DSC-424_Midterm_Exam

Sanket Patil

2024-02-14

Question 1:

1) How are they applying Factor Analysis?

–> In this study, the researchers applied exploratory factor analysis (EFA) to examine the underlying structure of technostress among primary school teachers. EFA is a statistical technique used to identify the underlying structure of a set of variables and to group them into meaningful dimensions or factors. Steps:

i) **Item Selection**: The researchers started by selecting 28 items related to technostress from previous literature.

ii) **Expert Verification**: The modified and translated items were sent to experts for verification of content validity, face validity, and criterion validity.

iii) **Pilot Study**: A pilot study was conducted with 106 primary school teachers to collect data using the newly developed questionnaire.

iv) **Exploratory Factor Analysis (EFA):**

a) Data Preparation: The researchers looked at the data collected from the pilot study. They used a method called principal component analysis to explore the data and find patterns.

b) KMO and Bartlett's Test: They checked if the data was good enough for their analysis by doing two tests. The tests told them that the data was good and could be used for their analysis.

c) Factor Extraction: They tried to find the main factors or groups in the data that explained most of the differences between the responses. They kept factors that were really important, based on a certain value.

d) Factor Rotation: They adjusted the factors they found to make them easier to understand. This helped them see clearer patterns in the data.

e) Factor Interpretation: They looked at each item in the questionnaire to see which factor it belonged to. They kept items that fit well with a factor and made sense.

f) Dimensionality Determination: They found five main groups or dimensions in the data that explained technostress among primary school teachers.

g) Reliability Analysis: They checked if the items they kept in the questionnaire were consistent and reliable. If items were consistent, they were more confident that their questionnaire accurately measured technostress. The researchers used factor analysis to figure out the different aspects of technostress experienced by primary school teachers. They found five main dimensions of technostress and created a dependable tool to measure it.

2) **What kind of rotation do they use?**

–> The researchers used Varimax rotation in the Exploratory Factor Analysis (EFA) procedure. Varimax rotation is a popular orthogonal rotation method that aims to maximize the variance of the squared loadings on each factor, making it easier to interpret the factors. In the context of factor analysis, rotation helps simplify the pattern of loadings and makes it easier to understand the relationships between variables and factors. The Varimax rotation method is commonly used when the factors are expected to be uncorrelated, which is a common assumption in many factor analysis applications.

3) **How many components do they concentrate on in their analysis? How did they arrive at these number of components?**

–> In their analysis, the researchers focused on five components. They arrived at this number of components through an Exploratory Factor Analysis (EFA) procedure, specifically employing Principal Component Analysis (PCA) with Varimax rotation.

4) **Explain the breakdown of the components and the significance of their names**.

–> Technical Oriented: This means teachers feeling stressed about using technology because it's tricky. They might struggle with computer programs, fixing broken equipment, or learning new tech stuff. Profession Oriented: This is about teachers feeling stressed because they worry about how well they're using technology in their job. They might feel pressure to be really good with tech, keep up with new teaching tools, or make sure they're using tech in the best way for teaching. Personal Oriented: This is when teachers feel stressed because technology makes their personal life busy or overwhelming. They might feel tired or frustrated from always being connected or having too much to do because of technology. Social Oriented: This is about stress from how technology affects relationships and social life. Teachers might worry about balancing work and personal time because of technology, or they might feel pressure from others to use technology in certain ways. Teaching-Learning Process Oriented: This means stress from using technology to teach and learn. Teachers might find it hard to adapt lessons for online learning, deal with students' different tech skills, or manage distractions caused by technology in class. The names of these components are important because they help us understand different ways teachers feel stressed about using technology. By breaking down technostress into these categories, the study helps us see that it's not just one type of stress, but many. This gives us a better idea of how technology affects teachers' feelings and work. Also, these categories help researchers organize the information they collected in the study. They can use these categories to see which areas of technostress are the most concerning for teachers. This can help them come up with ideas to help teachers deal with stress from technology better.

5) **How do they evaluate the stability of the components (i.e. factorability)?**

–> To evaluate the stability of the components or factorability, the researchers employed varous statistical methods and criteria: Kaiser-Meyer-Olkin (KMO) Measure of Sampling Adequacy: The KMO measure checks if our data is good for studying patterns. It tells us if the connections between different things we're studying are strong enough. If the KMO value is above 0.6, it means our data is probably good for studying these patterns. In our study, the KMO value was 0.884, so our data looked good.

Bartlett's Test of Sphericity: Bartlett's test checks if the connections between different things we're studying are strong enough for us to find meaningful patterns. If the result of Bartlett's test is significant (usually with a p-value less than 0.05), it means our data is suitable for finding these patterns. In our study, Bartlett's test showed a significant result with a p-value less than 0.001, indicating our data was good for finding patterns.

Eigenvalues: Eigenvalues tell us how much each pattern we find explains the differences in our data. If an eigenvalue is above 1.0, it means that pattern is important and explains a lot of the differences. In our study, we looked at eigenvalues for each pattern we found, and those above 1.0 were considered important.

Factor Loadings: Factor loadings show how much each thing we're studying relates to the patterns we found. Higher factor loadings mean stronger connections. We looked at factor loadings to see which things were most strongly connected to each pattern. Items with high factor loadings, usually above 0.55, were seen as important for that pattern.

By checking these things, the researchers made sure the patterns they found were reliable. The high KMO value, significant Bartlett's test, eigenvalues above 1.0, and strong factor loadings showed that the patterns they found were trustworthy representations of what they were studying, which in this case was technostress.

6) **Do they use these components in later analysis, such as regression? If so, what do they discovery?**

–> The researchers not used these components directly in this study. But they might use these components identified through exploratory factor analysis (EFA) in further analysis such as regression. In regression analysis, these components can serve as independent variables to predict or explain variability in a dependent variable

Technical oriented : In regression analysis, they may find that higher levels of stress in this dimension are associated with higher overall technostress levels among teachers.

Profession oriented : Regression analysis might reveal that stress in this dimension significantly predicts technostress, indicating that the professional context plays a role in teachers' experiences of technostress.

Personal oriented : Regression analysis could show that personal factors significantly contribute to technostress levels, highlighting the importance of considering individual differences in understanding and addressing technostress.

Social oriented : Regression analysis may demonstrate that social factors play a significant role in shaping technostress experiences among teachers.

Teaching-learning process oriented : Regression analysis might reveal that challenges or difficulties in adapting teaching methods to technology contribute to overall technostress levels among teachers.

The researchers would likely investigate how each of these dimensions contributes to technostress among primary school teachers. They might conduct regression analyses to examine the relationship between these dimensions and technostress, controlling for relevant covariates.

By looking at how these different aspects, like technical problems or pressures from work, relate to technostress in teachers, researchers can figure out what exactly makes teachers stressed about using technology. This helps them find ways to help teachers deal with this stress better in schools.

7) What overall conclusions does Principal Component Analysis allow them to draw?

–> PCA helps researchers understand technostress among primary school teachers better by breaking it down into different parts:

Different Aspects: Technostress is not just one thing; it's made up of five main parts: technology-related stress, stress from professional demands, stress from personal experiences, stress from social interactions, and stress related to teaching and learning.

Understanding the Factors: By using PCA, researchers can figure out how much each of these parts contributes to overall technostress. This helps them see which parts are most important and how they all fit together.

Finding Important Items: PCA also helps researchers figure out which specific questions or statements are most important for measuring technostress. Some questions might be really good at showing technostress, while others might not be as helpful.

Checking if it's Reliable: Researchers also use PCA to check if their questions are consistent and stable. If they are, it means the questions are good at measuring technostress in different situations.

Making Sure it Works: Finally, PCA helps researchers make sure that the questions they're asking really do measure technostress among primary school teachers. If the analysis shows that the questions are valid and reliable, then they can trust the results they get from using them.

Overall, PCA helps researchers understand technostress better so they can find ways to help teachers deal with it and feel better at work.

Question 2

```
# Define the matrices and vectors
Z <- matrix(c(1, 1, 1, 1, 9,5,-3,11), nrow=4, byrow=FALSE)
Z
```

```
##      [,1] [,2]
## [1,]    1    9
## [2,]    1    5
## [3,]    1   -3
## [4,]    1   11
```

```
Y <-  matrix(c(-1, 6, 0, 8), nrow=4, byrow=FALSE)
Y
```

```
##      [,1]
## [1,]   -1
## [2,]    6
## [3,]    0
## [4,]    8
```

```
M <- matrix(c(1, 11, 0,
              42, 52, 35,
              0, 9, 3), nrow=3, byrow=TRUE)
M
```

```
##      [,1] [,2] [,3]
## [1,]    1   11    0
## [2,]   42   52   35
## [3,]    0    9    3
```

```
N <- matrix(c(-10,-10,0,
              0,10,20,
              10,20,10), nrow=3, byrow=TRUE)
N
```

```
##      [,1] [,2] [,3]
## [1,]  -10  -10    0
## [2,]    0   10   20
## [3,]   10   20   10
```

```
v <- matrix(c(-11, 11, 22), nrow=3)
v
```

```
##      [,1]
## [1,]  -11
## [2,]   11
## [3,]   22
```

```
w <- matrix(c(8,-2,4), nrow=3)
w
```

```
##      [,1]
## [1,]    8
## [2,]   -2
## [3,]    4
```

v.w (dot product)

```
v_dot_w <- sum(v * w)
v_dot_w
```

```
## [1] -22
```

Scalar multiplication of -3 with w

```
neg_3_w <- -3 * w
neg_3_w
```

```
##      [,1]
## [1,]  -24
## [2,]    6
## [3,]  -12
```

Matrix-vector multiplication of M and v

```
M_times_v <- M %*% v
M_times_v
```

```
##      [,1]
## [1,]  110
## [2,]  880
## [3,]  165
```

Matrix addition of M and N

```
M_plus_N <- M + N
M_plus_N
```

```
##      [,1] [,2] [,3]
## [1,]   -9    1    0
## [2,]   42   62   55
## [3,]   10   29   13
```

Matrix subtraction of M and N

```
M_minus_N <- M - N
M_minus_N
```

```
##      [,1] [,2] [,3]
## [1,]   11   21    0
## [2,]   42   42   15
## [3,]  -10  -11   -7
```

Z transpose times Z

```
Z_transpose_times_Z <- crossprod(Z)
Z_transpose_times_Z
```

```
##      [,1] [,2]
## [1,]    4   22
## [2,]   22  236
```

Inverse of Z_transpose_times_Z

```
Z_transpose_times_Z_inv <- solve(Z_transpose_times_Z)
Z_transpose_times_Z_inv

##             [,1]         [,2]
## [1,]  0.51304348 -0.047826087
## [2,] -0.04782609  0.008695652
```

Z transpose times Y

```
Z_transpose_times_Y <- t(Z) %*% Y
Z_transpose_times_Y

##      [,1]
## [1,]   13
## [2,]  109
```

Calculate B

```
B <- Z_transpose_times_Z_inv %*% Z_transpose_times_Y
B

##          [,1]
## [1,] 1.456522
## [2,] 0.326087
```

Determinant of Z_transpose_times_Z

```
det_Z_transpose_times_Z <- det(Z_transpose_times_Z)
det_Z_transpose_times_Z

## [1] 460
```

—-

Question No. 3:

What are the different ways of treating missing values? Give examples that show the benefits or disadvantages of using these different strategies.

–> Treating missing values is a crucial step in data preprocessing, and there are several strategies to handle them. Each strategy has its own benefits and disadvantages, and the choice depends on the specific characteristics of the dataset and the goals of the analysis.

Below are some of the ways of treating missing values:

1) **Deletion:** Rows or columns containing missing values are entirely removed from the dataset. Advantages: Simple and straightforward Disadvantages: It can lead to loss of valuable information, especially if the missing values are not randomly distributed. This approach may result in biased analysis if the missing data is related to the outcome of interest. Example: Suppose we have a dataset of customer reviews for a product, and one of the columns is "Age" where some rows have missing values. If we delete rows with missing age values, we will lose valuable information about customers' age demographics, which could be important for targeted marketing campaigns.

2) **Imputation**: We can replace missing values with the help of mean, median, mode, or we can also predict the missing values with the help of algorithm such as KNN. Advantages: Retains all observations in the dataset, prevents information loss, and maintains sample size. Disadvantages: Imputed values may introduce bias or distort the original distribution of the variable. Example: If we have numeric variables, we can replace missing values with mean or median and if we have categorical variable, we can replace missing values by mode.

3) **Prediction Models:** We can predict the missing values using machine learning algorithms with the help of other features in the dataset. Advantages: Utilizes relationships between variables to make more accurate predictions. Can handle complex patterns in missing data. Disadvantages: Requires computational resources and may overfit the data. Not suitable for large datasets with high NA values. Example: If temperature readings are missing for certain days, a machine learning model trained on other weather variables like humidity, pressure, and wind speed can predict the missing temperature values.

4) **Flagging and Encoding:** We can create an additional binary indicator variable to signify if the value was missing or not. Advantages: Preserves the information that a value was missing, allowing models to account for the missingness pattern. Can be combined with imputation methods. Disadvantages: Increases the dimensionality of the dataset and may introduce noise if the missingness pattern is not informative.

5) **Domain Knowledge:** Use domain knowledge or expert judgment to fill in missing values based on context. Advantages: It incorporates subject matter expertise into the imputation process, leading to more meaningful results. Disadvantages: Subjective and may introduce bias if the expert judgment is incorrect or inconsistent. Also it is time consuming as it will require manual efforts. Example: In a healthcare dataset, if a patient's weight is missing, a medical professional may use their knowledge of the patient's medical history, demographics, and health condition to estimate a reasonable weight value.

