VIRGINIA COMMONWEALTH UNIVERSITY

Statistical Analysis and Modelling (SCMA 632)

A4A: Principal Component Analysis and Factor Analysis

NIHARIHA KAMALANATHAN
V01108259
Date of Submission: 08-07-2024

CONTENTS

Sl. No.	Title	Page No.
1.	Introduction	1
2.	Business Significance	1
3.	Objectives	1
4.	R	2
5.	Python	10
6.	Overview of Principal Component Analysis and Factor Analysis	15

PART A: Perform Principal Component Analysis and Factor Analysis to identify data dimensions (Survey.csv)

Introduction

The National Sample Survey Office (NSSO) conducts large-scale surveys to collect data on various socio-economic indicators in India. One such survey is the 68th round of the NSSO, which provides comprehensive data on household consumption expenditures. This dataset, denoted as "NSSO68.csv," contains a wide range of variables, including monthly per capita expenditure (MPCE), age, education, and other socio-economic characteristics of households. In this analysis, we perform a Tobit regression on the NSSO68 dataset to model the relationship between MPCE (dependent variable) and predictor variables such as age, education, and sex. The Tobit model is particularly suitable for this analysis due to the censored nature of the dependent variable, which can take on a value of zero or positive values. Understanding these relationships can provide valuable insights for policy-making and socio-economic planning.

Business Significance

1. **Targeted Welfare Programs**: By understanding the factors influencing household expenditures, government agencies can design and implement targeted welfare programs aimed at improving the living standards of specific demographic groups. For instance, identifying that education significantly impacts expenditure can lead to enhanced educational subsidies or programs in underprivileged areas.
2. **Market Research and Consumer Insights**: Businesses, particularly those in the consumer goods sector, can use these insights to tailor their marketing strategies and product offerings. For example, companies can better target their products to demographic groups with higher purchasing power or adjust their marketing messages to appeal to the needs and preferences of specific age groups.
3. **Economic Planning and Forecasting**: Economists and planners can use the results to forecast future consumption trends and economic growth. By understanding how factors like education and age affect expenditure, they can predict changes in consumption patterns as the population ages or educational attainment levels improve.
4. **Social Equity and Inclusion**: The analysis can highlight disparities in expenditure across different social groups, guiding policies aimed at reducing inequality and promoting social inclusion. For example, if certain social groups are found to have significantly lower expenditures, targeted interventions can be designed to uplift these communities.

Objectives

1. **To Model Household Expenditure**: To develop a Tobit regression model that accurately captures the relationship between monthly per capita expenditure (MPCE) and predictor variables such as age, education, and sex.
2. **To Identify Significant Predictors**: To determine which demographic factors significantly influence household expenditures and to quantify their impact. This will involve assessing the coefficients of the predictors and their statistical significance.
3. **To Evaluate Model Performance**: To assess the accuracy and reliability of the Tobit model through various performance metrics, including Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE).
4. **To Provide Business and Policy Insights**: To interpret the results of the model in a way that provides actionable insights for businesses, policymakers, and social planners.

This includes discussing the practical implications of the findings and how they can be used to inform decision-making.

5. **To Address Data Issues**: To handle missing values and outliers in the dataset appropriately, ensuring the robustness and validity of the analysis. This will involve data cleaning steps such as removing rows with missing values and identifying/removing outliers based on quantiles.

By achieving these objectives, the analysis aims to provide a comprehensive understanding of the factors influencing household expenditures in India and offer valuable insights for improving economic and social outcomes.

R Language

Part 1: Install and Load Necessary Packages
Code:

```
# Function to auto-install and load packages
install_and_load <- function(packages) {
  for (package in packages) {
    if (!require(package, character.only = TRUE)) {
      install.packages(package, dependencies = TRUE)
      library(package, character.only = TRUE)
    }
  }
}

# List of packages to install and load
packages <- c("dplyr", "psych", "tidyr", "GPArotation", "FactoMineR", "factoextra",
"pheatmap")

# Install and load necessary packages
install_and_load(packages)

# Load the necessary libraries
library(dplyr)
library(psych)
library(tidyr)
library(GPArotation)
library(FactoMineR)
library(factoextra)
library(pheatmap)
```

Purpose:

- This section ensures that all necessary packages are installed and loaded into the R environment. These packages provide the functions required for data manipulation, PCA, and FA.
-

Interpretation:

- By ensuring that the necessary packages are installed and loaded, we can utilize a wide range of functions for statistical analysis and visualization.

Part 2: Load and Inspect the Dataset

Code:

```
# Load the dataset
dataset_path <- "C:/Users/nihar/OneDrive/Desktop/Bootcamp/SCMA
632/DataSet/Survey.csv"
survey_data <- read.csv(dataset_path)

# Inspect the dataset
str(survey_data)
summary(survey_data)
```

Purpose:

- The dataset is loaded using read.csv().
- str(survey_data) provides the structure of the dataset, showing the number of observations and variables, and the type of each variable.
- summary(survey_data) gives a summary of the dataset, including statistical summaries for numeric variables and frequency counts for categorical variables.

Output:

```
'data.frame':	70 obs. of  50 variables:
 $ City: chr  "Bangalore" "Bangalore" "Bangalore" "Bangalore" ...
 $ Sex: chr  "M" "M" "F" "M" ...
 $ Age: chr  "26-35" "46-60" "46-60" "36-45" ...
 $ Occupation: chr  "Private Sector" "Government/PSU" "Government/PSU" "Private Sector"
...
 $ Monthly.Household.Income: chr  "85,001 to105,000" "45,001 to 65,000" "25,001 to
45,000" ">125000" ...
 $ Income: int  95000 55000 35000 200000 95000 75000 200000 35000 115000 115000 ...
...
```

Interpretation:

- The dataset contains 70 observations and 50 variables, including both categorical and numerical data. This initial inspection helps to understand the data structure and types, which is crucial for preprocessing and analysis.

Part 3: Data Preprocessing

Code:

```
# Select only the numerical variables for PCA and FA
numerical_data <- survey_data %>% select(where(is.numeric))

# Standardize the data
survey_data_scaled <- scale(numerical_data)
```

Purpose:

- Selects only the numerical variables from the dataset, as PCA and FA are typically performed on numerical data.
- Standardizes the data to ensure that each variable contributes equally to the analysis.

