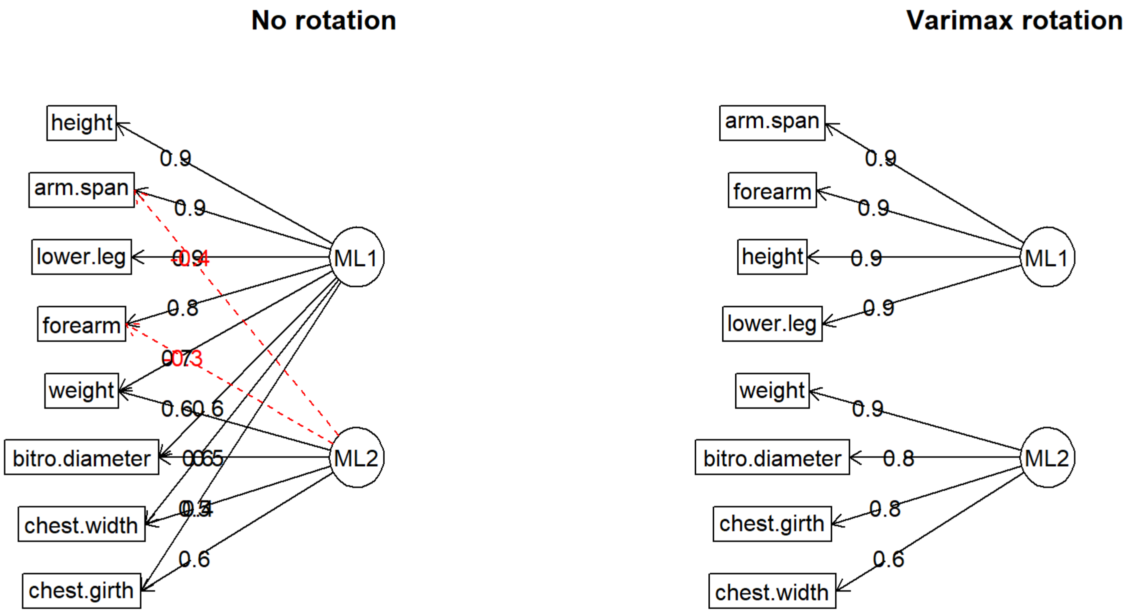
```
# Varimax rotation
fa.out.varimax <- fa(r = Harman23.cor$cov, nfactors = 2,
   n.obs = Harman23.cor$n.obs, fm = "ml",
   rotate = "varimax")
print(fa.out.varimax$loadings, digits = 2, cutoff = 0.3)
```

```
##
## Loadings:
##                 ML1  ML2
## height          0.86
## arm.span        0.93
## forearm         0.90
## lower.leg       0.86
## weight               0.93
## bitro.diameter       0.77
## chest.girth          0.75
## chest.width          0.62
##
##                  ML1  ML2
## SS loadings     3.33 2.62
## Proportion Var 0.42 0.33
## Cumulative Var 0.42 0.74
```

We can see that the unrotated loadings (left panel above) do not group the variables in a clearly. The are multiple variables which are correlated with both the factors. In contrast, the varimax rotated loadings (righ panel above) clearly groups variables into two groups.

Such patterns can be clearly seen if draw diagrams for the fitted models. We show the diagrams below. This can be done using the `fa.diagram()` function in the `psych` package.

```
par(mfrow = c(1,2))
# no rotation
fa.diagram(fa.out.none, cut=0.3, simple = F, main = "No rotation")
# varimax
fa.diagram(fa.out.varimax, cut=0.3, simple = F, main = "Varimax rotation")
```



No rotation

Varimax rotation

Estimating the Factor Scores

Once we estimate the loadings, $\hat{\lambda}_{ij}$, the next step is to estimate the *factor scores* for each individual.

- The factor scores are low-dimensional summaries of the individual data vectors
- The scores can be used in further analysis, e.g., regression, classification, clustering.
- The scores are often regarded as more reliable measures of the underlying latent factors compared to the observed variables.

Typically, we can not "estimate" the factor scores (as they are random variables). Instead we "predict" them. Define the vector of factors as $f = (f_1, \ldots, f_k)^T$ for a k-factor model. Under the MLE approach (i.e., under the assumption of normality of the data), we can show that, *conditional on the observed X's*, f follows a multivariate normal distribution:

$$N(\Lambda^T \Sigma^{-1} \boldsymbol{X}, (\Lambda^T \Psi^{-1} \Lambda + I)^{-1})$$

Thus we can predict the factor scores by the mean of the distribution above. Specifically, suppose we have observed data vector $\boldsymbol{x}_i$. Then the corresponding factor scores will be

$$\hat{\Lambda}^T \boldsymbol{S}^{-1} \boldsymbol{x}_i,$$

where $\hat{\Lambda}$ are the estimated loadings, $\boldsymbol{S}$ is the sample covariance matrix (since we do not know $\boldsymbol{\Sigma}$). Thus we can get a vector of k scores (one for each factor) for *each individual* in the dataset.

