Getting Started with Exploratory Factor Analysis

Clay Ford

Spring 2018

Agenda

- Conceptual overview of Exploratory Factor Analysis (EFA)
- How to implement in R
- How to interpret
- How to visualize
- Tips and warnings along the way

A review of covariance and correlation

- EFA involves modeling a covariance or correlation matrix
- Covariance: measure of linear association between two variables
- Correlation: standardized measure of linear association between two variables
- Positive values mean a positive relationship (as one increases, so does the other)
- Negative values mean a negative relationship (as one increases, the other decreases)
- Covariances and correlations are usually displayed in a *matrix*

Example of covariance matrix

Ability and Intelligence Tests (see `?ability.cov` in R)

	general	picture	blocks	maze	reading	vocab
general	24.641	5.991	33.520	6.023	20.755	29.701
picture	5.991	6.700	18.137	1.782	4.936	7.204
blocks	33.520	18.137	149.831	19.424	31.430	50.753
maze	6.023	1.782	19.424	12.711	4.757	9.075
reading	20.755	4.936	31.430	4.757	52.604	66.762
vocab	29.701	7.204	50.753	9.075	66.762	135.292

- Notice the matrix is square (equal number of rows and columns)
- Notice the matrix is symmetric (lower left and top right values are equal)
- The diagonal values (top left corner to bottom right corner) are variances
- The off-diagonal values are covariances

Example of correlation matrix

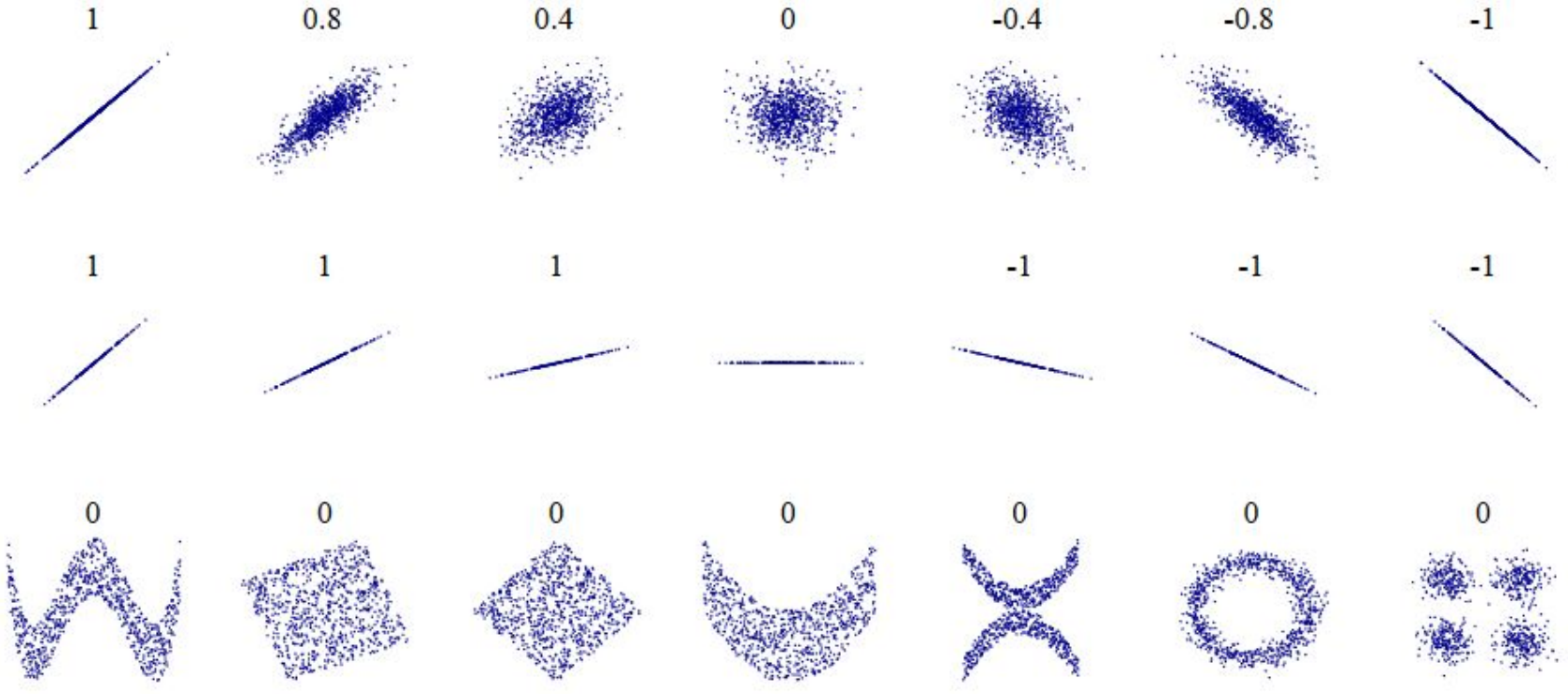
Same data as previous slide but now expressed as correlations.

	general	picture	blocks	maze	reading	vocab
general	1.000	0.466	0.552	0.340	0.576	0.514
picture	0.466	1.000	0.572	0.193	0.263	0.239
blocks	0.552	0.572	1.000	0.445	0.354	0.356
maze	0.340	0.193	0.445	1.000	0.184	0.219
reading	0.576	0.263	0.354	0.184	1.000	0.791
vocab	0.514	0.239	0.356	0.219	0.791	1.000