Question 4:

Explain how to use R to check for the four assumptions of linear regression.

–> To check for the four assumptions of linear regression in R, we can follow below steps:

1) **Linearity between variables:**

Check for linearity between the independent variables and the dependent variable.

```
library(ggplot2)
```

Let’s assume df is our dataframe and y is our dependent variable with X1 and X2 as independent variables.

```
# Plot each independent variable against the dependent variable

ggplot(df, aes(x = x1, y = y)) + geom_point() + geom_smooth(method = “lm”, se = FALSE)
ggplot(df, aes(x = x2, y = y)) + geom_point() + geom_smooth(method = “lm”, se = FALSE)
```

2) **There should be less or no multicolinearity**:

To check correlation, we can use below function in R. cor(df) With the help of corrplot library, we can visualize the correlation matrix. corrplot(correlation_matrix, method = “circle”)

If the correlation value is greater than 0.7 or 0.8, we can say that variable is having high correlation value.

Check the Variance Inflation Factor (VIF):

Calculate VIF for each independent variable to assess multicollinearity.

```
library(car)

vif_values <- vif(lm(y ~ ., data = df))  # y is dependent variable and df is dataframe

print(vif_values)
```

If the VIF value is greater than 10, we can say that there is multicollinearity present. We can remove those variables.

3) **Error should be normally distributed**:

```
# Extract residuals
residuals <- resid(model)

# Plot histogram of residuals

hist(residuals, main = “Histogram of Residuals”, xlab = “Residuals”) skewness(residuals)

# Q-Q plot of residuals

qqnorm(residuals) qqline(residuals)
```

Resiuduals should follow a normal destribution with ideal skewness value of 0. We can check this with the help of above sample code in R.

4) **Homoscedasticity**:

Check if residuals have constant variance across different levels of the independent variables.

Extract residuals

```
residuals <- residuals(model)    # Model is trained regression model
```

Extract fitted values

```
fitted_values <- fitted(model)
```

Create a data frame for plotting

```
plot_data <- data.frame(Fitted = fitted_values, Residuals = residuals)
```

Plot residuals against fitted values

```
ggplot(plot_data, aes(x = Fitted, y = Residuals)) + geom_point() + geom_hline(yintercept = 0,
linetype = “dashed”, color = “red”) + labs(title = “Residuals vs Fitted Values Plot”, x = “Fitted
Values”, y = “Residuals”)
```

If the spread of residuals is roughly constant across all levels of fitted values, homoscedasticity is met. If the spread of residuals varies systematically across different levels of fitted values, heteroscedasticity may be present. The residuals should exhibit a random pattern around zero when plotted against the predicted values.

Question 5:

What are the advantages and disadvantages of using ridge and lasso regressions? Give examples of when you would use ridge compared to when you would use lasso regression.

–>

Ridge Regression and Lasso Regression are both regularization techniques used in linear regression. These techniques are used to address the multicollinearity and prevent overfitting.

Ridge Regression:

Ridge regression is a type of linear regression that adds a penalty term to the ordinary least squares (OLS) method, which helps to shrink the coefficients towards zero. This penalty term is proportional to the square of the coefficients, hence we call it “ridge” or L2, and it’s controlled by a parameter called lambda (λ).

Advantages: Ridge regression helps to mitigate multicollinearity, which occurs when independent variables are highly correlated with each other. It works well even when the number of predictors is greater than the number of observations. This will prevent overfitting.

Disadvantages: Ridge regression does not perform variable selection, hence it keeps all predictors in the model regardless of their importance. This can make the model less interpretable. It may not be suitable for scenarios where identifying the most influential predictors is essential.

Example: Suppose we are building a model to predict housing prices based on various features like square footage, number of bedrooms, and distance to amenities. If some of these features are highly correlated, ridge regression can effectively handle this correlation and produce more reliable predictions.

Lasso Regression:

Lasso regression, short for Least Absolute Shrinkage and Selection Operator, is another form of linear regression that adds a penalty term to the OLS method. However, unlike ridge regression, lasso uses the absolute values of the coefficients as the penalty term. Hence it is also called as L1.

Advantages: Lasso regression performs both parameter shrinkage and variable selection, making it useful for models with a large number of predictors. It tends to shrink less important coefficients to zero, which will effectively eliminate them from the model. It can generate more interpretable models by automatically selecting the most relevant predictors.

Disadvantages: Lasso regression can be sensitive to outliers in the data, potentially leading to biased coefficient estimates.

Example: Consider a scenario where we are analyzing customer data to predict their likelihood of purchasing a product. We have numerous customer attributes such as age, income, and purchase history. Lasso regression can help identify the most influential factors in predicting purchase behavior while disregarding less relevant variables, resulting in a more concise and interpretable model.

Question 6:

A researcher has run a factor analysis and found some of the factors to be correlated to each other and other factors, which are independent of each other. What type of rotation matrix should the researcher be using to properly interpret the factors?

--> The researcher should use an orthogonal rotation matrix to properly interpret the factors. Orthogonal rotation methods, such as Varimax or Quartimax, ensure that the resulting factors are uncorrelated with each other, making the interpretation of each factor more straightforward.

When factors are correlated, it can be challenging to understand the unique contribution of each factor to the underlying constructs being measured. Orthogonal rotation helps in simplifying the factor structure by maximizing the variance of factor loadings within each factor while minimizing the variance of factor loadings across factors, thus enhancing the interpretability of the factors.

In contrast, oblique rotation methods allow for factors to be correlated with each other, which can sometimes be more realistic depending on the underlying theoretical framework. However, in cases where factors are meant to be independent or when simpler interpretation is desired, orthogonal rotation is typically preferred.

The researcher should use an orthogonal rotation matrix to properly interpret the factors. Orthogonal rotation methods, such as Varimax or Quartimax, ensure that the resulting factors are uncorrelated with each other, making the interpretation of each factor more straightforward.

When factors are correlated, it can be challenging to understand the unique contribution of each factor to the underlying constructs being measured. Orthogonal rotation helps in simplifying the factor structure by maximizing the variance of factor loadings within each factor while minimizing the variance of factor loadings across factors, thus enhancing the interpretability of the factors.

In contrast, oblique rotation methods allow for factors to be correlated with each other, which can sometimes be more realistic depending on the underlying theoretical framework. However, in cases where factors are meant to be independent or when simpler interpretation is desired, orthogonal rotation is typically preferred.

Question 7:

What are the advantages and disadvantages of using exploratory factor analysis versus principal component analysis? –>

Exploratory Factor Analysis (EFA):

Advantages:

- Helps to understand the underlying structure or patterns in your data.
- Identifies latent (hidden) variables that may not be directly observed.
- Provides insight into relationships between variables.
- Allows for the testing of theoretical models.

Disadvantages:

- Requires a larger sample size for accurate results.
- More complex interpretation compared to principal component analysis.
- Assumes that variables are normally distributed.
- Results can be sensitive to different extraction methods and rotation techniques.

Principal Component Analysis (PCA):

Advantages:

- Reduces the curse of dimensionality.
- Simplifies data by reducing dimensionality while retaining most of the variation.
- Easy to understand and interpret.
- Less stringent assumptions compared to EFA. Useful for data compression and visualization.

Disadvantages:

- May not always capture underlying factors if correlations between variables are weak.
- Does not differentiate between common and unique variance.
- Assumes linear relationships between variables.
- May not be suitable for identifying latent variables.

Question 8:

You are conducting a study to predict what a student's grade will be in a class using linear regression. How would you analyze this study? What would you write in your statistical analysis plan? If you do not have enough information, what questions would you need to ask to obtain the information to run your analysis?

–> 1) **Data Collection**: Gather data from various sources, such as student records, course evaluations, and academic performance databases. Collect information on student demographics (e.g., age, gender, ethnicity), academic history (e.g., GPA, standardized test scores), and course-related variables (e.g., attendance, participation, homework scores). Take final grade as target variable.

2) **Data Cleaning and Preprocessing:** Check for unique values in all columns. Remove variables which have unique identifiers such as StudentId, Roll Number etc. Check for missing data in the collected variables and decide on appropriate strategies for handling missing values (e.g., imputation, deletion). Examine the distribution of numerical variables and identify outliers that may need to be addressed. Convert categorical variables into dummy variables if necessary to include them in the regression model.

3) **Exploratory Data Analysis (EDA):** Visualize the relationships between predictor variables (e.g., study hours, previous grades) and the target variable (final grade) using scatter plots, histograms, and correlation matrices. Explore potential multicollinearity among predictor variables to ensure they are not highly correlated with each other, as this could affect the stability and interpretability of the regression coefficients.

4) **Model Building:** Split the dataset into training and testing sets to evaluate the performance of the regression model. Select appropriate predictor variables based on theoretical considerations, domain knowledge, and statistical significance. Fit a linear regression model using the selected predictor variables and the final grade as the target variable. Consider including interaction terms or polynomial terms if there is evidence of nonlinear relationships between predictors and the target variable.

5) **Model Evaluation:** Assess the goodness of fit of the regression model using metrics such as R-squared, adjusted R-squared, and root mean squared error (RMSE). Examine the normality of residuals and homoscedasticity to ensure that the assumptions of linear regression are met. Evaluate the performance of the model on the test set to determine its predictive accuracy and generalizability to new data.

6) **Interpretation:** Interpret the coefficients of the regression model to understand the direction and strength of the relationships between predictor variables and the final grade.

Questions to obtain more information:

1) What kinds of information do we have about students in the dataset?

2) Is there anything we should worry about regarding the quality of the data or how it was collected?

3) Are there any other things we need to think about that might affect our results?

--We're asking if there might be other factors we need to consider, like if some students had different opportunities or experiences that could change their grades.

4) Do we know anything about how the class was taught or how students were graded?

--We're wondering if there's any extra information about how the class worked or how students were evaluated that could help us understand the grades better.

5) Are there any rules or concerns about privacy or being fair to the students when we use this data?

--We need to make sure that we are following the rules and being respectful to the students' privacy when we use their information for our study.

Question 9:

Load necessary libraries

```
library(tidyverse)
```

```
## Warning: package 'ggplot2' was built under R version 4.3.2
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ dplyr     1.1.3     ✓ readr     2.1.4
## ✓ forcats   1.0.0     ✓ stringr   1.5.0
## ✓ ggplot2   3.4.4     ✓ tibble    3.2.1
## ✓ lubridate 1.9.3     ✓ tidyr     1.3.0
## ✓ purrr     1.0.2
## ── Conflicts ──────────────────────────────────────────
tidyverse_conflicts() ──
## ✕ dplyr::filter() masks stats::filter()
## ✕ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(dplyr)
library(modeest)  # for the mfv function to find the mode
library(caret)    # For correlation
```

```
## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(fastDummies)
```

```
## Thank you for using fastDummies!
## To acknowledge our work, please cite the package:
## Kaplan, J. & Schlegel, B. (2023). fastDummies: Fast Creation of Dummy (Binary) Columns and Rows
from Categorical Variables. Version 1.7.1. URL: https://github.com/jacobkap/fastDummies,
https://jacobkap.github.io/fastDummies/.
```

Importing data in R

```
df <- read.csv("D:/Assignments_Depaul/DSC_424_Advance_Data_Analysis/Midterm
Exam/home_prices.csv", header = TRUE)
dim(df)
```

```
## [1] 545  13
```

```
head(df)
```

```
##   price_of_house area_of_house number_of_bedrooms number_of_bathrooms
## 1       13300000          7420                  4                   2
## 2       12250000          8960                  4                   4
## 3       12250000          9960                  3                   2
## 4       12215000          7500                  4                   2
## 5       11410000          7420                  4                   1
## 6       10850000          7500                  3                   3
##   Number_of_house_stories On_mainroad Has_guestroom Has_basement
## 1                       3         yes            no           no
## 2                       4         yes            no           no
## 3                       2         yes            no          yes
## 4                       2         yes            no          yes
## 5                       2         yes           yes          yes
## 6                       1         yes            no          yes
##   Has_hotwaterheating Has_airconditioning Number_of_parking_spaces
## 1                  no                 yes                        2
## 2                  no                 yes                        3
## 3                  no                  no                        2
## 4                  no                 yes                        3
## 5                  no                 yes                        2
## 6                  no                 yes                        2
##   in_preferred_area   is_furnished
## 1               yes      furnished
## 2                no      furnished
## 3               yes semi-furnished
## 4               yes      furnished
## 5                no      furnished
## 6               yes semi-furnished
```

Display summary statistics

```
summary(df)
```

```
##  price_of_house     area_of_house   number_of_bedrooms number_of_bathrooms
##  Min.   : 1750000   Min.   : 1650   Min.   :1.000      Min.   :1.000
##  1st Qu.: 3430000   1st Qu.: 3600   1st Qu.:2.000      1st Qu.:1.000
##  Median : 4340000   Median : 4600   Median :3.000      Median :1.000
##  Mean   : 4766729   Mean   : 5151   Mean   :2.965      Mean   :1.286
##  3rd Qu.: 5740000   3rd Qu.: 6360   3rd Qu.:3.000      3rd Qu.:2.000
##  Max.   :13300000   Max.   :16200   Max.   :6.000      Max.   :4.000
##  Number_of_house_stories On_mainroad        Has_guestroom
##  Min.   :1.000           Length:545         Length:545
##  1st Qu.:1.000           Class :character   Class :character
##  Median :2.000           Mode  :character   Mode  :character
##  Mean   :1.806
##  3rd Qu.:2.000
##  Max.   :4.000
##  Has_basement       Has_hotwaterheating Has_airconditioning
##  Length:545         Length:545          Length:545
##  Class :character   Class :character    Class :character
##  Mode  :character   Mode  :character    Mode  :character
##
##
##
##  Number_of_parking_spaces in_preferred_area  is_furnished
##  Min.   :0.0000           Length:545         Length:545
##  1st Qu.:0.0000           Class :character   Class :character
##  Median :0.0000           Mode  :character   Mode  :character
##  Mean   :0.6936
##  3rd Qu.:1.0000  
##  Max.   :3.0000  
```

