Principal Component Analysis and Exploratory Factor Analysis

Statistical Data Analysis — Lesson 3

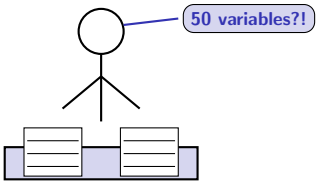
Prof. Dr. Jörg Osterrieder

School of Engineering and Computer Science

The 50-Question Problem

Scenario: You survey 200 students on 50 personality questions.

- Each student is described by 50 numbers
- How do you visualize 50-dimensional data?
- Which questions are really measuring the same thing?
- Can we simplify without losing meaning?



Today we learn two techniques that turn 50 confusing variables into a handful of meaningful ones.

What We'll Learn Today

Learning Objectives:

1. **Understand what PCA does and why** — reducing many variables to a few summary components
2. **Walk through PCA step by step with numbers** — from raw data to principal components
3. **Understand EFA and how it differs from PCA** — finding hidden causes behind observed data
4. **Know when to use PCA vs EFA** — choosing the right tool for your research question

Running Examples

- **PCA:** 5 students measured on Height, Weight, and Age
- **EFA:** 10-item restaurant satisfaction survey

By the end of this lesson, you will be able to perform and interpret both PCA and EFA.

Why Reduce Dimensions?

From many variables to few meaningful summaries

The Curse of Dimensionality

The numbers speak for themselves:

- With **50 questions**: $\frac{50 \times 49}{2} = 1{,}225$ pairwise correlations
- With **5 components**: $\frac{5 \times 4}{2} = 10$ correlations
- That's a **99.2% reduction!**

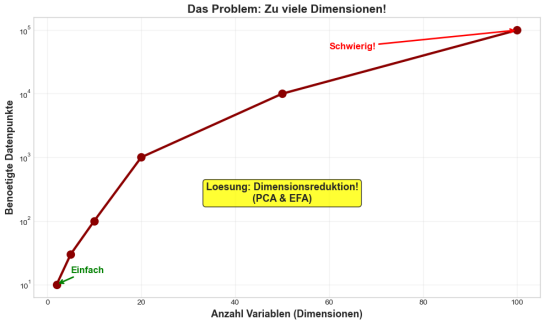
Discovery

Quick — how many pairwise correlations for 10 variables?

The Curse of Dimensionality

The numbers speak for themselves:

- With **50 questions**: $\frac{50 \times 49}{2} = 1{,}225$ pairwise correlations
- With **5 components**: $\frac{5 \times 4}{2} = 10$ correlations
- That's a **99.2% reduction!**



Discovery

Quick — how many pairwise correlations for 10 variables?
Answer: $\frac{10 \times 9}{2} = 45$

Cross-reference: We will use PCA as preprocessing for clustering in Lesson 4.

Dimensionality reduction is not about throwing away data — it is about finding what matters.

Two Approaches to Simplification

PCA: The Photographer

"Find the best summary angles"

- Finds directions of maximum spread
- Creates new **composite** variables
- Goal: **compress** the data
- No assumptions about hidden causes

EFA: The Detective

"Find hidden causes"

- Looks for **latent factors** behind the data
- Explains **why** variables correlate
- Goal: **understand** the structure
- Assumes hidden factors exist

	PCA	**EFA**
Metaphor	Camera angles	Detective work
Output	Components	Factors
Purpose	Summarize	Explain

PCA asks "what is the best summary?" — EFA asks "what is the hidden cause?"

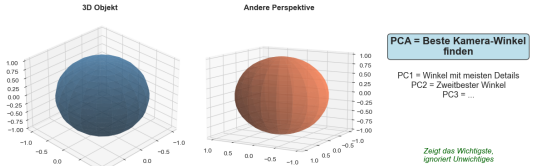
Principal Component Analysis

Finding the best summary of your data

What Is PCA?

The camera analogy:

- Imagine photographing a 3D sculpture
- Some angles capture almost everything
- Other angles are redundant
- PCA finds the **best angles** automatically



Definition

A **principal component** is a new variable that summarizes several original variables. It is a weighted combination (linear mix) of the originals.

PCA creates new variables that capture the most "spread" (variance) in your data.

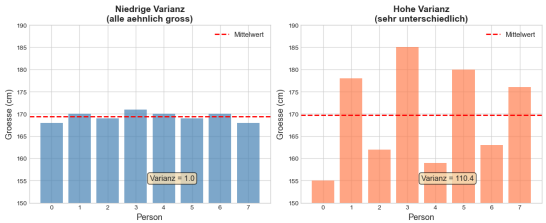
Variance = Information

Which variable is more informative?

- **Height**: ranges 150–190 cm
  → lots of spread → informative
- **Hair count**: 99,990–100,010
  → almost no spread → uninformative

Key insight: A variable that barely changes tells us nothing about differences between people.

PCA keeps directions with **high variance** and discards directions with **low variance**.



Variance = spread = information. PCA maximizes variance in each successive component.

Correlation = Redundancy

When two variables track each other:

- Height and arm span: $r = 0.95$
- Knowing one almost tells you the other
- They carry **redundant** information
- One summary could replace both

Discovery

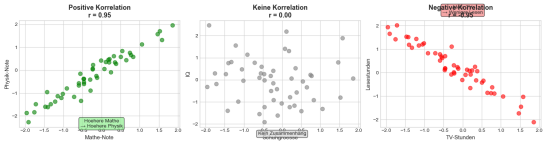
Which pair could be replaced by a single summary?

(a) Height & shoe size ($r = 0.87$)
(b) Height & IQ ($r = 0.02$)

Correlation = Redundancy

When two variables track each other:

- Height and arm span: $r = 0.95$
- Knowing one almost tells you the other
- They carry **redundant** information
- One summary could replace both



Discovery

Which pair could be replaced by a single summary?

(a) Height & shoe size ($r = 0.87$)
(b) Height & IQ ($r = 0.02$)

Answer: (a) — high correlation means high redundancy.

High correlation between variables means PCA can compress them into fewer components.

The Correlation Matrix

Our running example: 5 students measured on Height, Weight, and Age (standardized).

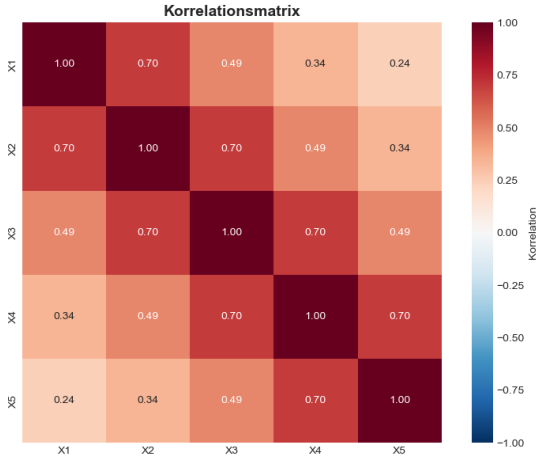
Definition

Covariance measures how two variables move together. Positive = same direction; negative = opposite.

