Factor Analysis

Max Turgeon

STAT 4690–Applied Multivariate Analysis

Latent variable models

- With PCA, we saw how we could reduce the dimension of data using the eigenvectors of the sample covariance matrix.
- Conversely, we could construe PCA has a generative model, where the principal components give rise to the observed data.
- **Latent Variable Models** formalise this idea:
    - Latent (i.e. unobserved) variables $\mathbf{F}$ give rise to observed data $\mathbf{Y}$ through a *specified* model.

Factor Analysis i

- **Factor Analysis** is a special kind of latent variable model.
- Let $\mathbf{Y}$ be a p-dimensional vector with mean μ and covariance matrix Σ.
- Let $\mathbf{F}$ be a m-dimensional *latent* vector.
- The *orthogonal factor model* is given by

$$\mathbf{Y} - \mu = L\mathbf{F} + \mathbf{E},$$

where L is a $p \times m$ *matrix of factor loadings*, and $\mathbf{E}$ is a p-dimensional vector of *errors*.

Factor Analysis ii

- $\mathbf{F}$ are also called *common factors*; $\mathbf{E}$ are also called *specific factors*.
- **Note**: This is essentially a multivariate regression model, but where the covariates are unobserved.

Assumptions i

- The model above is generally not identifiable, since there are too many parameters.
- We therefore need to impose further restrictions:
    - $E(\mathbf{F}) = 0$
    - $\mathrm{Cov}(\mathbf{F}) = I$
    - $E(\mathbf{E}) = 0$
    - $\mathrm{Cov}(\mathbf{E}) = \Psi = \begin{pmatrix} \psi_1 & 0 & \cdots & 0 \\ 0 & \psi_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \psi_p \end{pmatrix}$
    - $\mathrm{Cov}(\mathbf{F}, \mathbf{E}) = 0$

Assumptions ii

- In other words:
- Both common and specific factors have mean zero;
- They are uncorrelated;
- The common factors are mutually uncorrelated and standardised;
- The specific factors each affect only one observed variable.

Structured Covariance i

- As a consequence of these assumptions, we can derive an assumption on the structure of $\Sigma = \mathrm{Cov}(\mathbf{Y})$:

$$\begin{aligned}
\Sigma &= E\left((\mathbf{Y}-\mu)(\mathbf{Y}-\mu)^T\right)\\
&= E\left((L\mathbf{F}+\mathbf{E})(L\mathbf{F}+\mathbf{E})^T\right)\\
&= LE(\mathbf{F}\mathbf{F}^T)L + E(\mathbf{E}\mathbf{F}^T)L^T + LE(\mathbf{F}\mathbf{E}^T) + E(\mathbf{E}\mathbf{E}^T)\\
&= LIL^T + 0L^T + L0 + \Psi\\
&= LL^T + \Psi.
\end{aligned}$$

Structured Covariance ii

- Similarly, we can show that

$$\mathrm{Cov}(\mathbf{Y}, \mathbf{F}) = L.$$

- If we write ℓ_{ij} for the (i, j)-th element of L, we see that

$$\mathrm{Var}(Y_i) = \sum_{k=1}^{m} \ell_{ik}^2 + \psi_i.$$

- Crucially, these assumptions are **testable**. In other words, we can check whether they are reasonable for our data.

Example i

- Let's look at an example where there is no solution.
- Assume $p = 3$, $m = 1$, with

$$\Sigma = \begin{pmatrix} 1 & 0.9 & 0.7 \\ 0.9 & 1 & 0.4 \\ 0.7 & 0.4 & 1 \end{pmatrix}.$$

- From our assumptions on the covariance structure, we derive a system of equations

$$\begin{aligned} 1 = \ell_{11}^2 + \psi_1 \quad 0.9 &= \ell_{11}\ell_{21} \quad 0.7 = \ell_{11}\ell_{31} \\ 1 &= \ell_{22}^2 + \psi_2 \quad 0.4 = \ell_{21}\ell_{31} \\ & \qquad\qquad\quad 1 = \ell_{33}^2 + \psi_3 \end{aligned}$$