- Correlations range from -1 to 1
- Correlation of 1 means a perfectly positive linear relationship
- Correlation of -1 means a perfectly negative linear relationship
- Correlation of 0 means no linear relationship
- There are 1's on the diagonal since since variables are perfectly correlated with themselves

Be cautious with correlation

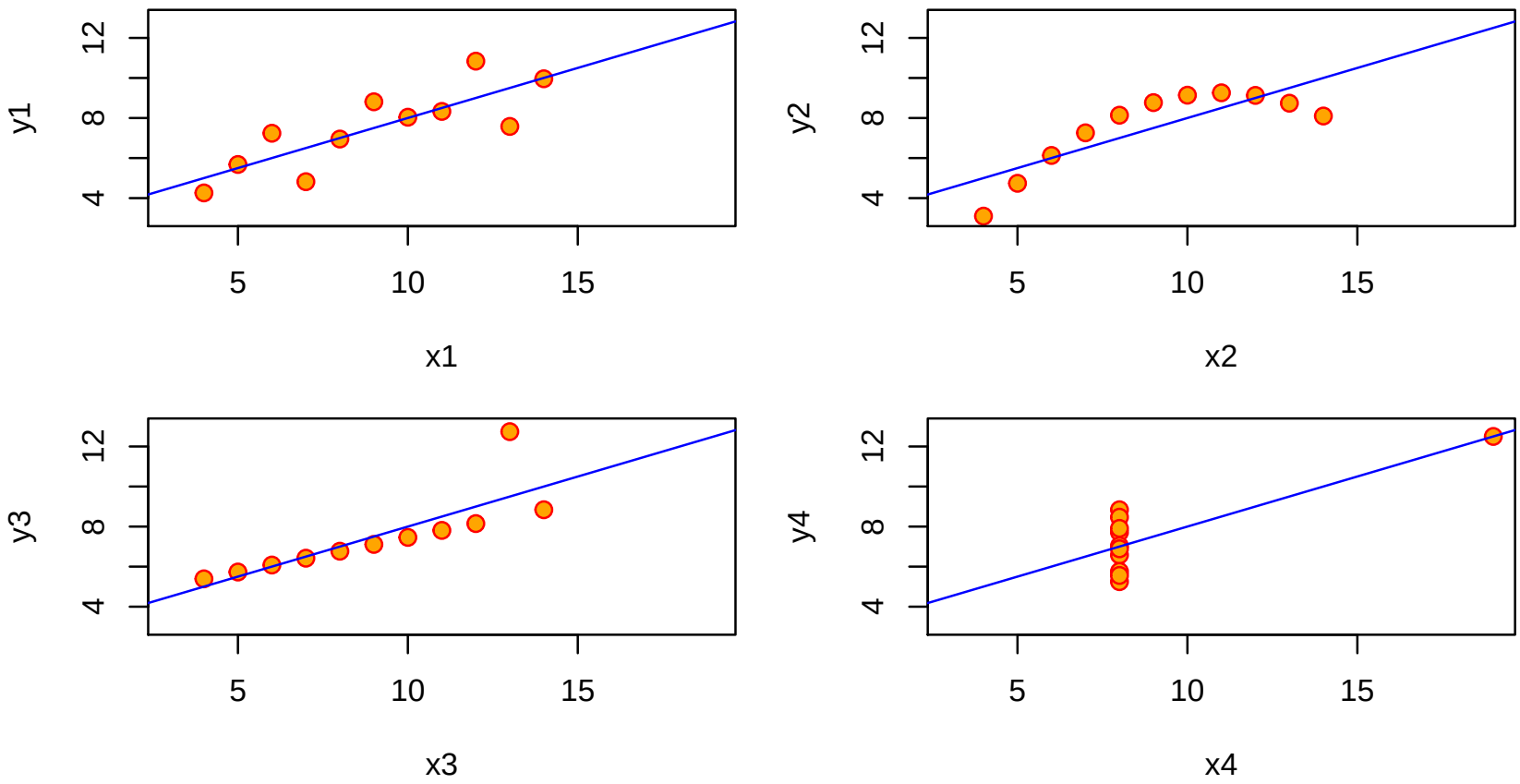
Correlation measures strength of *linear* association. Low correlation doesn't always mean "no relationship". Below: data on bottom row has a relationship not captured with correlation.



Source: https://en.wikipedia.org/wiki/Correlation_and_dependence

Be cautious with correlation

Correlation measures strength of *linear* association. High correlation doesn't always mean "linear relationship". Below: Four sets of data with the same correlation of 0.816.



