Factor Analysis

Nathaniel E. Helwig

Assistant Professor of Psychology and Statistics
University of Minnesota (Twin Cities)

Updated 16-Mar-2017

Copyright

Outline of Notes

1) Background
- Overview
- FA vs PCA

2) Factor Analysis Model
- Model Form
- Parameter Estimation
- Factor Rotation
- Factor Scores

3) Some Extensions
- Oblique Factor Model
- Confirmatory Factor Analysis

4) Decathlon Example
- Data Overview
- Estimate Factor Loadings
- Estimate Factor Scores
- Oblique Rotation

Background

Definition and Purposes of FA

Factor Analysis (FA) assumes the covariation structure among a set of variables can be described via a linear combination of unobservable (latent) variables called factors.

There are three typical purposes of FA:

1. Data reduction: explain covariation between p variables using $r < p$ latent factors
2. Data interpretation: find features (i.e., factors) that are important for explaining covariation (exploratory FA)
3. Theory testing: determine if hypothesized factor structure fits observed data (confirmatory FA)

Difference between FA and PCA

FA and PCA have similar themes, i.e., to explain covariation between variables via linear combinations of other variables.

However, there are distinctions between the two approaches:

- FA assumes a statistical model that describes covariation in observed variables via linear combinations of latent variables
- PCA finds uncorrelated linear combinations of observed variables that explain maximal variance (no latent variables here)

FA refers to a statistical model, whereas PCA refers to the eigenvalue decomposition of a covariance (or correlation) matrix.

Factor Analysis Model

Factor Model with m Common Factors

$\boldsymbol{X} = (X_1, \ldots, X_p)'$ is a random vector with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$.

The Factor Analysis model assumes that

$$\boldsymbol{X} = \boldsymbol{\mu} + \mathbf{L}\boldsymbol{F} + \boldsymbol{\epsilon}$$

where

- $\mathbf{L} = \{\ell_{jk}\}_{p \times m}$ denotes the matrix of factor loadings
    - ℓ_{jk} is the loading of the j-th variable on the k-th common factor
- $\boldsymbol{F} = (F_1, \ldots, F_m)'$ denotes the vector of latent factor scores
    - F_k is the score on the k-th common factor
- $\boldsymbol{\epsilon} = (\epsilon_1, \ldots, \epsilon_p)'$ denotes the vector of latent error terms
    - ϵ_j is the j-th specific factor

Orthogonal Factor Model Assumptions

The orthogonal FA model assumes the form

$$\boldsymbol{X} = \boldsymbol{\mu} + \mathbf{L}\boldsymbol{F} + \boldsymbol{\epsilon}$$

and adds the assumptions that

- $\boldsymbol{F} \sim (\mathbf{0}, \mathbf{I}_m)$, i.e., the latent factors have mean zero, unit variance, and are uncorrelated
- $\boldsymbol{\epsilon} \sim (\mathbf{0}, \boldsymbol{\Psi})$ where $\boldsymbol{\Psi} = \mathrm{diag}(\psi_1, \ldots, \psi_p)$ with ψ_j denoting the j-th specific variance
- ϵ_j and F_k are independent of one another for all pairs j, k

Orthogonal Factor Model Implied Covariance Structure

The implied covariance structure for $\boldsymbol{X}$ is

$$\begin{aligned}\mathrm{Var}(\boldsymbol{X}) &= E[(\boldsymbol{X}-\boldsymbol{\mu})(\boldsymbol{X}-\boldsymbol{\mu})'] \\ &= E[(\mathbf{L}\boldsymbol{F}+\boldsymbol{\epsilon})(\mathbf{L}\boldsymbol{F}+\boldsymbol{\epsilon})'] \\ &= E[\mathbf{L}\boldsymbol{F}\boldsymbol{F}'\mathbf{L}'] + E[\mathbf{L}\boldsymbol{F}\boldsymbol{\epsilon}'] + E[\boldsymbol{\epsilon}\boldsymbol{F}'\mathbf{L}'] + E[\boldsymbol{\epsilon}\boldsymbol{\epsilon}'] \\ &= \mathbf{L}E[\boldsymbol{F}\boldsymbol{F}']\mathbf{L}' + \mathbf{L}E[\boldsymbol{F}\boldsymbol{\epsilon}'] + E[\boldsymbol{\epsilon}\boldsymbol{F}']\mathbf{L}' + E[\boldsymbol{\epsilon}\boldsymbol{\epsilon}'] \\ &= \mathbf{L}\mathbf{L}' + \boldsymbol{\Psi}\end{aligned}$$

where $E[\boldsymbol{F}\boldsymbol{F}'] = \mathbf{I}_m$, $E[\boldsymbol{F}\boldsymbol{\epsilon}'] = \mathbf{0}_{m\times p}$, $E[\boldsymbol{\epsilon}\boldsymbol{F}'] = \mathbf{0}_{p\times m}$, and $E[\boldsymbol{\epsilon}\boldsymbol{\epsilon}'] = \boldsymbol{\Psi}$.

