Principal Component Analysis and Exploratory Factor Analysis

PCA and EFA: Dimension Reduction and Latent Structures

Statistical Data Analysis

Lesson A2 (Advanced Module)

December 30, 2025

Learning Objectives

Principal Component Analysis (PCA)

Big picture: Finding the "main directions" in your data

- Understand the mathematical foundations of PCA
- Explain and apply the core concepts
- Perform PCA for data reduction and interpret results
- Understand eigenvalues and eigenvectors geometrically

Exploratory Factor Analysis (EFA)

Big picture: Discovering hidden causes behind observed patterns

- Know the differences between PCA and EFA
- Explain and apply the basics of exploratory factor analysis
- Perform principal axis factoring
- Conduct exploratory factor analysis in R

Both methods reduce dimensions, but have different goals

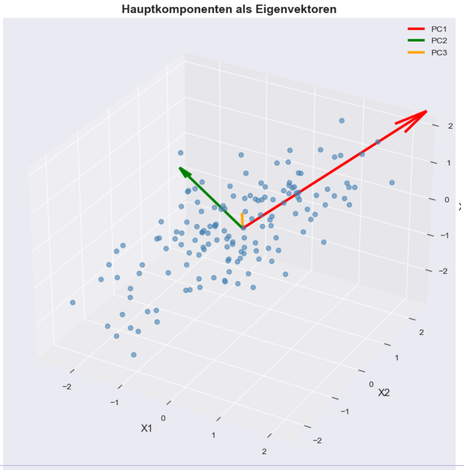
Motivation: The Curse of Dimensionality

The Problem ("Too many variables!")

- High-dimensional data is hard to visualize
- Many correlated variables (redundancy)
- "Curse of Dimensionality" (distances become meaningless)
- Multicollinearity in regressions

The Solution

- Dimension reduction: fewer variables, same information
- Identify latent structures (hidden patterns)
- Minimize information loss



Goal: Represent complex data in fewer dimensions

Part 1:
Principal Component Analysis (PCA)

What is Principal Component Analysis?

Intuition ("Find the main directions of spread")
Think of a cloud of data points. PCA finds:

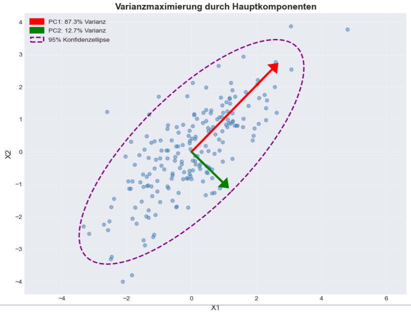
- The direction of maximum spread (PC1)
- The perpendicular direction with next-most spread (PC2)
- And so on...

Principal Components

- PC1: Captures maximum variance
- PC2: Second-largest variance (orthogonal to PC1)
- PC3, PC4, ... and so on

Applications

- Data visualization
- Feature reduction
- Noise filtering
- Exploratory analysis



PCA finds orthogonal directions of maximum variance

Covariance Matrix: The Starting Point

Definition ("How do variables move together?")

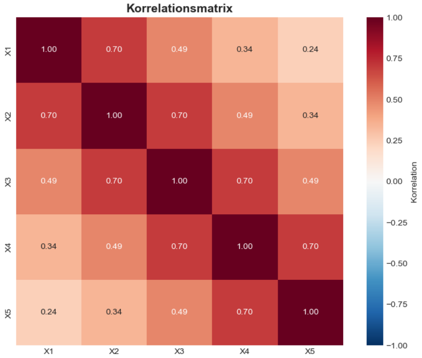
$$\mathbf{\Sigma} = \frac{1}{n-1}\mathbf{X}^T\mathbf{X}$$

Properties

- Symmetric: $\Sigma_{ij} = \Sigma_{ji}$
- Diagonal: Variances (spread of each variable)
- Off-diagonal: Covariances (how pairs move together)

Correlation Matrix

$$\mathbf{R} = \text{Corr}(\mathbf{X})$$



Red = positive, Blue = negative, White = no correlation

The covariance matrix contains all information about linear relationships

Computing Eigenvalues - Step 1

Kovarianzmatrix Σ:

2.0	0.8
0.8	1.0

Schritt 1: Eigenwertgleichung aufstellen

$$\det(\Sigma - \lambda I) = 0$$

Eigenvalue Problem: $\mathbf{\Sigma v} = \lambda \mathbf{v}$ (directions stretched by λ, not rotated)
Rearranged: $(\mathbf{\Sigma} - \lambda \mathbf{I})\mathbf{v} = \mathbf{0}$

Eigenvectors show directions; eigenvalues give variance in each direction

Computing Eigenvalues - Step 2

Schritt 2: Charakteristisches Polynom

```
[  2-λ   0.8  ]
[  0.8   1-λ  ]
```

$$(2-\lambda)(1-\lambda) - 0.8^2 = 0$$

$$\boxed{\lambda^2 - 3\lambda + 1.36 = 0}$$

For a non-trivial solution: $\det(\mathbf{\Sigma} - \lambda \mathbf{I}) = 0$

This gives a polynomial of degree p (number of variables)

The roots of the characteristic polynomial are the eigenvalues

Computing Eigenvalues - Step 3

Schritt 3: Eigenwerte berechnen

$$\lambda = \frac{3 \pm \sqrt{9 - 5.44}}{2}$$

λ□ = 2.443 (groesster Eigenwert)

λ□ = 0.557 (kleinster Eigenwert)

Varianz erklaert: 81.4% + 18.6% = 100%

Interpretation ("How much variance does each PC capture?")

- λ_1: Largest eigenvalue = Variance of PC1
- λ_2: Second largest = Variance of PC2
- Sum of all eigenvalues = Total variance

Eigenvalues determine the importance of each principal component

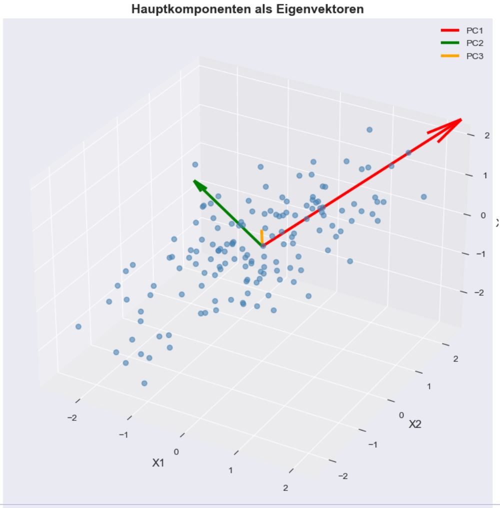
Eigenvectors: A Geometric View

What are eigenvectors? ("The compass directions")

- Eigenvectors = directions of spread
- Arrows show PC axes
- Length proportional to $\sqrt{\lambda_i}$ (standard deviation)
- Orthogonal (perpendicular) to each other

Key Properties

- Unit length: $||\mathbf{v}_i|| = 1$
- Orthogonal: $\mathbf{v}_i^T \mathbf{v}_j = 0$ for $i \neq j$
- Form a new coordinate system



Eigenvectors define a new orthogonal coordinate system for your data

Variance Maximization: The Core Concept

The Optimization Problem

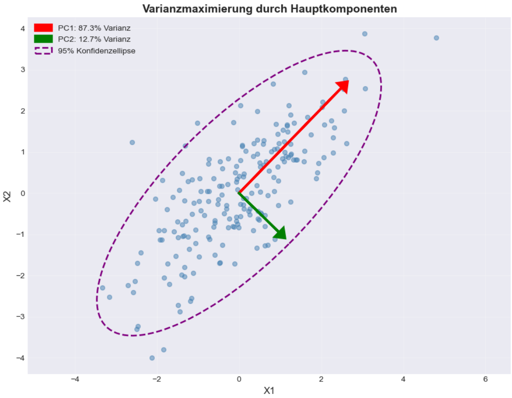
PC1 maximizes:

$$\max_{\mathbf{w}} \mathrm{Var}(\mathbf{Xw})$$

subject to $||\mathbf{w}|| = 1$

The Solution

- $\mathbf{w}$ is the eigenvector for $\lambda_{\max}$
- PC1 explains maximum variance
- PC2 explains maximum remaining variance
- Orthogonality is guaranteed



- Red line: PC1 (maximum variance)
- Green line: PC2 (second largest)
- Ellipse: 95% confidence region

PCA finds directions of maximum spread in the data

Principal Components as Linear Combinations

Mathematical Form

Principal component i:

$$PC_i = w_{i1}X_1 + w_{i2}X_2 + \cdots + w_{ip}X_p$$

Matrix form:

$$\mathbf{PC} = \mathbf{XW}$$

where $\mathbf{W}$ is the loading matrix.