Groups of correlations

- Let's say we have a correlation matrix with groups of variables that are highly correlated among themselves but not so much with variables in a different group.
- Then perhaps each group of variables represent a single underlying construct, or *factor*, that is responsible for the observed correlations?

	general	picture	blocks	maze	reading	vocab
general	1.000	0.466	0.552	0.340	0.576	0.514
picture	0.466	1.000	0.572	0.193	0.263	0.239
blocks	0.552	0.572	1.000	0.445	0.354	0.356
maze	0.340	0.193	0.445	1.000	0.184	0.219
reading	0.576	0.263	0.354	0.184	1.000	0.791
vocab	0.514	0.239	0.356	0.219	0.791	1.000

And now to Exploratory Factor Analysis

- EFA attempts to describe, *if possible*, the covariance/correlation relationships among many variables in terms of a few underlying, *but unobservable*, random quantities called *factors*. (Johnson and Wichern, 2007)
- On the previous slide it appeared that two *latent factors* could be responsible for the groups of correlations within the matrix
- EFA helps us investigate the possibility that there are one or more factors generating our covariance matrix

EFA models the covariance matrix

- EFA says we can use the following matrix algebra formula to model our covariance matrix:

$$\Sigma = LL' + \Psi$$

- Σ (sigma) is our covariance matrix
- L is a matrix of unobserved factors, called *loadings*. It will have the same number of rows as our covariance matrix but *fewer columns*. The number of columns is the number of factors.
- Ψ (psi, pronounced "sigh") are the variances unique to each variable in our covariance matrix (ie, error). These are called *uniquenesses*.
- EFA estimates L and Ψ for a specified number of groups

Performing EFA in R

- We can carry out EFA in R using the `factanal` function
- The `psych` package also provides the `fa` function that has a few more options
- The most basic usage is to give the functions a correlation matrix and specify the number of factors
- For example, say we have a correlation matrix called `cor_matrix` and we want to perform an EFA for 2 factors:
  - `factanal(covmat = cor_matrix, factors = 2)`
  - `fa(r = cor_matrix, nfactors = 2)`

What `factanal` returns

```
Uniquenesses:
general picture  blocks     maze reading   vocab
  0.455   0.589   0.218    0.769   0.052   0.334

Loadings:
        Factor1 Factor2
general 0.499   0.543
picture 0.156   0.622
blocks  0.206   0.860
maze    0.109   0.468
reading 0.956   0.182
vocab   0.785   0.225

                Factor1 Factor2
ss loadings       1.858   1.724
Proportion Var    0.310   0.287
Cumulative Var    0.310   0.597
```

What did we just look at?

- The *uniquenesses* are unexplained variability (0,1); we hope they're small, say less than 0.3
- The *loadings* are the variables' correlation with the unobserved factors; we hope they're large on some factors and small on the rest
- On the previous slide the first factor could be interpreted as "verbal comprehension" while the second could be "spatial reasoning"
- *SS Loadings* are the loadings squared and then summed; old-fashioned rule-of-thumb: "keep" a factor if SS Loadings $>$ 1
- *Proportion Var* = *SS Loadings* / # *of vars*
- *Cumulative Var* summarizes how well the loadings are summarizing the original covariance matrix; Cumulative Var of 0.597 says the two factors summarize about 60% of the covariance matrix

Looking as residuals after using `factanal`

- We can use our EFA results to created an estimated covariance matrix: $\hat{\Sigma} = \hat{L}\hat{L}' + \hat{\Psi}$
- We can then subtract the estimated covariance matrix from the observed covariance matrix to get residuals: $\Sigma - \hat{\Sigma}$
- Lots of small residuals mean our EFA model is doing a good job of modeling the observed covariance matrix

```
f.out <- factanal(covmat = cor_matrix, factors = 2)
L <- f.out$loadings
Psi <- diag(f.out$uniquenesses)
# calculate residuals
cor_matrix - (L %*% t(L) + Psi)
```

Looking at residuals after using `fa`

- The `fa` function calculates residuals for us, but does it a little differently
- It does not add the uniquenesses when fitting the estimated covariance matrix
- Hence the uniquenesses are on the diagonal

```
library(psych)
fa.out <- fa(r = cor_matrix, nfactors = 2)
residuals(fa.out)
```

- Let's go to R!

How many factors?

- This is probably the most important decision to make when doing EFA
- According to Johnson and Wichern (2007), the decision is typically made based on some combination of
  1. proportion of variance explained
  2. subject-matter knowledge
  3. "reasonableness" of the results
- We'll discuss a few other statistically motivated options later in the workshop

Estimation and Rotation

- In the `R` script we noticed that `factanal` and `fa` returned two different answers
- They each use different default *estimation* and *rotation* methods
- Estimation refers to how the loadings and uniquenesses are estimated
- Rotation refers to multiplying the loadings by a "rotation" matrix, that helps clarify the structure of the loadings matrix (ie, easier to interpret)

More on estimation

- `factanal` uses *maximum likelihood estimation*. This assumes the latent factors and uniquenesses are multivariate normal. This is the only estimation option for `factanal`.
- `fa` uses the *minimum residual* algorithm. It also provides several other estimation procedures, including maximum likelihood. Specify using the `fm` argument.
- According to `fa` documentation: "There are many ways to do factor analysis, and maximum likelihood procedures are probably the most commonly preferred."
- Johnson and Wichern (2007) recommend maximum likelihood approach.
- If the factor model is appropriate ($\Sigma = LL' + \Psi$), then it doesn't really matter which estimation method you use; they should all produce consistent results.