Correlation matrix (standardized covariances):

$$\mathbf{R} = \begin{pmatrix} 1.00 & 0.92 & -0.82 \\ 0.92 & 1.00 & -0.93 \\ -0.82 & -0.93 & 1.00 \end{pmatrix}$$

Worked: $r(\text{H,W}) = 0.92 \rightarrow$ tall students tend to be heavier.



The correlation matrix is the input to PCA — it encodes all pairwise relationships.

Discovery: Which Direction Captures the Most Spread?

Look at the scatter plot on the right.

Discovery

Imagine drawing a single straight line through this cloud of points.

Which direction captures the **most spread**?

Draw an arrow through the **longest axis** of the ellipse.

Discovery: Which Direction Captures the Most Spread?

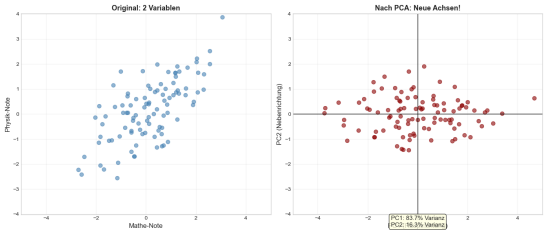
Look at the scatter plot on the right.

Discovery

Imagine drawing a single straight line through this cloud of points.

Which direction captures the **most spread**?

Draw an arrow through the **longest axis** of the ellipse.



Reveal: That arrow IS the first principal component (PC1)!

- PC1 = direction of maximum variance
- PC2 = perpendicular to PC1, maximum remaining variance

PCA automatically finds the direction that captures the most "spread" in the data.

Eigenvalues: How Important Is Each Component?

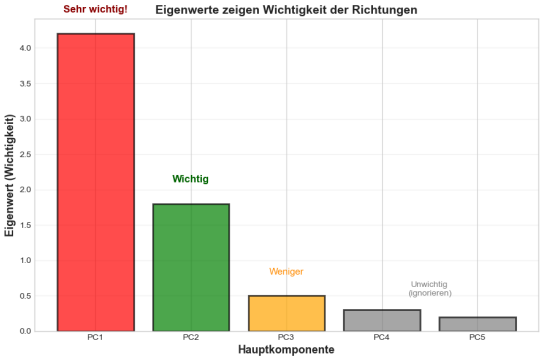
Definition

An **eigenvalue** is a number telling how much variance a component captures — like an importance score.

Our running example (3 variables → 3 eigenvalues):

Component	λ	% Variance
PC1	2.78	92.7%
PC2	0.18	6.0%
PC3	0.04	1.2%
Total	3.00	100%

Worked: $\lambda_1 = 2.78$, so PC1 captures $\frac{2.78}{3.00} = 92.7\%$ of the total variance.



Eigenvalues always sum to the number of variables. Larger eigenvalue = more important component.

Eigenvectors: The Recipe for Each Component

Definition

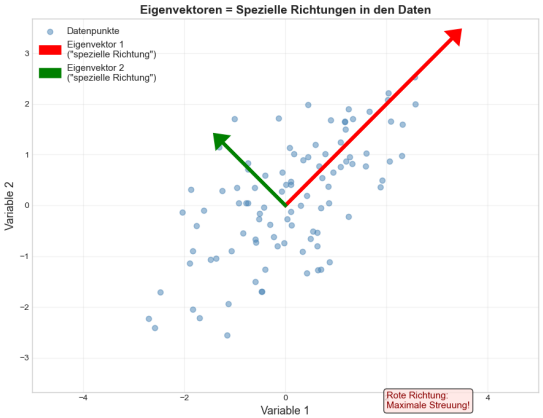
An **eigenvector** is the recipe for mixing original variables into a component — the "weights" or "loadings" for each variable.

PC1 eigenvector: $[0.57,\ 0.59,\ -0.57]$

Interpretation:

- Height: $+0.57$ (positive weight)
- Weight: $+0.59$ (positive weight)
- Age: -0.57 (negative weight)

$\rightarrow$ Height and Weight go together; Age goes in the opposite direction.
$\rightarrow$ PC1 measures **"body size vs. maturity."**



Eigenvectors tell you what each component means — which variables contribute and in which direction.

PCA in 4 Steps

The complete PCA procedure:

1. **Standardize** the data (subtract mean, divide by SD)
2. **Compute** the correlation matrix
3. **Eigendecompose** — find eigenvalues and eigenvectors
4. **Project** — multiply data by eigenvectors to get component scores

Discovery

What happens if one variable is measured in centimeters and another in kilograms?

PCA in 4 Steps

The complete PCA procedure:

1. **Standardize** the data (subtract mean, divide by SD)
2. **Compute** the correlation matrix
3. **Eigendecompose** — find eigenvalues and eigenvectors
4. **Project** — multiply data by eigenvectors to get component scores

Discovery

What happens if one variable is measured in centimeters and another in kilograms?

The cm variable has a much larger range and would **dominate** the analysis. That is why Step 1 (standardize) is essential — it puts all variables on the same scale.

Steps 1–2 you do yourself; Steps 3–4 the computer does for you.

Steps 1–2: Standardize and Compute Correlations

Step 1: Standardized data ($z = \frac{x-\bar{x}}{s}$)

Student	Height	Weight	Age
Alice	1.08	1.00	−0.46
Bob	−0.72	−1.20	1.37
Carol	0.45	0.80	−0.91
Dave	−1.35	−0.90	0.73
Eve	0.54	0.30	−0.73

Each column now has mean≈ 0, SD≈ 1.

Step 2: Correlation matrix

$$\mathbf{R} = \begin{pmatrix} 1.00 & 0.92 & -0.82 \\ 0.92 & 1.00 & -0.93 \\ -0.82 & -0.93 & 1.00 \end{pmatrix}$$

Reading the matrix:

- r(H,W) $= 0.92$: strong positive
- r(W,A) $= -0.93$: strong negative
- r(H,A) $= -0.82$: strong negative

All three variables are **highly correlated** $\rightarrow$ good candidate for PCA.

Strong correlations signal redundancy — PCA can compress these 3 variables effectively.