This implies that the covariance between $\boldsymbol{X}$ and $\boldsymbol{F}$ has the form

$$\begin{aligned}\mathrm{Cov}(\boldsymbol{X},\boldsymbol{F}) &= E[(\boldsymbol{X}-\boldsymbol{\mu})\boldsymbol{F}'] \\ &= E[(\mathbf{L}\boldsymbol{F}+\boldsymbol{\epsilon})\boldsymbol{F}'] = \mathbf{L}\end{aligned}$$

Variance Explained by Common Factors

The portion of variance of the j-th variable that is explained by the m common factors is called the communality of the j-th variable:

$$\underbrace{\sigma_{jj}}_{\text{Var}(X_j)} = \underbrace{h_j^2}_{\text{communality}} + \underbrace{\psi_j}_{\text{uniqueness}}$$

where

- σ_{jj} is the variance of X_j (i.e., the j-th diagonal of $\boldsymbol{\Sigma}$)
- $h_j^2 = (\mathbf{LL}')_{jj} = \ell_{j1}^2 + \ell_{j2}^2 + \cdots + \ell_{jm}^2$ is the communality of X_j
- ψ_j is the specific variance (or uniqueness) of X_j

Note that the communality h_j^2 is the sum of squared loadings for X_j.

Principal Components Solution for Factor Analysis

Note that the parameters of interest are the factor loadings $\mathbf{L}$ and specific variances on the diagonal of $\boldsymbol{\Psi}$.

For $m < p$ common factors, the PCA solution estimates $\mathbf{L}$ and $\boldsymbol{\Psi}$ as

$$\hat{\mathbf{L}} = \left[\lambda_1^{1/2}\mathbf{v}_1, \lambda_2^{1/2}\mathbf{v}_2, \ldots, \lambda_m^{1/2}\mathbf{v}_m\right]$$
$$\hat{\psi}_j = \sigma_{jj} - \hat{h}_j^2$$

where $\boldsymbol{\Sigma} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}'$ is the eigenvalue decomposition of $\boldsymbol{\Sigma}$, and $\hat{h}_j^2 = \sum_{k=1}^m \hat{\ell}_{jk}^2$ is the estimated communality of the j-th variable.

Proportion of total sample variance explained by the k-th factor is

$$R_k^2 = \frac{\sum_{j=1}^p \hat{\ell}_{jk}^2}{\sum_{j=1}^p \sigma_{jj}} = \frac{\left(\lambda_k^{1/2}\mathbf{v}_k\right)'\left(\lambda_k^{1/2}\mathbf{v}_k\right)}{\sum_{j=1}^p \sigma_{jj}} = \frac{\lambda_k}{\sum_{j=1}^p \sigma_{jj}}$$

Iterated Principal Axis Factoring Method

Assume we are applying FA to a sample correlation matrix $\mathbf{R}$

$$\mathbf{R} - \boldsymbol{\Psi} = \mathbf{L}\mathbf{L}'$$

and we have some initial estimate of the specific variance $\hat{\psi}_j$.

- Can use $\hat{\psi}_j = 1/r^{jj}$ where r^{jj} is the j-th diagonal of $\mathbf{R}^{-1}$

The iterated principal axis factoring algorithm:

1. Form $\tilde{\mathbf{R}} = \mathbf{R} - \hat{\boldsymbol{\Psi}}$ given current $\hat{\psi}_j$ estimates
2. Update $\tilde{\mathbf{L}} = \left[\tilde{\lambda}_1^{1/2}\tilde{\mathbf{v}}_1, \tilde{\lambda}_2^{1/2}\tilde{\mathbf{v}}_2, \ldots, \tilde{\lambda}_m^{1/2}\tilde{\mathbf{v}}_m\right]$ where $\tilde{\mathbf{R}} = \tilde{\mathbf{V}}\tilde{\boldsymbol{\Lambda}}\tilde{\mathbf{V}}'$ is the eigenvalue decomposition of $\tilde{\mathbf{R}}$
3. Update $\hat{\psi}_j = 1 - \sum_{k=1}^m \tilde{\ell}_{jk}^2$

Maximum Likelihood Estimation for Factor Analysis

Suppose $\mathbf{x}_i \overset{\text{iid}}{\sim} \mathrm{N}(\boldsymbol{\mu}, \mathbf{L}\mathbf{L}' + \boldsymbol{\Psi})$ is a multivariate normal vector.

The log-likelihood function for a sample of n observations has the form

$$LL(\boldsymbol{\mu}, \mathbf{L}, \boldsymbol{\Psi}) = -\frac{np\log(2\pi)}{2} + \frac{n\log(|\boldsymbol{\Sigma}^{-1}|)}{2} - \frac{\sum_{i=1}^{n}(\mathbf{x}_i - \boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}(\mathbf{x}_i - \boldsymbol{\mu})}{2}$$

where $\boldsymbol{\Sigma} = \mathbf{L}\mathbf{L}' + \boldsymbol{\Psi}$. Use an iterative algorithm to maximize LL.

Benefit of ML solution: there is a simple relationship between FA solution for **S** (covariance matrix) and **R** (correlation matrix).