Key Points

- Weights w_{ij} come from eigenvectors
- Each PC is a linear combination of ALL variables
- Scores = transformed data points

Example

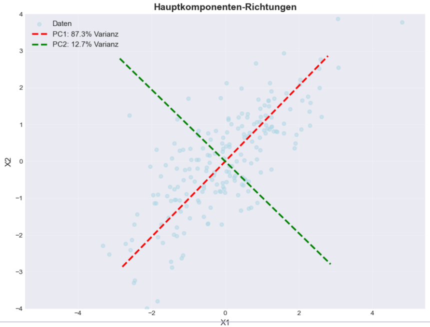
$$PC_1 = 0.45X_1 + 0.42X_2 + 0.38X_3 + \cdots$$

Back-transformation

Reconstruct original data:

$$\mathbf{X} \approx \mathbf{PC}_k \mathbf{W}_k^T$$

where $k < p$ (dimension reduction)



PC1 and PC2 define a new rotated coordinate system

PCA Algorithm: Step by Step

The Steps

1. **Center the data**
   $\tilde{\mathbf{X}} = \mathbf{X} - \bar{\mathbf{X}}$
2. **(Optional) Standardize**
   If variables have different units
3. **Compute covariance matrix**
   $\boldsymbol{\Sigma} = \frac{1}{n-1}\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}$
4. **Eigenvalue decomposition**
   $\boldsymbol{\Sigma}\mathbf{v}_i = \lambda_i\mathbf{v}_i$
5. **Sort**
   By eigenvalues (descending)
6. **Transform**
   $\mathbf{PC} = \tilde{\mathbf{X}}\mathbf{W}$

Alternative: SVD

Singular Value Decomposition:

$$\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^T$$

- More numerically stable
- Used by `prcomp()`
- Eigenvalues: $\lambda_i = d_i^2/(n-1)$

Computational Aspects

- $O(p^3)$ for eigendecomposition
- $O(np^2)$ for SVD
- Sparse PCA for large datasets
- Approximate methods available

SVD is the preferred numerical method

R Code: prcomp()

Basic Usage

```
# Perform PCA
pca_result <- prcomp(data,
                     scale. = TRUE)

# Summary
summary(pca_result)

# Eigenvalues
pca_result$sdev^2

# Loadings (Rotation)
pca_result$rotation

# Scores
pca_result$x
```

Parameters

- `scale. = TRUE`: Standardize
- `center = TRUE`: Center (default)

Output Components

- `$sdev`: Standard deviations
- `$rotation`: Loading matrix
- `$x`: PC scores
- `$center`, `$scale`: Centering/scaling values

prcomp() uses SVD and is numerically stable

R Output: prcomp() Results

Summary Output

```
Importance of components:
                          PC1    PC2    PC3
Standard deviation     1.8091 1.2022 0.9847
Proportion of Variance 0.4536 0.2003 0.1344
Cumulative Proportion  0.4536 0.6539 0.7883
```

Interpretation

- PC1 explains 45.4% of variance
- PC2 explains 20.0% of variance
- Together: 65.4%
- PC3 adds another 13.4%

Computing Variance Explained

```
# Eigenvalues
eigenvalues <- pca_result$sdev^2

# Proportion
var_prop <- eigenvalues /
            sum(eigenvalues)

# Cumulative
cumsum(var_prop)
```

Rules of Thumb

- Keep enough PCs for $\geq$ 80% variance
- Or: Kaiser criterion ($\lambda > 1$)
- Or: Elbow in Scree Plot

Choose as few components as possible, as many as needed

Scree Plot: Theory

What is a Scree Plot?