Steps 3–4: Eigendecomposition and Projection

Step 3: Eigendecomposition (computer finds these)

	λ	% Var	Cum. %
PC1	2.78	92.7%	92.7%
PC2	0.18	6.0%	98.8%
PC3	0.04	1.2%	100%

Eigenvectors:
PC1 $= [0.57,\ 0.59,\ -0.57]$
PC2 $= [-0.72,\ 0.03,\ -0.69]$

Step 4: Projection (compute scores)

PC1 score for Alice:

$$\begin{aligned} &= 0.57 \times 1.08 + 0.59 \times 1.00 \\ &\quad + (-0.57) \times (-0.46) \\ &= 0.62 + 0.59 + 0.26 \\ &= \mathbf{1.47} \end{aligned}$$

Interpretation: Alice scores high on PC1 $\rightarrow$ she is taller and heavier for her age group.

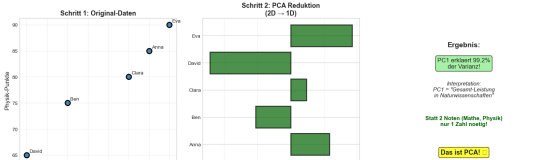
We reduced **3 numbers** to **1 number** that captures 92.7% of the information!

Projection is just a weighted sum — multiply each variable by its eigenvector weight and add up.

Complete Toy Example: Before and After PCA

Before PCA: 5 students $\times$ 3 variables $=$ 15 numbers

	H	W	A
Alice	1.08	1.00	−0.46
Bob	−0.72	−1.20	1.37
Carol	0.45	0.80	−0.91
Dave	−1.35	−0.90	0.73
Eve	0.54	0.30	−0.73



After PCA: 1 PC captures **92.7%**

$\rightarrow$ We can use PC1 scores for downstream analysis (e.g., clustering in Lesson 4).

PCA compressed 3 variables into 1 component that retains 92.7% of the original information.

PCA Checkpoint

What we have learned so far:

1. **PCA finds new axes** that align with the directions of maximum variance in the data
2. **Eigenvalue = importance score** — tells how much variance each component captures
3. **Eigenvector = recipe** — tells which original variables contribute to each component
4. **Fewer PCs = simpler data** — we keep the important components and drop the rest

Key Formula

$$\text{PC1 score} = w_1 \times z_1 + w_2 \times z_2 + \cdots + w_p \times z_p$$

where w_i are the eigenvector weights and z_i are the standardized variables.

PCA is a linear transformation — it rotates your data to align with the directions of maximum spread.

Checkpoint Quiz

Problem

A PCA on 4 variables yields eigenvalues: $\lambda_1 = 4.0$, $\lambda_2 = 2.0$, $\lambda_3 = 0.5$, $\lambda_4 = 0.3$.

Question 1: What percentage of variance does PC1 explain?

Checkpoint Quiz

Problem

A PCA on 4 variables yields eigenvalues: $\lambda_1 = 4.0$, $\lambda_2 = 2.0$, $\lambda_3 = 0.5$, $\lambda_4 = 0.3$.

Question 1: What percentage of variance does PC1 explain?

Answer: Total $= 4.0 + 2.0 + 0.5 + 0.3 = 6.8$. $\quad$ PC1 $= \dfrac{4.0}{6.8} = \mathbf{58.8}\%$

Question 2: How many PCs do you need to reach at least 80%?

Checkpoint Quiz

Problem

A PCA on 4 variables yields eigenvalues: $\lambda_1 = 4.0$, $\lambda_2 = 2.0$, $\lambda_3 = 0.5$, $\lambda_4 = 0.3$.

Question 1: What percentage of variance does PC1 explain?

Answer: Total $= 4.0 + 2.0 + 0.5 + 0.3 = 6.8$. $\quad$ PC1 $= \dfrac{4.0}{6.8} = \mathbf{58.8}\%$

Question 2: How many PCs do you need to reach at least 80%?

Answer: PC1 + PC2 $= \dfrac{4.0 + 2.0}{6.8} = \dfrac{6.0}{6.8} = \mathbf{88.2}\% \geq 80\%$.
Need **2 components**.

Note: Eigenvalues do not have to sum to the number of variables when extracting from a covariance matrix (here they sum to 6.8). When computed from a **correlation** matrix, they always sum to p.

These are the three key questions to ask after every PCA: how much, how many, what do they mean?

How Many Components? Three Rules of Thumb

There is no single "correct" answer — use multiple criteria together:

1. **Scree Plot Elbow Rule:** Plot eigenvalues. Keep components **before the elbow** (where the curve flattens).
2. **Kaiser's Rule** ($\lambda > 1$): Keep components with eigenvalue greater than 1 (they explain more than a single variable).
3. **Cumulative Variance** $\geq 80\%$: Keep enough components to explain at least 80% of total variance.

In practice

Apply all three. If they disagree, prefer the scree plot and consider your research context.

No single rule is perfect — convergence across methods gives you more confidence in your choice.

The Scree Plot

Definition

A **scree plot** graphs eigenvalues (vertical axis) against component number (horizontal axis). The "elbow" marks where additional components add little information.

Our running example:

- $\lambda_1 = 2.78$ (tall bar)
- $\lambda_2 = 0.18$ (short bar)
- $\lambda_3 = 0.04$ (tiny bar)

Discovery

Where is the elbow?

The Scree Plot

Definition

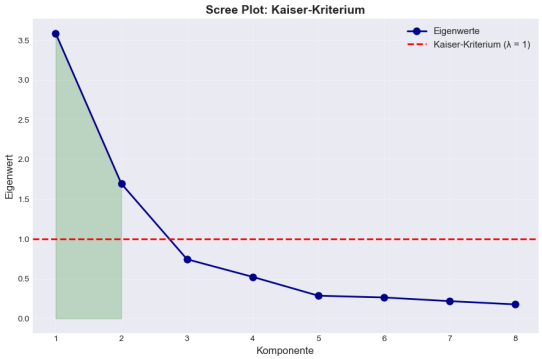
A **scree plot** graphs eigenvalues (vertical axis) against component number (horizontal axis). The "elbow" marks where additional components add little information.

Our running example:

- $\lambda_1 = 2.78$ (tall bar)
- $\lambda_2 = 0.18$ (short bar)
- $\lambda_3 = 0.04$ (tiny bar)

Discovery

Where is the elbow?
After component 1 — the drop from 2.78 to 0.18 is dramatic.



The scree plot gets its name from the "scree" (rubble at the base of a cliff) — keep what is above the rubble.

