I. **Factor Analysis (FA)** is a statistical method used to uncover the underlying structure of a set of variables. It aims to explain correlations among many variables in terms of a smaller number of unobservable (latent) variables called factors¹. Here are the steps for performing factor analysis in R:

1. \*\*Data Preparation\*\*:

- Standardize the data (scale the variables) to ensure they are on a common scale and have equal weight in the analysis.

- For example, you can use the `scale()` function to prepare your data.

2. \*\*Factor Extraction\*\*:

- Identify factors based on their ability to explain variance in the data.

- Common methods for extracting factors include Principal Components Analysis (PCA), Maximum Likelihood Estimate (MLE), and Minimum Residuals (MR).

3. \*\*Factor Rotation\*\*:

- Rotate the factors to make their interpretation easier.

- Varimax rotation is a common method that maximizes the variance of factor loadings.

4. \*\*Factor Interpretation\*\*:

- Interpret the factors and their loadings (correlation between each variable and each factor).

- Loadings represent how strongly each variable is associated with each factor.

To demonstrate, let's use the built-in `iris` dataset in R. First, load the dataset and scale the variables. Then, determine the number of factors to extract (e.g., using the Kaiser criterion). Finally, perform factor analysis with rotation (e.g., Varimax) and interpret the results¹. If you need further assistance or have specific questions, feel free to ask! 😊

Source: Conversation with Copilot, 7/25/2024

(1) Factor Analysis in R programming - GeeksforGeeks. https://www.geeksforgeeks.org/factor-analysis-in-r-programming/.

(2) Exploratory Factor Analysis in R (Example) - Statistics Globe. https://statisticsglobe.com/exploratory-factor-analysis-r.

(3) Factor Analysis in R Course | DataCamp. https://www.datacamp.com/courses/factor-analysis-in-r.

(4) MFA in R Using FactoMineR: Quick Scripts and Videos. http://www.sthda.com/english/articles/22-principal-component-methods-videos/73-mfa-in-r-using-factominer-quick-scripts-and-videos/.

(5) Quick-R: Factor Analysis. https://www.statmethods.net/advstats/factor.html.

II. **Varimax rotation method** - In the R programming language the varimax method is implemented in several packages including stats (function varimax( )), or in contributed packages including GPArotation or psych.

III. **Principal Components Analysis (PCA)** – In R, the functions princomp and prcomp can be used for principal component analysis; prcomp uses singular value decomposition which generally gives better numerical accuracy.

IV. **The Kaiser – Guttman** method was also employed to determine the number of factors with eigenvalues greater than 1.

Returns the count of the number of eigenvalues greater than 1 in a correlation matrix. This value is often referred to as the "Kaiser", "Kaiser-Guttman", or "Guttman-Kaiser" rule for determining the number of components or factors in a correlation matrix.

The **Kaiser-Guttman criterion** is a popular factor retention rule in **Principal Component Analysis (PCA)**. [It suggests retaining as many principal components (PCs) as there are sample eigenvalues greater than 11](https://r.qcbs.ca/workshop09/book-en/principal-component-analysis.html). Here’s how you can calculate it in R:

1. **Load Data**: First, load your dataset. [For example, let’s use the varechem dataset containing measurements of chemical properties of soil samples1](https://r.qcbs.ca/workshop09/book-en/principal-component-analysis.html). You can select specific variables from this dataset.
2. **Standardize Data**: Standardize the data to have mean zero and unit variance. This step ensures that all variables are on the same scale:

|  |
| --- |
| **R** |
| # Load the datasets package  library(datasets)  # Load the varechem dataset  data(varechem)  # Select data (e.g., first two variables N and P)  data <- varechem[, 1:2]  # Standardize data  data\_std <- scale(data) |