Checking the class of the columns

```
column_types <- sapply(df, class)
print(column_types)
```

```
##           price_of_house            area_of_house       number_of_bedrooms 
##                "integer"                "integer"                "integer" 
##      number_of_bathrooms  Number_of_house_stories              On_mainroad 
##                "integer"                "integer"              "character" 
##            Has_guestroom             Has_basement      Has_hotwaterheating 
##              "character"              "character"              "character" 
##      Has_airconditioning Number_of_parking_spaces        in_preferred_area 
##              "character"                "integer"              "character" 
##             is_furnished 
##              "character" 
```

Count the number of categorical and numerical variables

```
num_categorical <- sum(column_types == "factor" | column_types == "character")
num_numerical <- sum(column_types == "numeric" | column_types == "integer")
```

Print the results

```
cat("Number of Categorical Variables:", num_categorical, "\n")
```

```
## Number of Categorical Variables: 7
```

```
cat("Number of Numerical Variables:", num_numerical, "\n")
```

```
## Number of Numerical Variables: 6
```

Checking number of unique values

```
unique_counts <- sapply(df, function(x) length(unique(x)))
```

Print the number of unique values for each column

```
print(unique_counts)
```

```
##           price_of_house            area_of_house       number_of_bedrooms 
##                      219                      284                        6 
##      number_of_bathrooms  Number_of_house_stories              On_mainroad 
##                        4                        4                        2 
##            Has_guestroom             Has_basement      Has_hotwaterheating 
##                        2                        2                        2 
##      Has_airconditioning Number_of_parking_spaces        in_preferred_area 
##                        2                        4                        2 
##          is_furnished
##                     3
```

Checking if data has NA values columnwise

```
na_percentages <- colMeans(is.na(df)) * 100
na_percentages
```

```
##         price_of_house          area_of_house      number_of_bedrooms
##                     0                      0                      0
##     number_of_bathrooms  Number_of_house_stories            On_mainroad
##                     0                      0                      0
##          Has_guestroom           Has_basement      Has_hotwaterheating
##                     0                      0                      0
##     Has_airconditioning Number_of_parking_spaces       in_preferred_area
##                     0                      0                      0
##          is_furnished
##                     0
```

Calculate the percentage of rows with NA

```
percentage_na_rows <- mean(apply(df, 1, function(row) any(is.na(row)))) * 100
print(percentage_na_rows)
```

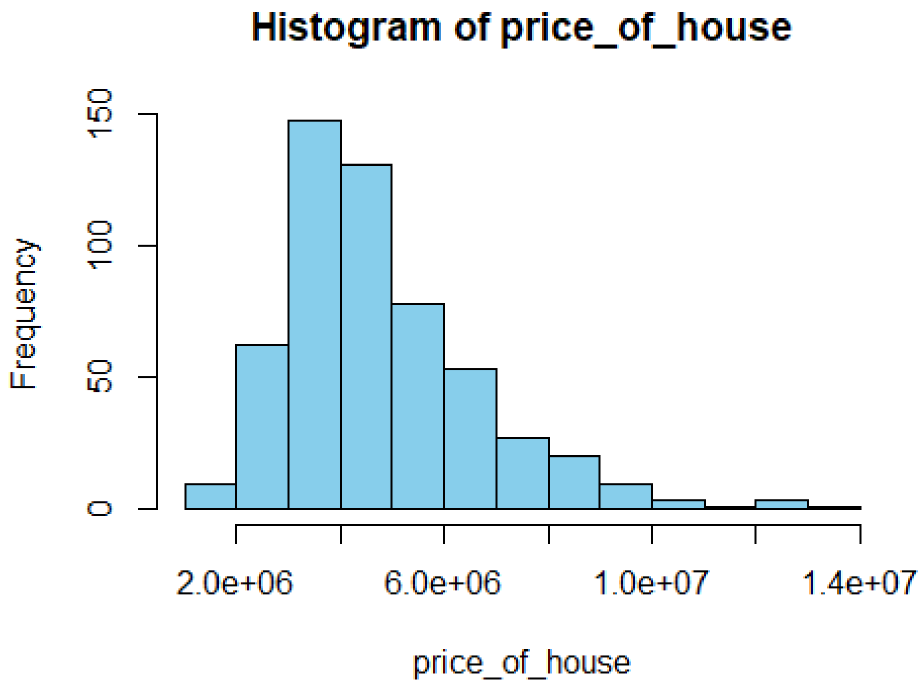
```
## [1] 0
```

No Missing values present in the data frame.

—————————————————— Target Variable Analysis

Plotting histogram of target variable

```
hist(df$price_of_house, main = "Histogram of price_of_house", xlab = "price_of_house", col =
"skyblue", border = "black")
```



```
price_of_house_skewness <- skewness(df$price_of_house)

cat("Skewness of Sale_Price:", price_of_house_skewness, "\n")

## Skewness of Sale_Price: 1.205574
```

Finding outliers

Calculate Z-scores

```
z_scores <- scale(df$price_of_house)
```

Set a threshold (e.g., 3 or -3)

```
threshold <- 3
```

Identify outliers

```
outliers <- which(abs(z_scores) > threshold)
```

Print the indices of outliers

```
cat("Indices of outliers in Sale_Price:", outliers, "\n")

## Indices of outliers in Sale_Price: 1 2 3 4 5 6
```

Print the values of outliers

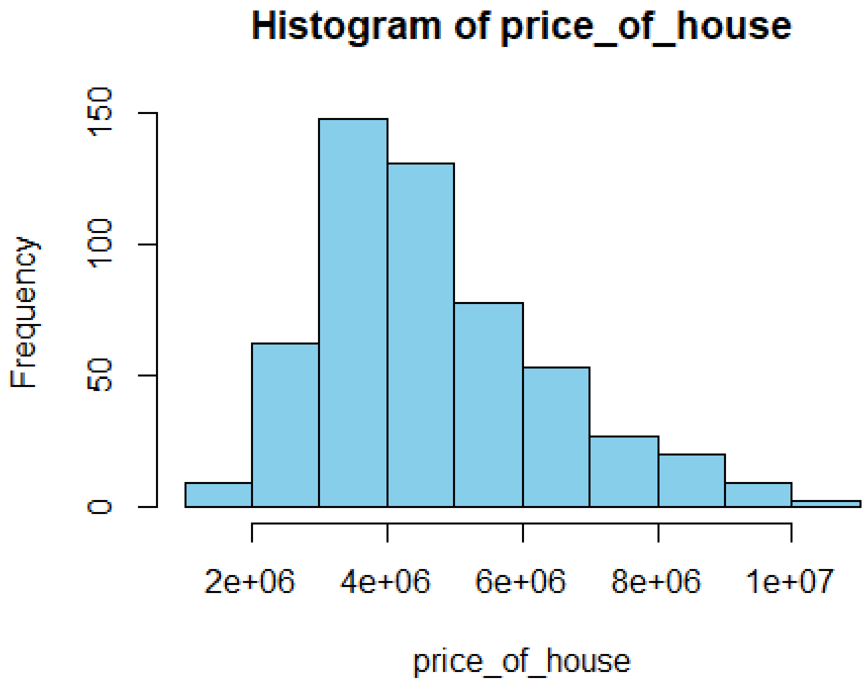
```
cat("Values of outliers in Sale_Price:", df$price_of_house[outliers], "\n")

## Values of outliers in Sale_Price: 13300000 12250000 12250000 12215000 11410000 10850000
```

Remove rows with outliers

```
df <- df[-outliers, ]
```

Print information about removed rows

```
cat("Number of rows removed:", length(outliers), "\n")

## Number of rows removed: 6

hist(df$price_of_house, main = "Histogram of price_of_house", xlab = "price_of_house", col =
"skyblue", border = "black")
```



```
skewness(df$price_of_house)

## [1] 0.846508
```

Identify numeric and categorical columns

```
numeric_cols <- sapply(df, is.numeric)
categorical_cols <- sapply(df, function(x) is.factor(x) | is.character(x))
```

Create df_numeric and df_categorical

```
df_numeric <- df[, numeric_cols]
df_categorical <- df[, categorical_cols]
```

———————————— Correlation check ————————————

-

Question 1:

Checking Correlation

```
correlation_matrix <- cor(df_numeric)
correlation_matrix
```

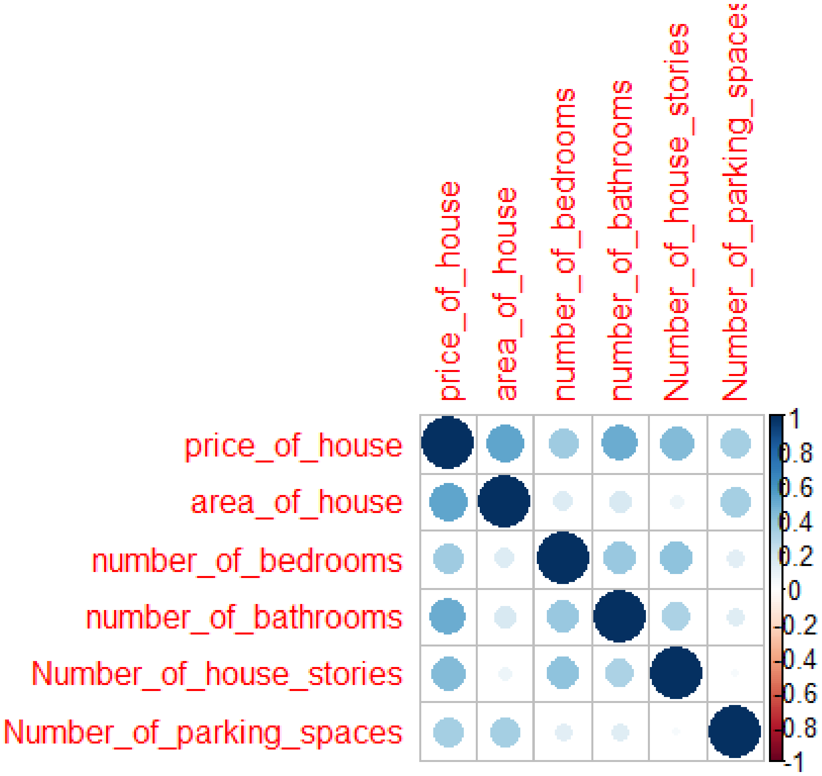
```
##                          price_of_house area_of_house number_of_bedrooms
## price_of_house                1.0000000    0.52905264          0.3583313
## area_of_house                 0.5290526    1.00000000          0.1413818
## number_of_bedrooms            0.3583313    0.14138182          1.0000000
## number_of_bathrooms           0.4912675    0.16840040          0.3727310
## Number_of_house_stories       0.4331775    0.07464968          0.4037353
## Number_of_parking_spaces      0.3384209    0.33489866          0.1206985
##                          number_of_bathrooms Number_of_house_stories
## price_of_house                     0.4912675              0.43317750
## area_of_house                      0.1684004              0.07464968
## number_of_bedrooms                 0.3727310              0.40373530
## number_of_bathrooms                1.0000000              0.31904802
## Number_of_house_stories            0.3190480              1.00000000
## Number_of_parking_spaces           0.1365822              0.03027760
##                          Number_of_parking_spaces
## price_of_house                          0.3384209
## area_of_house                           0.3348987
## number_of_bedrooms                      0.1206985
## number_of_bathrooms                     0.1365822
## Number_of_house_stories                 0.0302776
## Number_of_parking_spaces                1.0000000
```

```
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.3.2
```

```
## corrplot 0.92 loaded
```

```
corrplot(correlation_matrix, method = "circle")
```



- By checking correlation matrix, we can clearly see that all the variables have either less or moderate correlation with each other as well as with target variable.
- Hence, no need to remove any of the variable as no veriable is highly correlated.
- There are some variables with very less correlation values but we will try converting those variables into factors as there might be any non-linear relations between those variables and target variable because they have less number of unique values.

