VIRGINIA COMMONWEALTH UNIVERSITY

Statistical analysis and modelling (SCMA 632)

A4.1: Principal Component Analysis, Factor Analysis

SAYA SANTHOSH

V01101901

Date of Submission: 08-07-2024

CONTENTS

Sl. No.	Title	Page No.
1.	Introduction	**1**
2.	Results and Interpretations using R	**2**
3.	Results and Interpretations using Python	**7**
4.	Recommendations	**11**
5.	Codes	**11**
6.	References	**17**

Introduction

Factor analysis and principal component analysis :
This assignment uses the "Survey.csv" dataset, which includes a variety of socio-demographic and housing-related parameters like age, income, number of rooms, and proximity to facilities, to investigate Principal Component Analysis (PCA) and Factor Analysis (FA). Reducing the dataset's dimensionality and locating underlying data structures are the main goals. In order to facilitate data visualization and comprehension, PCA will be used to reduce the original variables into a smaller collection of uncorrelated components that preserve the majority of the variation. Simultaneously, FA will provide latent factors that provide a deeper comprehension of the inherent patterns in the data by explaining observed correlations among variables. By adopting these methods, the study hopes to improve data interpretability, streamline complicated datasets, and offer useful insights into housing and sociodemographic choices. This study shows the value of PCA and FA in extracting significant information from complex datasets, while also clarifying important data dimensions through their meticulous use.

Objectives :

- Simplifies data by reducing the number of variables.
- Detects hidden patterns in data.
- Makes complex data easier to understand.
- Creates visual representations of data components.
- Draws valuable conclusions from data analysis.

Business Significance :

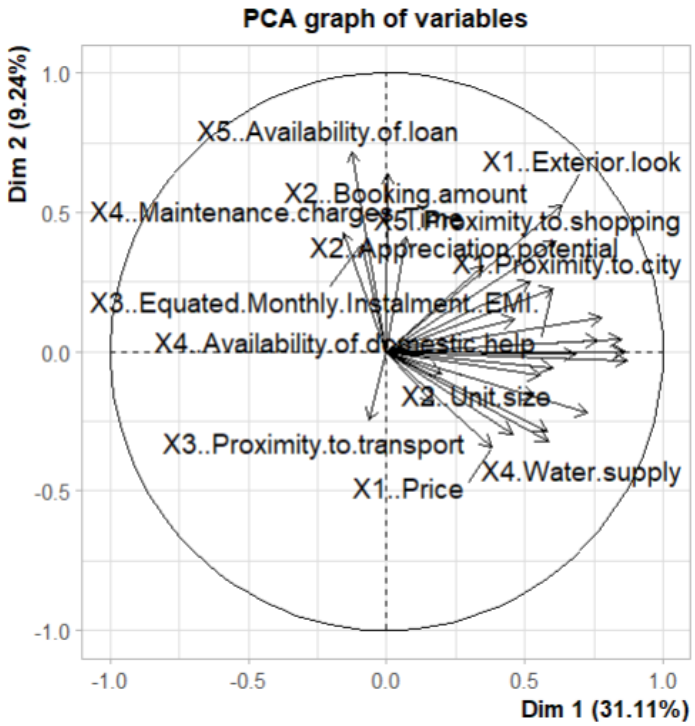
Organizations can extract valuable business insights from complicated socio-demographic and housing-related data by applying Principal Component Analysis (PCA) and Factor Analysis (FA) to the "Survey.csv" dataset. These methods simplify the data and increase its accessibility for strategic decision-making by lowering its dimensionality. Businesses may

more precisely understand consumer preferences and behaviors by identifying underlying causes and primary components. This helps them develop marketing strategies that are more focused, enhance their product offers, and allocate resources more efficiently. In the end, using data-driven insights, this analytical method improves the capacity to meet market demands, raise customer happiness, and create competitive advantage.