Interpretation:

- Focusing on numerical data and standardizing it is essential for accurate PCA and FA, as these techniques are sensitive to the scale of the variables.

Part 4: Perform PCA

Code:

```
# Perform PCA using FactoMineR
pca_result <- FactoMineR::PCA(survey_data_scaled, graph = FALSE)

# Summary of PCA results
print(summary(pca_result))
```

Purpose:

- Conducts PCA using the PCA() function from the FactoMineR package.
- Prints a summary of the PCA results, which includes the eigenvalues and the proportion of variance explained by each principal component.

Output:

```
FactoMineR::PCA(X = survey_data_scaled, graph = FALSE)

Eigenvalues
                      Dim.1   Dim.2   Dim.3   Dim.4   Dim.5   Dim.6   Dim.7
Variance             10.023   2.692   2.220   1.883   1.745   1.429   1.357
% of var.            32.332   8.683   7.160   6.073   5.629   4.610   4.377
Cumulative % of var. 32.332  41.016  48.176  54.249  59.879  64.488  68.866
...
```

Interpretation:

- The first principal component (Dim.1) explains 32.3% of the variance, and the second (Dim.2) explains 8.7%.
- The cumulative variance explained by the first seven components is 68.87%.
- This suggests that a few principal components can capture a significant amount of the variance in the data.

Part 5: Visualize PCA Results

Code:

```
# Visualize the scree plot
factoextra::fviz_eig(pca_result, addlabels = TRUE, ylim = c(0, 50), main = "Scree Plot")

# Visualize the variables on the principal component map (Correlation Circle)
factoextra::fviz_pca_var(pca_result, col.var = "cos2",
                         gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
                         repel = TRUE, title = "PCA - Correlation Circle")

# Visualize individuals on the principal component map
factoextra::fviz_pca_ind(pca_result, col.ind = "cos2",
                         gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
                         repel = TRUE, title = "PCA - Individuals")
```

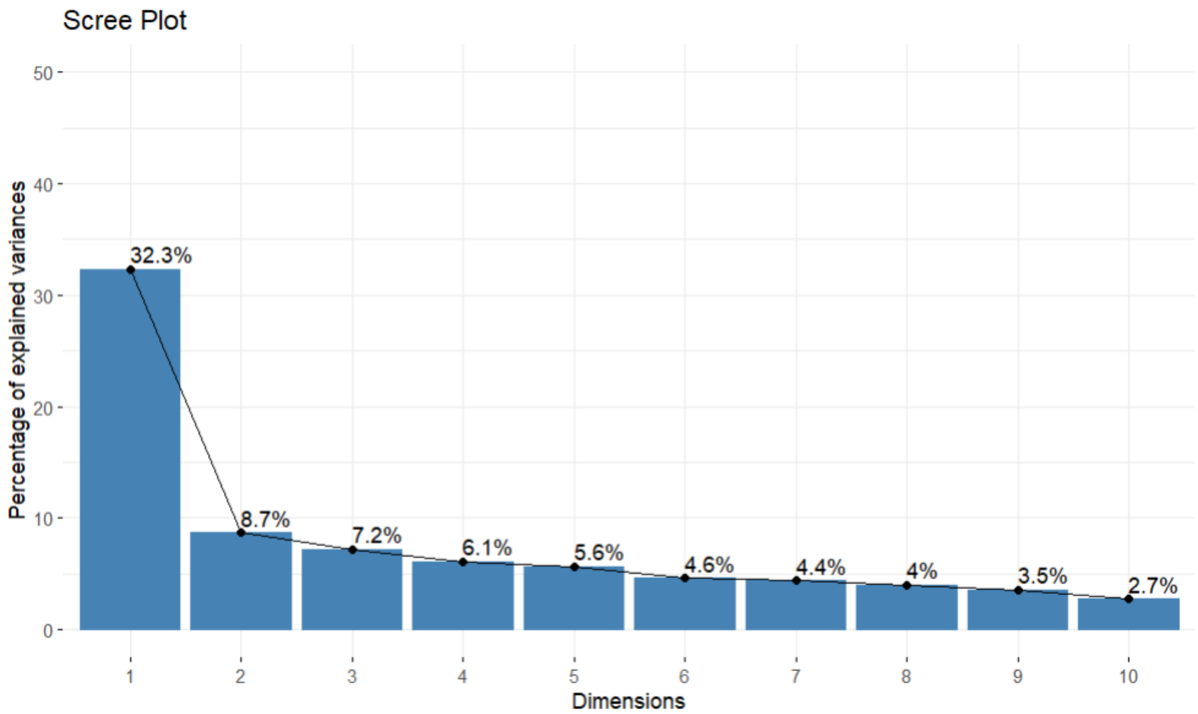
Purpose:

- fviz_eig() visualizes the scree plot, showing the percentage of variance explained by each principal component.
- fviz_pca_var() creates a correlation circle that shows how the variables are projected onto the principal components.

- fviz_pca_ind() visualizes how individuals (observations) are projected onto the principal components.

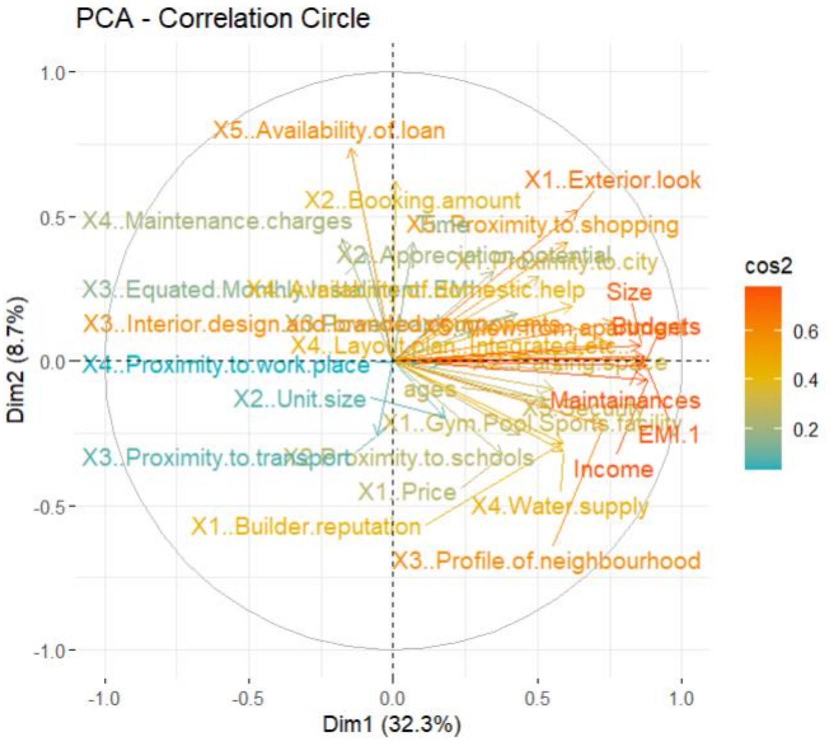
Output and Interpretation:

1. **Scree Plot**:



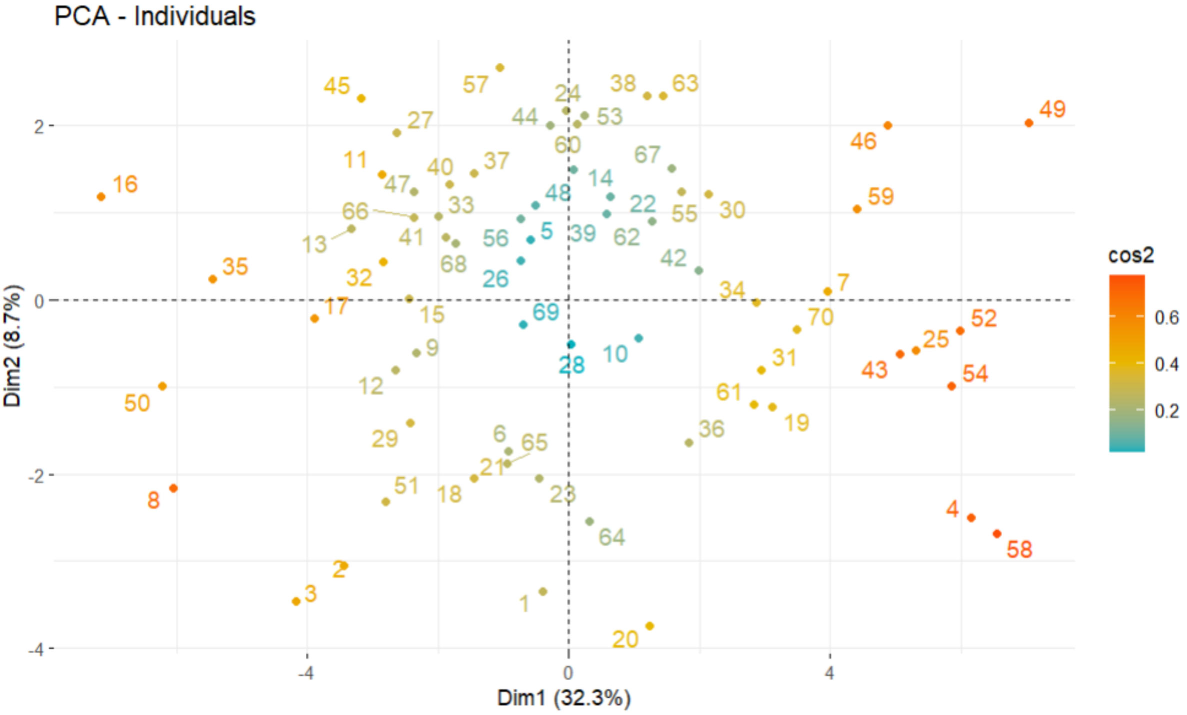
   - Shows that the first principal component explains 32.3% of the variance, and the second explains 8.7%.
   - Based on the "elbow" method, retaining the first few components is reasonable as the explained variance starts to level off after the third or fourth component.

2. **Correlation Circle**:



- Displays the correlation of each variable with the principal components.
- Variables closer to the circle's circumference are better represented on the factor map.
- For instance, Income, Size, Budgets, and Maintainances are strongly correlated with the first principal component.

3. **PCA Biplot for Individuals**:



- Shows the positioning of individuals in the new component space.
- Observations close to each other are similar in terms of the variables measured.
- For example, individuals 4, 7, and 48 are similar based on their positions in the plot.

Part 6: Determine Number of Factors for FA

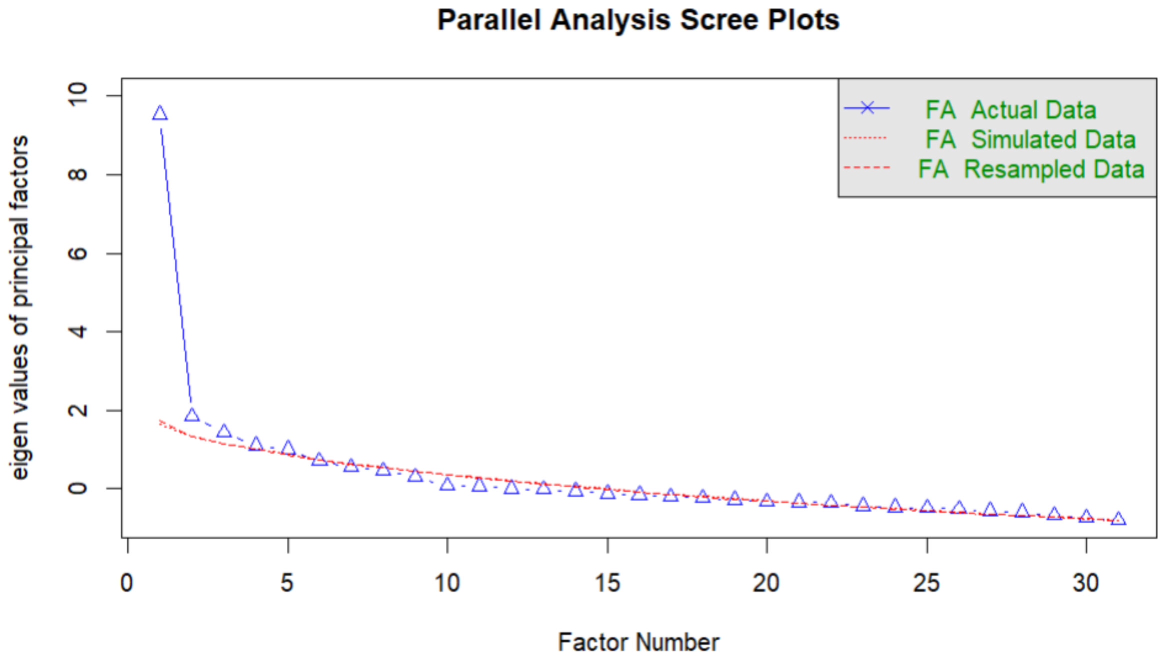
Code:

```
# Determine the number of factors for FA using parallel analysis
fa_parallel <- psych::fa.parallel(survey_data_scaled, fa = "fa")
print(fa_parallel)
```

