PRACTICE #04

Factor Analysis

(*Keyword:* Factor Analysis*)*

I. Goals

- Students can use **Python** with **scikit-learn** library to implement Factor Analysis.

II. Introduction

Factor Analysis is a statistical technique used to uncover the underlying structure in a set of variables. It is commonly used in psychology, sociology, market research, and other fields where understanding the relationships between variables is crucial.

- Preprocess the data: This might involve handling missing values, scaling the variables, and checking for normality.
- Conduct Exploratory Factor Analysis (EFA):
  - Choose a suitable method for factor extraction (e.g., Principal Component Analysis, Maximum Likelihood).
  - Determine the number of factors to retain using techniques like Kaiser's criterion or the scree plot.
  - Interpret the extracted factors based on the loadings on the original variables.
- Evaluate the model fit: Use fit indices like the Kaiser-Meyer-Olkin (KMO) measure of sampling adequacy and Bartlett's test of sphericity to assess the suitability of the data for factor analysis.
- Key Concepts:
  - **Factors**: Factors are underlying variables that cannot be directly observed but are inferred from observed variables. For example, in a study measuring academic performance, factors like intelligence, motivation, and study habits might influence test scores.
  - **Eigenvalues and Eigenvectors**: Eigenvalues represent the variance explained by each factor, while eigenvectors represent the direction or pattern of the variables. Eigenvalues greater than 1 are considered significant in Factor Analysis.
  - **Factor Loading**: Factor loading indicates the correlation between each variable and the underlying factor. Higher factor loadings suggest stronger relationships between variables and factors.

III. Content

1. Dataset
   - Origin: https://vincentarelbundock.github.io/Rdatasets/csv/psych/bfi.csv
   - Description: BFI (dataset based on personality assessment project), which was collected using a 6-point response scale: 1 Very Inaccurate, 2 Moderately Inaccurate, 3 Slightly Inaccurate 4 Slightly Accurate, 5 Moderately Accurate, and 6 Very Accurate.

2. Based on the **Additional guidance** below**,** students report the following requests
   - Interpretation of the extracted factors in your own words, explaining what each factor represents based on the variable loadings.
   - Understand model fit indices and their implications.
   - Conclusion, summarizing the key insights gained from the analysis.

3. After finishing these "**Request**", answer the following research questions
   1. What are factors in Factor Analysis, and why are they important?
   2. Explain the significance of eigenvalues and eigenvectors in Factor Analysis.
   3. Compare the Factor Analysis Vs. Principle Component Analysis.
   4. Provide examples of real-world applications where Factor Analysis can be useful.

IV. Requirements

1. The directory structure of the compressed submission
   - *doc*: report files include *MSSV_report_p04.doc/docx/latex source* **and** *MSSV_report_p04.pdf*
   - *source*: contains entire source code, removed temporary files, intermediate compiled files if it has... code *(Jupyter Notebook is accepted, but a separate report file is still required)*
   - *bonus*: optional, for plus points, if available

   Additional files *(if has, and in the case of large size)*: these could be provided by URL in a text file inside the corresponding directory *(Google Drive or Microsoft OneDrive).*

   In the case of the total zip file being too big, students could also submit a single text file with personal info and a URL for downloading, but any changes in the URL content after the assignment's due date could be considered 0 points (same as no submit). So, please be careful in this way.
2. Other requirements

- The report should be presented clearly and intuitively: a self-scoring table (assessment) of the results of the work compared to the corresponding requirements (0-100%), a list of the functions included in the program with proof images, summarize the usage and implementation (for example: through pseudo-code, description of methods, or how to do it, *do not copy the source code into the report*).
- The source code needs to be commented on the corresponding lines.

V. Additional guidance

Please be aware that the source code could be different due to various versions from dependencies. Students should describe detailed issues or any modifications it has, it could be considered as a plus point.

1. Required Libraries

```
# Import required libraries
# import pandas as pd
from sklearn.datasets import load_iris
from factor_analyzer import FactorAnalyzer
import matplotlib.pyplot as plt
```

2. Process Data

```
df= pd.read_csv("bfi.csv")

df.columns

# Index(['A1', 'A2', 'A3', 'A4', 'A5', 'C1', 'C2', 'C3', 'C4', 'C5', 'E1', 'E2',
       'E3', 'E4', 'E5', 'N1', 'N2', 'N3', 'N4', 'N5', 'O1', 'O2', 'O3', 'O4',
       'O5', 'gender', 'education', 'age'],
      dtype='object')

# Dropping unnecessary columns
df.drop(['gender', 'education', 'age'],axis=1,inplace=True)

# Dropping missing values rows
df.dropna(inplace=True)
```

3. Before you perform factor analysis, you need to evaluate the “factorability” of our dataset. Factorability means "can we found the factors in the dataset?". There are two methods to check the factorability or sampling adequacy:
    a. Bartlett’s Test
    b. Kaiser-Meyer-Olkin Test

```
from factor_analyzer.factor_analyzer import calculate_bartlett_sphericity
chi_square_value,p_value=calculate_bartlett_sphericity(df)
```

```
chi_square_value, p_value
# the p-value is 0

from factor_analyzer.factor_analyzer import calculate_kmo
kmo_all,kmo_model=calculate_kmo(df)
kmo_model

# KMO is 0.84.
```

Request 01: Students explain the meaning of chi_square_value, p_value, KMO values.

4. Choosing the Number of Factors

```
# Create factor analysis object and perform factor analysis
fa = FactorAnalyzer()
fa.analyze(df, 25, rotation=None)
# Check Eigenvalues
ev, v = fa.get_eigenvalues()
ev

# You can create scree plot using matplotlib
# plt.scatter(range(1,df.shape[1]+1),ev)
# plt.plot(range(1,df.shape[1]+1),ev)
# plt.title('Scree Plot')
# plt.xlabel('Factors')
# plt.ylabel('Eigenvalue')
# plt.grid()
# plt.show()
```

Request 02: Students explain the **eigenvalues** and base on that eigenvalues choose the best number of factor to do the Factor Analysis. Explain why you choose this number.

5. Performing Factor Analysis

```
# Create factor analysis object and perform factor analysis
fa = FactorAnalyzer()
fa.analyze(df, <the magic number you choose>, rotation="varimax")

fa.loadings

# Get variance of each factors
fa.get_factor_variance()
```

Request 03: Students look at the loadings table explain the significant of each factor versus each property. If there are factor(s) that has no “high loading” value, you can remove these and perform Factor Analysis again with the remain factor. Otherwise, explain the Factor Variance Table.