Results and Interpretation using R

```
> # Performing PCA using GPArotation
> library(GPArotation)
> pca_1 <- principal(sur_int, 5, n.obs = 70, rotate = "promax")
> print(pca_1)
Principal Components Analysis
Call: principal(r = sur_int, nfactors = 5, rotate = "promax", n.obs = 70)
Standardized loadings (pattern matrix) based upon correlation matrix
                                          RC1   RC5   RC3   RC2   RC4
h2   u2 com
X1.Proximity.to.city                    -0.02  0.36  0.62  0.22 -0.33 0
.71 0.29 2.5
X2.Proximity.to.schools                 -0.06  0.26  0.49 -0.12  0.16 0
.44 0.56 1.9
X3..Proximity.to.transport              -0.08  0.02 -0.22  0.16  0.76 0
.55 0.45 1.3
X4..Proximity.to.work.place             -0.34 -0.01  0.95  0.12 -0.02 0
.71 0.29 1.3
X5..Proximity.to.shopping                0.76 -0.12  0.12  0.23 -0.06 0
.65 0.35 1.3
X1..Gym.Pool.Sports.facility             0.47 -0.06  0.22 -0.13  0.23 0
.45 0.55 2.2
X2..Parking.space                        0.56  0.00  0.16 -0.17 -0.01 0
.46 0.54 1.4
X3.Power.back.up                         0.41 -0.27  0.57  0.09  0.00 0
.58 0.42 2.3
X4.Water.supply                          0.27  0.26  0.06  0.03  0.71 0
.76 0.24 1.6
X5.Security                              0.84 -0.16 -0.24 -0.15  0.32 0
.69 0.31 1.6
X1..Exterior.look                        0.72  0.21 -0.16  0.21 -0.35 0
.79 0.21 2.0
X2..Unit.size                           -0.16  0.61 -0.28 -0.20 -0.10 0
.39 0.61 1.9
X3..Interior.design.and.branded.components  0.55  0.28  0.13 -0.07 -0.06 0
.61 0.39 1.7
X4..Layout.plan..Integrated.etc..        0.25  0.43  0.28 -0.06 -0.14 0
.55 0.45 2.7
X5..View.from.apartment                  0.80  0.21 -0.16 -0.08 -0.04 0
.71 0.29 1.3
X1..Price                               -0.17  0.50  0.11  0.06  0.55 0
.52 0.48 2.3
```

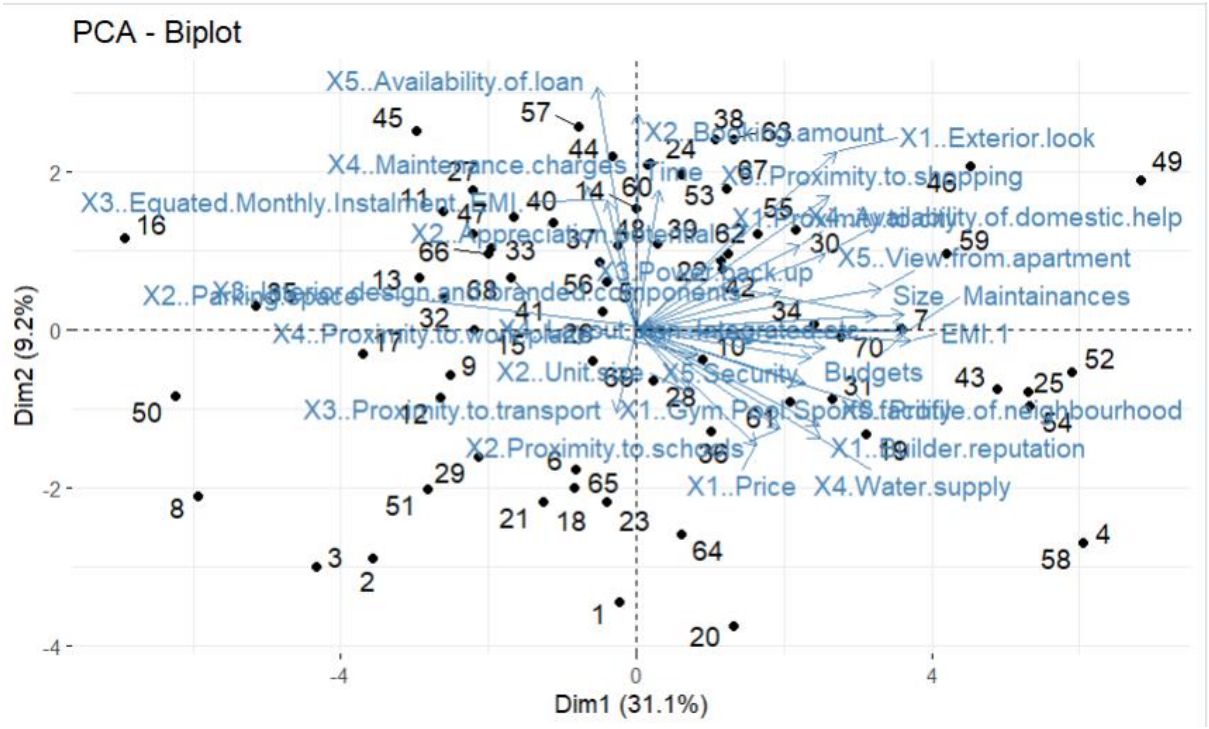
```
X2..Booking.amount                          0.12  0.05 -0.12  0.62 -0.09 0
.46 0.54 1.2
X3..Equated.Monthly.Instalment..EMI.       -0.10 -0.02  0.00  0.70  0.44 0
.53 0.47 1.7
X4..Maintenance.charges                     0.00 -0.06 -0.12  0.43 -0.03 0
.24 0.76 1.2
X5..Availability.of.loan                   -0.18 -0.08  0.31  0.89  0.03 0
.76 0.24 1.4
X1..Builder.reputation                     -0.04  0.80 -0.07 -0.09  0.23 0
.66 0.34 1.2
X2..Appreciation.potential                  0.12  0.44 -0.11  0.38  0.07 0
.36 0.64 2.3
X3..Profile.of.neighbourhood                0.55  0.37 -0.20 -0.20  0.27 0
.66 0.34 3.0
X4..Availability.of.domestic.help           0.90  0.07 -0.43 -0.07 -0.10 0
.70 0.30 1.5
Time                                        0.22 -0.06 -0.10  0.47  0.15 0
.27 0.73 1.8
Size                                        0.41  0.58  0.09  0.06  0.06 0
.76 0.24 1.9
Budgets                                     0.37  0.64  0.07  0.04  0.09 0
.80 0.20 1.7
Maintainances                               0.42  0.51  0.16  0.06  0.14 0
.77 0.23 2.4
EMI.1                                       0.36  0.57  0.23 -0.03  0.00 0
.81 0.19 2.1

                       RC1  RC5  RC3  RC2  RC4
SS loadings           5.93 4.28 2.52 2.47 2.14
Proportion Var        0.20 0.15 0.09 0.09 0.07
Cumulative Var        0.20 0.35 0.44 0.52 0.60
Proportion Explained  0.34 0.25 0.15 0.14 0.12
Cumulative Proportion 0.34 0.59 0.73 0.88 1.00

 With component correlations of
     RC1   RC5   RC3   RC2   RC4
RC1 1.00  0.40  0.43  0.00  0.09
RC5 0.40  1.00  0.28 -0.05 -0.01
RC3 0.43  0.28  1.00 -0.20  0.14
RC2 0.00 -0.05 -0.20  1.00 -0.27
RC4 0.09 -0.01  0.14 -0.27  1.00

Mean item complexity =  1.8
Test of the hypothesis that 5 components are sufficient.

The root mean square of the residuals (RMSR) is  0.07
 with the empirical chi square  289.69  with prob <  0.21

Fit based upon off diagonal values = 0.95
```