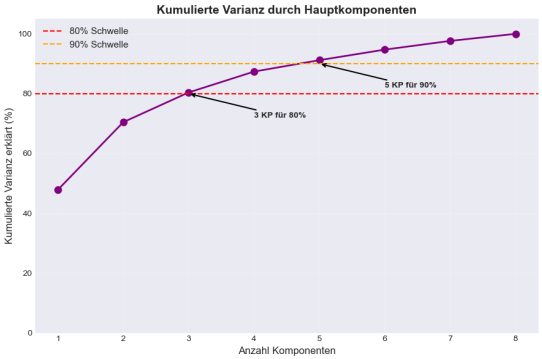
Kaiser's Rule and Cumulative Variance

Kaiser's Rule ($\lambda > 1$):

	λ	Keep?
PC1	2.78	✓ (> 1)
PC2	0.18	× (< 1)
PC3	0.04	× (< 1)

Cumulative variance:

	% Var	Cum. %
PC1	92.7%	92.7%
PC2	6.0%	98.8%
PC3	1.2%	100%



Verdict: All three methods agree:

- Scree elbow: after PC1
- Kaiser: only $\lambda_1 > 1$
- Cumulative: PC1 already $> 80\%$

$\rightarrow$ **Keep 1 component.**

When all criteria agree, the decision is clear. When they disagree, use judgment and domain knowledge.

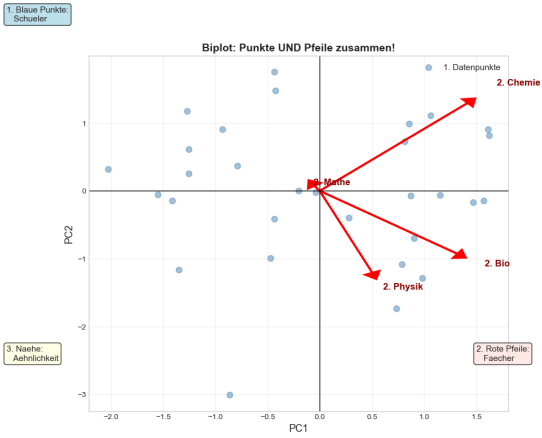
Reading the Biplot

A biplot shows observations AND variables together.

Step-by-step reading:

1. **Dots** = observations (students), positioned by their PC scores
2. **Arrows** = original variables, direction shows correlation with PCs
3. **Arrow length** = how well the variable is represented
4. **Arrow angle**: small angle between arrows → high positive correlation; 180° → negative correlation

Height and Weight arrows point the same way → $r = 0.92$.
Age arrow points opposite → $r(\text{W,A}) = -0.93$.



The biplot is the single most informative PCA graphic — learn to read it fluently.

The Loading Matrix

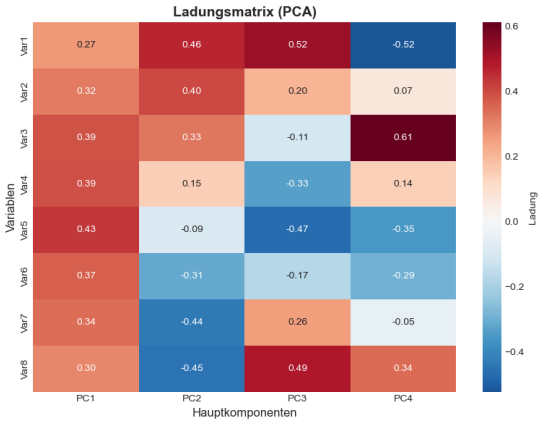
Loadings = correlations between variables and components.

Variable	PC1	PC2
Height	0.57	−0.72
Weight	0.59	0.03
Age	−0.57	−0.69

Interpreting PC1:

- Height: +0.57 (positive)
- Weight: +0.59 (positive)
- Age: −0.57 (negative)

→ PC1 = "**body size vs. maturity**"
High PC1 = tall, heavy, young.
Low PC1 = short, light, older.



Name your components based on the pattern of high loadings — this is where science meets statistics.

PCA in R: prcomp()

```r
# Create the data matrix
students <- data.frame(
  Height = c(1.08, -0.72, 0.45, -1.35, 0.54),
  Weight = c(1.00, -1.20, 0.80, -0.90, 0.30),
  Age    = c(-0.46, 1.37, -0.91, 0.73, -0.73)
)

# Run PCA (data already standardized)
pca <- prcomp(students, scale. = FALSE)

# View results
summary(pca)
# Standard deviation: 1.67  0.42  0.20
# Proportion:         0.927 0.060 0.012
# Cumulative:         0.927 0.988 1.000

# Loadings (rotation matrix)
pca$rotation
#         PC1    PC2    PC3
# H      0.57  -0.72   0.40
# W      0.59   0.03  -0.81
# A     -0.57  -0.69  -0.43
```

Key outputs:

- `summary()`: shows variance explained
- `pca$rotation`: eigenvectors (loadings)
- `pca$x`: PC scores for each student

Note: R reports *standard deviation*, not eigenvalue directly.

$\text{SD} = 1.67 \Rightarrow \lambda = 1.67^2 = 2.78$
$\Rightarrow$ explains $\frac{2.78}{3.00} = 92.7\%$.

Use `scale. = TRUE` when data is **not** pre-standardized.

prcomp() is the recommended R function for PCA — it uses SVD, which is numerically more stable.

Try It Yourself

Discovery Exercise

Open R (or RStudio) and run the following:

```r
# Load the famous iris dataset (built into R)
data(iris)

# Run PCA on the 4 numeric columns
pca_iris <- prcomp(iris[, 1:4], scale. = TRUE)

# View summary
summary(pca_iris)

# Plot the scree
plot(pca_iris, type = "l", main = "Iris Scree Plot")

# Biplot
biplot(pca_iris)
```

Questions to answer:

1. How many PCs have eigenvalue > 1? (Hint: square the standard deviations.)
2. What cumulative variance do the first 2 PCs explain?
3. In the biplot, which variables cluster together?

The iris dataset is a classic — 150 flowers, 4 measurements, 3 species. Perfect for practicing PCA.

Exploratory Factor Analysis

Finding hidden causes behind observed data

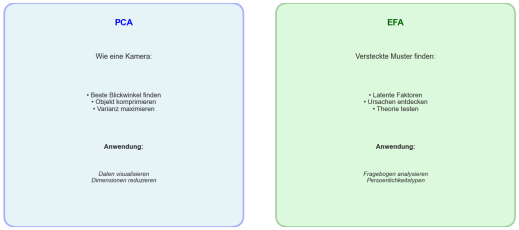
PCA Summarizes, EFA Explains

PCA: "What is the best summary?"

- Components are **computed from** the data
- Goal: compress into fewer variables
- No theory about causes

EFA: "What is the hidden cause?"

- Factors are **behind** the data
- Goal: explain why variables correlate
- Assumes latent structure exists



Definition

A **latent factor** is a hidden variable you cannot directly measure but that influences observable responses (e.g., "intelligence" influences test scores).

PCA: observed variables cause components. EFA: latent factors cause observed variables. The arrow is reversed.