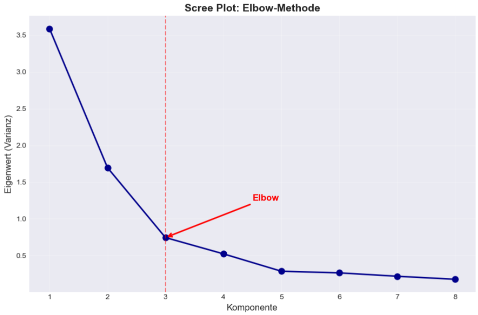
- Eigenvalues vs. component number
- Visualizes importance of each PC
- Helps choose how many to keep

Elbow Method

- Look for a "bend" in the curve
- After the elbow, curve flattens
- Keep components before the elbow
- Subjective interpretation

Kaiser Criterion

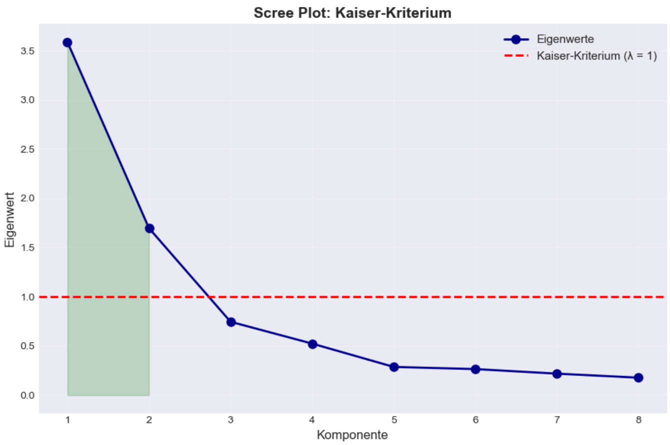
- Keep PCs with $\lambda > 1$
- Only for correlation matrix
- Conservative (often keeps too many)



Elbow at component 3

Combine multiple criteria for a robust decision

Scree Plot: Kaiser Criterion



Kaiser Criterion

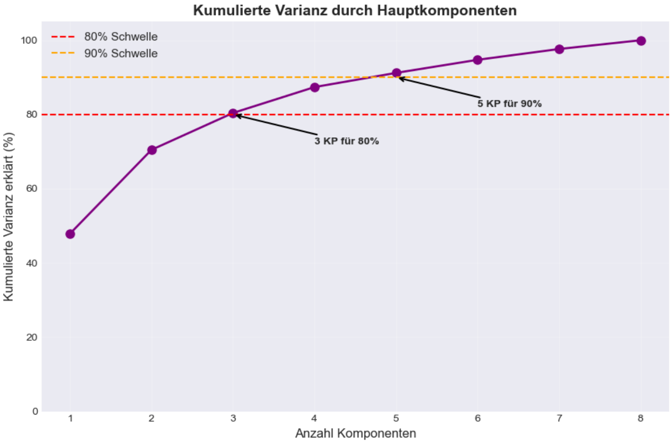
Keep components with $\lambda > 1$

Limitations

- Only for correlated data
- Tends to retain too many PCs

Green area: Components retained by Kaiser criterion

Cumulative Variance Explained



Interpretation

- Number of components vs. % variance
- Typical target: 80-90%

Decision

- 3 PCs for 80% variance
- 5 PCs for 90% variance

Cumulative variance shows information content

Biplot: What Does It Show?

Two Types of Information

1. **Observations** (points)
2. **Variables** (arrows)

Reading the Arrows

- Length = importance
- Direction = correlation with PCs
- Angle between arrows = correlation between variables