Interpretation:

The Principal Component Analysis (PCA) results indicate that the data is well-explained by five components, which together account for 60% of the total variance. Each component represents a combination of variables, with the first component (RC1) explaining 20% of the variance, and subsequent components (RC5, RC3, RC2, and RC4) contributing progressively less. The standardized loadings (pattern matrix) show how each variable correlates with the components, highlighting the importance of variables like "Proximity to city," "Gym/Pool/Sports facility," and "Security" in explaining the variance. The component correlations suggest some degree of relationship between the components, particularly between RC1, RC5, and RC3. The root mean square of the residuals (RMSR) is 0.07, indicating a good fit, and the empirical chi-square test supports the sufficiency of the five-component model with a high probability ($p > 0.21$).

```
> # Using factoextra to plot the PCA biplot
> library(factoextra)
> fviz_pca_biplot(pca_2, repel = TRUE)
```

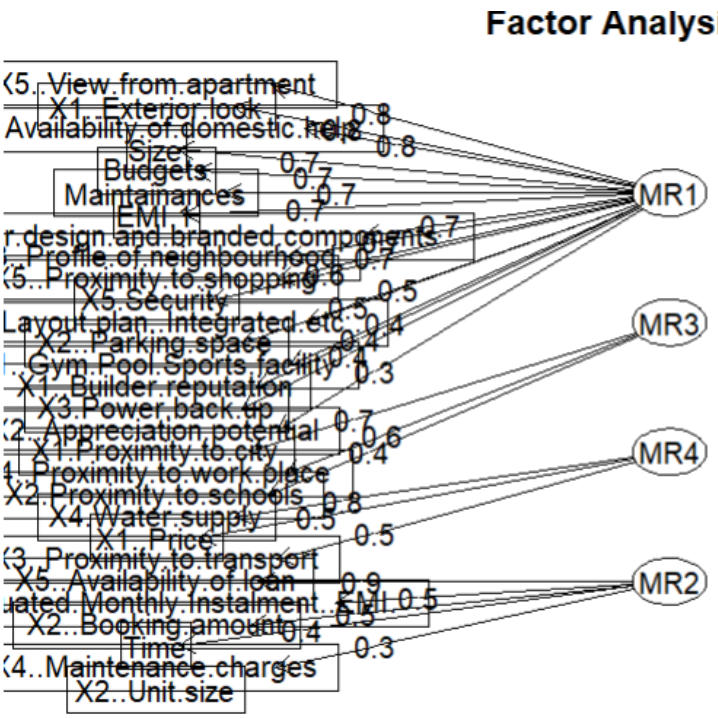


Interpretation:

Using the `factoextra` package, a PCA biplot was generated to visualize the principal components of the dataset. The biplot displays both the observations and variables on the same plot, helping to interpret the relationships between them. In this plot, the variables are represented as vectors, and their directions and lengths indicate their contributions to the principal components. Observations are plotted as points, and their proximity to the variable vectors suggests how strongly they are associated with those variables. This visualization aids in identifying clusters of observations and the most influential variables, enhancing the interpretability of the PCA results by providing a clear graphical representation of the data structure.

```
> # Factor Analysis
> factor_analysis<-fa(sur_int,nfactors = 4,rotate = "varimax")
> names(factor_analysis)
 [1] "residual"      "dof"           "chi"           "nh"            "rms"
 [6] "EPVAL"         "crms"          "EBIC"          "ESABIC"        "fit"
[11] "fit.off"       "sd"            "factors"       "complexity"    "n.ob
s"
[16] "objective"     "criteria"      "STATISTIC"     "PVAL"          "Call
"
[21] "null.model"    "null.dof"      "null.chisq"    "TLI"           "CFI"
[26] "RMSEA"         "BIC"           "SABIC"         "r.scores"      "R2"
[31] "valid"         "score.cor"     "weights"       "rotation"      "hype
rplane"
[36] "communality"   "communalities" "uniquenesses"  "values"        "e.va
lues"
```

```
[41] "loadings"      "model"         "fm"            "rot.mat"       "Stru
cture"
[46] "method"        "scores"        "R2.scores"     "r"             "np.o
bs"
[51] "fn"            "Vaccounted"    "ECV"
> print(factor_analysis$loadings,reorder=TRUE)
```