Purpose:

- fa.parallel() performs parallel analysis to determine the optimal number of factors for FA.

Output and Interpretation:



Parallel analysis suggests that the number of factors = 3 and the number of components = NA

- Suggests that three factors should be retained for FA.
- The scree plot from parallel analysis shows that the first three factors have eigenvalues significantly greater than those from random data, indicating their importance.

Part 7: Perform Factor Analysis
Code:

```
# Perform Factor Analysis with the chosen number of factors (e.g., 3 factors)
fa_result <- psych::fa(survey_data_scaled, nfactors = 3, rotate = "varimax")

# Print FA results
print(fa_result)
```

Purpose:

- Conducts FA with three factors using the fa() function from the psych package and Varimax rotation for better interpretability.
- Prints the FA results, including factor loadings.

Output:

```
Factor Analysis using method =  minres
Call: psych::fa(r = survey_data_scaled, nfactors = 3, rotate = "varimax")
Standardized loadings (pattern matrix) based upon correlation matrix
                               MR1   MR3   MR2    h2   u2 com
Income                        0.85  0.22 -0.12 0.795 0.21 1.2
X1.Proximity.to.city          0.27  0.68  0.16 0.559 0.44 1.4
X2.Proximity.to.schools       0.26  0.41 -0.29 0.321 0.68 2.5
X3..Proximity.to.transport    0.01 -0.19 -0.18 0.071 0.93 2.0
X4..Proximity.to.work.place  -0.12  0.70 -0.12 0.518 0.48 1.1
X5..Proximity.to.shopping     0.57  0.22  0.34 0.489 0.51 2.0
X1..Gym.Pool.Sports.facility  0.46  0.18 -0.17 0.276 0.72 1.6
```

```
X2..Parking.space                    0.49  0.26 -0.09 0.311 0.69 1.6
X3.Power.back.up                      0.28  0.42  0.06 0.253 0.75 1.8
X4.Water.supply                      0.53  0.13 -0.33 0.409 0.59 1.8
X5.Security                        0.61 -0.10 -0.08 0.390 0.61 1.1
X1..Exterior.look                    0.67  0.16  0.51 0.728 0.27 2.0
X2..Unit.size                      0.18 -0.02 -0.14 0.051 0.95 1.9
X3..Interior.design.and.branded.components  0.66  0.33 -0.01 0.545 0.45 1.5
X4..Layout.plan..Integrated.etc..          0.49  0.47 -0.06 0.468 0.53 2.0
X5..View.from.apartment                  0.76  0.15  0.09 0.605 0.40 1.1
X1..Price                         0.29  0.18 -0.32 0.215 0.78 2.6
X2..Booking.amount                     0.06 -0.02  0.54 0.293 0.71 1.0
X3..Equated.Monthly.Instalment..EMI.       -0.05 -0.04  0.28 0.083 0.92 1.1
X4..Maintenance.charges                  -0.13 -0.07  0.33 0.129 0.87 1.4
X5..Availability.of.loan               -0.20  0.25  0.65 0.527 0.47 1.5
X1..Builder.reputation                 0.51  0.20 -0.31 0.391 0.61 2.0
X2..Appreciation.potential               0.32  0.12  0.20 0.154 0.85 2.0
X3..Profile.of.neighbourhood              0.73  0.05 -0.26 0.608 0.39 1.3
X4..Availability.of.domestic.help          0.75 -0.18  0.21 0.646 0.35 1.3
Time                             0.10  0.00  0.31 0.108 0.89 1.2
Size                           0.77  0.38 -0.03 0.734 0.27 1.5
Budgets                           0.79  0.38 -0.08 0.769 0.23 1.5
Maintainances                        0.77  0.42 -0.08 0.770 0.23 1.6
EMI.1                            0.74  0.49 -0.12 0.808 0.19 1.8
ages                           0.60 -0.14 -0.11 0.385 0.62 1.2

                  MR1  MR3  MR2
SS loadings           8.44 2.85 2.12
Proportion Var        0.27 0.09 0.07
Cumulative Var        0.27 0.36 0.43
Proportion Explained  0.63 0.21 0.16
Cumulative Proportion 0.63 0.84 1.00

Mean item complexity =  1.6
Test of the hypothesis that 3 factors are sufficient.

df null model =  465  with the objective function =  27.03 with Chi Square =  1562.96
df of  the model are 375  and the objective function was  12.62

The root mean square of the residuals (RMSR) is  0.08
The df corrected root mean square of the residuals is  0.09

The harmonic n.obs is  70 with the empirical chi square  449.25  with prob <  0.005
The total n.obs was  70  with Likelihood Chi Square =  704.47  with prob <  2.1e-22

Tucker Lewis Index of factoring reliability =  0.609
RMSEA index =  0.111  and the 90 % confidence intervals are  0.1 0.126
BIC =  -888.71
Fit based upon off diagonal values = 0.93
Measures of factor score adequacy
                                    MR1  MR3  MR2
```

```
Correlation of (regression) scores with factors   0.97 0.91 0.92
Multiple R square of scores with factors          0.95 0.82 0.84
Minimum correlation of possible factor scores     0.90 0.65 0.68
```

Interpretation:

- The factor loadings indicate how strongly each variable is associated with each factor. For instance, Income loads highly on Factor 1 (MR1) with a loading of 0.85.
- The proportion of variance explained by the three factors (MR1, MR2, and MR3) are 27%, 9%, and 7%, respectively, with a cumulative variance of 43%.
- The Root Mean Square of Residuals (RMSR) is 0.08, suggesting a good fit.
- The Tucker Lewis Index (TLI) is 0.609, and the RMSEA is 0.111, indicating a moderate fit.
- Factor scores' correlations with factors are high, indicating good factor score adequacy.