Example ii

- From $0.7 = \ell_{11}\ell_{31}$ and $0.4 = \ell_{21}\ell_{31}$, we get

$$\ell_{21} = \frac{0.4}{0.7}\ell_{11}.$$

- But since $0.9 = \ell_{11}\ell_{21}$, we can conclude that

$$\ell_{11} = \pm 1.255.$$

- However, since the first component Y_1 has unit variance, $\ell_{11} = \mathrm{Corr}(Y_1, F_1)$, and therefore the correlation is out of bounds.
- Similarly, we get

$$\psi_1 = 1 - \ell_{11}^2 = 1 - 1.575 = -0.575.$$

Example iii

- But since ψ_1 is the variance of the first error term, we once again get a non-sensical solution.

Factor Rotation i

- Even with our assumptions above, our model is still not uniquely identified.
- Let T be an $m \times m$ orthogonal matrix. We have

$$\begin{aligned}\mathbf{Y} - \mu &= L\mathbf{F} + \mathbf{E} \\ &= LTT^T\mathbf{F} + \mathbf{E} \\ &= \tilde{L}\tilde{\mathbf{F}} + \mathbf{E},\end{aligned}$$

where $\tilde{L} = LT$ and $\tilde{\mathbf{F}} = T^T\mathbf{F}$.

Factor Rotation ii

- Both models lead to the same covariance matrix:

$$\Sigma = LL^T + \Psi = LTT^T L^T + \Psi = \tilde{L}\tilde{L}^T + \Psi.$$

- As we will see, this will turn out to be a blessing in disguise:
  - We will impose a uniqueness condition to get one solution.
  - Then we will rotate our solution using T to improve interpretation.

Estimation–Principal Component Method i

- Recall the spectral decomposition of the covariance matrix:
$$\Sigma = \sum_{i=1}^{p} \lambda_i w_i w_i^T,$$
with $\lambda_1 \geq \cdots \geq \lambda_p$.
- If we let W be the matrix whose i-th column is $\sqrt{\lambda_i} w_i$, we can rewrite the spectral decomposition as
$$\Sigma = WW^T.$$
- In other words, if we let $m = p$ and $\Psi = 0$, we see that we recover the orthogonal factor model with $L = W$.

Estimation–Principal Component Method ii

- Of course, this is not very satisfactory, as the dimension of the common factors is the same as that of the original data.
- Instead, we select $m < p$ using one of the methods we discussed with PCA and we approximate

$$\Sigma \approx \sum_{i=1}^{m} \lambda_i w_i w_i^T.$$

- If we let L be the $p \times m$ matrix whose i-th column is $\sqrt{\lambda_i} w_i$, we can estimate Ψ as follows:

$$\psi_i = \sigma_{ii} - \sum_{j=1}^{m} \ell_{ij}^2.$$

Estimation–Principal Component Method iii

Algorithm

1. Let $\hat{\lambda}_1 \cdots > \hat{\lambda}_p$ and $\hat{w}_1, \ldots, \hat{w}_p$ be the eigenvalues and eigenvectors of the covariance matrix S_n.
2. Select m using one of the PCA criteria.
3. Estimate $\hat{L}$ with the $p \times m$ matrix whose i-th column is $\sqrt{\hat{\lambda}_i}\hat{w}_i$.
4. Estimate $\hat{\Psi}$ with the diagonal elements of $S_n - \hat{L}\hat{L}^T$.

Example i

```r
library(psych)

dim(bfi)
```

```
## [1] 2800   28
```

```r
names(bfi)
```

Example ii

```
##  [1] "A1"        "A2"        "A3"        "A4"
##  [6] "C1"        "C2"        "C3"        "C4"
## [11] "E1"        "E2"        "E3"        "E4"
## [16] "N1"        "N2"        "N3"        "N4"
## [21] "O1"        "O2"        "O3"        "O4"
## [26] "gender"    "education" "age"
```