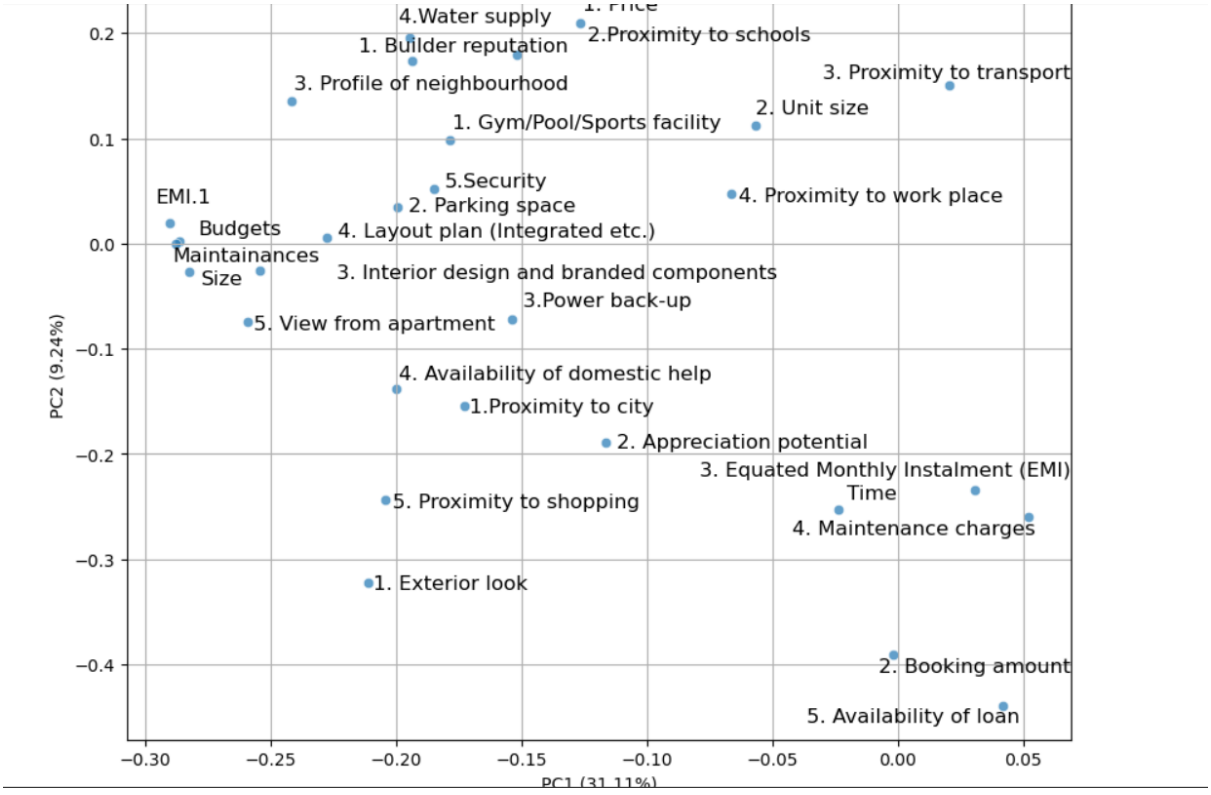


Interpretation:

The Factor Analysis conducted with four factors using the Varimax rotation method identifies the underlying structure of the dataset. Each factor (MR1, MR2, MR3, MR4) is represented by the loadings of various variables, showing how strongly each variable correlates with the factor. For example, MR1 has high loadings for variables like "Exterior look," "Availability of domestic help," and "View from apartment," indicating these variables are closely related. MR2, MR3, and MR4 have their own sets of significant loadings, showing different patterns of relationships. The analysis shows that these four factors together explain 47.6% of the total variance, with MR1 explaining the largest proportion. This dimensionality reduction helps simplify the complex data into fewer underlying factors, facilitating easier interpretation and insights into the main influences and groupings within the data.

Results and Interpretation using Python

```
# PCA biplot function
def biplot_scores(pca, X):
    plt.figure(figsize=(10, 8))
    sns.scatterplot(x=pca.components_[0], y=pca.components_[1], alpha=0.7)
    texts = []
    for i, txt in enumerate(X.columns):
        texts.append(plt.text(pca.components_[0][i], pca.components_[1][i], txt, fontsize=12))
    adjust_text(texts)

    xlabel = f'PC1 ({explained_variance[0]:.2f}%)'
    ylabel = f'PC2 ({explained_variance[1]:.2f}%)'

    plt.xlabel(xlabel)
    plt.ylabel(ylabel)
    plt.title('PCA Biplot')
    plt.grid(True)
    plt.show()

# Plot the PCA biplot
biplot_scores(pca, sur_int)
```