Running Example: Restaurant Satisfaction Survey

10 survey items rated 1–5 by 300 customers:

1. "The food tasted delicious"
2. "The menu had good variety"
3. "Portions were generous"
4. "The ingredients were fresh"
5. "The waiter was friendly"
6. "Service was fast"
7. "Staff were knowledgeable"
8. "The atmosphere was pleasant"
9. "The restaurant was clean"
10. "The décor was attractive"

Discovery

Which items do you think belong together?

Running Example: Restaurant Satisfaction Survey

10 survey items rated 1–5 by 300 customers:

1. "The food tasted delicious"
2. "The menu had good variety"
3. "Portions were generous"
4. "The ingredients were fresh"
5. "The waiter was friendly"
6. "Service was fast"
7. "Staff were knowledgeable"
8. "The atmosphere was pleasant"
9. "The restaurant was clean"
10. "The décor was attractive"

Discovery

Which items do you think belong together?

- Items 1–4: **Food quality**
- Items 5–7: **Service quality**
- Items 8–10: **Ambiance**

EFA will discover these groups **automatically** from the correlation pattern!

EFA works by detecting clusters of correlated items and inferring hidden factors behind each cluster.

The Factor Model with Numbers

The model: Each observed item = weighted sum of factor scores + error.

$$X_i = \lambda_{i1} F_1 + \lambda_{i2} F_2 + \lambda_{i3} F_3 + e_i$$

Worked examples (after Varimax rotation):

$$\begin{aligned}
\text{Item 1 (Taste)} &= \mathbf{0.85} \times \text{Food} + 0.10 \times \text{Service} + 0.05 \times \text{Ambiance} + e_1 \\
\text{Item 5 (Friendly)} &= 0.12 \times \text{Food} + \mathbf{0.82} \times \text{Service} + 0.10 \times \text{Ambiance} + e_5 \\
\text{Item 10 (Décor)} &= 0.08 \times \text{Food} + 0.05 \times \text{Service} + \mathbf{0.85} \times \text{Ambiance} + e_{10}
\end{aligned}$$

Reading the loadings:

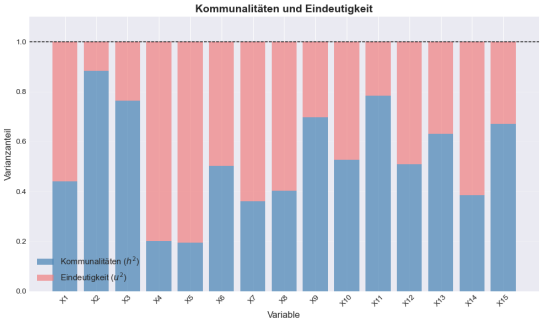
- Each item loads **strongly** on one factor (highlighted) and **weakly** on others
- "Taste" is 85% driven by Food quality, with minimal influence from Service or Ambiance

The factor loadings (λ) tell you how strongly each hidden factor influences each observed item.

Communality: How Well Is Each Item Explained?

Definition

Communality (h^2) is the proportion of an item's variance explained by *all* factors combined. h^2 = sum of squared loadings.



Worked examples:

Item 1 (Taste):
$h^2 = 0.85^2 + 0.10^2 + 0.05^2$
$= 0.7225 + 0.0100 + 0.0025 = \mathbf{0.7350}$
$\rightarrow$ 73.5% explained by the 3 factors.

Item 5 (Friendly):
$h^2 = 0.12^2 + 0.82^2 + 0.10^2$
$= 0.0144 + 0.6724 + 0.0100 = \mathbf{0.6968}$
$\rightarrow$ 69.7% explained.

Item 10 (Décor):
$h^2 = 0.08^2 + 0.05^2 + 0.85^2$
$= 0.0064 + 0.0025 + 0.7225 = \mathbf{0.7314}$
$\rightarrow$ 73.1% explained.

Rule of thumb:

- $h^2 > 0.50$: item fits well
- $h^2 < 0.30$: consider dropping

Communality = shared variance. The remainder ($1 - h^2$) is unique variance plus error.

How Many Factors? Parallel Analysis

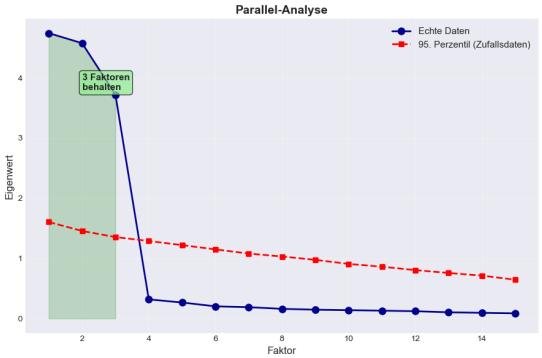
Problem: Kaiser's rule ($\lambda > 1$) and scree plots are subjective. Parallel analysis is more rigorous.

How it works:

1. Generate random data (same n and p)
2. Extract eigenvalues from the random data
3. Repeat 1000 times; take the 95th percentile
4. Keep factors whose real eigenvalue exceeds the random threshold

Our example:

- 3 real eigenvalues > random threshold
- 4th eigenvalue < random threshold
- → Retain **3 factors**



Parallel analysis is the gold standard for determining the number of factors — it controls for sampling noise.

Why Rotate? Making Factors Interpretable

Unrotated factors:

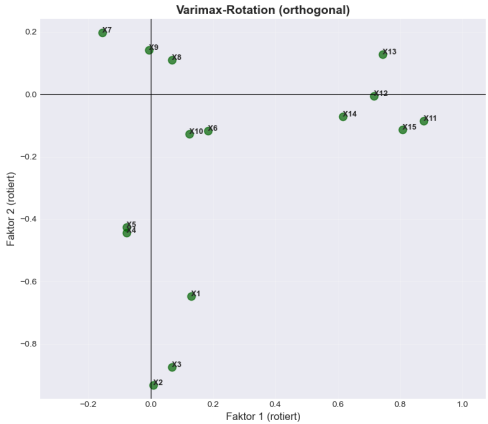
Item	F1	F2
Taste	0.60	0.50
Décor	0.55	0.45

→ Items load on **both** factors. Hard to interpret!

After Varimax rotation:

Item	F1	F2
Taste	**0.85**	0.10
Décor	0.08	**0.85**

→ Each item loads on **one** factor. Clear!



Discovery

Which is easier to interpret?