1. **Compute Covariance Matrix**: Next, compute the covariance matrix of the standardized data:

|  |
| --- |
| **R** |
| cov\_matrix <- cov(data\_std) |

1. **Extract Eigenvalues**: Calculate the eigenvalues associated with the principal components:

|  |
| --- |
| **R** |
| eigenvalues <- eigen(cov\_matrix)$values |

1. **Apply Kaiser-Guttman Criterion**: Retain the principal components with eigenvalues greater than 1:

|  |
| --- |
| **R** |
| retained\_pcs <- eigenvalues[eigenvalues > 1] |

**V. The Kaiser-Meyer-Olkin (KMO)** **test** is a statistical measure to determine how suited data is for [factor analysis](https://en.wikipedia.org/wiki/Factor_analysis). The test measures sampling adequacy for each variable in the model and the complete model. The statistic is a measure of the proportion of variance among variables that might be common variance. The higher the proportion, the higher the KMO-value, the more suited the data is to factor analysis.[[1]](https://en.wikipedia.org/wiki/Kaiser%E2%80%93Meyer%E2%80%93Olkin_test#cite_note-ibm-1)

Kaiser proposed that a KMO > 0.9 was marvelous, in the 0.80s, meritorious, in the 0.70s, middling, in the 0.60s, mediocre, in the 0.50s, miserable, and less than 0.5 would be unacceptable. [[3]](https://en.wikipedia.org/wiki/Kaiser%E2%80%93Meyer%E2%80%93Olkin_test#cite_note-k2-3) In general, KMO values between 0.8 and 1 indicate the sampling is adequate. KMO values less than 0.6 indicate the sampling is not adequate and that remedial action should be taken. In contrast, others set this cutoff value at 0.5.[[5]](https://en.wikipedia.org/wiki/Kaiser%E2%80%93Meyer%E2%80%93Olkin_test#cite_note-review1974-5) A KMO value close to zero means that there are large partial correlations compared to the sum of correlations. In other words, there are widespread correlations which would be a large problem for factor analysis.[[1]](https://en.wikipedia.org/wiki/Kaiser%E2%80%93Meyer%E2%80%93Olkin_test#cite_note-ibm-1)

Example in R:

|  |
| --- |
| library(psych)  set.seed(5L)  five.samples <- data.frame("A"=rnorm(100), "B"=rnorm(100), "C"=rnorm(100),                      "D"=rnorm(100), "E"=rnorm(100))  cor(five.samples)  KMO(five.samples) |
| Kaiser-Meyer-Olkin factor adequacy  Call: KMO(r = five.samples)  Overall MSA = 0.53  MSA for each item =  A B C D E  0.52 0.56 0.52 0.48 0.54 |

VI. The **Bartlett’s Test of Sphericity** is a statistical test that compares an observed correlation matrix to the identity matrix. Essentially, it checks whether there is redundancy between variables that can be summarized with a few factors. Here’s how it works:

1. **Correlation Matrix**: A correlation matrix shows the correlation coefficients between variables. Each value in the matrix represents the correlation between two variables, ranging from -1 to 1. The further a value is from 0, the stronger the correlation between the variables.
2. **Identity Matrix**: An identity matrix has 1s along the diagonal and 0s elsewhere. If the numbers in this matrix represent correlation coefficients, it means that each variable is perfectly orthogonal (uncorrelated) to every other variable.
3. **Purpose of Bartlett’s Test**: We conduct Bartlett’s Test of Sphericity to ensure that the correlation matrix of our dataset significantly diverges from the identity matrix. If it does, data reduction techniques like Principal Component Analysis (PCA) or factor analysis can meaningfully compress the data.
4. **How to Conduct Bartlett’s Test of Sphericity in R**:
   * Use the cortest.bartlett() function from the psych library.
   * Syntax: cortest.bartlett(R, n)
     + R: Correlation matrix of the dataset.
     + n: Sample size.
   * Example code:

|  |
| --- |
| **R** |
| # Create a fake dataset (replace with your own data)  set.seed(0)  data <- data.frame(A = rnorm(50, 1, 4), B = rnorm(50, 3, 6), C = rnorm(50, 5, 8))  # Conduct Bartlett's Test  result <- cortest.bartlett(cor(data), nrow(data))  # Check p-value  if (result$p < 0.05) {  cat("Dataset suitable for data reduction technique.\n")  } else {  cat("Consider other approaches.\n")  } |

VII. Another technique used was the **k-means clustering method**, which helped identify different segments in religious tourism.

**VIII. The Kruskal-Wallis** test was used to find differences between the means of segments.

[R](https://en.wikipedia.org/wiki/R_(programming_language)) base-package has an implement of this test using kruskal.test.

**IX. The Mann-Whitney U test** was used to determine where these differences were found.

A black background with pink text

Description automatically generated