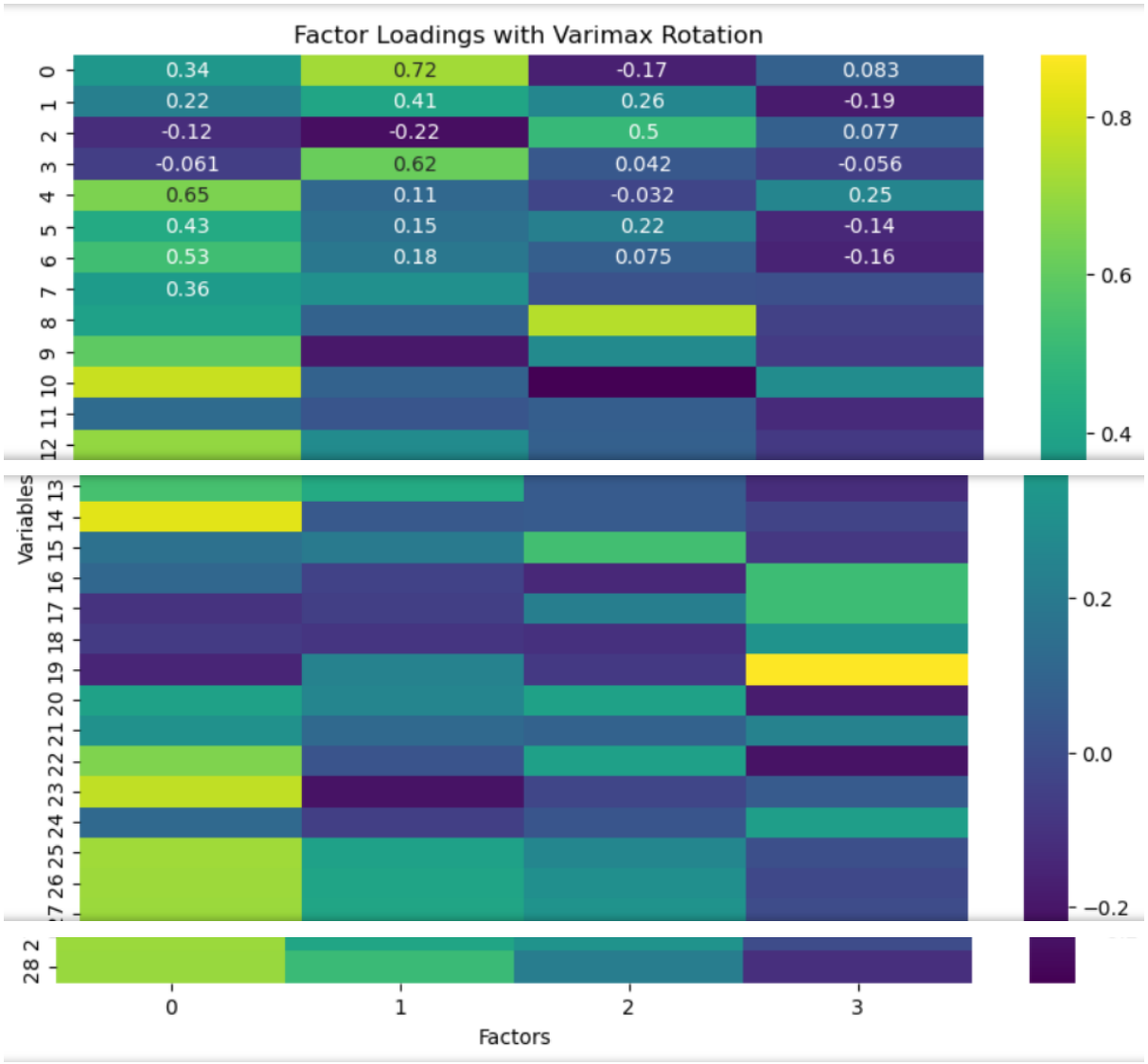


Interpretation:

The biplot visualizes the results of Principal Component Analysis (PCA) on the given dataset. The plot displays two principal components, PC1 and PC2, which explain 31.11% and 9.24%

of the variance in the data, respectively. Each point on the plot represents a variable from the dataset, and the position indicates the contribution of that variable to the principal components.Variables such as "Builder reputation," "Profile of neighborhood," and "View from apartment" have significant contributions to PC1, while variables like "Proximity to transport," "Water supply," and "Price" are more aligned with PC2. The proximity of points to each other indicates their correlation; for instance, "Budgets," "Maintainances," and "Size" are close together, suggesting they are correlated. Additionally, points farther from the origin have a higher contribution to the explained variance.This visualization helps in identifying patterns and relationships between variables, as well as understanding the underlying structure of the dataset.

```
# Plotting the factor loadings
def plot_factor_loadings(loadings, title):
    plt.figure(figsize=(10, 8))
    sns.heatmap(loadings, annot=True, cmap='viridis')
    plt.title(title)
    plt.xlabel('Factors')
    plt.ylabel('Variables')
    plt.show()

plot_factor_loadings(varimax_loadings, 'Factor Loadings with Varimax Rotation')
```