Why Rotate? Making Factors Interpretable

Unrotated factors:

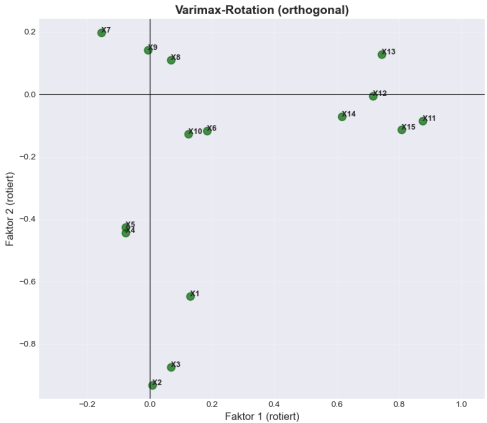
Item	F1	F2
Taste	0.60	0.50
Décor	0.55	0.45

→ Items load on **both** factors. Hard to interpret!

After Varimax rotation:

Item	F1	F2
Taste	**0.85**	0.10
Décor	0.08	**0.85**

→ Each item loads on **one** factor. Clear!



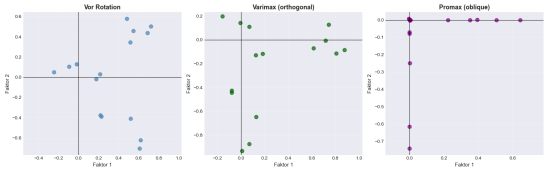
Discovery

Which is easier to interpret? The rotated one — each item belongs to one factor.

Rotation does not change the total variance explained — it redistributes it to make factors clearer.

Varimax vs. Promax Rotation

Varimax (Orthogonal)	Promax (Oblique)
Factors are **uncorrelated**	Factors may be **correlated**
Simpler interpretation	More realistic
Constrains factor angles to 90°	Allows any angle
Use when: factors should be independent	Use when: factors may overlap



Example:

- "Food" and "Service" in restaurants are likely **correlated** (good restaurants excel at both)
- → Promax may be more appropriate here

Start with Varimax for simplicity; switch to Promax if you suspect factors are correlated.

Interpreting the Full Loading Matrix

Complete loading matrix (after Varimax):

Item	Food	Service	Ambiance
1. Taste	**0.85**	0.10	0.05
2. Variety	**0.78**	0.15	0.12
3. Portions	**0.72**	0.08	0.10
4. Freshness	**0.80**	0.05	0.15
5. Friendly	0.12	**0.82**	0.10
6. Fast	0.08	**0.75**	0.05
7. Knowledge	0.15	**0.70**	0.18
8. Atmosphere	0.10	0.12	**0.80**
9. Clean	0.05	0.20	**0.73**
10. Décor	0.08	0.05	**0.85**

Naming the factors:

- **Factor 1:** Items 1–4 load high → **"Food Quality"**
- **Factor 2:** Items 5–7 load high → **"Service Quality"**
- **Factor 3:** Items 8–10 load high → **"Ambiance"**

Rule: Highlight loadings > 0.50; consider cross-loadings > 0.30 as problematic.

Clean simple structure: every item belongs to exactly one factor — easy to interpret and name.

Factor naming is a substantive judgment — the statistics tell you which items cluster, you give the cluster a name.

Good vs. Bad Simple Structure

Definition

Simple structure means each item loads strongly on exactly one factor and weakly on all others.

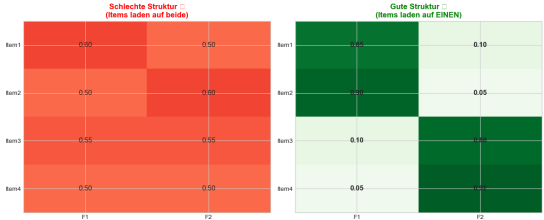
Good simple structure:

	F1	F2	F3
Item A	**0.82**	0.05	0.10
Item B	0.08	**0.79**	0.12
Item C	0.11	0.06	**0.85**

Bad simple structure:

	F1	F2	F3
Item A	0.55	0.48	0.30
Item B	0.40	0.42	0.45
Item C	0.35	0.50	0.38

→ Items load on **multiple** factors. Uninterpretable.



What to do about bad structure:

- Try a different rotation
- Try fewer or more factors
- Remove cross-loading items
- Reconsider your survey design

Simple structure is the goal of rotation — each item should "belong" to exactly one factor.

EFA in R: `fa()` from the psych Package

```r
library(psych)

# Check sampling adequacy
KMO(data)       # KMO > 0.6 is acceptable
cortest.bartlett(cor(data), n = nrow(data))

# Run EFA with 3 factors, Varimax rotation
efa <- fa(data, nfactors = 3,
          rotate = "varimax",
          fm = "pa")  # principal axis factoring

# View loadings (suppress small ones)
print(efa$loadings, cutoff = 0.3)

# Communalities
efa$communalities

# Parallel analysis to decide nfactors
fa.parallel(data)
```

Definition

KMO (Kaiser-Meyer-Olkin) measures whether correlations are strong enough for factor analysis. Range: 0–1.

KMO interpretation:

- > 0.90: Marvelous
- 0.80–0.90: Meritorious
- 0.60–0.80: Acceptable
- < 0.60: Unacceptable — do not run EFA

Bartlett's test: $p < 0.05$ means correlations are significantly different from zero.

Always check KMO and Bartlett's test before running EFA — they confirm your data is suitable.

When to Use Which?

PCA vs. EFA — choosing the right tool

Discovery: Match the Scenario

For each scenario, decide: PCA or EFA?

1. You have 50 gene expression variables and need to reduce them to a few features for a regression model.
2. You designed a 20-item personality questionnaire and want to discover what traits it measures.
3. You want to create a composite index from 12 economic indicators.
4. You want to test whether your anxiety scale really measures two distinct constructs (cognitive vs. somatic anxiety).

Discovery: Match the Scenario

For each scenario, decide: PCA or EFA?

1. You have 50 gene expression variables and need to reduce them to a few features for a regression model.
2. You designed a 20-item personality questionnaire and want to discover what traits it measures.
3. You want to create a composite index from 12 economic indicators.
4. You want to test whether your anxiety scale really measures two distinct constructs (cognitive vs. somatic anxiety).

Answers:

1. **PCA** — goal is data compression for downstream modeling
2. **EFA** — goal is discovering latent traits behind the items
3. **PCA** — goal is creating a summary index
4. **EFA** — goal is testing a hypothesis about latent structure

---

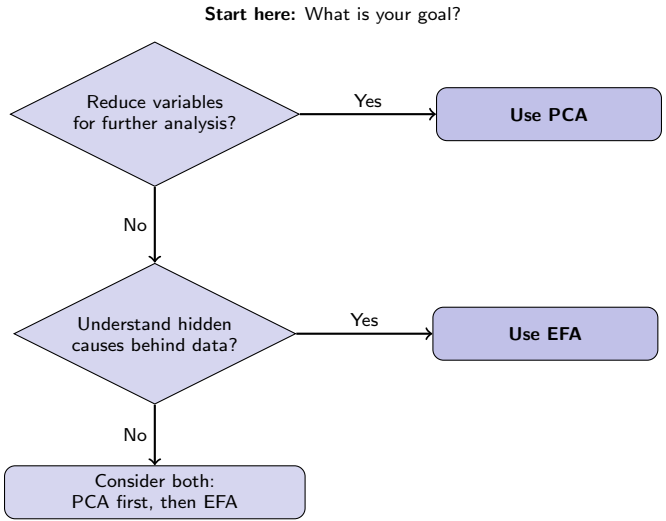
PCA = compression for further analysis. EFA = understanding hidden structure in survey/test data.