- If $\hat{\theta}$ is the MLE of θ, then $g(\hat{\theta})$ is the MLE of $g(\theta)$

Rotating Points in Two Dimensions

Suppose we have $\mathbf{z} = (x, y)' \in \mathbb{R}^2$, i.e., points in 2D Euclidean space.

A 2×2 orthogonal rotation of (x, y) of the form

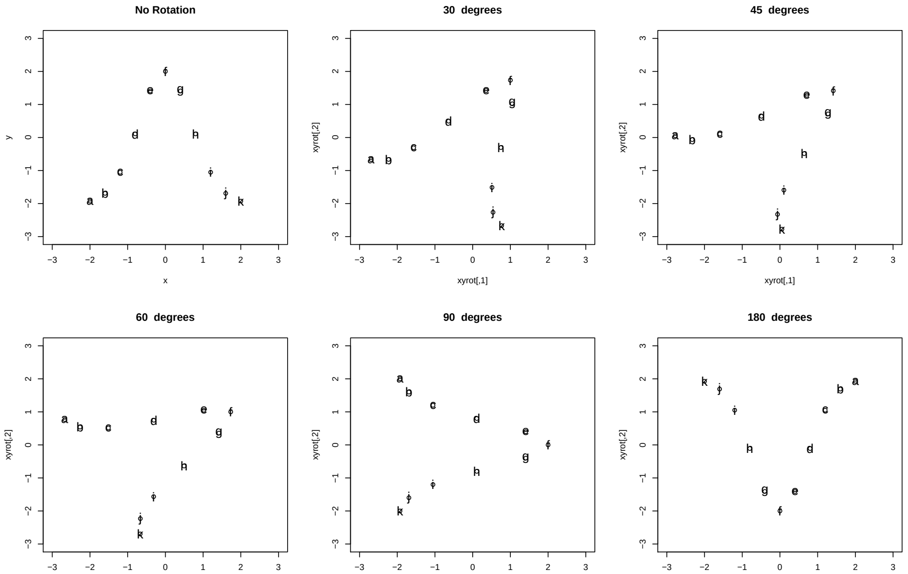
$$\begin{pmatrix} x^* \\ y^* \end{pmatrix} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

rotates (x, y) counter-clockwise around the origin by an angle of θ and

$$\begin{pmatrix} x^* \\ y^* \end{pmatrix} = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

rotates (x, y) clockwise around the origin by an angle of θ.

Visualization of 2D Clockwise Rotation



Visualization of 2D Clockwise Rotation (R Code)

```
rotmat2d <- function(theta){
  matrix(c(cos(theta),sin(theta),-sin(theta),cos(theta)),2,2)
}
x <- seq(-2,2,length=11)
y <- 4*exp(-x^2) - 2
xy <- cbind(x,y)
rang <- c(30,45,60,90,180)
dev.new(width=12,height=8,noRStudioGD=TRUE)
par(mfrow=c(2,3))
plot(x,y,xlim=c(-3,3),ylim=c(-3,3),main="No Rotation")
text(x,y,labels=letters[1:11],cex=1.5)
for(j in 1:5){
  rmat <- rotmat2d(rang[j]*2*pi/360)
  xyrot <- xy%*%rmat
  plot(xyrot,xlim=c(-3,3),ylim=c(-3,3))
  text(xyrot,labels=letters[1:11],cex=1.5)
  title(paste(rang[j]," degrees"))
}
```

Orthogonal Rotation in Two Dimensions

Note that the 2×2 rotation matrix

$$\mathbf{R} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}$$

is an orthogonal matrix for all θ:

$$\begin{aligned} \mathbf{R}'\mathbf{R} &= \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \\ &= \begin{pmatrix} \cos^2(\theta) + \sin^2(\theta) & \cos(\theta)\sin(\theta) - \cos(\theta)\sin(\theta) \\ \cos(\theta)\sin(\theta) - \cos(\theta)\sin(\theta) & \cos^2(\theta) + \sin^2(\theta) \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \end{aligned}$$

Orthogonal Rotation in Higher Dimensions

Suppose we have a data matrix $\mathbf{X}$ with p columns.

- Rows of $\mathbf{X}$ are coordinates of points in p-dimensional space
- Note: when $p = 2$ we have situation on previous slides

A $p \times p$ orthogonal rotation is an orthogonal linear transformation.

- $\mathbf{R}'\mathbf{R} = \mathbf{R}\mathbf{R}' = \mathbf{I}_p$ where $\mathbf{I}_p$ is $p \times p$ identity matrix
- If $\tilde{\mathbf{X}} = \mathbf{X}\mathbf{R}$ is rotated data matrix, then $\tilde{\mathbf{X}}\tilde{\mathbf{X}}' = \mathbf{X}\mathbf{X}'$
- Orthogonal rotation preserves relationships between points

Rotational Indeterminacy of Factor Analysis Model

Suppose $\mathbf{R}$ is an orthogonal rotation matrix, and note that

$$\begin{aligned} \boldsymbol{X} &= \mu + \mathbf{L}\boldsymbol{F} + \epsilon \\ &= \mu + \tilde{\mathbf{L}}\tilde{\boldsymbol{F}} + \epsilon \end{aligned}$$

where

- $\tilde{\mathbf{L}} = \mathbf{L}\mathbf{R}$ are the rotated factor loadings
- $\tilde{\boldsymbol{F}} = \mathbf{R}'\boldsymbol{F}$ are the rotated factor scores

Note that $\tilde{\mathbf{L}}\tilde{\mathbf{L}}' = \mathbf{L}\mathbf{L}'$, so we can orthogonally rotate the FA solution without changing the implied covariance structure.