Calculate VIF scores

```
library(car)

## Loading required package: carData

##
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':
##
##     recode

## The following object is masked from 'package:purrr':
##
##     some

vif_scores <- vif(lm(formula = df$price_of_house ~ ., data = df_numeric))
```

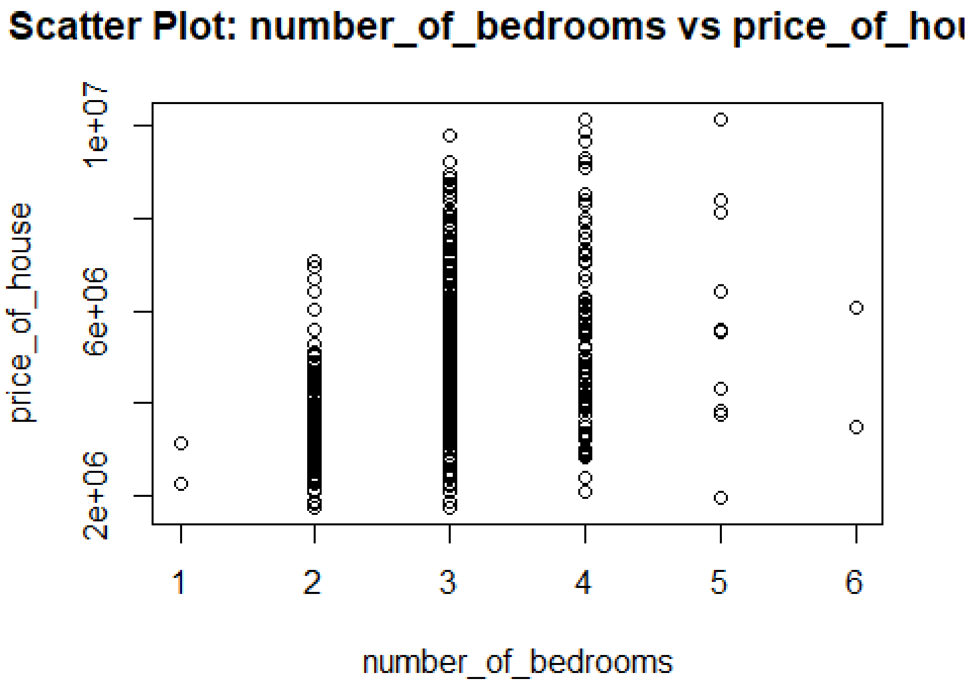
Print VIF scores

```
print(vif_scores)

##          area_of_house       number_of_bedrooms      number_of_bathrooms
##                1.151090                1.311500                1.234034
## Number_of_house_stories Number_of_parking_spaces
##                1.245955                1.139732
```

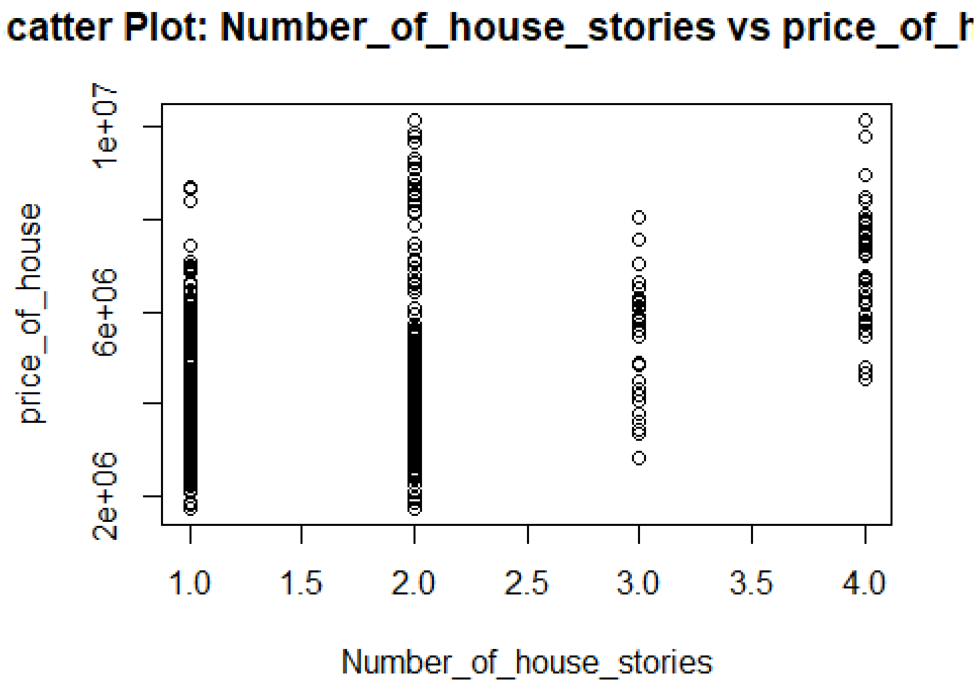
All the variables have vif value less than 10 hence we can say there is no multicollinearity.

——————————————- Converting variables to factors

```
plot(df$number_of_bedrooms, df$price_of_house,
     xlab = "number_of_bedrooms", ylab = "price_of_house",
     main = "Scatter Plot: number_of_bedrooms vs price_of_house")
```



```
plot(df$Number_of_house_stories, df$price_of_house,
     xlab = "Number_of_house_stories", ylab = "price_of_house",
     main = "Scatter Plot: Number_of_house_stories vs price_of_house")
```



```
df$number_of_bedrooms <- factor(df$number_of_bedrooms)
df$number_of_bathrooms <- factor(df$number_of_bathrooms)
df$Number_of_house_stories <- factor(df$Number_of_house_stories)
df$Number_of_parking_spaces <- factor(df$Number_of_parking_spaces)

sapply(df, class)
```

```
##           price_of_house            area_of_house       number_of_bedrooms 
##                "integer"                "integer"                 "factor" 
##      number_of_bathrooms  Number_of_house_stories              On_mainroad 
##                 "factor"                 "factor"              "character" 
##           Has_guestroom             Has_basement      Has_hotwaterheating 
##              "character"              "character"              "character" 
##      Has_airconditioning Number_of_parking_spaces        in_preferred_area 
##              "character"                 "factor"              "character" 
##             is_furnished 
##              "character"
```

Combining Data

Identify numeric and categorical columns again

```
numeric_cols <- sapply(df, is.numeric)
categorical_cols <- sapply(df, function(x) is.factor(x) | is.character(x))
```

Create df_numeric and df_categorical

```
df_numeric <- df[, numeric_cols]
df_categorical <- df[, categorical_cols]
```

Creating dummy variables

```
New_df <- cbind(df_numeric, df_categorical)

df_combined_dummies <- New_df %>% model.matrix(~ . - 1, data = .) %>%  as.data.frame()
dim(df_combined_dummies)

## [1] 539  24
```

Splitting Data

Creating a train/test partition

```
set.seed(123)
splitIndex <- createDataPartition(df_combined_dummies$price_of_house, p = 0.8, list = FALSE)
df_train <- df_combined_dummies[splitIndex, ]
df_test <- df_combined_dummies[-splitIndex, ]

dim(df_train)

## [1] 433  24

dim(df_test)

## [1] 106  24
```

Question 2:

Apply linear regression

```
Initial_model <- lm(price_of_house ~ ., data=df_train)
summary(Initial_model)

##
## Call:
## lm(formula = price_of_house ~ ., data = df_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2816744  -632024   -22657   471110  4066065
##
## Coefficients: (1 not defined because of singularities)
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 2.718e+06  7.289e+05   3.729 0.000219 ***
## area_of_house                 2.329e+02  2.525e+01   9.225  < 2e-16 ***
## number_of_bedrooms1          -9.359e+05  9.897e+05  -0.946 0.344893
## number_of_bedrooms2          -1.041e+06  7.092e+05  -1.468 0.142821
## number_of_bedrooms3          -7.348e+05  7.028e+05  -1.045 0.296444
## number_of_bedrooms4          -8.001e+05  7.082e+05  -1.130 0.259246
## number_of_bedrooms5          -5.148e+05  7.793e+05  -0.661 0.509240
## number_of_bedrooms6                  NA         NA      NA       NA
## number_of_bathrooms2          8.087e+05  1.259e+05   6.425 3.67e-10 ***
## number_of_bathrooms3          1.754e+06  3.715e+05   4.722 3.20e-06 ***
## Number_of_house_stories2      2.701e+05  1.256e+05   2.151 0.032053 *
## Number_of_house_stories3      6.412e+05  2.090e+05   3.067 0.002303 **
## Number_of_house_stories4      1.631e+06  2.274e+05   7.173 3.45e-12 ***
## On_mainroadyes                5.204e+05  1.416e+05   3.675 0.000270 ***
## Has_guestroomyes              3.918e+05  1.364e+05   2.873 0.004282 **
## Has_basementyes               3.023e+05  1.139e+05   2.653 0.008283 **
## Has_hotwaterheatingyes        1.102e+06  2.275e+05   4.846 1.79e-06 ***
## Has_airconditioningyes        7.275e+05  1.127e+05   6.453 3.10e-10 ***
## Number_of_parking_spaces1     3.052e+05  1.233e+05   2.476 0.013706 *
## Number_of_parking_spaces2     5.451e+05  1.362e+05   4.003 7.43e-05 ***
## Number_of_parking_spaces3    -3.432e+05  3.874e+05  -0.886 0.376198
## in_preferred_areayes          5.086e+05  1.248e+05   4.074 5.54e-05 ***
## `is_furnishedsemi-furnished`  8.064e+04  1.220e+05   0.661 0.508936
## is_furnishedunfurnished      -3.877e+05  1.295e+05  -2.993 0.002928 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 977200 on 410 degrees of freedom
## Multiple R-squared:   0.69,  Adjusted R-squared:  0.6734
## F-statistic: 41.48 on 22 and 410 DF,  p-value: < 2.2e-16
```

- R-squared value is 0.69, indicating that approximately 69% of the variance in house prices is accounted for by the predictor variables in the model.
- Adjusted R-squared value is 0.6734. The adjusted R-squared value adjusts the R-squared value for the number of predictors in the model, providing a more accurate measure of model fit, especially when comparing models with different numbers of predictors.
- The F-statistic tests the overall significance of the regression model by comparing the variance explained by the model to the variance not explained. The low p-value (< 2.2e-16) associated with the F-statistic suggests that the regression model is statistically significant, indicating that at least one of the predictor variables has a non-zero coefficient.
- The table under “Coefficients” provides information about the significance of individual predictor variables. Variables with p-values less than the 0.05 are considered statistically significant.
- Variables with p-values marked with asterisks (***) are highly significant
- The “Estimate” column provides the estimated coefficients (beta coefficients) of the predictor variables. These coefficients represent the change in the dependent variable for a one-unit change in the predictor variable, holding all other variables constant.
- For significant predictor variables, the beta coefficients indicate the direction and magnitude of the relationship between the predictor variable and the dependent variable. Positive coefficients indicate a positive relationship (increase in predictor variable leads to an increase in the dependent variable), while negative coefficients indicate a negative relationship (increase in predictor variable leads to a decrease in the dependent variable).