PCA vs. EFA: Side-by-Side Comparison

Aspect	PCA	EFA
Goal	Data compression	Discover latent structure
Output	Components	Factors
Direction	Data $\rightarrow$ components	Factors $\rightarrow$ data
Unique variance	All variance used	Shared variance only
Error term	None	Explicit (e_i)
Rotation	Optional	Essential
Number of outputs	= number of variables	Researcher chooses
R function	`prcomp()`	`psych::fa()`
Typical use	Feature reduction, indices	Scale development, theory
Assumptions	None (descriptive)	Latent factors exist

Neither is "better" — they answer different questions. Choose based on your research goal.

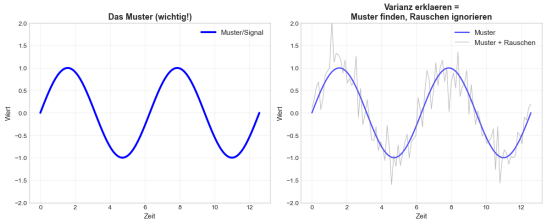
Decision Flowchart: PCA or EFA?



When in doubt, run both and compare — they often reveal complementary insights about your data.

Key Takeaways

1. **PCA** finds new axes that maximize variance — it **compresses** data without assuming hidden causes
2. **EFA** discovers latent factors that **explain** why variables correlate — it assumes hidden structure
3. **Eigenvalues** tell you importance; **eigenvectors/loadings** tell you meaning
4. Use scree plot, Kaiser's rule, and cumulative variance (or parallel analysis) to decide **how many** components/factors to keep



PCA = best summary. EFA = hidden explanation. Both reduce dimensionality; they differ in why.

What We Computed Today

PCA running example (5 students, 3 variables):

- Standardized data $\rightarrow$ Correlation matrix $\rightarrow$ Eigenvalues [2.78, 0.18, 0.04]
- PC1 captures 92.7% of variance with weights $[0.57, 0.59, -0.57]$
- Alice's PC1 score: $0.57 \times 1.08 + 0.59 \times 1.00 + (-0.57) \times (-0.46) = 1.47$

EFA running example (10-item restaurant survey):

- 3 factors discovered: Food Quality, Service Quality, Ambiance
- Each item loads strongly on exactly one factor (simple structure achieved)
- Communalities range from 69.7% to 73.5% — all items well-explained

Both methods reduced complex data to a few meaningful dimensions.

The worked examples show PCA and EFA are complementary tools for dimensionality reduction.

Connections to Other Lessons

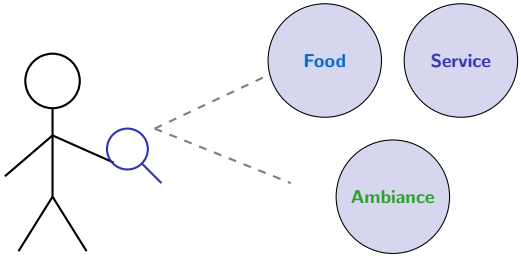
- **Lesson 1 (Regression):** Use PC scores as predictors to avoid multicollinearity when original variables are highly correlated.
- **Lesson 4 (Clustering):** PCA is commonly used as a **preprocessing step** before clustering — reduce 50 variables to 5 PCs, then cluster on those PCs.
- **Lesson 5 (Time Series):** Factor models appear in **state-space models** and dynamic factor analysis for multivariate time series.

Practical Workflow

1. Run PCA or EFA to reduce dimensions
2. Use the resulting scores as inputs for regression (L1), clustering (L4), or time series (L5)
3. Report both the variance explained **and** the interpretation of components/factors

Dimensionality reduction is rarely an end in itself — it prepares data for further analysis.

Questions?



Thank you!

Principal Component Analysis and Exploratory Factor Analysis
Statistical Data Analysis — Lesson 3

PCA finds the best summary angles; EFA finds the hidden causes. Now you know both.

Appendix

Mathematical details and additional reference material

A2: Eigenvalue Decomposition — Worked 2×2 Example

Given the 2×2 correlation matrix: $\mathbf{R} = \begin{pmatrix} 1.0 & 0.8 \\ 0.8 & 1.0 \end{pmatrix}$

Step 1: Solve $\det(\mathbf{R} - \lambda \mathbf{I}) = 0$

$$\det \begin{pmatrix} 1-\lambda & 0.8 \\ 0.8 & 1-\lambda \end{pmatrix} = (1-\lambda)^2 - 0.64 = 0$$

$(1-\lambda)^2 = 0.64 \Rightarrow 1 - \lambda = \pm 0.8 \Rightarrow \lambda_1 = 1.8,\ \lambda_2 = 0.2$

Step 2: Find eigenvectors. For $\lambda_1 = 1.8$:

$$\begin{pmatrix} -0.8 & 0.8 \\ 0.8 & -0.8 \end{pmatrix} \mathbf{v} = 0 \Rightarrow v_1 = v_2 \Rightarrow \mathbf{v}_1 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \approx \begin{pmatrix} 0.71 \\ 0.71 \end{pmatrix}$$

Interpretation: PC1 $= 0.71X_1 + 0.71X_2$ (equal-weight average), explaining $\frac{1.8}{2.0} = 90\%$ of variance.

The eigenvalue equation $\det(\mathbf{R} - \lambda \mathbf{I}) = 0$ is solved by the computer — this shows what happens inside.

A3: Singular Value Decomposition and PCA

Any data matrix $\mathbf{X}$ $(n \times p)$ can be decomposed as:

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$$

Components of SVD:

- $\mathbf{U}$ $(n \times n)$: left singular vectors (observation patterns)
- $\mathbf{\Sigma}$ $(n \times p)$: diagonal matrix of singular values
- $\mathbf{V}$ $(p \times p)$: right singular vectors = **eigenvectors of $\mathbf{X}^\top\mathbf{X}$**

Connection to PCA:

- Eigenvalues: $\lambda_i = \frac{\sigma_i^2}{n-1}$ where σ_i are singular values
- Eigenvectors: columns of $\mathbf{V}$
- PC scores: $\mathbf{T} = \mathbf{X}\mathbf{V} = \mathbf{U}\mathbf{\Sigma}$

Why SVD? It is numerically more stable than computing $\mathbf{X}^\top\mathbf{X}$ directly. This is why `prcomp()` uses SVD internally.