- 0: positively correlated
- 90: uncorrelated
- 180: negatively correlated

Reading the Points

- Position = PC scores
- Proximity = similarity
- Clusters = groups

Points + Arrows Together

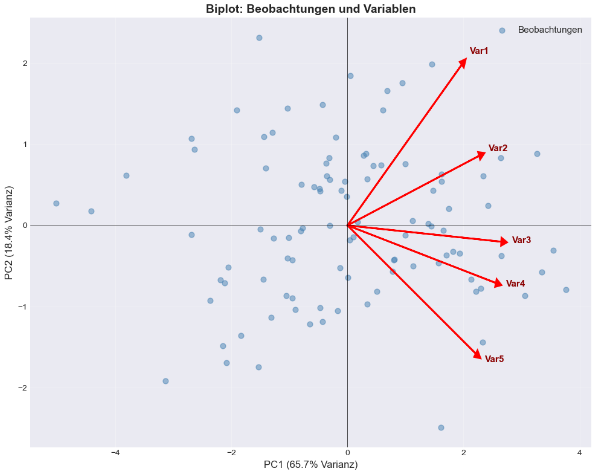
- Project point onto arrow
- ≈ value for that variable

Limitations

- Only 2D (first 2 PCs)
- Scaling matters

The biplot is the most powerful PCA visualization tool

Biplot: Comprehensive Example



Observations

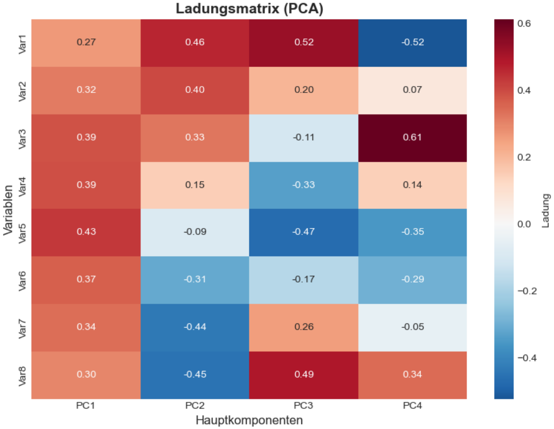
- Blue points: data points
- Spread shows variability

Variables

- Red arrows: 5 variables
- Var1, Var2 correlated with PC1

PC1 explains horizontal spread, PC2 explains vertical spread

Loading Matrix Visualized



Reading the Heatmap

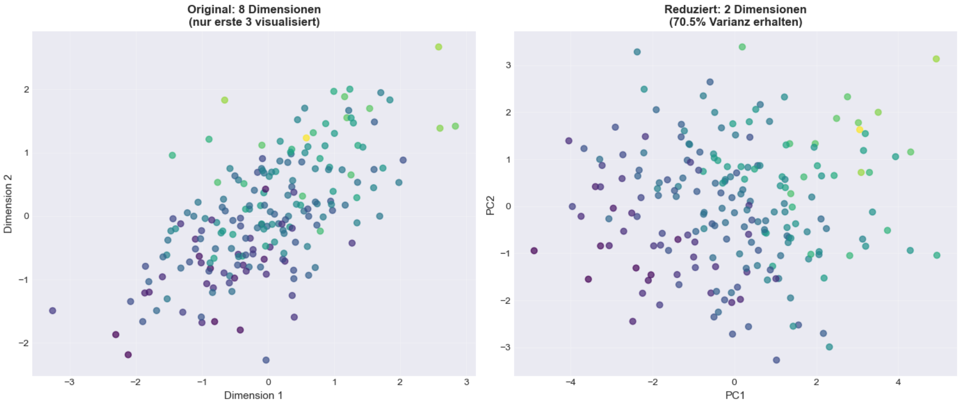
- Rows: Variables
- Columns: PCs
- Color: Loading strength

Look for Patterns

- Which variables load on PC1?
- Is there simple structure?

Heatmap reveals the structure of the loading matrix

PCA for Data Reduction



Before: 8 Dimensions

- Hard to visualize
- Redundant information

After: 2 Dimensions

- Easy to visualize
- 70.7% variance retained

PCA compresses data with minimal information loss

R Code: princomp() vs. prcomp()

princomp()

```r
# Eigenvalue decomposition
pca_princomp <- princomp(data,
                         cor = TRUE)

# Loadings
loadings(pca_princomp)

# Scores
pca_princomp$scores
```

Differences

- Eigenvalue decomposition of $\boldsymbol{\Sigma}$
- Less numerically stable
- Traditional method
- Compatibility with old software

Comparison

	prcomp	princomp
Method	SVD	Eigen
Stability	High	Medium
Speed	Fast	Slower
Recommended	Yes	No

Recommendation

- Use `prcomp()`
- More numerically stable
- Modern standard
- Same results (except rounding)

prcomp() is the preferred method in R

Part 2:
Exploratory Factor Analysis (EFA)

EFA: Discovering Latent Structures

Core Idea ("What hidden causes explain these correlations?")