We will perform backward elimination model to select significant variables

Perform backward elimination using stepwise regression

```
backward_model <- step(Initial_model, direction = "backward")

## Start:  AIC=11966.66
## price_of_house ~ area_of_house + number_of_bedrooms1 + number_of_bedrooms2 +
##     number_of_bedrooms3 + number_of_bedrooms4 + number_of_bedrooms5 +
##     number_of_bedrooms6 + number_of_bathrooms2 + number_of_bathrooms3 +
##     Number_of_house_stories2 + Number_of_house_stories3 + Number_of_house_stories4 +
##     On_mainroadyes + Has_guestroomyes + Has_basementyes + Has_hotwaterheatingyes +
##     Has_airconditioningyes + Number_of_parking_spaces1 + Number_of_parking_spaces2 +
##     Number_of_parking_spaces3 + in_preferred_areayes + `is_furnishedsemi-furnished` +
##     is_furnishedunfurnished
##
##
## Step:  AIC=11966.66
## price_of_house ~ area_of_house + number_of_bedrooms1 + number_of_bedrooms2 +
##     number_of_bedrooms3 + number_of_bedrooms4 + number_of_bedrooms5 +
##     number_of_bathrooms2 + number_of_bathrooms3 + Number_of_house_stories2 +
##     Number_of_house_stories3 + Number_of_house_stories4 + On_mainroadyes +
##     Has_guestroomyes + Has_basementyes + Has_hotwaterheatingyes +
##     Has_airconditioningyes + Number_of_parking_spaces1 + Number_of_parking_spaces2 +
##     Number_of_parking_spaces3 + in_preferred_areayes + `is_furnishedsemi-furnished` +
##     is_furnishedunfurnished
##
##                              Df  Sum of Sq        RSS   AIC
## - number_of_bedroom5          1 4.1675e+11 3.9197e+14 11965
## - `is_furnishedsemi-furnished`  1 4.1735e+11 3.9197e+14 11965
## - Number_of_parking_spaces3     1 7.4948e+11 3.9230e+14 11966
## - number_of_bedrooms1          1 8.5398e+11 3.9240e+14 11966
## - number_of_bedrooms3          1 1.0437e+12 3.9259e+14 11966
## - number_of_bedrooms4          1 1.2189e+12 3.9277e+14 11966
## <none>                                    3.9155e+14 11967
## - number_of_bedrooms2          1 2.0586e+12 3.9361e+14 11967
## - Number_of_house_stories2      1 4.4190e+12 3.9597e+14 11970
## - Number_of_parking_spaces1     1 5.8526e+12 3.9740e+14 11971
## - Has_basementyes              1 6.7228e+12 3.9827e+14 11972
## - Has_guestroomyes             1 7.8809e+12 3.9943e+14 11973
## - is_furnishedunfurnished       1 8.5558e+12 4.0011e+14 11974
## - Number_of_house_stories3      1 8.9849e+12 4.0053e+14 11974
## - On_mainroadyes               1 1.2895e+13 4.0444e+14 11979
## - Number_of_parking_spaces2     1 1.5300e+13 4.0685e+14 11981
## - in_preferred_areayes          1 1.5853e+13 4.0740e+14 11982
## - number_of_bathrooms3          1 2.1298e+13 4.1285e+14 11988
## - Has_hotwaterheatingyes        1 2.2427e+13 4.1398e+14 11989
## - number_of_bathrooms2          1 3.9422e+13 4.3097e+14 12006
## - Has_airconditioningyes        1 3.9764e+13 4.3131e+14 12006
## - Number_of_house_stories4      1 4.9139e+13 4.4069e+14 12016
## - area_of_house                1 8.1266e+13 4.7282e+14 12046
##
## Step:  AIC=11965.12
## price_of_house ~ area_of_house + number_of_bedrooms1 + number_of_bedrooms2 +
##     number_of_bedrooms3 + number_of_bedrooms4 + number_of_bathrooms2 +
##     number_of_bathrooms3 + Number_of_house_stories2 + Number_of_house_stories3 +
##     Number_of_house_stories4 + On_mainroadyes + Has_guestroomyes +
##     Has_basementyes + Has_hotwaterheatingyes + Has_airconditioningyes +
##     Number_of_parking_spaces1 + Number_of_parking_spaces2 + Number_of_parking_spaces3 +
##     in_preferred_areayes + `is_furnishedsemi-furnished` + is_furnishedunfurnished
##
##                              Df  Sum of Sq        RSS   AIC
## - `is_furnishedsemi-furnished`  1 4.0772e+11 3.9237e+14 11964
## - number_of_bedrooms1          1 4.4406e+11 3.9241e+14 11964
## - Number_of_parking_spaces3     1 7.4378e+11 3.9271e+14 11964
## - number_of_bedrooms3          1 9.7442e+11 3.9294e+14 11964
## - number_of_bedrooms4          1 1.3220e+12 3.9329e+14 11965
## <none>                                    3.9197e+14 11965
## - number_of_bedrooms2          1 3.2921e+12 3.9526e+14 11967
## - Number_of_house_stories2      1 4.4204e+12 3.9639e+14 11968
## - Number_of_parking_spaces1     1 6.0264e+12 3.9799e+14 11970
## - Has_basementyes              1 6.5324e+12 3.9850e+14 11970
## - Has_guestroomyes             1 7.8632e+12 3.9983e+14 11972
## - is_furnishedunfurnished      1 8.5595e+12 4.0053e+14 11972
## - Number_of_house_stories3     1 8.8742e+12 4.0084e+14 11973
## - On_mainroadyes               1 1.3410e+13 4.0538e+14 11978
## - Number_of_parking_spaces2    1 1.5312e+13 4.0728e+14 11980
## - in_preferred_areayes         1 1.5871e+13 4.0784e+14 11980
## - number_of_bathrooms3         1 2.0929e+13 4.1289e+14 11986
## - Has_hotwaterheatingyes       1 2.2110e+13 4.1408e+14 11987
## - Has_airconditioningyes       1 3.9462e+13 4.3143e+14 12005
## - number_of_bathrooms2         1 3.9551e+13 4.3152e+14 12005
## - Number_of_house_stories4     1 4.9032e+13 4.4100e+14 12014
## - area_of_house                1 8.0873e+13 4.7284e+14 12044
## 
## Step:  AIC=11963.57
## price_of_house ~ area_of_house + number_of_bedrooms1 + number_of_bedrooms2 + 
##     number_of_bedrooms3 + number_of_bedrooms4 + number_of_bathrooms2 + 
##     number_of_bathrooms3 + Number_of_house_stories2 + Number_of_house_stories3 + 
##     Number_of_house_stories4 + On_mainroadyes + Has_guestroomyes + 
##     Has_basementyes + Has_hotwaterheatingyes + Has_airconditioningyes + 
##     Number_of_parking_spaces1 + Number_of_parking_spaces2 + Number_of_parking_spaces3 + 
##     in_preferred_areayes + is_furnishedunfurnished
## 
##                             Df  Sum of Sq        RSS   AIC
## - number_of_bedrooms1        1 4.6349e+11 3.9284e+14 11962
## - Number_of_parking_spaces3  1 7.7898e+11 3.9315e+14 11962
## - number_of_bedrooms3        1 8.5337e+11 3.9323e+14 11962
## - number_of_bedrooms4        1 1.1778e+12 3.9355e+14 11963
## <none>                                    3.9237e+14 11964
## - number_of_bedrooms2        1 3.1178e+12 3.9549e+14 11965
## - Number_of_house_stories2   1 4.4797e+12 3.9685e+14 11966
## - Number_of_parking_spaces1  1 5.8068e+12 3.9818e+14 11968
## - Has_basementyes            1 6.5515e+12 3.9893e+14 11969
## - Has_guestroomyes           1 7.7094e+12 4.0008e+14 11970
## - Number_of_house_stories3   1 8.8070e+12 4.0118e+14 11971
## - On_mainroadyes             1 1.3209e+13 4.0558e+14 11976
## - Number_of_parking_spaces2  1 1.5242e+13 4.0762e+14 11978
## - in_preferred_areayes       1 1.5651e+13 4.0803e+14 11978
## - is_furnishedunfurnished    1 1.6766e+13 4.0914e+14 11980
## - number_of_bathrooms3       1 2.1146e+13 4.1352e+14 11984
## - Has_hotwaterheatingyes     1 2.2137e+13 4.1451e+14 11985
## - Has_airconditioningyes     1 3.9062e+13 4.3144e+14 12003
## - number_of_bathrooms2       1 3.9209e+13 4.3158e+14 12003
## - Number_of_house_stories4   1 4.8713e+13 4.4109e+14 12012
## - area_of_house              1 8.0667e+13 4.7304e+14 12042
## 
## Step:  AIC=11962.08
## price_of_house ~ area_of_house + number_of_bedrooms2 + number_of_bedrooms3 + 
##     number_of_bedrooms4 + number_of_bathrooms2 + number_of_bathrooms3 + 
##     Number_of_house_stories2 + Number_of_house_stories3 + Number_of_house_stories4 + 
##     On_mainroadyes + Has_guestroomyes + Has_basementyes + Has_hotwaterheatingyes + 
##     Has_airconditioningyes + Number_of_parking_spaces1 + Number_of_parking_spaces2 + 
##     Number_of_parking_spaces3 + in_preferred_areayes + is_furnishedunfurnished
```

```
##
##                              Df  Sum of Sq        RSS   AIC
## - number_of_bedrooms3        1 5.0172e+11 3.9334e+14 11961
## - Number_of_parking_spaces3  1 7.5291e+11 3.9359e+14 11961
## - number_of_bedrooms4        1 8.0289e+11 3.9364e+14 11961
## <none>                                    3.9284e+14 11962
## - number_of_bedrooms2        1 2.6574e+12 3.9549e+14 11963
## - Number_of_house_stories2   1 5.0176e+12 3.9785e+14 11966
## - Number_of_parking_spaces1  1 5.8831e+12 3.9872e+14 11966
## - Has_basementyes            1 6.7616e+12 3.9960e+14 11968
## - Has_guestroomyes           1 7.6780e+12 4.0052e+14 11968
## - Number_of_house_stories3   1 9.2019e+12 4.0204e+14 11970
## - On_mainroadyes             1 1.3215e+13 4.0605e+14 11974
## - Number_of_parking_spaces2  1 1.5372e+13 4.0821e+14 11977
## - in_preferred_areayes       1 1.5606e+13 4.0844e+14 11977
## - is_furnishedunfurnished    1 1.6602e+13 4.0944e+14 11978
## - number_of_bathrooms3       1 2.1709e+13 4.1455e+14 11983
## - Has_hotwaterheatingyes     1 2.2426e+13 4.1526e+14 11984
## - Has_airconditioningyes     1 3.9297e+13 4.3213e+14 12001
## - number_of_bathrooms2       1 3.9789e+13 4.3263e+14 12002
## - Number_of_house_stories4   1 4.9129e+13 4.4197e+14 12011
## - area_of_house              1 8.2108e+13 4.7495e+14 12042
##
## Step:  AIC=11960.64
## price_of_house ~ area_of_house + number_of_bedrooms2 + number_of_bedrooms4 +
##     number_of_bathrooms2 + number_of_bathrooms3 + Number_of_house_stories2 +
##     Number_of_house_stories3 + Number_of_house_stories4 + On_mainroadyes +
##     Has_guestroomyes + Has_basementyes + Has_hotwaterheatingyes +
##     Has_airconditioningyes + Number_of_parking_spaces1 + Number_of_parking_spaces2 +
##     Number_of_parking_spaces3 + in_preferred_areayes + is_furnishedunfurnished
##
##                              Df  Sum of Sq        RSS   AIC
## - number_of_bedrooms4        1 3.3988e+11 3.9368e+14 11959
## - Number_of_parking_spaces3  1 7.7842e+11 3.9412e+14 11960
## <none>                                    3.9334e+14 11961
## - Number_of_house_stories2   1 5.1960e+12 3.9854e+14 11964
## - number_of_bedrooms2        1 5.2915e+12 3.9863e+14 11964
## - Number_of_parking_spaces1  1 5.7927e+12 3.9913e+14 11965
## - Has_basementyes            1 6.7741e+12 4.0011e+14 11966
## - Has_guestroomyes           1 7.6275e+12 4.0097e+14 11967
## - Number_of_house_stories3   1 9.2116e+12 4.0255e+14 11969
## - On_mainroadyes             1 1.2957e+13 4.0630e+14 11973
## - Number_of_parking_spaces2  1 1.5161e+13 4.0850e+14 11975
## - in_preferred_areayes       1 1.5315e+13 4.0865e+14 11975
## - is_furnishedunfurnished    1 1.6349e+13 4.0969e+14 11976
## - Has_hotwaterheatingyes     1 2.2961e+13 4.1630e+14 11983
## - number_of_bathrooms3       1 2.3340e+13 4.1668e+14 11984
## - Has_airconditioningyes     1 3.9169e+13 4.3251e+14 12000
## - number_of_bathrooms2       1 4.1014e+13 4.3435e+14 12002
## - Number_of_house_stories4   1 4.8832e+13 4.4217e+14 12009
## - area_of_house              1 8.4248e+13 4.7759e+14 12043
```

```
## 
## Step:  AIC=11959.01
## price_of_house ~ area_of_house + number_of_bedrooms2 + number_of_bathrooms2 + 
##     number_of_bathrooms3 + Number_of_house_stories2 + Number_of_house_stories3 + 
##     Number_of_house_stories4 + On_mainroadyes + Has_guestroomyes + 
##     Has_basementyes + Has_hotwaterheatingyes + Has_airconditioningyes + 
##     Number_of_parking_spaces1 + Number_of_parking_spaces2 + Number_of_parking_spaces3 + 
##     in_preferred_areayes + is_furnishedunfurnished
## 
##                             Df  Sum of Sq        RSS   AIC
## - Number_of_parking_spaces3  1 7.7807e+11 3.9446e+14 11958
## <none>                                    3.9368e+14 11959
## - Number_of_house_stories2   1 4.8682e+12 3.9855e+14 11962
## - number_of_bedrooms2        1 5.0912e+12 3.9877e+14 11963
## - Number_of_parking_spaces1  1 5.8264e+12 3.9951e+14 11963
## - Has_basementyes            1 6.8994e+12 4.0058e+14 11964
## - Has_guestroomyes           1 7.5003e+12 4.0118e+14 11965
## - Number_of_house_stories3   1 9.0214e+12 4.0270e+14 11967
## - On_mainroadyes             1 1.3118e+13 4.0680e+14 11971
## - Number_of_parking_spaces2  1 1.5059e+13 4.0874e+14 11973
## - in_preferred_areayes       1 1.5624e+13 4.0930e+14 11974
## - is_furnishedunfurnished    1 1.6136e+13 4.0981e+14 11974
## - number_of_bathrooms3       1 2.3163e+13 4.1684e+14 11982
## - Has_hotwaterheatingyes     1 2.3238e+13 4.1692e+14 11982
## - Has_airconditioningyes     1 3.9388e+13 4.3307e+14 11998
## - number_of_bathrooms2       1 4.0799e+13 4.3448e+14 12000
## - Number_of_house_stories4   1 4.8553e+13 4.4223e+14 12007
## - area_of_house              1 8.3909e+13 4.7759e+14 12041
## 
## Step:  AIC=11957.86
## price_of_house ~ area_of_house + number_of_bedrooms2 + number_of_bathrooms2 + 
##     number_of_bathrooms3 + Number_of_house_stories2 + Number_of_house_stories3 + 
##     Number_of_house_stories4 + On_mainroadyes + Has_guestroomyes + 
##     Has_basementyes + Has_hotwaterheatingyes + Has_airconditioningyes + 
##     Number_of_parking_spaces1 + Number_of_parking_spaces2 + in_preferred_areayes + 
##     is_furnishedunfurnished
## 
##                             Df  Sum of Sq        RSS   AIC
## <none>                                    3.9446e+14 11958
## - number_of_bedrooms2        1 4.7826e+12 3.9924e+14 11961
## - Number_of_house_stories2   1 5.4136e+12 3.9987e+14 11962
## - Number_of_parking_spaces1  1 6.4260e+12 4.0088e+14 11963
## - Has_basementyes            1 6.8545e+12 4.0131e+14 11963
## - Has_guestroomyes           1 7.5366e+12 4.0199e+14 11964
## - Number_of_house_stories3   1 9.5378e+12 4.0399e+14 11966
## - On_mainroadyes             1 1.2835e+13 4.0729e+14 11970
## - is_furnishedunfurnished    1 1.5682e+13 4.1014e+14 11973
## - in_preferred_areayes       1 1.6119e+13 4.1058e+14 11973
## - Number_of_parking_spaces2  1 1.6438e+13 4.1090e+14 11974
## - Has_hotwaterheatingyes     1 2.3237e+13 4.1769e+14 11981
## - number_of_bathrooms3       1 2.3307e+13 4.1776e+14 11981
## - Has_airconditioningyes     1 3.9795e+13 4.3425e+14 11998
## - number_of_bathrooms2       1 4.1373e+13 4.3583e+14 11999
## - Number_of_house_stories4   1 4.7974e+13 4.4243e+14 12006
## - area_of_house              1 8.3146e+13 4.7760e+14 12039
```

Summary of final model after backward elimination

```
summary(backward_model)
```

```
## 
## Call:
## lm(formula = price_of_house ~ area_of_house + number_of_bedrooms2 + 
##     number_of_bathrooms2 + number_of_bathrooms3 + Number_of_house_stories2 + 
##     Number_of_house_stories3 + Number_of_house_stories4 + On_mainroadyes + 
##     Has_guestroomyes + Has_basementyes + Has_hotwaterheatingyes + 
##     Has_airconditioningyes + Number_of_parking_spaces1 + Number_of_parking_spaces2 + 
##     in_preferred_areayes + is_furnishedunfurnished, data = df_train)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -2786793  -647885   -23983   461631  4032461 
## 
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)               2019040.05  192181.39  10.506  < 2e-16 ***
## area_of_house                 230.86      24.65   9.364  < 2e-16 ***
## number_of_bedrooms2       -294082.48  130944.74  -2.246 0.025238 *  
## number_of_bathrooms2       812316.26  122976.25   6.605 1.21e-10 ***
## number_of_bathrooms3      1797366.10  362532.46   4.958 1.04e-06 ***
## Number_of_house_stories2   286784.87  120024.01   2.389 0.017321 *  
## Number_of_house_stories3   654299.34  206303.45   3.172 0.001629 ** 
## Number_of_house_stories4  1593071.30  223968.67   7.113 4.99e-12 ***
## On_mainroadyes             514626.38  139878.23   3.679 0.000265 ***
## Has_guestroomyes           382236.24  135580.61   2.819 0.005043 ** 
## Has_basementyes            303916.67  113036.49   2.689 0.007462 ** 
## Has_hotwaterheatingyes    1115111.00  225257.58   4.950 1.08e-06 ***
## Has_airconditioningyes     723155.01  111626.75   6.478 2.62e-10 ***
## Number_of_parking_spaces1  315843.01  121326.45   2.603 0.009565 ** 
## Number_of_parking_spaces2  557990.60  134013.85   4.164 3.81e-05 ***
## in_preferred_areayes       509324.57  123531.22   4.123 4.52e-05 ***
## is_furnishedunfurnished   -420683.22  103444.19  -4.067 5.70e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 973800 on 416 degrees of freedom
## Multiple R-squared:  0.6877, Adjusted R-squared:  0.6757 
## F-statistic: 57.26 on 16 and 416 DF,  p-value: < 2.2e-16
```

- area_of_house: For every one unit increase in the area of the house, the price_of_house is estimated to increase by $230.86, on average.