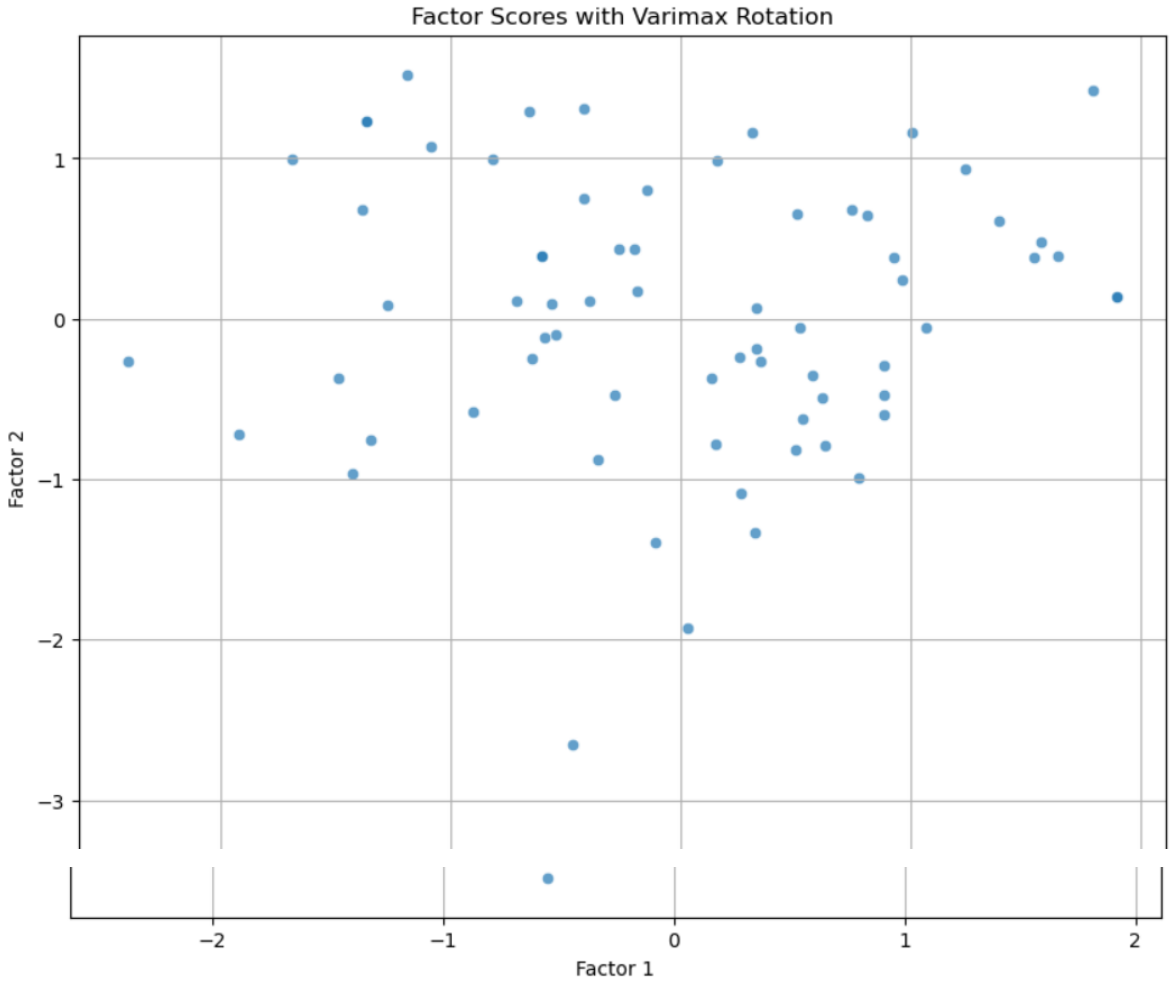


Interpretation:

The heatmap visualizes the factor loadings from a factor analysis with Varimax rotation. Each cell represents the loading of a variable (rows) on a factor (columns), with color intensity indicating the magnitude and direction of the loading.The heatmap includes four factors, labeled as 0, 1, 2, and 3, with variables listed along the vertical axis and their loadings on each factor shown across the horizontal axis. Higher loadings, closer to ±1, indicate a stronger relationship between the variable and the factor. For instance, the first variable (row 0) has high loadings on factor 1 (0.72) and moderate loadings on factor 0 (0.34), indicating a strong association with factor 1. Similarly, the variable in row 4 has a high loading on factor 0 (0.65), suggesting it is

strongly associated with that factor. The color gradient displays positive loadings in green to yellow and negative loadings in blue to purple, allowing for quick visual identification of relationships.

```
# Plotting factor scores
plt.figure(figsize=(10, 8))
sns.scatterplot(x=varimax_factor_scores[:, 0], y=varimax_factor_scores[:, 1], alpha=0.7)
plt.xlabel('Factor 1')
plt.ylabel('Factor 2')
plt.title('Factor Scores with Varimax Rotation')
plt.grid(True)
plt.show()
```



Recommendation

The R script automates the installation and loading of essential packages for data manipulation and statistical analysis, ensuring a smooth workflow for handling a survey dataset. It reads the data, checks for missing values, and selects specific columns for analysis. By employing both GPArotation and FactoMineR packages for PCA, it provides flexibility in methods and visualization, with factoextra enhancing interpretability through biplots. For factor analysis, the script uses the psych package with varimax rotation, offering a clear structure of factor loadings and scores. To improve, consider adding detailed comments explaining each step and a section for data cleaning and preprocessing. Overall, the well-structured script provides a solid foundation for conducting PCA and FA on survey data.