- Explain observed variables by few *latent factors*
- Factors are not directly measurable
- Focus on shared (common) variance
- Theory-driven or exploratory

Examples

- Intelligence → IQ test items
- Personality → questionnaire items
- Customer satisfaction → ratings
- Economic factors → indicators

EFA vs. PCA

EFA	PCA
Latent factors	Components
Common variance	Total variance
Causal model	Descriptive
Error term	No error
Theory-based	Data reduction

In Common
Both reduce dimensions

EFA searches for underlying causes of correlations

The Factor Model: Mathematical Form

Basic Equation

$$\mathbf{X} = \mathbf{\Lambda F} + \mathbf{e}$$

where:

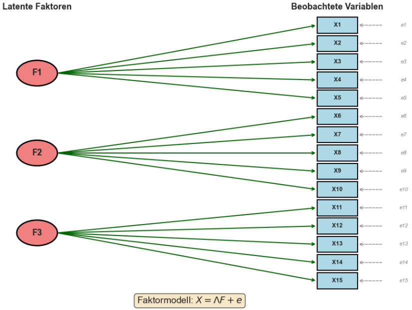
- $\mathbf{X}$: Observed variables ($p \times 1$)
- $\mathbf{\Lambda}$: Loading matrix ($p \times m$)
- $\mathbf{F}$: Latent factors ($m \times 1$)
- $\mathbf{e}$: Error term ($p \times 1$)

For Variable i

$$X_i = \lambda_{i1} F_1 + \lambda_{i2} F_2 + \cdots + \lambda_{im} F_m + e_i$$

Assumptions

- $\mathbf{F}$ standardized: $E[\mathbf{F}] = 0$, $\text{Var}[\mathbf{F}] = 1$
- $\mathbf{e}$ uncorrelated with $\mathbf{F}$
- e_i uncorrelated with e_j (usually)



Path diagram shows causal structure: factors cause observed values

Communalities and Uniqueness

Communality (h_i^2)
Proportion of variance in X_i explained by factors:

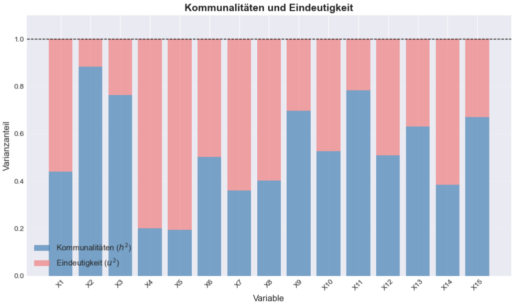
$$h_i^2 = \lambda_{i1}^2 + \lambda_{i2}^2 + \cdots + \lambda_{im}^2$$

Uniqueness (u_i^2)
Proportion of variance NOT explained:

$$u_i^2 = 1 - h_i^2$$

Interpretation

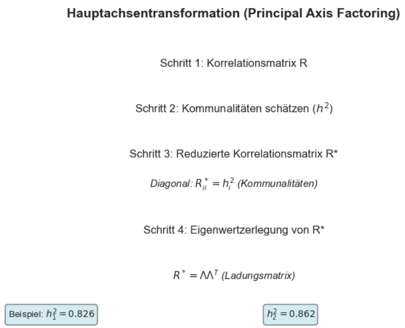
- h_i^2 near 1: well explained
- h_i^2 near 0: poorly explained
- u_i^2 contains measurement error + specific variance