- number_of_bedrooms2: For houses with two bedrooms compared to houses with one bedroom, the price_of_house is estimated to decrease by $294,082.48, on average. number_of_bathrooms2: For houses with two bathrooms compared to houses with one bathroom, the price_of_house is estimated to increase by $812,316.26, on average.
- number_of_bathrooms3: For houses with three bathrooms compared to houses with one bathroom, the price_of_house is estimated to increase by $1,797,366.10, on average. Number_of_house_stories2: For houses with two stories compared to houses with one story, the price_of_house is estimated to increase by $286,784.87, on average.
- Number_of_house_stories3: For houses with three stories compared to houses with one story, the price_of_house is estimated to increase by $654,299.34, on average.
- Number_of_house_stories4: For houses with four stories compared to houses with one story, the price_of_house is estimated to increase by $1,593,071.30, on average.
- On_mainroadyes: For houses located on a main road compared to those not on a main road, the price_of_house is estimated to increase by $514,626.38, on average.
- Has_guestroomyes: For houses with a guest room compared to those without, the price_of_house is estimated to increase by $382,236.24, on average.
- Has_basementyes: For houses with a basement compared to those without, the price_of_house is estimated to increase by $303,916.67, on average.
- Has_hotwaterheatingyes: For houses with hot water heating compared to those without, the price_of_house is estimated to increase by $1,115,111.00, on average.
- Has_airconditioningyes: For houses with air conditioning compared to those without, the price_of_house is estimated to increase by $723,155.01, on average. Number_of_parking_spaces1: For houses with one parking space compared to those without, the price_of_house is estimated to increase by $315,843.01, on average. Number_of_parking_spaces2: For houses with two parking spaces compared to those without, the price_of_house is estimated to increase by $557,990.60, on average.
- in_preferred_areayes: For houses in a preferred area compared to those not in a preferred area, the price_of_house is estimated to increase by $509,324.57, on average. is_furnishedunfurnished: For houses that are unfurnished compared to those that are fully furnished, the price_of_house is estimated to decrease by $420,683.22, on average.

Equation:

price_of_house = 2019040.05 + (230.86 * area_of_house) - (294082.48 * number_of_bedrooms2) + (812316.26 * number_of_bathrooms2) + (1797366.10 * number_of_bathrooms3) + (286784.87 * Number_of_house_stories2) + (654299.34 * Number_of_house_stories3) + (1593071.30 * Number_of_house_stories4) + (514626.38 * On_mainroadyes) + (382236.24 * Has_guestroomyes) + (303916.67 * Has_basementyes) + (1115111.00 * Has_hotwaterheatingyes) + (723155.01 * Has_airconditioningyes) + (315843.01

* Number_of_parking_spaces1) + (557990.60 * Number_of_parking_spaces2) + (509324.57 * in_preferred_areayes) + (-420683.22 * is_furnishedunfurnished)

——————————————- Lassso Regression ——————————

Question 3:

Load the glmnet package

```
library(glmnet)

## Warning: package 'glmnet' was built under R version 4.3.2

## Loading required package: Matrix

##
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack

## Loaded glmnet 4.1-8
```

Fit the Lasso regression model

```
lasso_model <- cv.glmnet(as.matrix(df_train[, -1]), df_train$price_of_house, alpha = 1)
```

Print the summary of the Lasso model

```
print(lasso_model)

##
## Call:  cv.glmnet(x = as.matrix(df_train[, -1]), y = df_train$price_of_house,      alpha = 1)
##
## Measure: Mean-Squared Error
##
##     Lambda Index   Measure        SE Nonzero
## min  18307    43 1.029e+12 9.641e+10      21
## 1se  97697    25 1.111e+12 1.194e+11      15
```

Display optimal lambda value

```
best_lambda <- lasso_model$lambda.min
print(paste("Optimal lambda:", best_lambda))

## [1] "Optimal lambda: 18306.5455270837"
```

Display coefficients

```
lasso_coef <- coef(lasso_model, s = best_lambda)
print(lasso_coef)
```

```
## 24 x 1 sparse Matrix of class "dgCMatrix"
##                                      s1
## (Intercept)                2124920.4225
## area_of_house                  229.2099
## number_of_bedrooms1         -50099.1069
## number_of_bedrooms2        -322321.1074
## number_of_bedrooms3               .    
## number_of_bedrooms4               .    
## number_of_bedrooms5         147308.2030
## number_of_bedrooms6         447695.2267
## number_of_bathrooms2        802949.3437
## number_of_bathrooms3       1651464.1550
## Number_of_house_stories2    202807.0400
## Number_of_house_stories3    523164.4478
## Number_of_house_stories4   1493825.4022
## On_mainroadyes              507692.4891
## Has_guestroomyes            374194.2584
## Has_basementyes             270222.4864
## Has_hotwaterheatingyes     1044977.4455
## Has_airconditioningyes      711440.9699
## Number_of_parking_spaces1   261268.1354
## Number_of_parking_spaces2   502843.8799
## Number_of_parking_spaces3  -229182.4152
## in_preferred_areayes        486585.3877
## is_furnishedsemi-furnished   32639.1436
## is_furnishedunfurnished    -395605.3677
```

The results of the Lasso regression are different from the initial linear regression model. Lasso regression introduces a penalty term that encourages sparsity in the coefficients, leading to some coefficients being exactly zero. This is evident in the output where some coefficients are shown as “.” indicating zero.

Benifit:

- The benefit of using Lasso regression for this research question is that it automatically selects the most important features by shrinking the less important ones to zero.
- Lasso regression made coefficients of variables number_of_bedrooms3 and number_of_bedrooms4 to 0. Hence lasso regression performed variable selection here.

Disadvantages:

- The cost of using Lasso regression is that it may discard some potentially useful variables, leading to a simpler but less interpretable model. Moreover, the choice of the regularization parameter (lambda) needs to be optimized, which might require cross-validation.

———————————————— EDA -

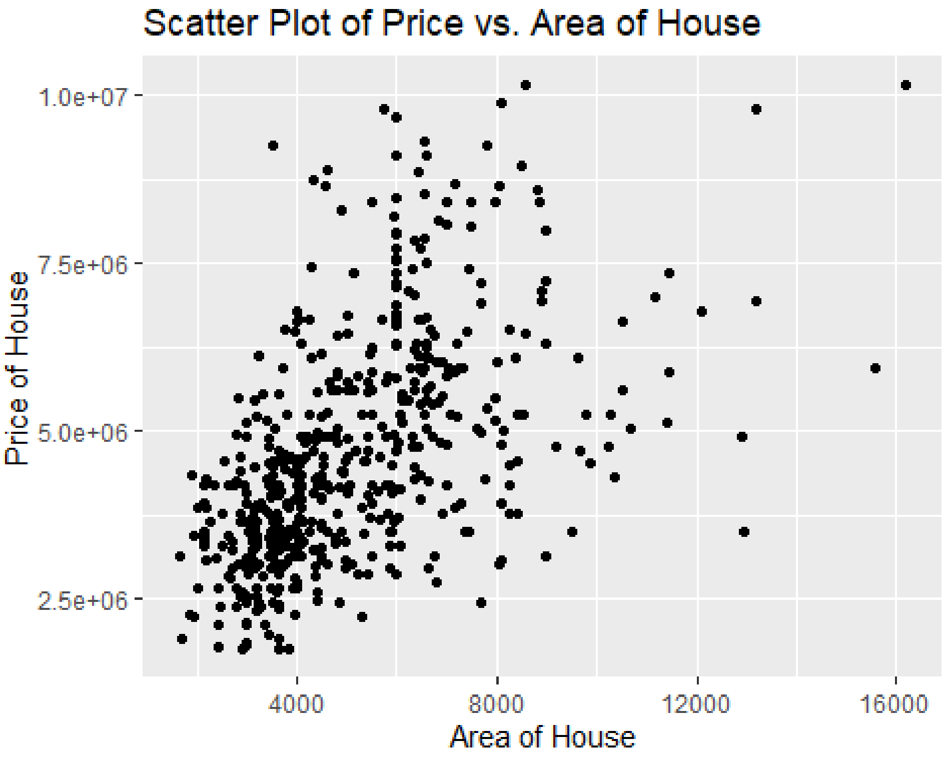
Question 4:

```
colnames(New_df)
```

```
##  [1] "price_of_house"          "area_of_house"
##  [3] "number_of_bedrooms"      "number_of_bathrooms"
##  [5] "Number_of_house_stories" "On_mainroad"
##  [7] "Has_guestroom"           "Has_basement"
##  [9] "Has_hotwaterheating"     "Has_airconditioning"
## [11] "Number_of_parking_spaces" "in_preferred_area"
## [13] "is_furnished"
```

Scatter plot between area_of_house and price_of_house

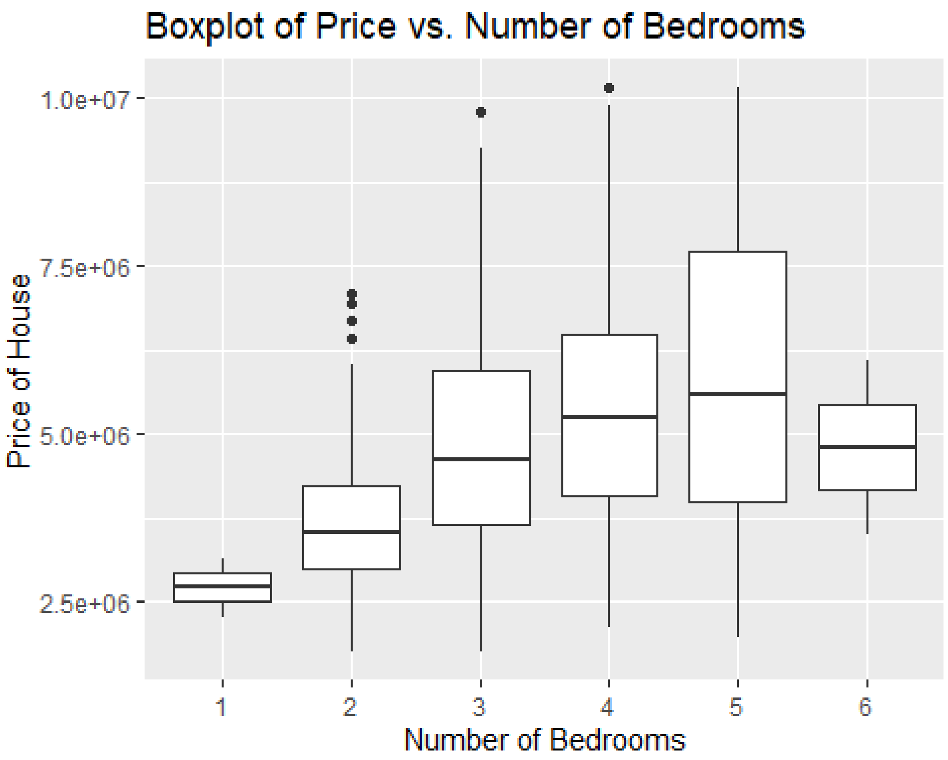
```
ggplot(New_df, aes(x = area_of_house, y = price_of_house)) +
  geom_point() +
  labs(x = "Area of House", y = "Price of House") +
  ggtitle("Scatter Plot of Price vs. Area of House")
```



- By looking at the scatterplot, we can see that there is moderate positive linear relation between the varialbes. Hence the variable area_of_house will be a significant variable while predicting the price.