Maximum likelihood estimation

- MLE allows us to perform a chi-square hypothesis test for the number of factors
- The null hypothesis: number of specified factors is sufficient to model the observed covariance matrix
- A high p-value provides evidence in support of the null
- A low p-value (say less than 0.05) provides evidence against the null
- Both `factanal` and `fa` conduct the test; must specify the number of observations
  - `factanal(covmat = ability.cor, factors = 2, n.obs = 112)`
  - `fa(r = ability.cor, nfactors = 2, fm = "mle", n.obs = 112)`
- Beware: test is sensitive to number of subjects; more subjects leads to lower p-values and thinking you need more factors than you really do

More on rotation

- Johnson and Wichern (2007) liken rotation to "sharpening the focus of a microscope" to see more detail
- Loadings that have been rotated give the same representation and produce the same estimated covariance matrix
- Let $\hat{L}^*$ represent rotated loadings. Then

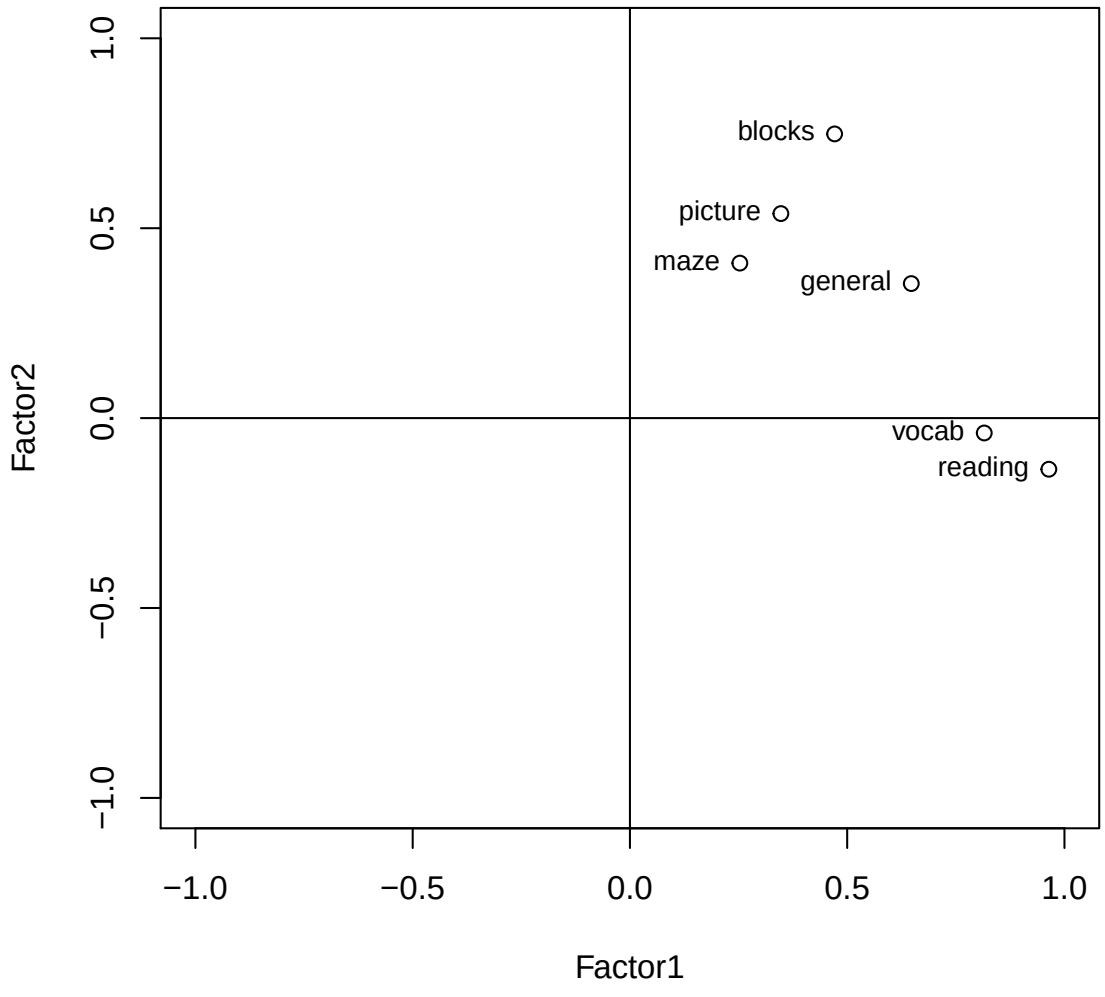
$$\hat{L}\hat{L}' + \hat{\Psi} = \hat{L^*}\hat{L^*}' + \hat{\Psi} = \hat{\Sigma}$$

- Ideally we would like variables to load high on one factor and have small loadings on the remaining factors
- Rotation often helps us achieve this