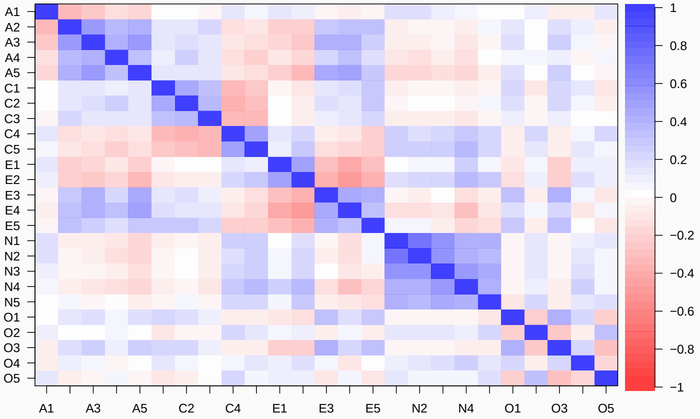
```r
library(tidyverse)

data <- bfi %>%
  select(-gender, -education, -age) %>%
  filter(complete.cases(.))
```

Example iii

```
cor.plot(cor(data))
```

Example iv


A1
A2
A3
A4
A5
C1
C2
C3
C4
C5
E1
E2
E3
E4
E5
N1
N2
N3
N4
N5
O1
O2
O3
O4
O5
A1
A3
A5
C2
C4
E1
E3
E5
N2
N4
O1
O3
O5
1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1

Example v

```r
decomp <- prcomp(data)
summary(decomp)$importance[,1:3]
```

```
##                             PC1      PC2      PC3
## Standard deviation     3.291635 2.451538 2.030393
## Proportion of Variance 0.215650 0.119620 0.082050
## Cumulative Proportion  0.215650 0.335270 0.417320
```

Example vi

```
cum_prop <- decomp %>%
  summary %>%
  .[["importance"]] %>%
  .["Cumulative Proportion",]

which(cum_prop > 0.8)[1]
```

```
## PC14
##   14
```

Example vii

```r
Lhat <- decomp$rotation[,1:14] %*%
  diag(decomp$sdev[1:14])
Psi_hat <- diag(cov(data) - tcrossprod(Lhat))
```

```r
# Sum squared error
sum((cov(data) - tcrossprod(Lhat) - diag(Psi_hat))^2)
```

```
## [1] 3.645694
```

```r
# Compare to the total variance
sum(diag(cov(data)))
```

Example viii

```
## [1] 50.24287
```

```r
# Our FA model explains:
sum(colSums(Lhat^2)/sum(diag(cov(data))))
```

```
## [1] 0.8160488
```

Comments

- In our example above, we saw that 14 factors explained 82% of the total variance.
- Two common default values for m in statistical softwares:
  - Number of positive eigenvalues of the sample covariance matrix.
  - Number of eigenvalues greater than one for the sample correlation matrix.
- In our example, the first criterion would lead to $m = p$, which is not very helpful

Example (cont'd) i

```
(m <- sum(eigen(cor(data))$values > 1))
```

```
## [1] 6
```

```
Lhat <- decomp$rotation[,seq_len(m)] %*%
  diag(decomp$sdev[seq_len(m)])
Psi_hat <- diag(cov(data) - tcrossprod(Lhat))

# Sum squared error
sum((cov(data) - tcrossprod(Lhat) - diag(Psi_hat))^2)
```

Example (cont'd) ii

```
## [1] 6.999035
```

```r
# Compare to the total variance
sum(diag(cov(data)))
```

```
## [1] 50.24287
```

```r
# Our FA model explains:
sum(colSums(Lhat^2)/sum(diag(cov(data))))
```

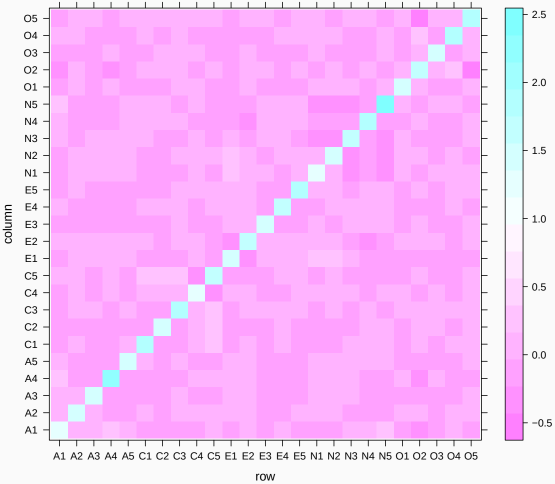
```
## [1] 0.5910586
```

Example (cont'd) iii

```r
# We can also  visualize the fit
Sn <- cov(data)
Sn_fit <- tcrossprod(Lhat) - diag(Psi_hat)

library(lattice)
levelplot(Sn - Sn_fit)
```