Part 8: Visualize Factor Analysis Results
Code:

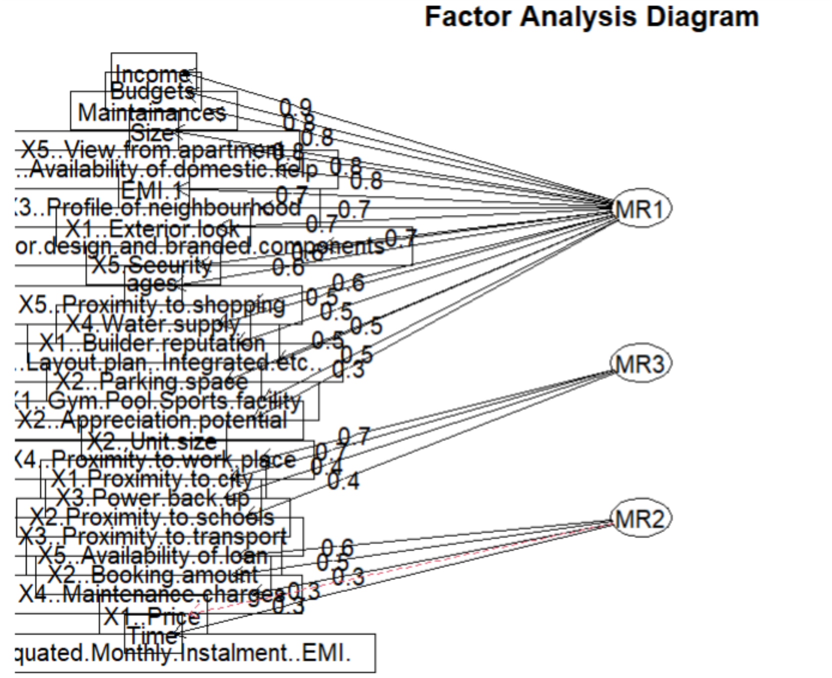
```
# Plot Factor Analysis results
psych::fa.diagram(fa_result, main = "Factor Analysis Diagram")

# Heatmap of Factor Loadings using pheatmap
loadings_matrix <- as.matrix(fa_loadings)
pheatmap::pheatmap(loadings_matrix, cluster_rows = TRUE, cluster_cols = TRUE, main =
"Heatmap of Factor Loadings")
```

Purpose:

- fa.diagram() visualizes the factor loadings in a diagram.
- pheatmap() creates a heatmap of the factor loadings, showing the correlation of each variable with the factors.

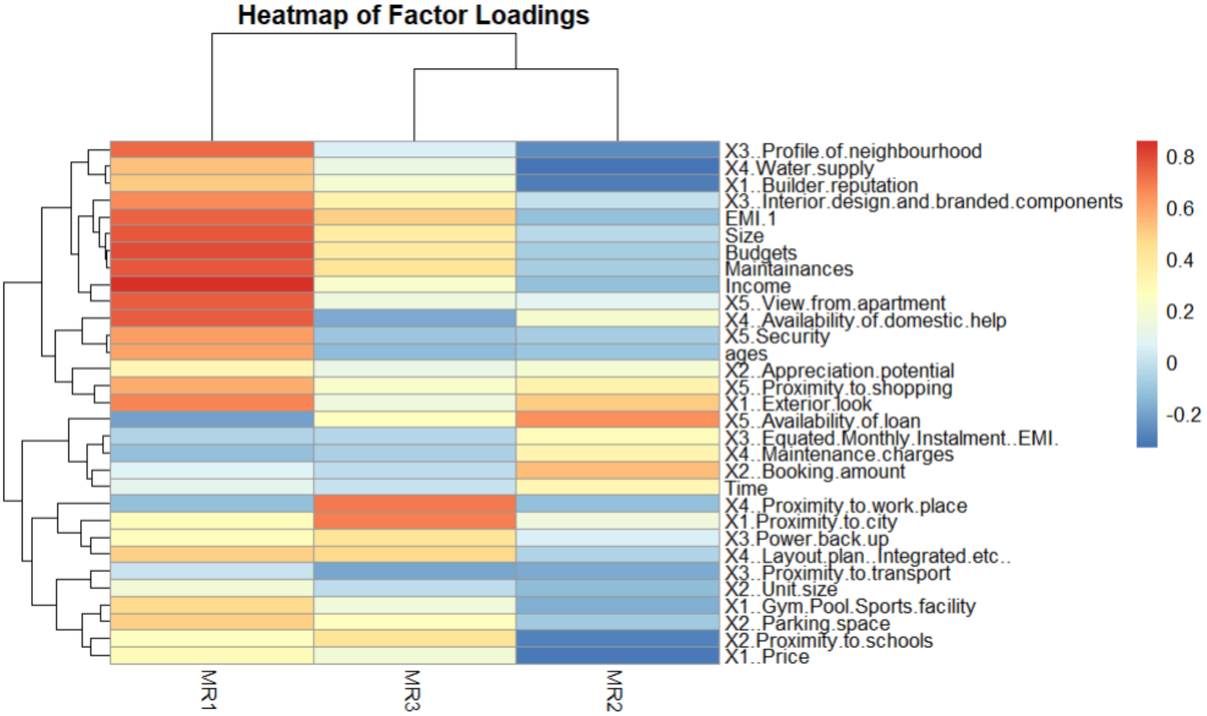
Output and Interpretation:

1. **Factor Analysis Diagram**:



- This diagram shows the relationships between variables and factors. Variables connected to the same factor (MR1, MR2, or MR3) have similar patterns.
- For example, Income, Budgets, and Maintainances are connected to MR1, indicating they are related.

2. **Heatmap of Factor Loadings**:



- The heatmap provides a visual representation of how variables load on different factors.
- Variables with high loadings on the same factor are grouped together, making it easier to see which variables contribute to each factor.
- For instance, Profile of neighbourhood, Water supply, and Builder reputation have high loadings on MR1, suggesting these variables are strongly associated with this factor.