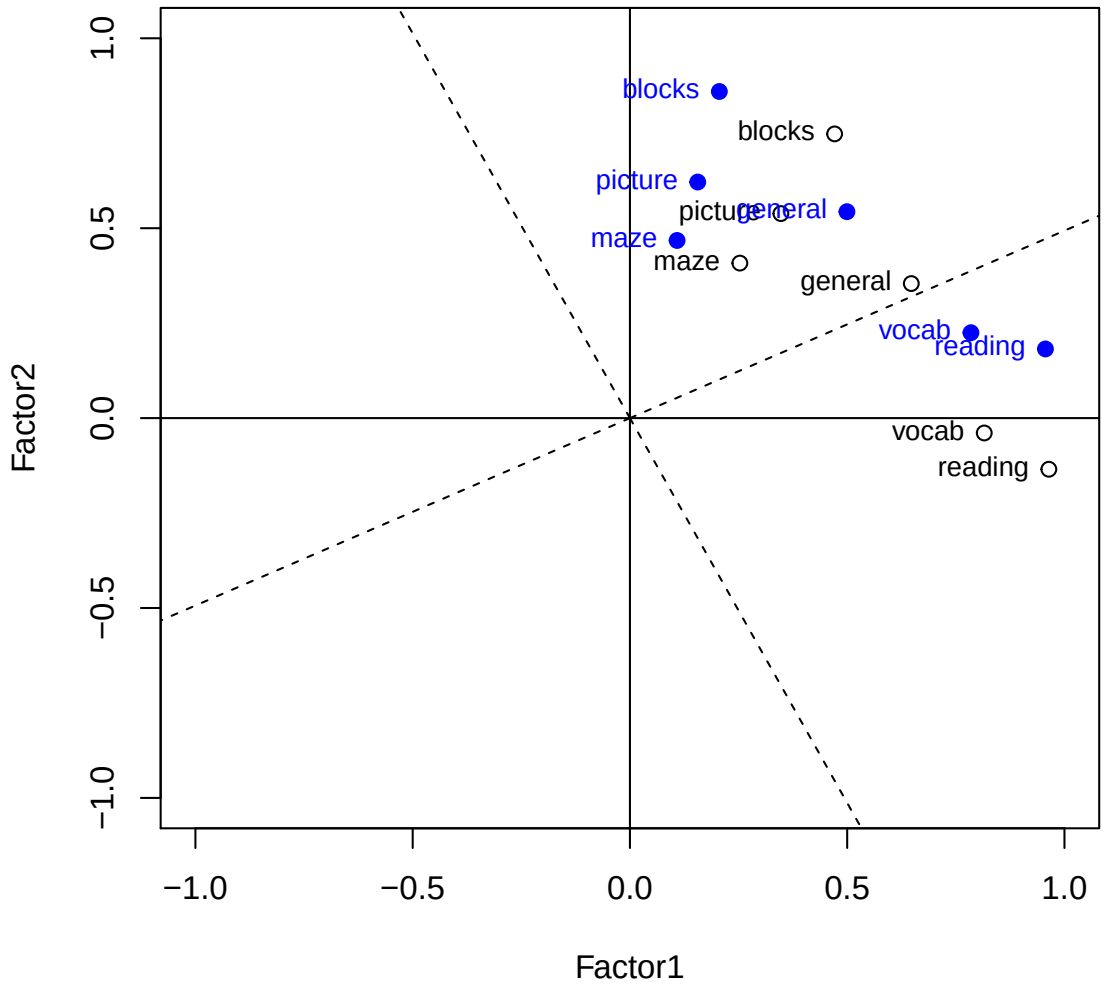
Two types of rotation

- There are two types of rotation:
  1. Orthogonal
  2. Oblique
- An Orthogonal rotation "rotates" fixed axes so they remain perpendicular; assumes uncorrelated factors
- An Oblique rotation "rotates" axes individually so they are not perpendicular; assumes correlated factors