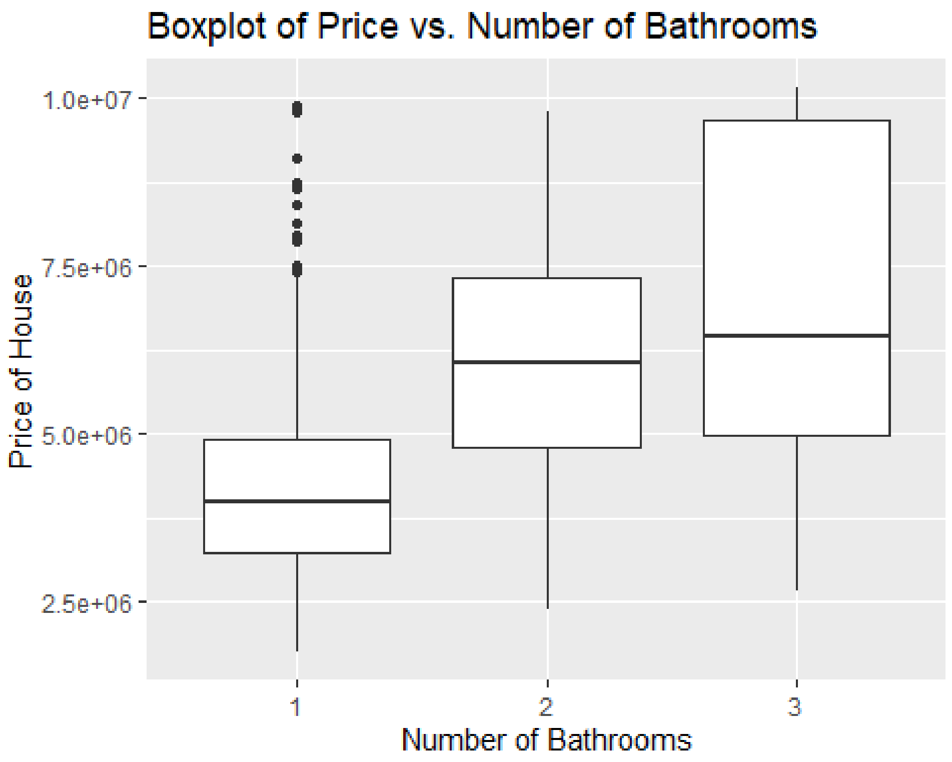
Boxplots

```
ggplot(New_df, aes(x = factor(number_of_bedrooms), y = price_of_house)) +
  geom_boxplot() +
  labs(x = "Number of Bedrooms", y = "Price of House") +
  ggtitle("Boxplot of Price vs. Number of Bedrooms")
```



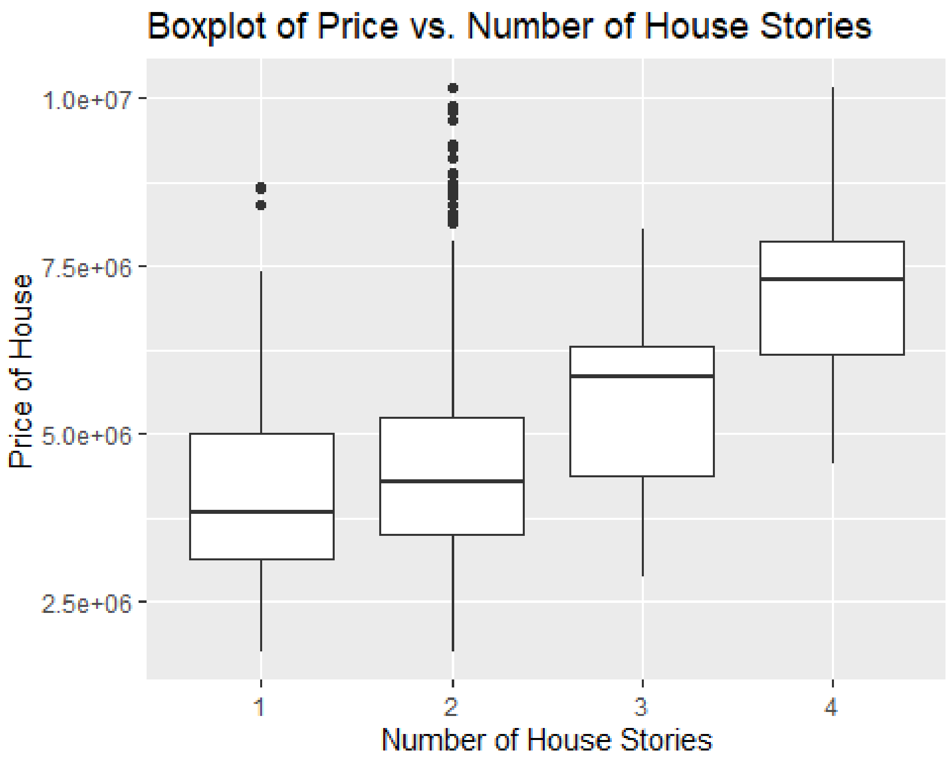
- In above plot, we can clearly see that the increase in median value of Number of bedrooms increases the price of the house.
- So that after converting the variable to factor, this variable might be significant for us to predict the price.

```
ggplot(New_df, aes(x = factor(number_of_bathrooms), y = price_of_house)) +
  geom_boxplot() +
  labs(x = "Number of Bathrooms", y = "Price of House") +
  ggtitle("Boxplot of Price vs. Number of Bathrooms")
```



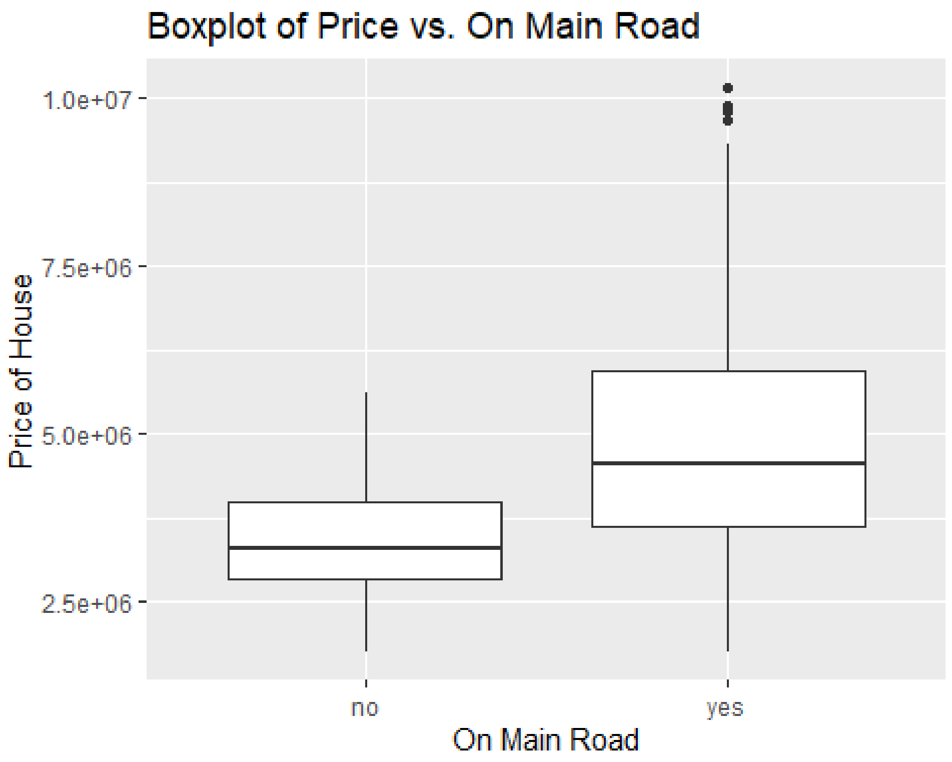
- Price increases as number of bathrooms increases. Number of bathrooms are higher as we increase the price of the house.

```
ggplot(New_df, aes(x = factor(Number_of_house_stories), y = price_of_house)) +
  geom_boxplot() +
  labs(x = "Number of House Stories", y = "Price of House") +
  ggtitle("Boxplot of Price vs. Number of House Stories")
```



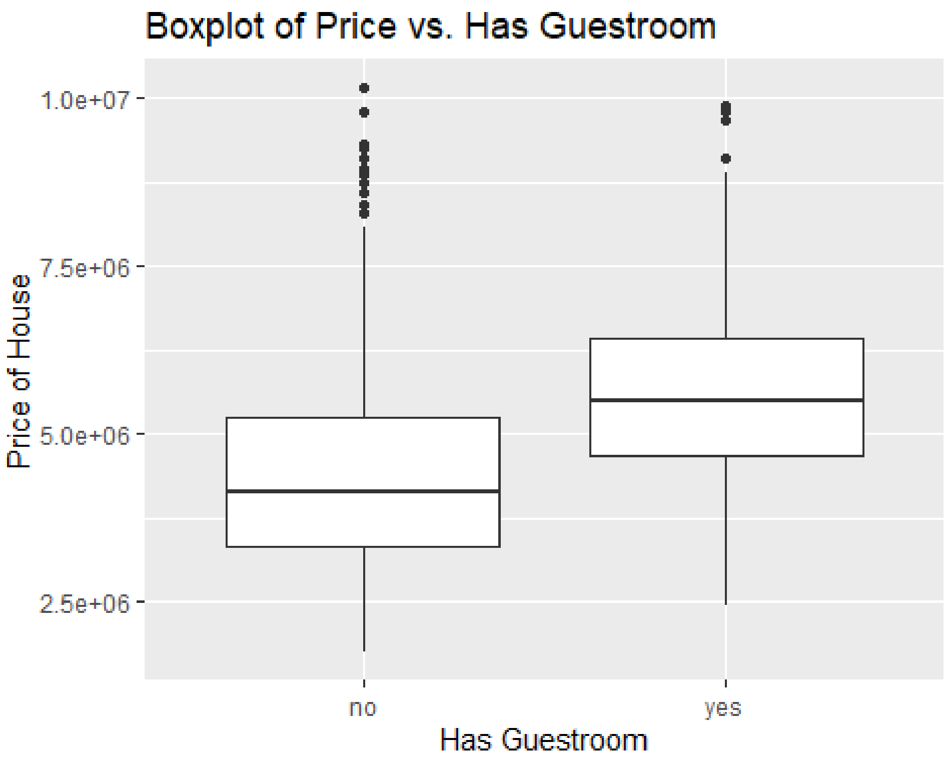
- Price is higher for higher number of house stories.

```
ggplot(New_df, aes(x = factor(On_mainroad), y = price_of_house)) +
  geom_boxplot() +
  labs(x = "On Main Road", y = "Price of House") +
  ggtitle("Boxplot of Price vs. On Main Road")
```



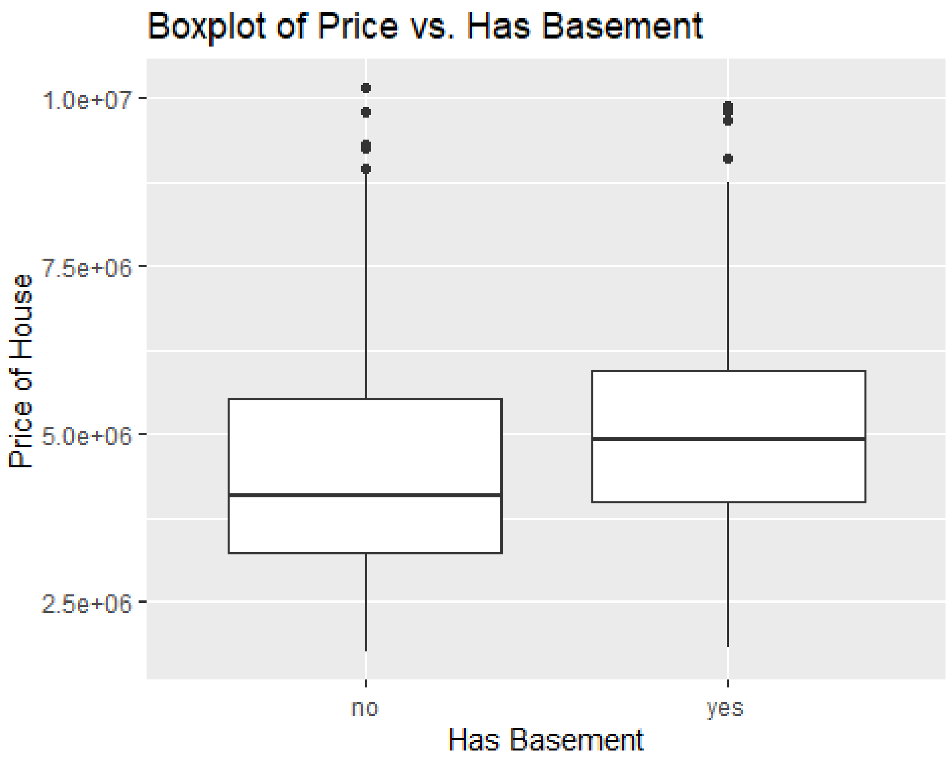
- The price of house is high if a house is on main road. If a house is not on main road, price of the house is less.

```
ggplot(New_df, aes(x = factor(Has_guestroom), y = price_of_house)) +
  geom_boxplot() +
  labs(x = "Has Guestroom", y = "Price of House") +
  ggtitle("Boxplot of Price vs. Has Guestroom")
```