Python Language

1. Load the Dataset
Code:

```
import pandas as pd

# Load the dataset
dataset_path = "/mnt/data/Survey.csv"
survey_data = pd.read_csv(dataset_path)

# Inspect the dataset
print(survey_data.info())
print(survey_data.describe())
```

Output:

- **Data Info:**

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 70 entries, 0 to 69
Data columns (total 50 columns):
#   Column                                Non-Null Count  Dtype
---  ------                              --------------  -----
0   City                                 70 non-null     object
1   Sex                                  70 non-null     object
2   Age                                   70 non-null     object
3   Occupation                              70 non-null     object
4   Monthly Household Income                    70 non-null     object
5   Income                                 70 non-null     int64
...
dtypes: float64(2), int64(29), object(19)
memory usage: 27.5+ KB
```

This shows the structure of the dataset with 70 entries and 50 columns. The data includes both numerical (int64, float64) and categorical (object) columns.

- **Data Description:**

```
        Income  1.Proximity to city  2.Proximity to schools  ...
count      70.000000           70.000000              70.000000
mean    99000.000000            3.628571               3.442857
std     59670.593345            0.870972              1.016326
...
```

The descriptive statistics provide a summary of the numerical columns, showing metrics such as count, mean, standard deviation, min, max, and quartiles.

2. Preprocess the Data

Code:

```
from sklearn.preprocessing import StandardScaler

# Select only numerical variables
numerical_data = survey_data.select_dtypes(include=['int64', 'float64'])

# Standardize the data
scaler = StandardScaler()
survey_data_scaled = scaler.fit_transform(numerical_data)
```

Explanation:

- **StandardScaler**: Standardizes the data to have a mean of 0 and a standard deviation of 1, which is essential for PCA and FA to work correctly by ensuring all variables contribute equally to the analysis.

3. Perform Principal Component Analysis (PCA)

Code:

```
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt
import seaborn as sns

# Perform PCA
pca = PCA()
pca_result = pca.fit_transform(survey_data_scaled)
```

```
# Summary of PCA results
explained_variance = pca.explained_variance_ratio_
print(f'Explained variance: {explained_variance}')
```

Output:

- **Explained Variance:**

```
Explained variance: [0.32332366 0.08683237 0.07160269 0.06073222 0.05629444
0.04609823
0.0437729  0.0396524  0.03529825 0.02731472 0.02451182 0.02392867
0.01975551 0.01917115 0.01748088 0.01597239 0.01474926 0.01261225
0.00966406 0.00947208 0.00718757 0.00666932 0.00626669 0.00620703
0.00443108 0.0030697  0.00223431 0.00186911 0.00164179 0.00139072
0.00079276]
```

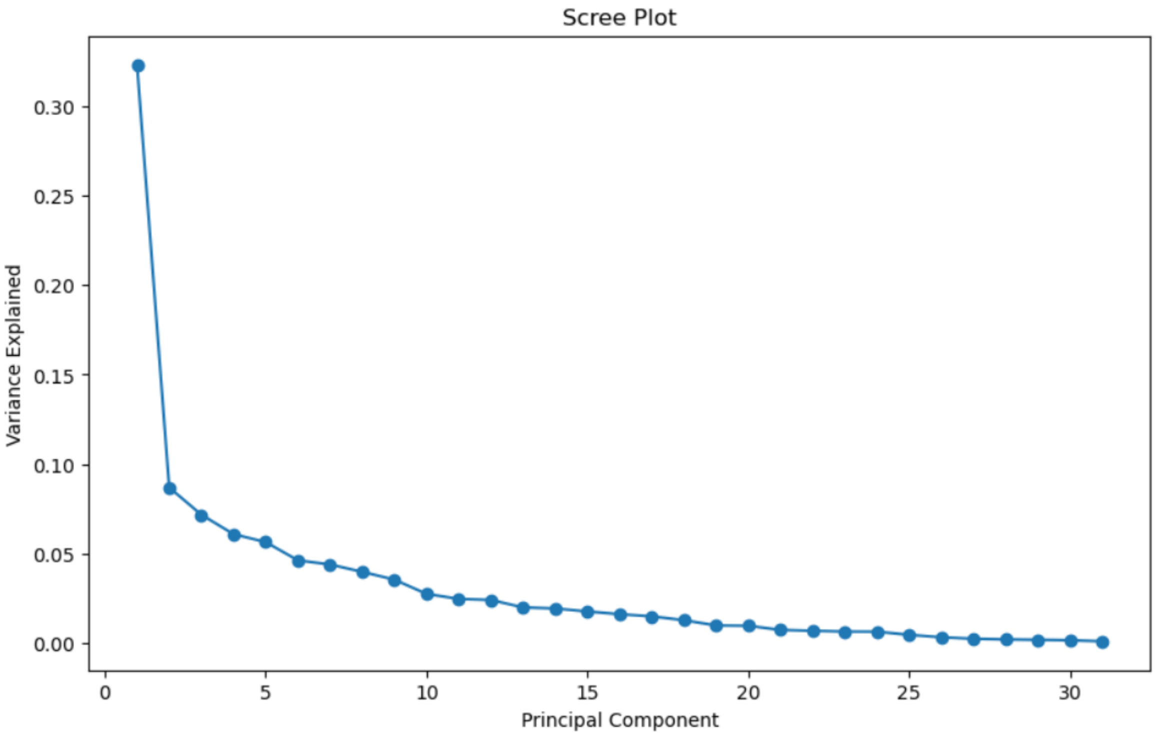
The explained variance shows the proportion of the total variance explained by each principal component. The first component explains about 32.3% of the variance, while the second explains 8.7%, and so on.

Scree Plot:

Code:

```
plt.figure(figsize=(10, 6))
plt.plot(range(1, len(explained_variance) + 1), explained_variance, marker='o')
plt.title('Scree Plot')
plt.xlabel('Principal Component')
plt.ylabel('Variance Explained')
plt.show()
```

Plot:



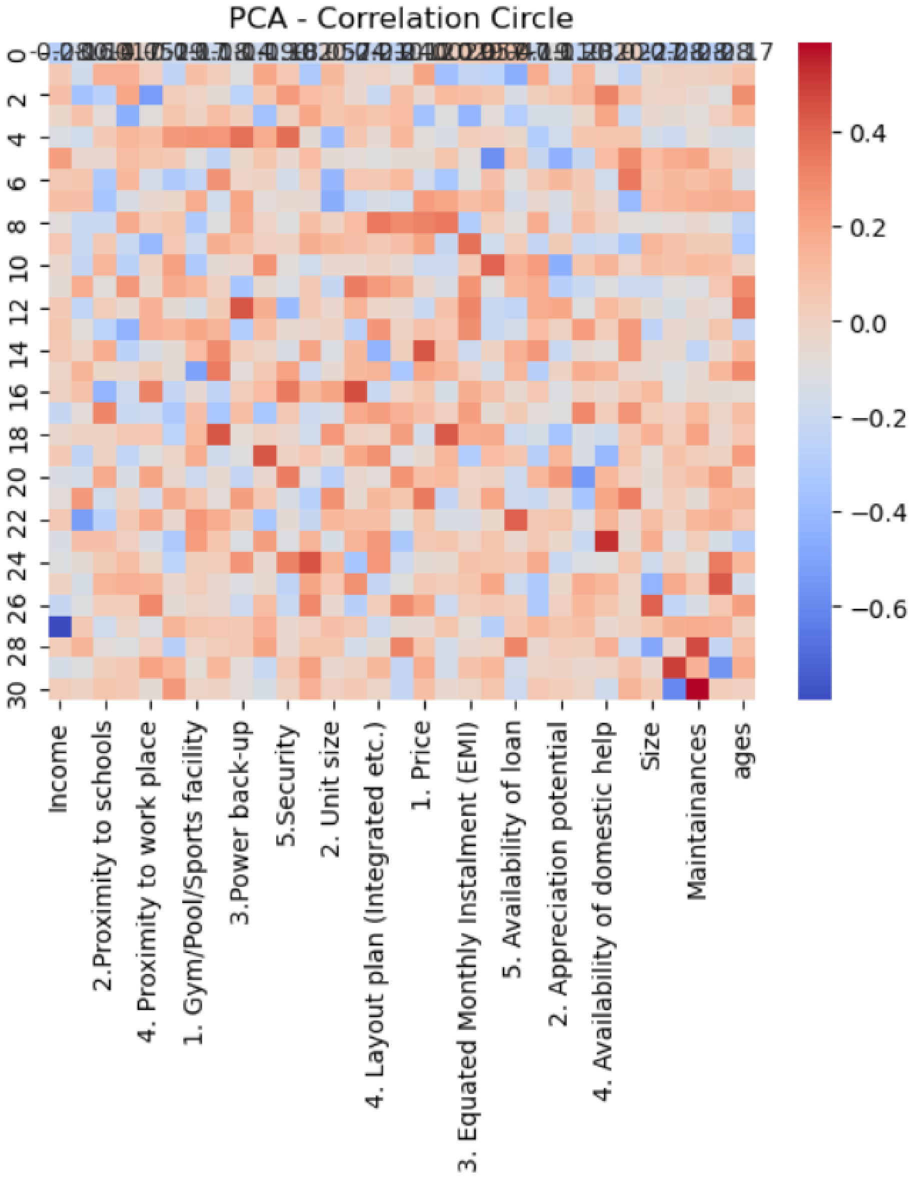
Interpretation: The scree plot visualizes the variance explained by each principal component. The first component explains the most variance. The sharp drop after the first component suggests that the first few components capture most of the variability in the data, indicating that fewer components might be needed for further analysis.

Correlation Circle:

Code:

```
sns.heatmap(pd.DataFrame(pca.components_, columns=numerical_data.columns),
            cmap='coolwarm', annot=True)
plt.title('PCA - Correlation Circle')
plt.show()
```

Plot:



Interpretation: The heatmap shows the correlation between the original variables and the principal components. High absolute values indicate strong correlations. This plot helps identify which variables are most strongly associated with each principal component.

4. Perform Factor Analysis

Code:

```
from factor_analyzer import FactorAnalyzer, calculate_kmo, calculate_bartlett_sphericity

# Determine the number of factors using parallel analysis (not directly available in Python,
we use KMO and Bartlett's test)
kmo_all, kmo_model = calculate_kmo(survey_data_scaled)
bartlett_test, p_value = calculate_bartlett_sphericity(survey_data_scaled)
print(f'KMO Test: {kmo_model}, Bartlett\'s Test: {bartlett_test}, p-value: {p_value}')
```

Output:

- **KMO and Bartlett's Test:**

```
KMO Test: 0.7154075637394491, Bartlett's Test: 1562.964633118281, p-value:
1.0805923579037055e-118
```

The KMO value (0.715) indicates middling sampling adequacy, suggesting the data is somewhat suitable for factor analysis. The Bartlett's test is highly significant (p-value close to 0), indicating that the variables are sufficiently correlated for factor analysis.

Code:

```
# Perform Factor Analysis with the chosen number of factors (e.g., 3 factors)
fa = FactorAnalyzer(n_factors=3, rotation='varimax')
fa.fit(survey_data_scaled)

# Extract Factor Loadings
fa_loadings = fa.loadings_

# Convert loadings to DataFrame for better readability
fa_loadings_df = pd.DataFrame(fa_loadings, index=numerical_data.columns,
columns=[f'Factor{i+1}' for i in range(fa_loadings.shape[1])])