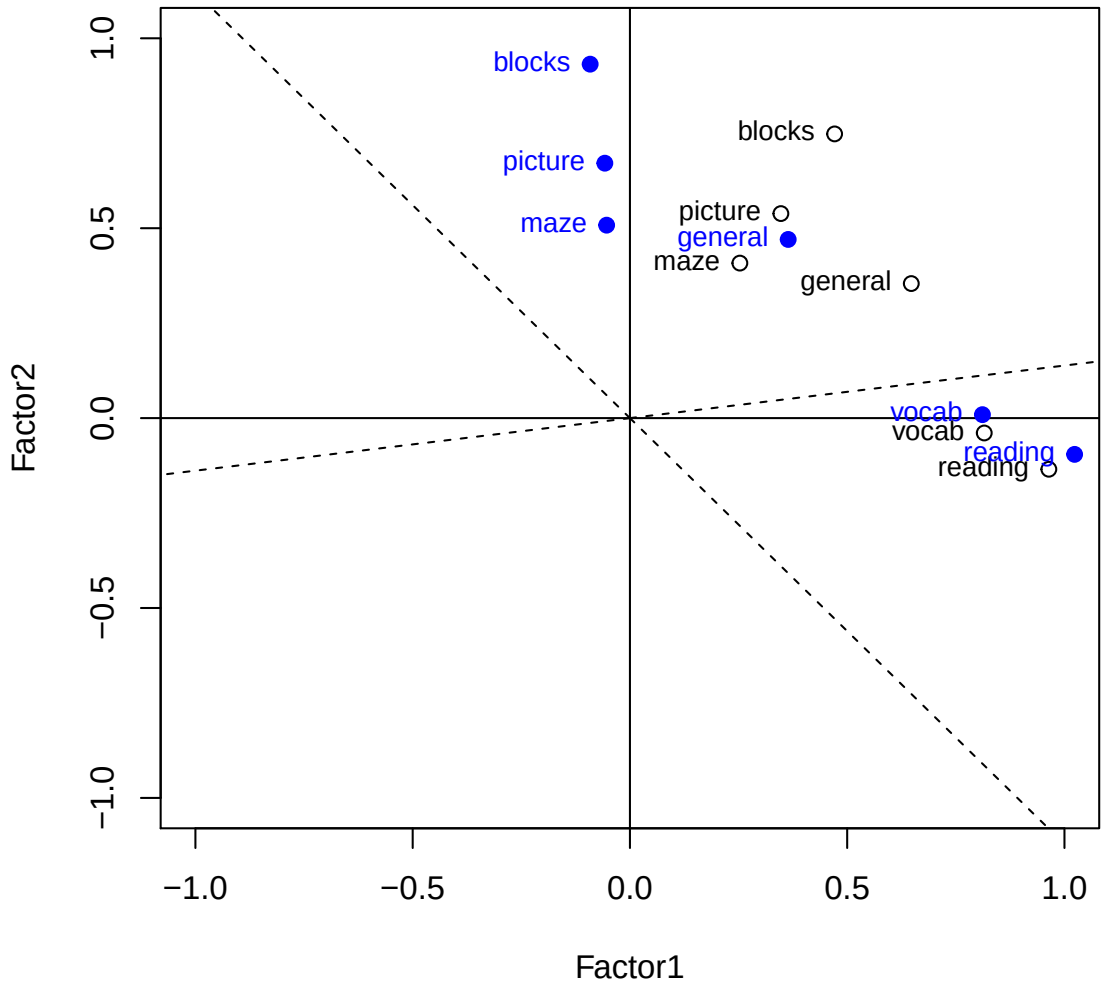
Before rotation



After Orthogonal Rotation



After Oblique Rotation



Orthogonal vs Oblique

- `factanal` performs *varimax* rotation by default (an Orthogonal rotation)
- `fa` performs *oblimin* rotation by default (an Oblique rotation)
- There are several different types of Orthogonal and Oblique rotations; `fa` provides 15 different rotation options!
- Preacher and MacCallum (2003) recommend using oblique rotations
- If factors are uncorrelated, an oblique rotation will be virtually the same as an orthogonal rotation

Specifying rotation

- For `factanal`, use the `rotation` argument
- For `fa`, use the `rotate` argument
- Examples:
  - `factanal(covmat = ability.cor, factors = 2, n.obs = 112, rotation = "promax")`
  - `fa(r = ability.cor, nfactors = 2, fm = "mle", n.obs = 112, rotate = "promax")`
- "promax" is an oblique transformation
- Base R only provides varimax and promax rotations for `factanal`
- The `GPArotation` package provides many more rotations
- Let's go to R!

Factor Scores

- Recall that EFA investigates the existence of unobserved *factors* such as intelligence, spatial reasoning, reading comprehension, depression, anxiety, etc
- The factors can't be directly measured, however using our model we can estimate their values
- These are called *factor scores*
- For example, we could use our EFA model to estimate someone's spatial reasoning and verbal comprehension *scores* given their test results
- Pairwise scatterplots of factor scores also helps identify outliers
- Factor scores can be used as inputs to a subsequent analysis

Estimating Factor Scores

- As you might guess, there are several ways to estimate factor scores
- `factanal` provides two methods: `"regression"` and `"Bartlett"`
- `fa` provides five methods
- Johnson and Wichern (2007) state neither `"regression"` nor `"Bartlett"` is uniformly superior
- The `fa` documentation makes no recommendation on which method to use

Estimating Factor Scores in R

- To have factor scores calculated for your data, you must have subject-level data available, not just a correlation matrix
- Scores are stored in the factor analysis object
- Say you have a data frame called `dat` with one row per subject:
  - `fa.out1 <- factanal(x = dat, factors = 2, scores = "regression")`
  - `fa.out2 <- fa(r = dat, nfactors = 2, scores = "regression")`
- To view or work with the scores
  - `fa.out1$scores`
  - `fa.out2$scores`

Estimating Factor Scores for new data

- Say we have test scores for an individual and we want to estimate her factor scores using our EFA model
- Regression formula:

$$\hat{\mathbf{f}}_j = \hat{\mathbf{L}}' \mathbf{R}^{-1} \mathbf{z}_j$$

- Bartlett formula:

$$\hat{\mathbf{f}}_j = (\hat{\mathbf{L}}' \hat{\Psi}^{-1} \hat{\mathbf{L}})^{-1} \hat{\mathbf{L}}' \hat{\Psi}^{-1} \mathbf{z}_j$$

Where $\mathbf{z}_j$ is a vector of standardized values

Estimating Factor Scores for new data - example

- Let's say we fit an EFA model with two factors
- Further, say we have *standardized* test scores for an individual (general, picture, blocks, etc)
- What are the person's *factor scores* using the regression method?

```
f.out <- factanal(covmat = ability.cor, factors = 2)
z <- c(.5, 0.75, 1.1, .79, 1.4, 1.2)
L <- f.out$loadings
Psi <- diag(f.out$uniquenesses)
# Regression
t(L) %*% solve(ability.cor) %*% z
```

```
##                [,1]
## Factor1 1.2395447
## Factor2 0.7501126
```

Estimating Factor Scores for new data - example

- What are the person's *factor scores* using the Bartlett method?

```
# Bartlett
solve(t(L) %*% solve(Psi) %*% L) %*%
  t(L) %*% solve(Psi) %*% z
```

```
##                [,1]
## Factor1 1.2748962
## Factor2 0.8466333
```

- Let's go to R!

Once again, how many factors?