Factor Rotation and Thurstone's Simple Structure

Factor rotation methods attempt to find some rotation of a FA solution that provides a more parsimonious interpretation.

Thurstone's (1947) simple structure describes an "ideal" factor solution

1. Each row of **L** contains at least one zero
2. Each column of **L** contains at least one zero
3. For each pair of columns of **L**, there should be several variables with small loadings on only one of the two factors
4. For each pair of columns of **L**, there should be several variables with small loadings on both factors if $m \geq 4$
5. For each pair of columns of **L**, there should be only a few variables with large loadings on both factors

Orthogonal Factor Rotation Methods

Many popular orthogonal factor rotation methods try to maximize

$$V(\mathbf{L}, \mathbf{R}|\gamma) = \frac{1}{p}\sum_{k=1}^{m}\left[\sum_{j=1}^{p}(\tilde{\ell}_{jk}/\tilde{h}_j)^4 - \frac{\gamma}{p}\left(\sum_{j=1}^{p}(\tilde{\ell}_{jk}/\tilde{h}_j)^2\right)^2\right]$$

where

- $\tilde{\ell}_{jk}$ is the rotated loading of the j-th variable on the k-th factor
- $\tilde{h}_j = \sqrt{\sum_{k=1}^{m}\tilde{\ell}_{jk}^2}$ is the square-root of the communality for X_j

Changing the γ parameter corresponds to different criertia

- $\gamma = 1$ corresponds to varimax criterion
- $\gamma = 0$ corresponds to quartimax criterion
- $\gamma = m/2$ corresponds to equamax criterion
- $\gamma = p(m-1)/(p+m-2)$ corresponds to parsimax criterion

Issues Related to Factor Scores

In FA, one may want to obtain estimates of the latent factor scores $\boldsymbol{F}$.

However, $\boldsymbol{F}$ is a random variable, so estimating realizations of $\boldsymbol{F}$ is different from estimating the parameters of the FA model ($\mathbf{L}$ and $\boldsymbol{\Psi}$).

- Note that $\mathbf{L}$ and $\boldsymbol{\Psi}$ are unknown constants at the population
- $\boldsymbol{F}$ is a random variable at the population

Estimation of FA scores makes sense if PCA solution is used, but one should proceed with caution otherwise.

Factor Score Indeterminacy (Controversial Topic)

To understand the problem, rewrite the FA model as

$$\begin{aligned}\boldsymbol{X} &= \boldsymbol{\mu} + \mathbf{L}\boldsymbol{F} + \boldsymbol{\epsilon} \\ &= \boldsymbol{\mu} + \begin{pmatrix}\mathbf{L} & \mathbf{I}_p\end{pmatrix}\begin{pmatrix}\boldsymbol{F} \\ \boldsymbol{\epsilon}\end{pmatrix} \\ &= \boldsymbol{\mu} + \mathbf{L}^*\boldsymbol{F}^*\end{aligned}$$

where $\mathbf{L}^*$ is a $p \times m + p$ matrix of common and specific factor loadings, and $\boldsymbol{F}^*$ is a $m + p \times 1$ vector of common and specific factor scores.

Given $\boldsymbol{\mu}$ and $\mathbf{L}$, we have $m + p$ unknowns (elements of $\boldsymbol{F}^*$) but only p equations available to solve for the unknowns.

- Fixing m and letting $p \to \infty$, the indeterminacy vanishes
- For finite p, there are an infinite number of $(\boldsymbol{F}, \boldsymbol{\epsilon})$ combinations

Estimating Factor Scores: Least Squares Method

Let $\hat{\mathbf{L}}$ and $\hat{\boldsymbol{\Psi}}$ denote estimates of $\mathbf{L}$ and $\boldsymbol{\Psi}$.

The weighted least squares estimate of the factor scores are

$$\hat{\mathbf{f}}_i = \left(\hat{\mathbf{L}}'\hat{\boldsymbol{\Psi}}^{-1}\hat{\mathbf{L}}\right)^{-1}\hat{\mathbf{L}}'\hat{\boldsymbol{\Psi}}^{-1}(\mathbf{x}_i - \bar{\mathbf{x}})$$

where

- $\mathbf{x}_i$ is the i-th subject's vector of data
- $\bar{\mathbf{x}} = (1/n)\sum_{i=1}^{n}\mathbf{x}_i$ is the sample mean

Note that if PCA is used to estimate $\hat{\mathbf{L}}$ and $\hat{\boldsymbol{\Psi}}$, then it is typical to use

$$\hat{\mathbf{f}}_i = \left(\hat{\mathbf{L}}'\hat{\mathbf{L}}\right)^{-1}\hat{\mathbf{L}}'(\mathbf{x}_i - \bar{\mathbf{x}})$$

which is the unweighted least squares estimate.