SVD and eigendecomposition give identical PCA results — SVD is preferred for numerical stability.

A4: Covariance Matrix vs. Correlation Matrix

Covariance matrix S:

- Uses raw (unstandardized) data
- Diagonal: variances s_i^2
- Variables with large variance dominate
- Use when: variables share the same unit and scale

Correlation matrix R:

- Uses standardized data (z-scores)
- Diagonal: all 1's
- All variables equally weighted
- Use when: variables have different units or scales

Example: If Height is in cm (variance ~ 100) and Weight in kg (variance ~ 25), Height dominates PC1.

In R:
`prcomp(X, scale.=FALSE)` $\rightarrow$ covariance
`prcomp(X, scale.=TRUE)` $\rightarrow$ correlation

Rule of thumb: When in doubt, use the **correlation** matrix (`scale.=TRUE`).

Most applications use the correlation matrix — it prevents variables with large variance from dominating.

A5: princomp() vs. prcomp() in R

Feature	prcomp()	princomp()
Algorithm	SVD	Eigendecomposition of cov
Numerical stability	Higher	Lower
Standardization	scale.=TRUE	Must do manually
Output: loadings	pca$rotation	pca$loadings
Output: scores	pca$x	pca$scores
Output: eigenvalues	pca$sdev^2	pca$sdev^2
Recommendation	**Preferred**	Legacy

```r
# Recommended approach
pca <- prcomp(data, scale. = TRUE)  # SVD-based, stable

# Legacy approach (avoid for new code)
pca_old <- princomp(data, cor = TRUE)  # eigendecomposition-based
```

Use prcomp() for all new analyses — it is more numerically stable and has a cleaner interface.

A6: Principal Axis Factoring Algorithm

How does EFA extract factors? Principal Axis Factoring (PAF):

1. Start with the correlation matrix **R**
2. Replace the diagonal (1's) with **initial communality estimates** (e.g., squared multiple correlations)
3. Extract eigenvalues/eigenvectors from this **reduced** matrix
4. Update communalities using the extracted loadings: $h_i^2 = \sum_j \lambda_{ij}^2$

Repeat steps 2–4 until communalities **converge** (change < 0.001).

Key difference from PCA:

- PCA uses the **full** correlation matrix (diagonal $= 1$)
- PAF uses a **reduced** matrix (diagonal $=$ communalities < 1)
- PAF analyzes only **shared variance**; PCA analyzes **total variance**

In R: `fa(data, fm="pa")` uses PAF; `fa(data, fm="ml")` uses maximum likelihood.

PAF iterates to separate shared variance from unique/error variance — this is what makes EFA different from PCA.

A7: KMO and Bartlett's Test — Details

Kaiser-Meyer-Olkin (KMO) measure:

$$\text{KMO} = \frac{\sum\sum_{i \neq j} r_{ij}^2}{\sum\sum_{i \neq j} r_{ij}^2 + \sum\sum_{i \neq j} a_{ij}^2}$$

where r_{ij} = correlation, a_{ij} = partial correlation.

Interpretation:

- High KMO: correlations are "diffuse" (shared factors)
- Low KMO: correlations are "concentrated" (pairwise only, no factors)

Bartlett's test of sphericity:

- H_0: Correlation matrix = identity (no correlations)
- H_1: At least some correlations $\neq 0$
- Must reject H_0 ($p < 0.05$) before running EFA

Decision matrix:

	KMO $\geq$ 0.6	**KMO $<$ 0.6**
Bartlett $p < .05$	Run EFA	Caution
Bartlett $p \geq .05$	Caution	Do not run

Both tests must pass before running EFA — KMO checks adequacy, Bartlett checks whether correlations exist.

A8: Promax Rotation and Factor Correlations

Promax rotation (oblique):

1. Start with the Varimax solution
2. Raise loadings to a power (κ, typically 4) to sharpen high/low contrast
3. Allow the factor axes to rotate freely (no 90° constraint)

Result: Two types of loadings:

- **Pattern matrix**: direct effects of factors on items (interpret this one)
- **Structure matrix**: correlations between items and factors (includes indirect effects)

Factor correlation matrix (Promax output):

	Food	**Service**	**Amb.**
Food	1.00	0.35	0.20
Service	0.35	1.00	0.28
Ambiance	0.20	0.28	1.00

Food and Service correlate at $r = 0.35$ — restaurants with good food often have good service too.

If factor correlations < 0.3: Varimax and Promax give similar results.

Use the pattern matrix for interpretation. Report factor correlations to show how factors relate.

A9: Complete R Code — PCA and EFA Walkthrough

PCA workflow:

```r
# 1. Load and inspect data
data <- read.csv("measurements.csv")
str(data); cor(data)

# 2. Run PCA
pca <- prcomp(data, scale. = TRUE)
summary(pca)

# 3. Scree plot
plot(pca, type = "l",
     main = "Scree Plot")
abline(h = 1, col = "red", lty = 2)

# 4. Biplot
biplot(pca, scale = 0,
       cex = 0.7)

# 5. Extract scores
scores <- pca$x[, 1:2]  # keep PC1-PC2
```

EFA workflow:

```r
library(psych)

# 1. Check adequacy
KMO(data)
cortest.bartlett(cor(data),
                 n = nrow(data))

# 2. Determine number of factors
fa.parallel(data, fa = "fa")

# 3. Run EFA
efa <- fa(data,
          nfactors = 3,
          rotate = "varimax",
          fm = "pa")

# 4. Inspect results
print(efa$loadings, cutoff = 0.3)
efa$communalities

# 5. Visualize
fa.diagram(efa)
```

These workflows are templates — adapt them to your specific data and research question.

A10: Further Reading and Resources

Textbooks:

- Field, A. (2018). *Discovering Statistics Using IBM SPSS Statistics*, 5th ed. — Ch. 17–18 cover PCA and EFA with clear examples.
- Tabachnick, B. & Fidell, L. (2019). *Using Multivariate Statistics*, 7th ed. — comprehensive treatment of factor analysis.

R packages and documentation:

- `psych` package: `fa()`, `fa.parallel()`, `KMO()`, `fa.diagram()` — the standard EFA toolkit in R.
- `factoextra` package: `fviz_pca_biplot()`, `fviz_screeplot()` — publication-quality PCA visualizations.

Key concepts to review:

- Linear algebra: eigenvectors, eigenvalues, matrix decomposition
- Correlation vs. covariance matrices
- Confirmatory Factor Analysis (CFA) — the next step after EFA for hypothesis testing

---

For CFA (confirmatory factor analysis), see the lavaan package in R — it tests whether your EFA structure holds.