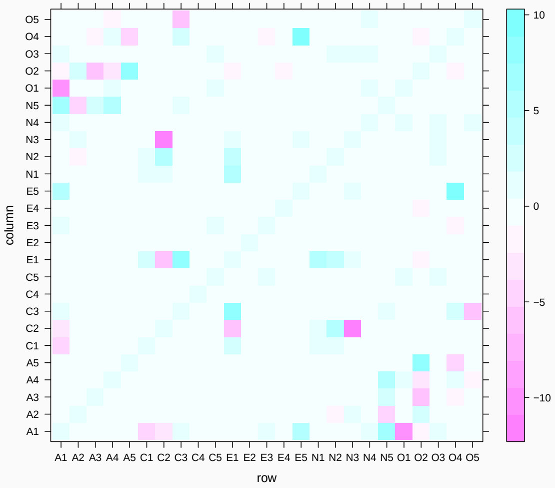
Example (cont'd) iv



Example (cont'd) v

```r
# Or if you prefer % difference
levelplot((Sn - Sn_fit)/Sn)
```

Example (cont'd) vi



Estimation–Maximum Likelihood Method i

- This estimation method assumes that both common factors $\mathbf{F}$ and specific factors $\mathbf{E}$ follow a multivariate normal distribution
    - $\mathbf{F} \sim N_m(0, I)$
    - $\mathbf{E} \sim N_p(0, \Psi)$
- From this assumption, it follows that $\mathbf{Y}$ also follows a multivariate normal distribution
    - $\mathbf{Y} \sim N_p(\mu, LL^T + \Psi)$
- Therefore, we can write down the likelihood in terms of both L and Ψ.

Estimation–Maximum Likelihood Method ii

- However, because of the factor rotation problem, we need to impose a constraint in order to obtain a unique solution:
    - $L^T\Psi^{-1}L = \Delta$ is diagonal.
- Given this assumption, the maximum likelihood estimates $\hat{L}$ and $\hat{\Psi}$ can be found using an iterative algorithm.
- We will not go into the details of the algorithm (but see Johnson & Wichern, Supplement 9A if interested).
    - Instead, we will rely on the `R` function `stats::factanal`.

Example (cont'd) i

```r
# CAREFUL: uses correlation matrix
fa_decomp <- factanal(data, factors = m,
                      rotation = 'none')

# Uniquenesses are the diagonal elements
# of the matrix Psi
Psi_mle <- fa_decomp$uniquenesses
Lmle <- fa_decomp$loadings
```

Example (cont'd) ii

```r
# We get an estimate of the correlation
R_mle <- tcrossprod(Lmle) - diag(Psi_mle)

sum((cor(data) - R_mle)^2)
```

```
## [1] 31.97525
```

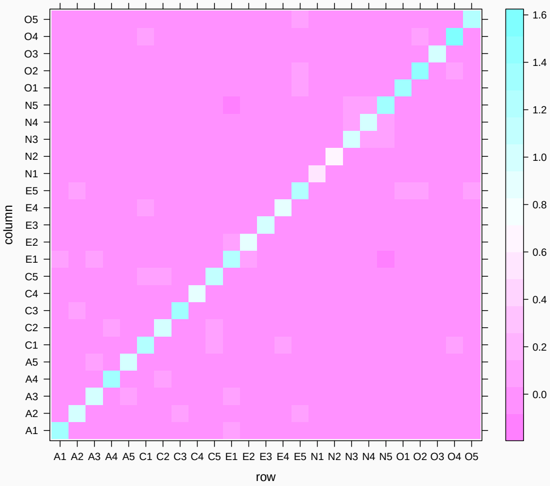
```r
# Our FA model explains:
sum(colSums(Lmle^2)/ncol(data))
```

```
## [1] 0.4498342
```