Estimating Factor Scores: Regression Method

Using the ML method, the joint distribution of $(\boldsymbol{X} - \mu, \boldsymbol{F})$ is multivariate normal with mean vector $\mathbf{0}_{p+m}$ and covariance matrix

$$\boldsymbol{\Sigma}^* = \begin{pmatrix} \mathbf{L}\mathbf{L}' + \boldsymbol{\Psi} & \mathbf{L} \\ \mathbf{L}' & \mathbf{I}_m \end{pmatrix}$$

which implies that the conditional distribution of $\boldsymbol{F}$ given $\boldsymbol{X}$ has

- $E(\boldsymbol{F}|\boldsymbol{X}) = \mathbf{L}'\left(\mathbf{L}\mathbf{L}' + \boldsymbol{\Psi}\right)^{-1}(\boldsymbol{X} - \mu)$
- $V(\boldsymbol{F}|\boldsymbol{X}) = \mathbf{I}_m - \mathbf{L}'\left(\mathbf{L}\mathbf{L}' + \boldsymbol{\Psi}\right)^{-1}\mathbf{L}$

The regression estimate of the factor scores have the form

$$\begin{aligned} \hat{\mathbf{f}}_i &= \hat{\mathbf{L}}'\left(\hat{\mathbf{L}}\hat{\mathbf{L}}' + \hat{\boldsymbol{\Psi}}\right)^{-1}(\mathbf{x}_i - \bar{\mathbf{x}}) \\ &= \left(\mathbf{I}_m + \hat{\mathbf{L}}'\hat{\boldsymbol{\Psi}}^{-1}\hat{\mathbf{L}}\right)^{-1}\hat{\mathbf{L}}'\hat{\boldsymbol{\Psi}}^{-1}(\mathbf{x}_i - \bar{\mathbf{x}}) \end{aligned}$$

Connecting Least Squares and Regression Methods

Note that there is a simple relationship between the weighted least squares estimate and the regression estimate

$$\hat{\mathbf{f}}_i^{(W)} = \left(\hat{\mathbf{L}}'\hat{\mathbf{\Psi}}^{-1}\hat{\mathbf{L}}\right)^{-1}\left(\mathbf{I}_m + \hat{\mathbf{L}}'\hat{\mathbf{\Psi}}^{-1}\hat{\mathbf{L}}\right)\hat{\mathbf{f}}_i^{(R)}$$
$$= \left(\mathbf{I}_m + (\hat{\mathbf{L}}'\hat{\mathbf{\Psi}}^{-1}\hat{\mathbf{L}})^{-1}\right)\hat{\mathbf{f}}_i^{(R)}$$

where $\hat{\mathbf{f}}_i^{(W)}$ and $\hat{\mathbf{f}}_i^{(R)}$ denote the WLS and REG estimates, respectively.

Note that this implies that $\|\hat{\mathbf{f}}_i^{(W)}\|^2 \geq \|\hat{\mathbf{f}}_i^{(R)}\|^2$ where $\|\cdot\|$ denotes the Euclidean norm.

Some Extensions

Oblique Factor Model Assumptions

The oblique FA model assumes the form

$$\boldsymbol{X} = \boldsymbol{\mu} + \mathbf{L}\boldsymbol{F} + \boldsymbol{\epsilon}$$

and adds the assumptions that

- $\boldsymbol{F} \sim (\mathbf{0}, \boldsymbol{\Phi})$, with $\mathrm{diag}(\boldsymbol{\Phi}) = \mathbf{1}_m$, i.e., the latent factors have mean zero, unit variance, and are correlated
- $\boldsymbol{\epsilon} \sim (\mathbf{0}, \boldsymbol{\Psi})$ where $\boldsymbol{\Psi} = \mathrm{diag}(\psi_1, \ldots, \psi_p)$ with ψ_j denoting the j-th specific variance
- ϵ_j and F_k are independent of one another for all pairs j, k

Oblique Factor Model Implied Covariance Structure

The implied covariance structure for $\boldsymbol{X}$ is

$$\begin{aligned}
\mathrm{Var}(\boldsymbol{X}) &= E[(\boldsymbol{X}-\boldsymbol{\mu})(\boldsymbol{X}-\boldsymbol{\mu})'] \\
&= E[(\mathbf{L}\boldsymbol{F}+\boldsymbol{\epsilon})(\mathbf{L}\boldsymbol{F}+\boldsymbol{\epsilon})'] \\
&= E[\mathbf{L}\boldsymbol{F}\boldsymbol{F}'\mathbf{L}'] + E[\mathbf{L}\boldsymbol{F}\boldsymbol{\epsilon}'] + E[\boldsymbol{\epsilon}\boldsymbol{F}'\mathbf{L}'] + E[\boldsymbol{\epsilon}\boldsymbol{\epsilon}'] \\
&= \mathbf{L}E[\boldsymbol{F}\boldsymbol{F}']\mathbf{L}' + \mathbf{L}E[\boldsymbol{F}\boldsymbol{\epsilon}'] + E[\boldsymbol{\epsilon}\boldsymbol{F}']\mathbf{L}' + E[\boldsymbol{\epsilon}\boldsymbol{\epsilon}'] \\
&= \mathbf{L}\boldsymbol{\Phi}\mathbf{L}' + \boldsymbol{\Psi}
\end{aligned}$$