Blue: Communality, Red: Uniqueness

Sum: $h^2 + u^2 = 1$ **for each variable**

Principal Axis Factoring



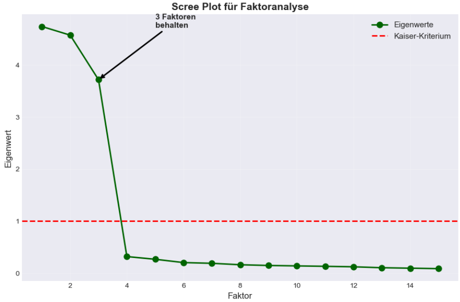
Iterative Procedure

1. Estimate initial communalities (e.g., R^2 from regressions)
2. Create reduced correlation matrix $\mathbf{R}^*$ (diagonal = communalities)
3. Eigenvalue decomposition of $\mathbf{R}^*$
4. Extract m factors
5. Compute new communalities
6. Repeat until convergence

Commonly used alternative to Maximum Likelihood

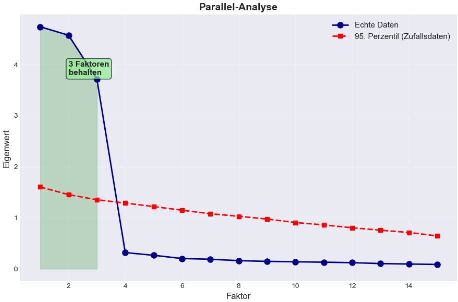
Determining Number of Factors

Scree Plot



- Elbow method
- Kaiser: $\lambda > 1$

Parallel Analysis



- Compare with random data
- More robust than Kaiser

Parallel analysis is the gold standard

Varimax Rotation

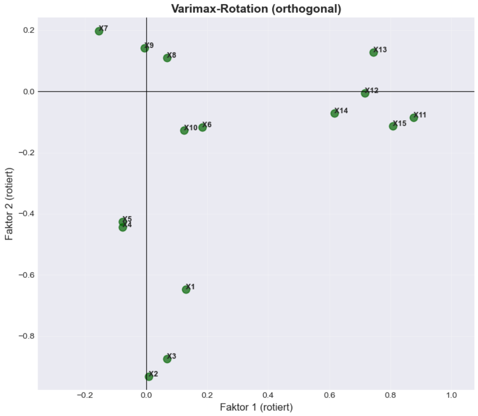
Why Rotation?

- Unrotated solution is hard to interpret
- Rotation improves clarity
- Does not change model fit

Varimax Criterion

Goal: "Simple structure"

- Make high loadings higher
- Make low loadings lower
- Orthogonal (factors uncorrelated)



Properties

- Factors remain uncorrelated
- Most commonly used

Varimax is the standard rotation method

Promax Rotation

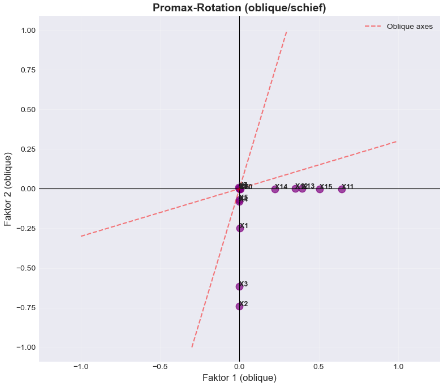
Oblique Rotation

- Factors are allowed to correlate
- More realistic in practice
- Starts with Varimax, then relaxes

When to Use Promax?

- Constructs are correlated
- Personality traits
- Cognitive abilities

Factor Correlations
Report the $\mathbf{\Phi}$ matrix!