Example (cont'd) iii

```r
levelplot(cor(data) - R_mle)
```

Example (cont'd) iv



Example (cont'd) v

```r
# To factor the covariance matrix
# Use psych::fa
fa_decomp <- psych::fa(data, nfactors = m,
                       rotate = "none",
                       covar = TRUE,
                       fm = "ml")

# Extract estimates
Psi_mle <- fa_decomp$uniquenesses
Lmle <- fa_decomp$loadings
```

Example (cont'd) vi

```r
# We get an estimate of the covariance
Sn_mle <- tcrossprod(Lmle) - diag(Psi_mle)

sum((Sn - Sn_mle)^2)
```

```
## [1] 128.025
```

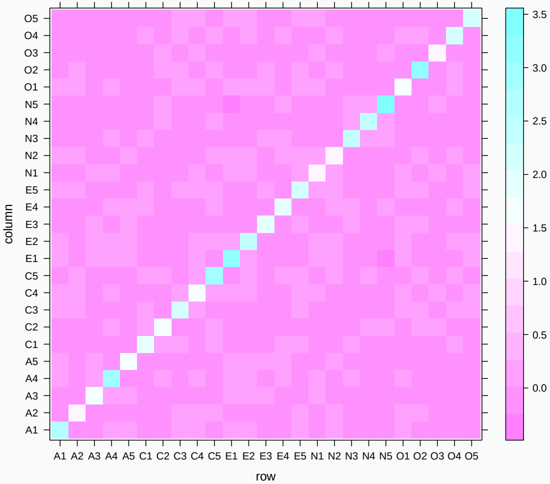
```r
# Our FA model explains:
sum(colSums(Lmle^2)/sum(diag(Sn)))
```

```
## [1] 0.459665
```

Example (cont'd) vii

```
levelplot(Sn - Sn_mle)
```

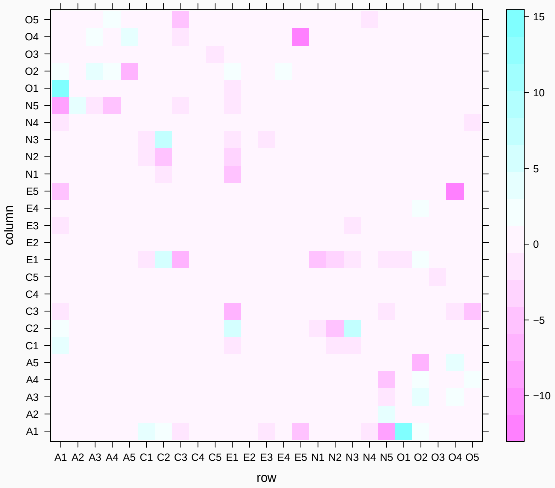
Example (cont'd) viii



Example (cont'd) ix

```r
# Compare MLE with PC estimate
levelplot((Sn_fit - Sn_mle)/Sn)
```

Example (cont'd) x



Comments about estimation i

- There are other methods of estimating the loadings and the "uniquenesses"
  - Ordinary Least Squares
  - Weighted OLS
  - Principal factor
- With so many choices of estimation methods, it can be hard to compare statistical softwares
  - And you also have to read the manual in order to know what is going on...
- You also always have the choice between factoring the covariance or the correlation matrix

Comments about estimation ii

    - Which one you choose depends on the commensurability of your variables (just like in PCA)
- Finally, it's always a good idea to compare the output of multiple estimation strategies
    - If your model is a good fit, you should get a similar answer regardless of the method.

Factor Rotation Redux i

- As we saw earlier, any orthgonal matrix T gives rise to the same factor analysis model

$$\Sigma = LL^T + \Psi = LTT^T L^T + \Psi = \tilde{L}\tilde{L}^T + \Psi.$$

- In other words, we cannot choose T to maximise the goodness of fit.
    - We need another criterion
- Intuitively, to ease interpretation, we want each variable to have large loadings for one factor and negligeable loadings for the other ones.