where $E[\boldsymbol{F}\boldsymbol{F}'] = \boldsymbol{\Phi}$, $E[\boldsymbol{F}\boldsymbol{\epsilon}'] = \mathbf{0}_{m\times p}$, $E[\boldsymbol{\epsilon}\boldsymbol{F}'] = \mathbf{0}_{p\times m}$, and $E[\boldsymbol{\epsilon}\boldsymbol{\epsilon}'] = \boldsymbol{\Psi}$.

This implies that the covariance between $\boldsymbol{X}$ and $\boldsymbol{F}$ has the form

$$\begin{aligned}
\mathrm{Cov}(\boldsymbol{X},\boldsymbol{F}) &= E[(\boldsymbol{X}-\boldsymbol{\mu})\boldsymbol{F}'] \\
&= E[(\mathbf{L}\boldsymbol{F}+\boldsymbol{\epsilon})\boldsymbol{F}'] = \mathbf{L}\boldsymbol{\Phi}
\end{aligned}$$

Factor Pattern Matrix and Factor Structure Matrix

For oblique factor models, the following vocabulary are common:

- $\mathbf{L}$ is called the factor pattern matrix
- $\mathbf{L\Phi}$ is called the factor structure matrix

The factor structure matrix $\mathbf{L\Phi}$ gives the covariance between the observed variables in $\boldsymbol{X}$ and the latent factors in $\boldsymbol{F}$.

If the factors are orthogonal, then $\mathbf{\Phi} = \mathbf{I}_m$ and the factor pattern and structure matrices are identical.

Oblique Factor Estimation (or Rotation)

To fit the oblique factor model, exploit the rotational indeterminacy.

- $\mathbf{L}\boldsymbol{F} = \tilde{\mathbf{L}}\tilde{\boldsymbol{F}}$ where $\tilde{\mathbf{L}} = \mathbf{L}\mathbf{T}$ and $\tilde{\boldsymbol{F}} = \mathbf{T}^{-1}\boldsymbol{F}$
- Note that $\mathbf{T}$ is some $m \times m$ nonsingular matrix

Let $\boldsymbol{\Phi} = \mathbf{V}_\phi \boldsymbol{\Lambda}_\phi \mathbf{V}'_\phi$ denote the eigenvalue decomposition of $\boldsymbol{\Phi}$

1. Define $\check{\mathbf{L}} = \mathbf{L}\mathbf{V}_\phi \boldsymbol{\Lambda}_\phi^{1/2}$ and $\check{\mathbf{F}} = \boldsymbol{\Lambda}_\phi^{-1/2}\mathbf{V}'_\phi \boldsymbol{F}$ so that $\boldsymbol{\Sigma} = \check{\mathbf{L}}\check{\mathbf{L}}' + \boldsymbol{\Psi}$
2. Fit orthogonal factor model to estimate $\check{\mathbf{L}}$ and $\boldsymbol{\Psi}$
3. Use oblique rotation method to rotate obtained solution

Popular oblique rotation methods include promax and quartimin.

- `R` package `GPArotation` has many options for oblique rotation

Exploratory versus Confirmatory Factor Analysis

Until now, we have assumed an exploratory factor analysis model, where **L** is just some unknown matrix with no particular form.

- All loadings ℓ_{jk} are freely esimated

In contrast, a confirmatory factor analysis model assumes that the factor loading matrix **L** has some particular structure.

- Some loadings ℓ_{jk} are constrained to zero

Confirmatory Factor Analysis (CFA) is a special type of structural equation modeling (SEM).

Examples of Different Factor Loading Patterns

Table: Possible patterns for loadings with $m = 2$ common factors.

	Unstructured		Discrete		Overlapping	
	$k = 1$	$k = 2$	$k = 1$	$k = 2$	$k = 1$	$k = 2$
$j = 1$	$*$	$*$	$*$	0	$*$	0
$j = 2$	$*$	$*$	$*$	0	$*$	0
$j = 3$	$*$	$*$	$*$	0	$*$	$*$
$j = 4$	$*$	$*$	0	$*$	$*$	$*$
$j = 5$	$*$	$*$	0	$*$	0	$*$
$j = 6$	$*$	$*$	0	$*$	0	$*$

Note. An entry of "$*$" denotes a non-zero factor loading.

Unstructured is EFA, whereas the Discrete and Overlapping are CFA.

Fitting and Evaluating Confirmatory Factor Models

Like EFA models, CFA models can be fit via either least squares or maximum likelihood estimation.

- Least squares is analogue of PCA fitting
- Maximum likelihood assumes multivariate normality

`R` package `sem` can be used to fit CFA models.

Most important part of CFA is evaluating and comparing model fit.