Clearlt, we need the full dataset (since we need $\boldsymbol{x}_i$ to obtain the scores. Using the `fa()` function on the Hemangioma data with three factor model, we can extract the scores as follows.

```
# Z is the standardized hemangioma data used before
fa.out <- fa(Z, nfactors = 3,
   n.obs = Harman23.cor$n.obs, fm = "ml",
   rotate = "varimax",
   scores = "regression") # get the factor scores
# extract the scores
scores <- fa.out$scores
dim(scores)
```

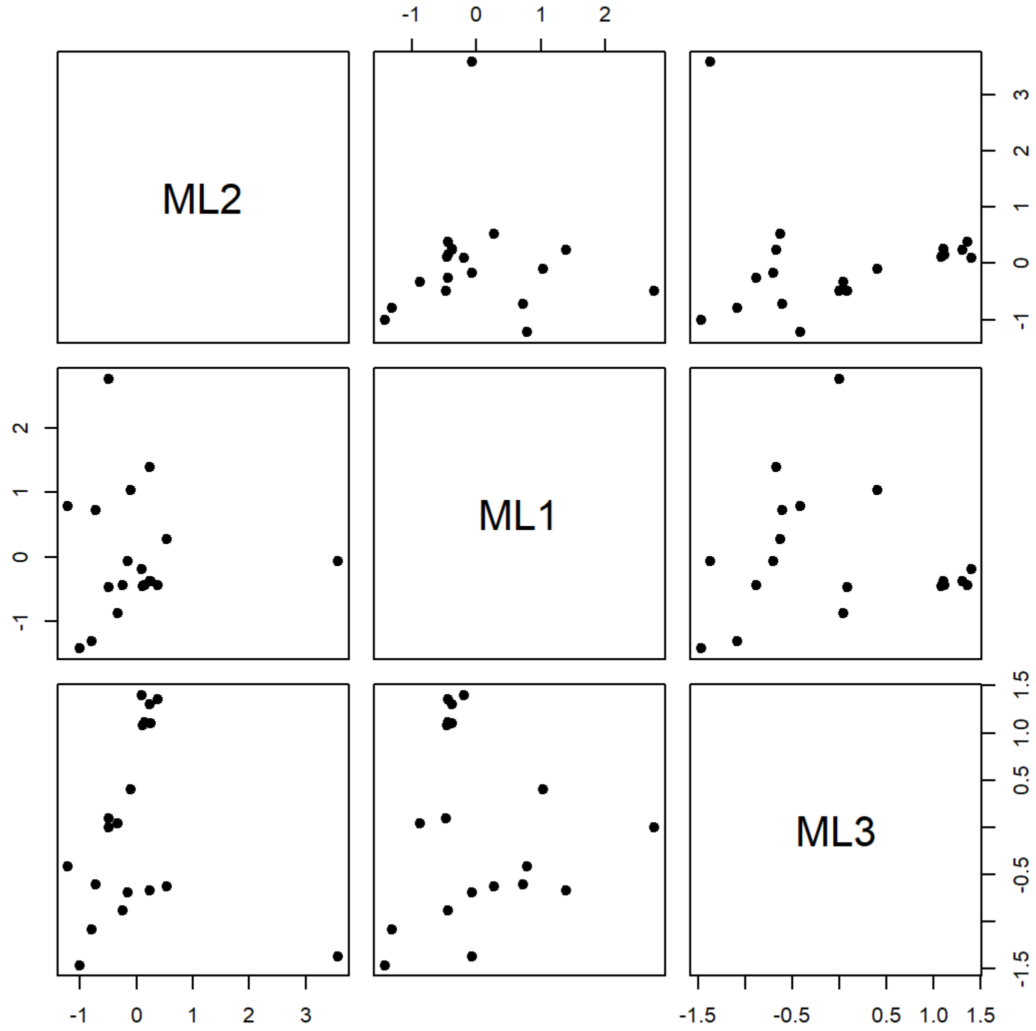
```
## [1] 19  3
```

```
scores
```

```
##              ML2         ML1          ML3
##  [1,] -0.1620326 -0.06918913 -0.700367508
##  [2,]  0.1476712 -0.44229872  1.112938030
##  [3,] -0.4954316 -0.47799927  0.086335836
##  [4,] -1.2152987  0.78997367 -0.421894700
##  [5,]  0.5291103  0.27449690 -0.635231228
##  [6,] -0.4851749  2.76808129 -0.004631437
##  [7,] -0.2512363 -0.44959842 -0.885961234
##  [8,]  0.2431462 -0.37529006  1.299771575
##  [9,]  0.2550536 -0.37578215  1.099547399
## [10,]  0.1130533 -0.45976013  1.079153421
## [11,]  3.5716584 -0.07583270 -1.375130537
## [12,]  0.1002018 -0.19651053  1.397011047
## [13,] -0.7272263  0.73116345 -0.609391288
## [14,]  0.3780132 -0.44442973  1.357119551
## [15,]  0.2365987  1.39409250 -0.675415397
## [16,] -0.7970907 -1.32223159 -1.089897101
## [17,] -0.3252087 -0.87717411  0.036753230
## [18,] -1.0120958 -1.42801801 -1.468407246
## [19,] -0.1037111  1.03630674  0.397697588
```

Since we fit three factors, and the dataset has 19 individuals, we obtain 3 scores for each individual (each row above). A scatterplot of the scores is below.

```
pairs(scores, pch=19)
```



Principal Components Analysis vs. Factor Analysis

Similarities

- Both techniques look for hidden structures in the data
- can be used for data reduction/scoring

Differences

- PCA does not assume any model for the covariance matrix (i.e, it is an unstructured fit). EFA posits a specific model (of the form $\Lambda\Lambda^T + \Psi$, i.e, a “low rank” matrix plus a diagonal matrix) and estimates the parameters.
- IN PCA, PCs are linear combinations of the variables. In factor model, the variables are modeled as a linear combinations of latent factors.
- PCA attempts capture most of the total variance. EFA tries to maximize the variance due to common factors.

- PCA, as it is, does not account for possible measurement errors in the observed variables. EFA can accomodate measurement errors in the variables through specific factors.
- In PCA, the PCs are orthogonal by construction. In EFA, there are *oblique factor models* that allow for correlation among factors.

Main page: **ST 437/537: Applied Multivariate and Longitudinal Data Analysis (https://maityst537.wordpress.ncsu.edu/)**