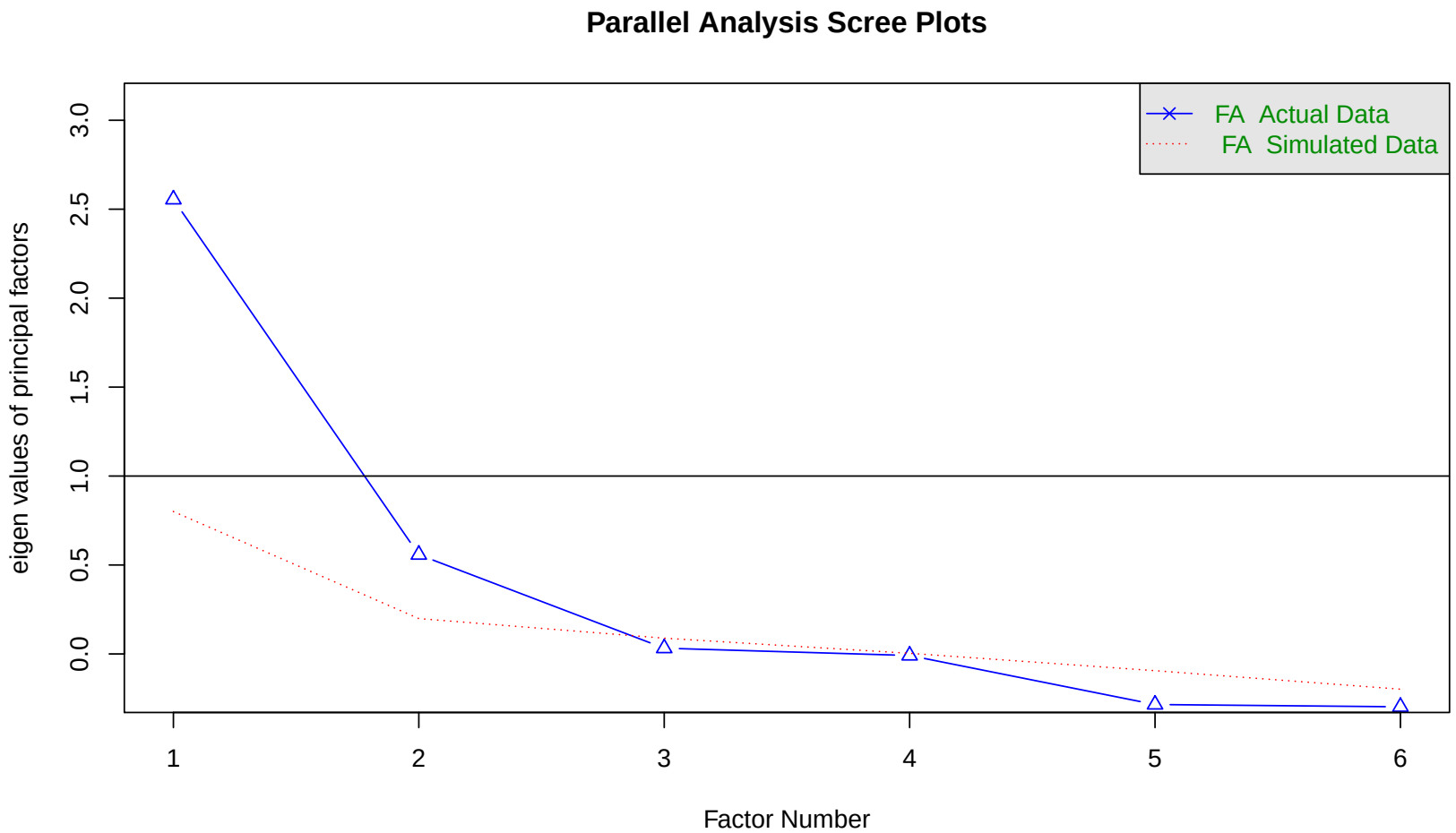
- Earlier we mentioned using proportion of variance explained and "reasonableness" of results
- Revelle reviews two other procedures:
  1. Parallel Analysis scree plots
  2. Very Simple Structure Criterion (VSS)
- Let's see how to use and interpret these procedures (without diving into how they work)

Parallel Analysis scree plots

- A Parallel Analysis scree plot involves eigenvalues and simulated data
- The details are beyond the scope of the workshop
- However the `psych` package makes it easy to run and interpret
- Use `fa.parallel` on the correlation matrix of your data
- How to interpret: the number of factors to retain is the number of triangles above the dotted red line
- The function will also provide helpful messages and warnings

Parallel Analysis scree plot example

```
library(psych)
fa.parallel(ability.cor, n.obs = 112, fa = "fa")
```