- Many fit indices exist for examining quality of CFA solution
- Should focus on cross-validation when comparing models

Decathlon Example

Men's Olympic Decathlon Data from 1988

Data from men's 1988 Olympic decathlon

- Total of $n = 34$ athletes
- Have $p = 10$ variables giving score for each decathlon event
- Have overall decathlon score also (`score`)

```
> decathlon[1:9,]
         run100 long.jump  shot high.jump run400 hurdle discus pole.vault javelin run1500 score
Schenk    11.25      7.43 15.48      2.27  48.90  15.13  49.28        4.7   61.32  268.95  8488
Voss      10.87      7.45 14.97      1.97  47.71  14.46  44.36        5.1   61.76  273.02  8399
Steen     11.18      7.44 14.20      1.97  48.29  14.81  43.66        5.2   64.16  263.20  8328
Thompson  10.62      7.38 15.02      2.03  49.06  14.72  44.80        4.9   64.04  285.11  8306
Blondel   11.02      7.43 12.92      1.97  47.44  14.40  41.20        5.2   57.46  256.64  8286
Plaziat   10.83      7.72 13.58      2.12  48.34  14.18  43.06        4.9   52.18  274.07  8272
Bright    11.18      7.05 14.12      2.06  49.34  14.39  41.68        5.7   61.60  291.20  8216
De.Wit    11.05      6.95 15.34      2.00  48.21  14.36  41.32        4.8   63.00  265.86  8189
Johnson   11.15      7.12 14.52      2.03  49.15  14.66  42.36        4.9   66.46  269.62  8180
```

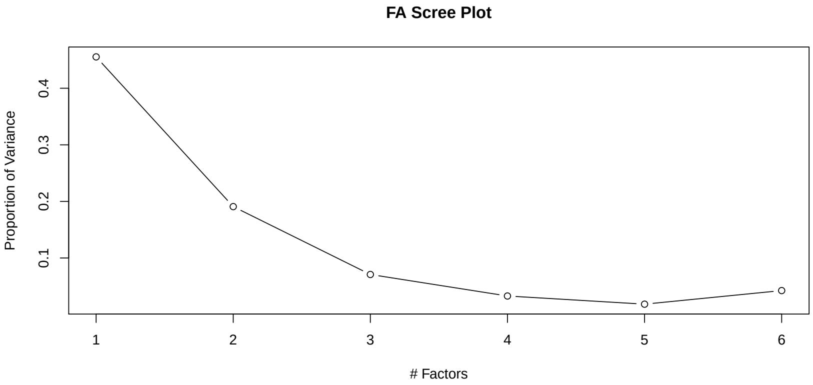
Resigning Running Events

For the running events (`run100`, `run400`, `run1500`, and `hurdle`), lower scores correspond to better performance, whereas higher scores represent better performance for other events.

To make interpretation simpler, we will resign the running events:

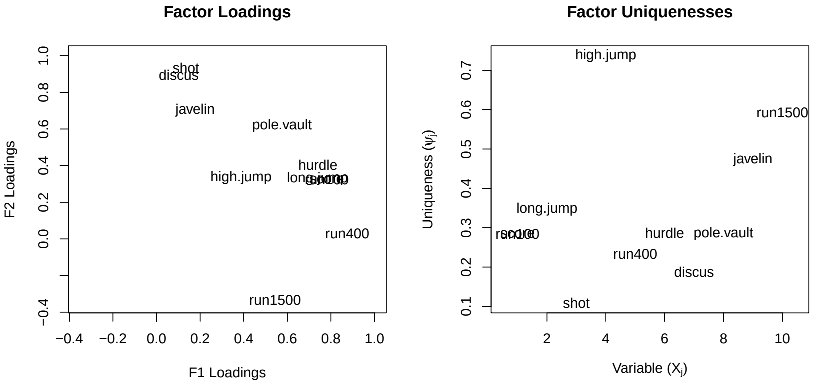
```
> decathlon[,c(1,5,6,10)] <- (-1)*decathlon[,c(1,5,6,10)]
> decathlon[1:9,]
          run100 long.jump  shot high.jump run400 hurdle discus pole.vault javelin run1500 score
Schenk    -11.25      7.43 15.48      2.27 -48.90 -15.13  49.28        4.7   61.32 -268.95  8488
Voss      -10.87      7.45 14.97      1.97 -47.71 -14.46  44.36        5.1   61.76 -273.02  8399
Steen     -11.18      7.44 14.20      1.97 -48.29 -14.81  43.66        5.2   64.16 -263.20  8328
Thompson  -10.62      7.38 15.02      2.03 -49.06 -14.72  44.80        4.9   64.04 -285.11  8306
Blondel   -11.02      7.43 12.92      1.97 -47.44 -14.40  41.20        5.2   57.46 -256.64  8286
Plaziat   -10.83      7.72 13.58      2.12 -48.34 -14.18  43.06        4.9   52.18 -274.07  8272
Bright    -11.18      7.05 14.12      2.06 -49.34 -14.39  41.68        5.7   61.60 -291.20  8216
De.Wit    -11.05      6.95 15.34      2.00 -48.21 -14.36  41.32        4.8   63.00 -265.86  8189
Johnson   -11.15      7.12 14.52      2.03 -49.15 -14.66  42.36        4.9   66.46 -269.62  8180
```