Factor Rotation Redux ii

    - `https://maxturgeon.ca/f19-stat4690/factor_rotation.gif`
- One common analytic criterion that formalises this idea is the **varimax criterion**.
    - Resulting loadings are called *varimax loadings*
- We have

$$\mathrm{VARIMAX} \propto \sum_{j=1}^{m} \begin{pmatrix} \text{Variance of squares of scales loadings} \\ \text{for } j\text{-th factor} \end{pmatrix}$$

- More precisely:
    - Let $\tilde{\ell}_{ij}$ be the (i,j)-th entry of the matrix $\tilde{L} = LT$. In other words, $\tilde{\ell}_{ij}$ depends on T.

Factor Rotation Redux iii

- Let $\tilde{h}_i^2 = \sum_{j=1}^m \tilde{\ell}_{ij}^2$.
- Define the scaled loadings $\tilde{\ell}_{ij}^* = \tilde{\ell}_{ij}/h_i$.
- The varimax criterion V is given as

$$V = \frac{1}{p}\sum_{j=1}^m \left(\sum_{i=1}^p \tilde{\ell}_{ij}^{*4} - \frac{1}{p}\left(\sum_{i=1}^p \tilde{\ell}_{ij}^{*2}\right)^2\right).$$

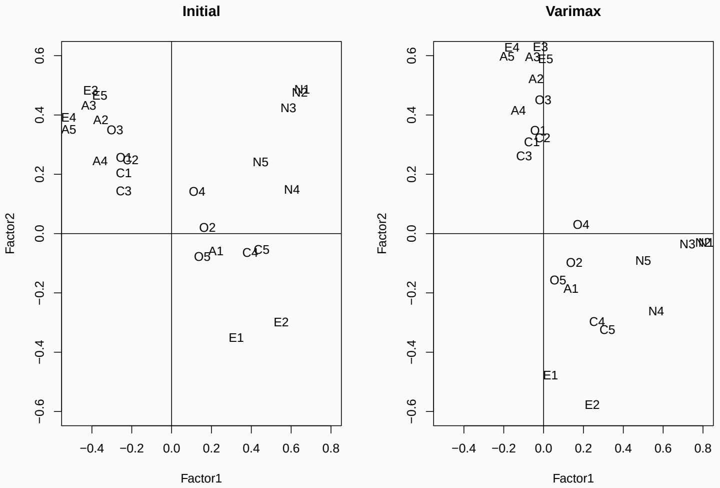
- In R, you can compute the rotated loadings using the `stats::varimax` function. Alternatively, the function `stats::factanal` can compute the rotation for you as part of the factor analysis (and so does `psych::fa`).

Example (cont'd) i

```r
# Let's start with m=2 for visualization
fa_decomp <- factanal(data, factors = 2,
                      rotation = 'none')

initial_loadings <- fa_decomp$loadings
varimax_loadings <- varimax(initial_loadings)
```

Example (cont'd) ii



Example (cont'd) iii

```r
# You can extract the matrix T
varimax_loadings$rotmat
```

```
##             [,1]       [,2]
## [1,] 0.7810310 -0.6244923
## [2,] 0.6244923  0.7810310
```

```r
# We can also get the angle of rotation
acos(varimax_loadings$rotmat[1,1])
```

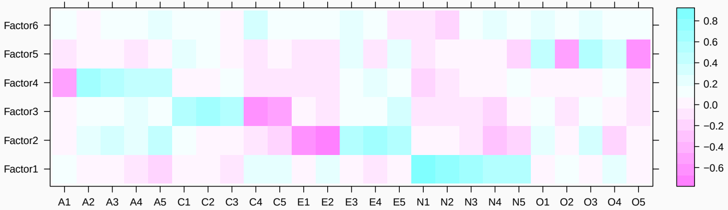
```
## [1] 0.6744813
```

Example (cont'd) iv

```r
# In more dimensions
fa_decomp <- factanal(data, factors = m,
                      rotation = 'varimax')

levelplot(unclass(fa_decomp$loadings),
          xlab = "", ylab = "")
```

Example (cont'd) v



Comments

- As with estimation, there are many more rotation methods.
  - See for example the help page `?GPArotation::rotations`
- One particular class of rotations are called *oblique*
  - The matrix T is no longer constrained to be orthogonal.
- Factor rotation is especially useful with loadings obtained through MLE
  - Recall the constraint on $L^T \Psi^{-1} L$ being diagonal
- Factor rotations are also sometimes used with PCA loadings.