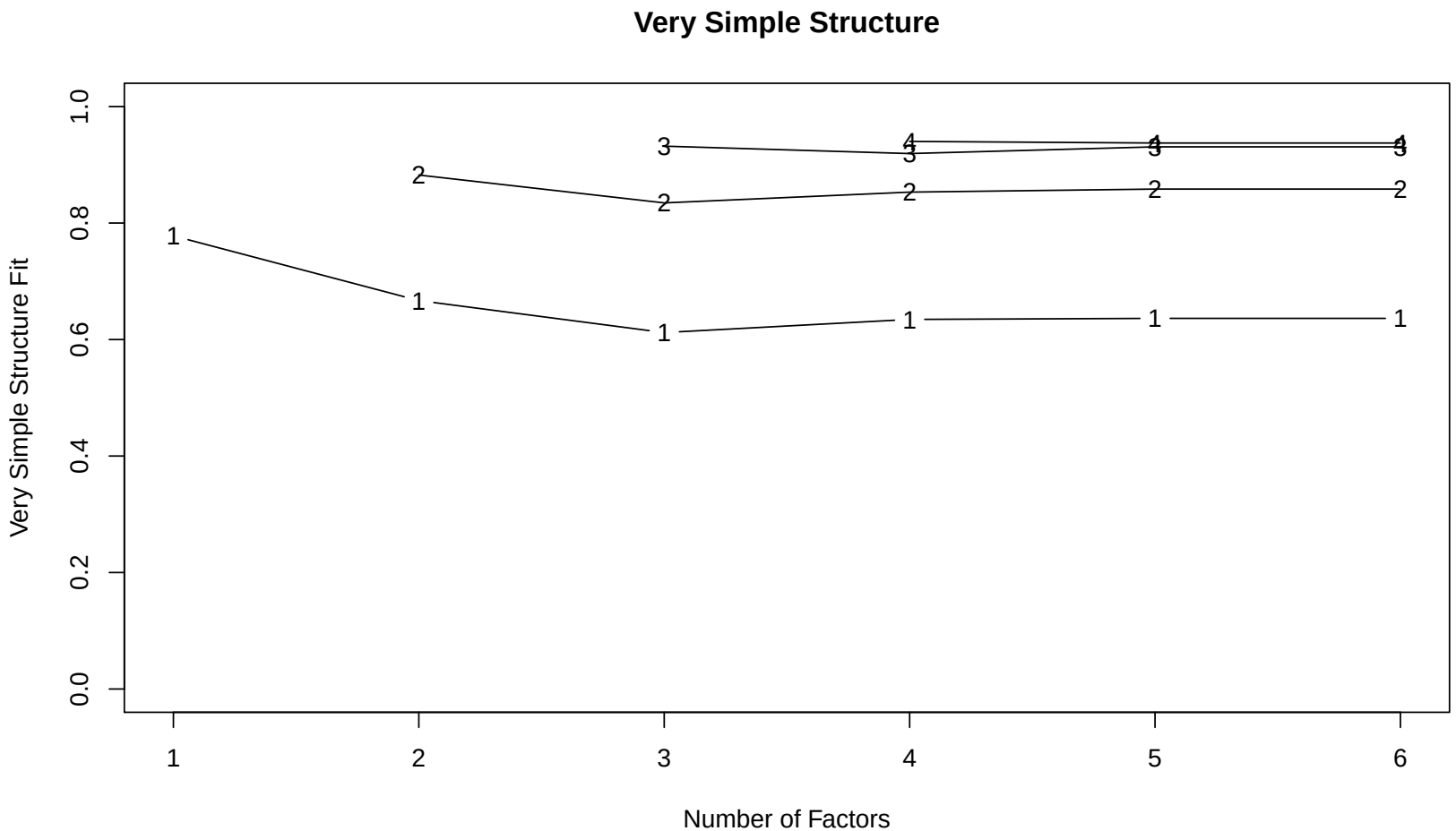


Very Simple Structure Criterion (VSS)

- When interpreting factor loadings we tend to focus on large loadings and ignore the small loadings
- We are essentially interpreting the loadings matrix as if it had a *simple structure*
- A simple structure consists of high loadings and 0s
- A simple structure with complexity 1 has one high loading and all 0s
- a simple structure with complexity 2 has two high loadings and all 0s, etc
- VSS allows us to compare solutions of varying complexity and for different number of factors
- Use `VSS` on the correlation matrix of your data
- How to interpret: peak criterion (on y-axis) for a given complexity corresponds to optimal number of factors (on x-axis)

VSS example

```
library(psych)
VSS(ability.cor, n.obs = 112)
```



EFA odds and ends

- Sometimes one or more uniquenesses will fall below 0. This is called a *Heywood case*. If it happens, perhaps try a different estimation method
- The `fa` function returns several measures of fit. Two of interest:
  - RMSEA index (values close to 0 suggest good model fit)
  - Tucker Lewis Index (values closer to 1 suggest good model fit)
- EFA models fit with the `fa` function can visualized with a path diagram using the `diagram` function in the `psych` package
- If your raw data consists of a mix of continuous, polytomous (limited set of whole numbers) and/or dichotomous values, use the `mixedCor` function in the `psych` package to calculate the correlation matrix
- For large data sets, split them in half and perform EFA on each part; compare the two results

Final thoughts

- Much more to EFA; this was just an intro
- "vast majority of attempted factor analyses do not yield clear-cut results." (Johnson and Wichern)
- If a factor analysis is successful, various combinations of estimations and rotations should result in the same conclusion
- Let's go to the R script!

References

Johnson, R. and Wichern, D. (2007) *Applied Multivariate Statistical Analysis, 6 ed*. Pearson Prentice Hall, New Jersey.

Preacher, K. and MacCallum, R. (2003) Repairing Tom Swift's Electric Factor Analysis Machine. *Understanding Statistics*, Volume 2, Issue 1.

Revelle, W. (in prep) *An introduction to psychometric theory with applications in R*. Springer. Working draft available at http://personality-project.org/r/book/

Thanks for coming

- For statistical consulting: statlab@virginia.edu
- Sign up for more workshops or see past workshops: http://data.library.virginia.edu/training/
- Register for the Research Data Services newsletter to be notified of new workshops: http://data.library.virginia.edu/newsletters/