Factor Analysis Scree Plot



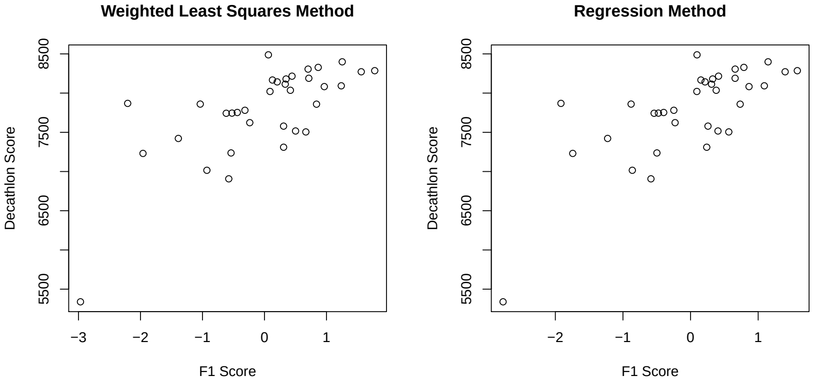

```
famods <- vector("list", 6)
for(k in 1:6)  famods[[k]] <- factanal(x=decathlon[,1:10], factors=k)
vafs <- sapply(famods, function(x) sum(x$loadings^2)) / nrow(famods[[1]]$loadings)
vaf.scree <- vafs - c(0, vafs[1:5])
plot(1:6, vaf.scree, type="b", xlab="# Factors",
     ylab="Proportion of Variance", main="FA Scree Plot")
```

FA Loadings: $m = 2$ Common Factors



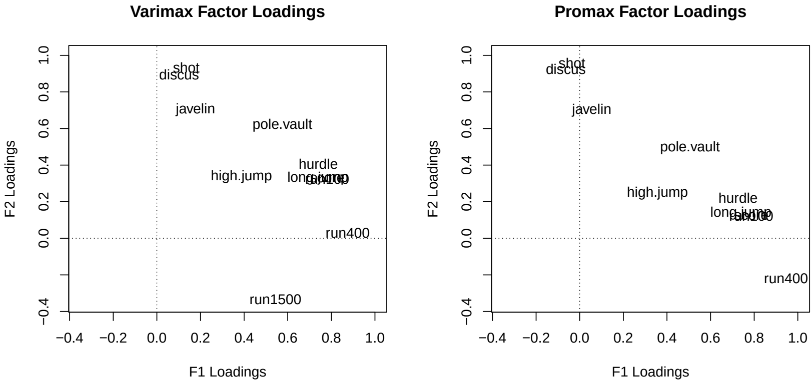

```
> famod <- factanal(x=decathlon[,1:10], factors=2)
> names(famod)
 [1] "converged"     "loadings"      "uniquenesses" "correlation"
 [5] "criteria"      "factors"       "dof"          "method"
 [9] "rotmat"        "STATISTIC"     "PVAL"         "n.obs"
[13] "call"
```

FA Scores: $m = 2$ Common Factors




```
> # refit model and get FA scores (NOT GOOD IDEA!!)
> famodW <- factanal(x=decathlon[,1:10], factors=2, scores="Bartlett")
> famodR <- factanal(x=decathlon[,1:10], factors=2, scores="regression")
> round(cor(decathlon$score, famodR$scores), 4)
     Factor1 Factor2
[1,]  0.7336  0.6735
> round(cor(decathlon$score, famodW$scores), 4)
     Factor1 Factor2
[1,]  0.7098  0.6474
```

FA with Oblique (Promax) Rotation




```
> famod.promax <- promax(famod$loadings)
> tcrossprod(solve(famod.promax$rotmat))     # correlation between rotated factor scores
          [,1]      [,2]
[1,] 1.0000000 0.4262771
[2,] 0.4262771 1.0000000
```

FA with Oblique (Promax) Rotation, continued

```
# compare loadings
> oldFAloadings <- famod$loadings
> newFAloadings <- famod$loadings %*% famod.promax$rotmat
> sum((newFAloadings - famod.promax$loadings)^2)
[1] 1.101632e-31

# compare reproduced data before and after rotation
> oldFAscores <- famodR$scores
> newFAscores <- oldFAscores %*% t(solve(famod.promax$rotmat))
> Xold <- tcrossprod(oldFAscores, oldFAloadings)
> Xnew <- tcrossprod(newFAscores, newFAloadings)
> sum((Xold - Xnew)^2)
[1] 3.370089e-30

# population and sample factor score covariance matrix (after rotation)
> tcrossprod(solve(famod.promax$rotmat))     # population
          [,1]      [,2]
[1,] 1.0000000 0.4262771
[2,] 0.4262771 1.0000000
> cor(newFAscores)                            # sample
          [,1]      [,2]
[1,] 1.0000000 0.4563499
[2,] 0.4563499 1.0000000
```