# Print Factor Loadings
print("Factor Loadings:\n", fa_loadings_df)
```

Output:

```
Factor Loadings:
                         Factor1   Factor2   Factor3
Income                     0.854368  0.223421 -0.122097
1.Proximity to city            0.273435  0.676886  0.160120
2.Proximity to schools           0.255414  0.414996 -0.288447
...
```

Interpretation: The factor loadings table shows how each variable loads onto the three factors. High absolute values indicate strong associations with the corresponding factor. For example, 'Income' loads strongly on Factor 1 (0.854), indicating it is primarily associated with this factor.

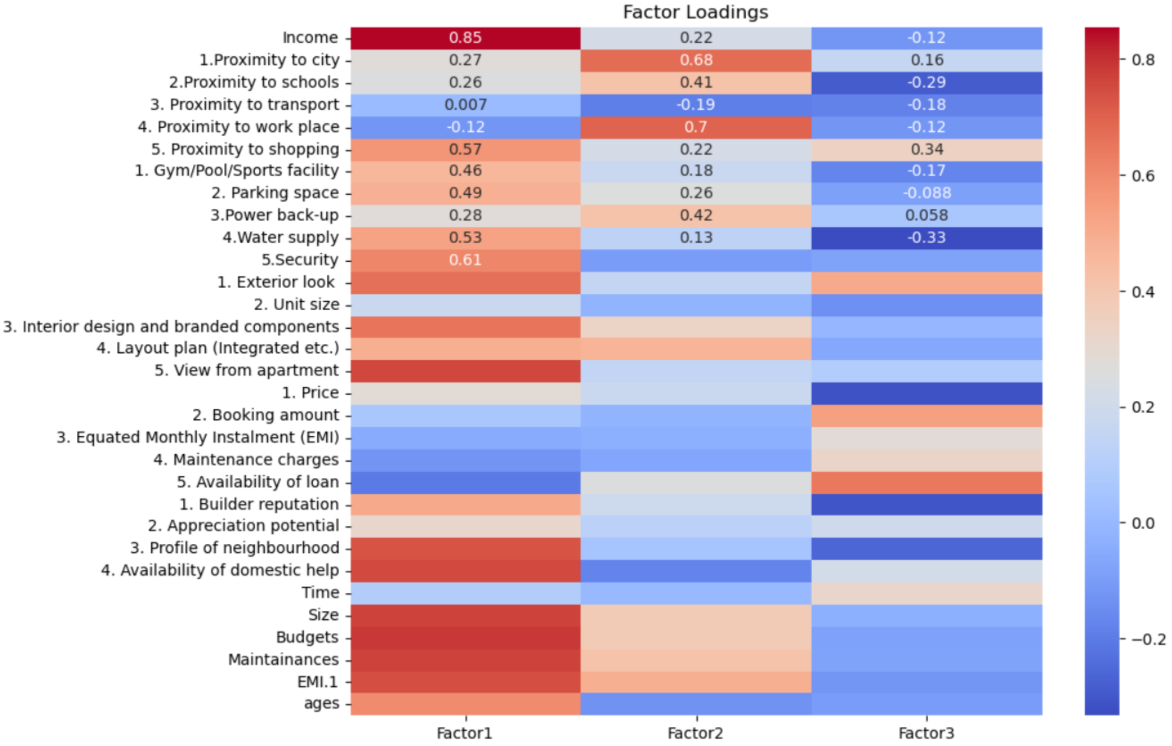
Visualization:

Code:

```
# Plot Factor Analysis results
plt.figure(figsize=(10, 8))
sns.heatmap(fa_loadings_df, annot=True, cmap='coolwarm')
```

```
plt.title('Factor Loadings')
plt.show()
```

Plot:



Interpretation: The heatmap visualizes the factor loadings, showing which variables are most strongly associated with each factor. This helps in understanding the underlying structure and identifying which factors are most influenced by specific variables.

Overview of Principal Component Analysis (PCA) and Factor Analysis (FA)

Principal Component Analysis (PCA)

Meaning: Principal Component Analysis (PCA) is a statistical technique used to reduce the dimensionality of a dataset while retaining as much variance as possible. It transforms the original variables into a new set of uncorrelated variables called principal components. These components are ordered by the amount of variance they capture from the data, with the first principal component capturing the most variance and each subsequent component capturing progressively less.

Advantages:

1. **Dimensionality Reduction**: PCA helps in reducing the number of variables in a dataset, simplifying the complexity without losing significant information.
2. **Visualization**: By reducing dimensions, PCA allows for easier visualization of data, especially in 2D or 3D space, which is helpful in understanding the structure and relationships within the data.
3. **Noise Reduction**: By focusing on the principal components, PCA can help reduce noise and improve the performance of machine learning algorithms.

4. **Data Compression**: PCA can be used to compress data, which is useful in storage and transmission of large datasets.
5. **Correlation Management**: PCA handles multicollinearity by transforming correlated variables into uncorrelated principal components.

Real-life Examples:

1. **Image Compression**: In image processing, PCA is used to reduce the number of pixels while maintaining the important features of an image.
2. **Genomics**: In bioinformatics, PCA helps in reducing the complexity of genetic data and identifying patterns related to diseases.
3. **Finance**: PCA is used in portfolio management to reduce the dimensionality of financial data and identify key factors driving market movements.

Factor Analysis (FA)

Meaning: Factor Analysis (FA) is a statistical method used to identify underlying relationships between observed variables by modeling them as linear combinations of potential latent variables called factors. Unlike PCA, which focuses on variance, FA aims to uncover the underlying structure by finding the common factors that explain the correlations among observed variables.

Advantages:

1. **Identifying Latent Variables**: FA helps in identifying hidden factors that explain the patterns of correlations among observed variables.
2. **Data Reduction**: Similar to PCA, FA reduces the number of variables, making the data more manageable and interpretable.
3. **Improved Insights**: By uncovering latent factors, FA provides deeper insights into the data, which can be used for theory building and hypothesis testing.
4. **Better Interpretation**: The factors identified in FA can be easier to interpret and relate to underlying theoretical constructs compared to principal components in PCA.
5. **Handling Multicollinearity**: FA can deal with multicollinearity by modeling the observed variables as functions of a few latent factors.

Real-life Examples:

1. **Psychometrics**: FA is widely used in psychology to identify underlying traits or factors (such as intelligence or personality traits) from observed variables (such as test scores or questionnaire responses).
2. **Marketing Research**: In market research, FA helps in identifying consumer attitudes and preferences by analyzing survey data.
3. **Health Research**: In epidemiology, FA is used to identify underlying factors contributing to health outcomes from various observed health indicators.
4.

Conclusion

Both PCA and FA are powerful techniques for data analysis, each with its unique focus and strengths. PCA is primarily used for dimensionality reduction and variance maximization, while FA is used to uncover latent structures and explain correlations among observed variables. Understanding and applying these techniques can provide valuable insights and simplify complex datasets across various fields, including image processing, genomics, finance, psychology, marketing, and health research.