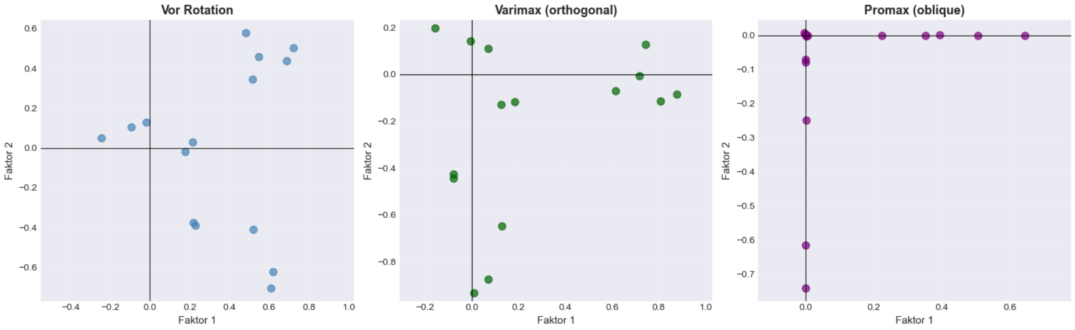
Oblique axes

Comparison

	Varimax	Promax
Correlation	No	Yes
Simplicity	Good	Better
Realism	-	+

Promax allows factor correlations

Rotation Comparison



Before Rotation

Items load on both factors

Varimax

Clear assignment, orthogonal axes

Promax

Clearest structure, oblique axes

Rotation dramatically improves interpretability

Interpreting Factor Loadings

Guidelines

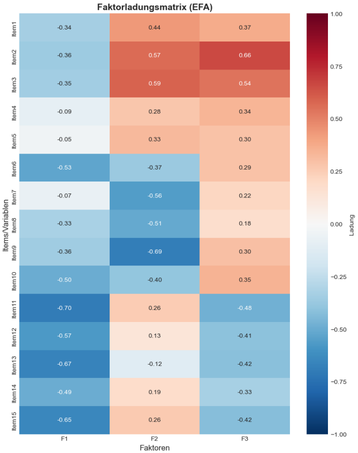
- $|\lambda| > 0.3$: minimal
- $|\lambda| > 0.5$: moderate
- $|\lambda| > 0.7$: strong

Cross-Loadings

- Item loads on ≥ 2 factors
- Problematic
- Consider removing item

Naming Factors

- Look at highest loadings
- Find common theme



Example

- F1: Items 1-5
- F2: Items 6-10
- F3: Items 11-15

Simple structure: Each item loads high on one factor only

Comparison and Decision Guide

PCA vs. EFA: When to Use Which?

Decision Tree

1. **Goal?**
   - Data reduction → PCA
   - Latent structure → EFA
2. **Theory available?**
   - Yes → EFA (or CFA)
   - No → PCA
3. **All variance important?**
   - Yes → PCA
   - No → EFA
4. **Measurement error relevant?**
   - Yes → EFA
   - No → PCA

Typical Applications

PCA

- Exploratory data analysis
- Dimension reduction for ML
- Data visualization
- Feature engineering
- Noise reduction

EFA

- Questionnaire development
- Scale construction
- Construct validation
- Theory generation
- Psychometrics

Both methods are complementary, not competing

PCA vs. EFA: Detailed Differences

Aspect	PCA	EFA
Goal	Maximize variance	Find latent factors
Variance	Total variance	Common variance
Model	$\mathbf{X} = \mathbf{WPC}$	$\mathbf{X} = \mathbf{\Lambda F} + \mathbf{e}$
Components	Linear combo of all X	Latent variables
Error term	No	Yes ($\mathbf{e}$)
Diagonal	Variances (σ_i^2)	Communalities (h_i^2)
Uniqueness	Yes	Not unique (rotation)
Rotation	Rarely	Essential
Interpretation	Components = data	Factors → data (causal)
R function	`prcomp()`, `princomp()`	`factanal()`, `fa()`
Choosing number	Scree, Kaiser, Variance%	Parallel, Scree, Theory

Key difference: PCA = descriptive, EFA = model with assumptions

Summary

Principal Component Analysis

- Eigenvalue decomposition of $\boldsymbol{\Sigma}$
- Eigenvectors = PC directions
- Eigenvalues = Variance in each direction
- Variance maximization
- Orthogonal projection
- Scree plot to choose number of PCs
- `prcomp()` in R (uses SVD)
- Biplot for visualization

When to Use PCA?

Data reduction, visualization, no theory about latent structure

Exploratory Factor Analysis

- Factor model with error term
- Focus on common variance
- Communalities (h^2) and uniqueness (u^2)
- Principal axis factoring
- Rotation is essential (Varimax, Promax)
- Parallel analysis to choose number
- `factanal()` or `psych::fa()`
- Interpret with theory in mind

When to Use EFA?

Latent constructs, scale construction, theory testing

Key takeaway: PCA for data reduction, EFA for latent structures

Thank You

Questions?

Next steps: Run R script `pca_efa_examples.R`