R Codes

```
# 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")

# Call the function
install_and_load(packages)
```

```
survey_df <- read.csv('C:\\Users\\sayas\\OneDrive\\New folder\\python projects\\Surv
ey.csv', header = TRUE)
dim(survey_df)
names(survey_df)
head(survey_df)
str(survey_df)

#is.na(survey_df)
sum(is.na(survey_df))
sur_int = survey_df[, 18:46]
str(sur_int)
dim(sur_int)

# Performing PCA using GPArotation
library(GPArotation)
pca_1 <- principal(sur_int, 5, n.obs = 70, rotate = "promax")
print(pca_1)

# Performing PCA using FactoMineR
library(FactoMineR)
pca_2 <- PCA(sur_int, scale.unit = TRUE)
summary(pca_2)

# Using factoextra to plot the PCA biplot
library(factoextra)
fviz_pca_biplot(pca_2, repel = TRUE)

# Factor Analysis
factor_analysis<-fa(sur_int,nfactors = 4,rotate = "varimax")
names(factor_analysis)
print(factor_analysis$loadings,reorder=TRUE)
fa.diagram(factor_analysis)
print(factor_analysis$communality)
print(factor_analysis$scores)
```

Python Codes

```
import pandas as pd
import numpy as np
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
#pip install factor-analyzer pandas numpy scikit-learn matplotlib seaborn adjustText
from factor_analyzer import FactorAnalyzer
import matplotlib.pyplot as plt
import seaborn as sns
from adjustText import adjust_text
# Load the dataset
survey_df = pd.read_csv('C:\\Users\\sayas\\OneDrive\\New folder\\python projects\\S
urvey.csv', header=0)

# Check dataset structure
print(survey_df.shape)
print(survey_df.columns)
print(survey_df.head())
print(survey_df.info())

# Remove NA values
print(survey_df.isna().sum())  # Check for NA values
survey_df = survey_df.dropna()  # Drop rows with NA values
# Select columns of interest for PCA and factor analysis
sur_int = survey_df.iloc[:, 17:46]

# Standardize the data
scaler = StandardScaler()
sur_int_scaled = scaler.fit_transform(sur_int)
# Performing PCA using scikit-learn
pca = PCA(n_components=5)
```

```
pca.fit(sur_int_scaled)

# Explained variance
explained_variance = pca.explained_variance_ratio_ * 100
print(f"Explained variance by each component: {explained_variance}")
# PCA biplot function
def biplot_scores(pca, X):
    plt.figure(figsize=(10, 8))
    sns.scatterplot(x=pca.components_[0], y=pca.components_[1], alpha=0.7)
    texts = []
    for i, txt in enumerate(X.columns):
        texts.append(plt.text(pca.components_[0][i], pca.components_[1][i], txt, fontsize
=12))
    adjust_text(texts)

    xlabel = f'PC1 ({explained_variance[0]:.2f}%)'
    ylabel = f'PC2 ({explained_variance[1]:.2f}%)'

    plt.xlabel(xlabel)
    plt.ylabel(ylabel)
    plt.title('PCA Biplot')
    plt.grid(True)
    plt.show()

# Plot the PCA biplot
biplot_scores(pca, sur_int)
# Performing Factor Analysis using GPArotation equivalent
fa = FactorAnalyzer(rotation='promax', n_factors=5)
fa.fit(sur_int_scaled)

# Get factor loadings
loadings = fa.loadings_
print(f"Factor Loadings:\n{loadings}")
```

```
# Get communalities
communalities = fa.get_communalities()
print(f"Communalities:\n{communalities}")

# Get factor scores
factor_scores = fa.transform(sur_int_scaled)
print(f"Factor Scores:\n{factor_scores}")

# Additional Factor Analysis with Varimax rotation
fa_varimax = FactorAnalyzer(rotation='varimax', n_factors=4)
fa_varimax.fit(sur_int_scaled)

# Get Varimax factor loadings
varimax_loadings = fa_varimax.loadings_
print(f"Varimax Factor Loadings:\n{varimax_loadings}")

# Get Varimax communalities
varimax_communalities = fa_varimax.get_communalities()
print(f"Varimax Communalities:\n{varimax_communalities}")

# Get Varimax factor scores
varimax_factor_scores = fa_varimax.transform(sur_int_scaled)
print(f"Varimax Factor Scores:\n{varimax_factor_scores}")
# Plotting the factor loadings
def plot_factor_loadings(loadings, title):
    plt.figure(figsize=(10, 8))
    sns.heatmap(loadings, annot=True, cmap='viridis')
    plt.title(title)
    plt.xlabel('Factors')
    plt.ylabel('Variables')
    plt.show()

plot_factor_loadings(varimax_loadings, 'Factor Loadings with Varimax Rotation')
# Plotting factor scores
```

```
plt.figure(figsize=(10, 8))
sns.scatterplot(x=varimax_factor_scores[:, 0], y=varimax_factor_scores[:, 1], alpha=0
.7)
plt.xlabel('Factor 1')
plt.ylabel('Factor 2')
plt.title('Factor Scores with Varimax Rotation')
plt.grid(True)
plt.show()
```

References

1. www.github.com