Principal Component Analysis (PCA) and Factor Analysis in R: Comprehensive Guide

Dani

1 Overview of PCA and Factor Analysis

Principal Component Analysis (PCA) and **Factor Analysis (FA)** are both multivariate techniques aimed at dimensionality reduction, but with different goals:

- **PCA** focuses on maximizing the variance captured by each component, making it ideal for visualizing high-dimensional data and identifying major sources of variation.
- **FA** seeks to identify latent factors that explain observed variable correlations, making it more suitable for uncovering hidden structures or constructs within the data.

1.1 Choosing Between PCA and FA

- **Use PCA when**:
  - You aim to capture as much variance as possible.
  - Data is numeric and relationships are linear.
  - You are more interested in reducing dimensionality for visualization purposes.
- **Use FA when**:
  - The goal is to identify underlying latent variables.
  - You want to explore hidden structures or groupings within the data.
  - Data might be non-normal or categorical, requiring alternative approaches like polychoric FA.

2 PCA on USArrests Data using `princomp`

- `princomp()` uses spectral decomposition on the covariance or correlation matrix.
- The choice between using a correlation matrix (`cor = TRUE`) or covariance matrix depends on whether the data needs to be standardized.
- Example:

```
data <- USArrests
pcUSA <- princomp(data, cor = TRUE)
summary(pcUSA)

# Eigenvalues of the principal components
eigs <- pcUSA$sdev^2
plot(eigs, type = "b", main = "Scree Plot")

# Loading matrix and biplot
pcUSA$loadings
biplot(pcUSA)
```

- **Interpretation**:

- The **Scree plot** helps visualize the proportion of variance explained by each component.
- `pcUSA$loadings` provides the contribution of each variable to the components.
- The first few components should ideally capture the majority of the variance.

2.1 PCA using `prcomp`

- `prcomp()` is preferred over `princomp()` when the dataset is large, as it uses Singular Value Decomposition (SVD).
- It provides a numerically stable solution, especially when dealing with highly correlated variables.
- Example:

```
prusa <- prcomp(data, scale. = TRUE)
summary(prusa)

# Eigenvalues
prusa$sdev^2

# Biplot to visualize the scores and loadings
biplot(prusa)
```

- **Key Consideration**:
  - Use ‘prcomp()‘ when numerical stability is a concern, especially for high-dimensional datasets.

3 Factor Analysis using `FactoMineR`

- `FactoMineR` is a versatile package for advanced factor analysis and PCA.
- It provides additional outputs like communalities, cosines, and variable contributions.
- Example on `USArrests`:

```
library(FactoMineR)
usafa <- PCA(USArrests)
summary(usafa)

# Accessing eigenvalues and variable loadings
usafa$eig
usafa$var$coord
```

- **Practical Tip**:
  - Use `FactoMineR` when you need to assess multiple dimensions, visualize variable contributions, or perform hierarchical clustering on principal components.

4 Bartlett’s Test and KMO Index using `psych`

- **Bartlett’s Test of Sphericity** tests if the correlation matrix is an identity matrix, indicating if the variables are sufficiently correlated to perform factor analysis.
- Example using the Food Price dataset:

```
library(psych)
R <- cor(food)
cortest.bartlett(R, n = nrow(food))
```

- **Kaiser-Meyer-Olkin (KMO) Index** assesses sampling adequacy.

- Values range from 0 to 1, with values above 0.6 indicating that the data is factorable.

```
kmo(food)
```

- **Interpretation of KMO**:
  - KMO > 0.9: Excellent
  - 0.7 > KMO > 0.9: Good to average
  - KMO < 0.5: Unacceptable

5 Advanced PCA on Food Price Data

- **Test of Factorability** is essential before PCA to ensure that the variables are correlated.
- PCA is performed using the correlation matrix when variables are on different scales.

```
pca_food <- princomp(food, cor = TRUE)
summary(pca_food)

# Loadings and visualization
plot(pca_food, type = "lines", main = "Scree Plot for Food Prices")
biplot(pca_food)
```

- **Interpretation**:
  - If the first two components explain a significant proportion of variance (e.g., ¿70%), use them for further analysis.
  - Biplots provide insight into how variables contribute to each component.

6 Component and Factor Loadings

- **Factor Loadings** indicate how much a variable contributes to a factor. Higher loadings suggest a stronger relationship with the factor.
- Example:

```
pcUSA$loadings
```

- **Communalities** measure the proportion of variance a variable shares with all components:

```
com1 <- (pcUSA$loadings[1,]*sqrt(eigs))^2
sum(com1[1:2])  # Sum of communalities for the first two components
```

7 Factor Analysis of Heptathlon Data

- Factor Analysis is performed on the `heptathlon` dataset, taking into account different directions for specific variables.
- Example:

```
data("heptathlon", package = "HSAUR")
pca_hep <- PCA(heptathlon, quanti.sup = 8)
plot(pca_hep$eig[,1], type = "o", main = "Scree Plot for Heptathlon
    Data")
```

- **Practical Tip**:
  - Supplementary variables and individuals (`ind.sup`, `quanti.sup`) can be included to analyze additional points without affecting the main analysis.