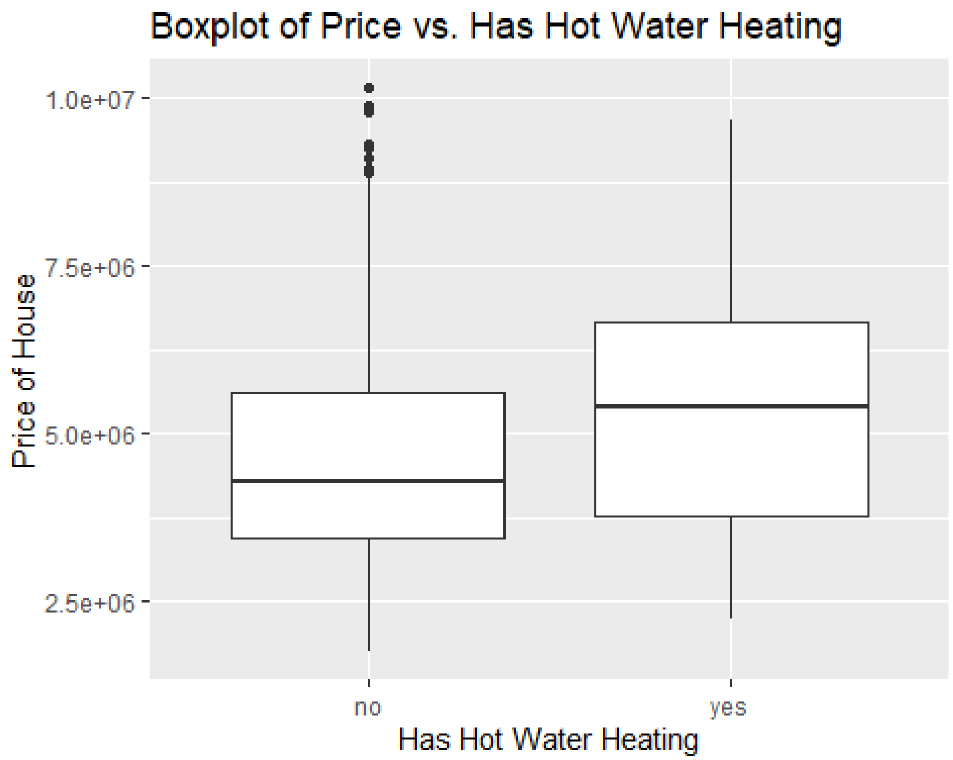
- If a house has guestroom, then price is high as compared to house without guestroom.

```
ggplot(New_df, aes(x = factor(Has_basement), y = price_of_house)) +
  geom_boxplot() +
  labs(x = "Has Basement", y = "Price of House") +
  ggtitle("Boxplot of Price vs. Has Basement")
```



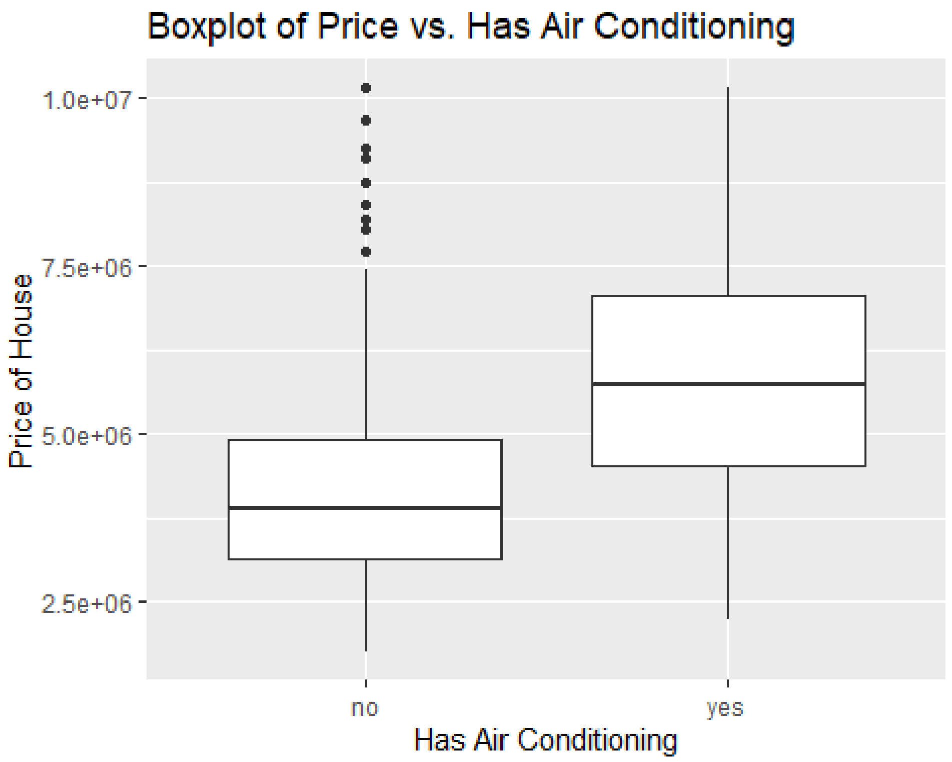
- Price of house is greater if a house has basement.

```
ggplot(New_df, aes(x = factor(Has_hotwaterheating), y = price_of_house)) +
  geom_boxplot() +
  labs(x = "Has Hot Water Heating", y = "Price of House") +
  ggtitle("Boxplot of Price vs. Has Hot Water Heating")
```



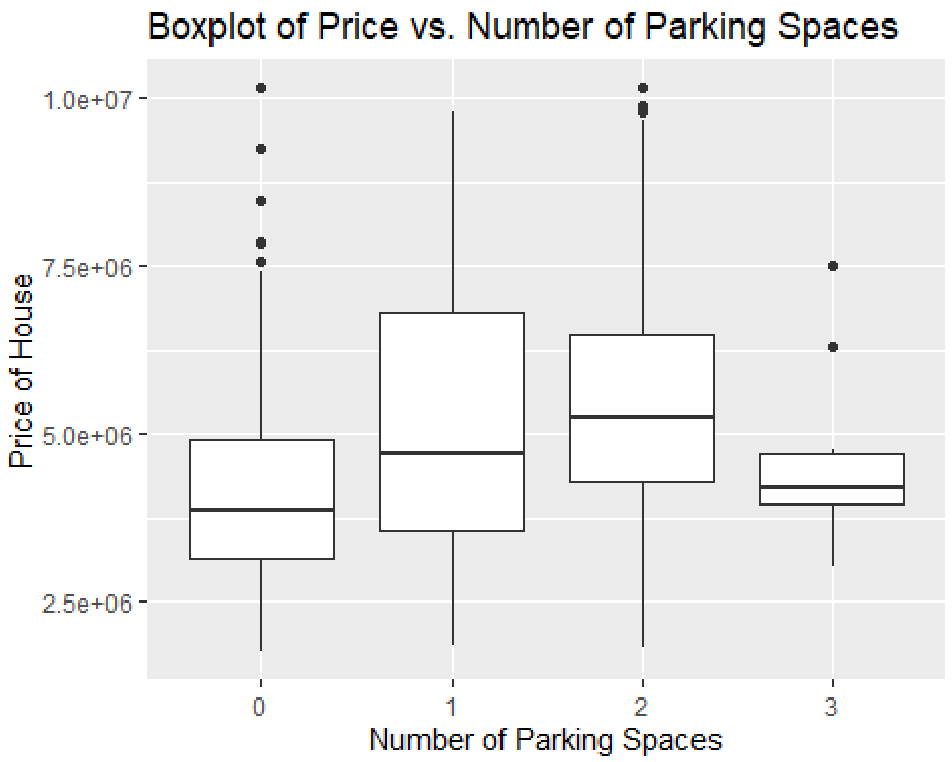
- If a house has hot water heating system, then the price is heigher as compared to house without water heating.

```
ggplot(New_df, aes(x = factor(Has_airconditioning), y = price_of_house)) +
  geom_boxplot() +
  labs(x = "Has Air Conditioning", y = "Price of House") +
  ggtitle("Boxplot of Price vs. Has Air Conditioning")
```



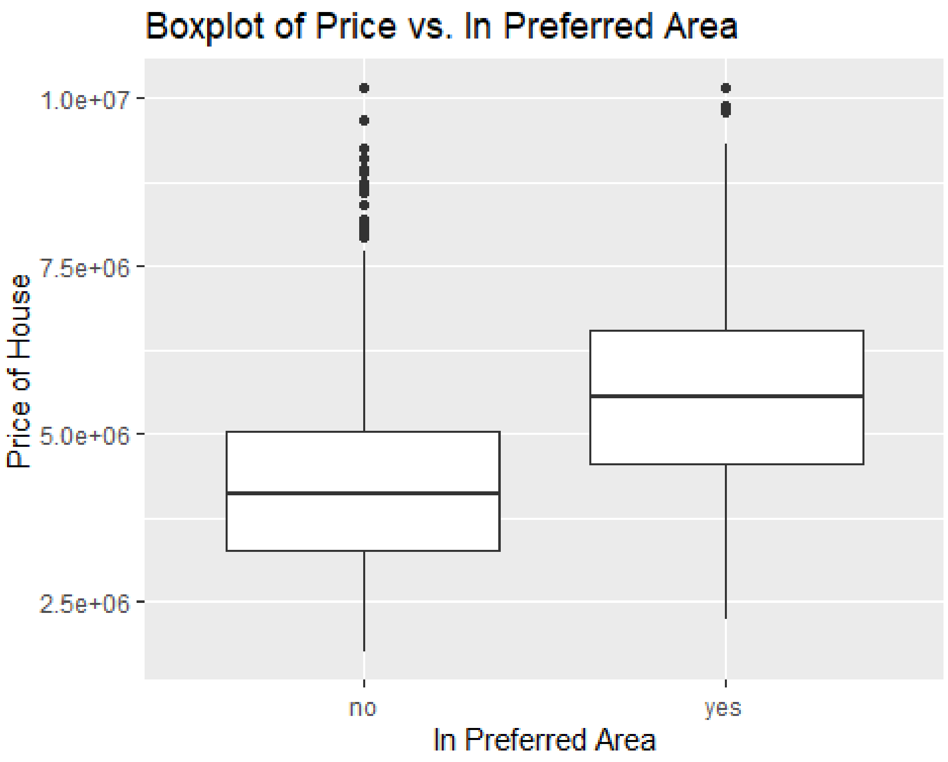
- Price is high for houses with Air Conditioning. The price is lower for houses without air conditioning.

```
ggplot(New_df, aes(x = factor(Number_of_parking_spaces), y = price_of_house)) +
  geom_boxplot() +
  labs(x = "Number of Parking Spaces", y = "Price of House") +
  ggtitle("Boxplot of Price vs. Number of Parking Spaces")
```



- Prices increases as we increase the number of parking spaces in a house.

```
ggplot(New_df, aes(x = factor(in_preferred_area), y = price_of_house)) +
  geom_boxplot() +
  labs(x = "In Preferred Area", y = "Price of House") +
  ggtitle("Boxplot of Price vs. In Preferred Area")
```



- If a house is in preferred area, then the house price is high.

———————— Now we will test our backward selection model

Making predictions on test data

```
predictions <- predict(backward_model, newdata = df_test)
dim(df_test)

## [1] 106  24
```

Calculate Mean Squared Error (MSE)

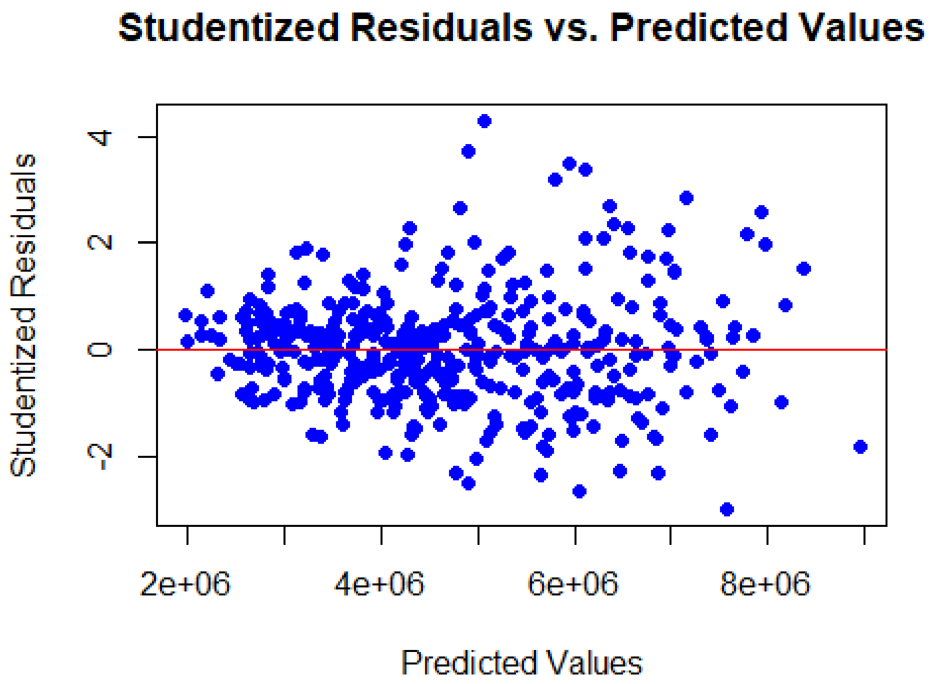
```
mse_initial <- mean((df_test$price_of_house - predictions)^2)
cat("Mean Squared Error (MSE):", mse_initial, "\n")

## Mean Squared Error (MSE): 890354373905
```

Calculate Mean Absolute Error (MAE)

```
mae_initial <- mean(abs(df_test$price_of_house - predictions))
cat("Mean Absolute Error (MAE):", mae_initial, "\n")

## Mean Absolute Error (MAE): 729161.6
```

Residual Analysis

```
residuals <- rstudent(backward_model)
predicted_values <- predict(backward_model)

plot(predicted_values, residuals, main="Studentized Residuals vs. Predicted Values",
    xlab="Predicted Values", ylab="Studentized Residuals", col="blue", pch=16)
abline(h=0, col="red")
```

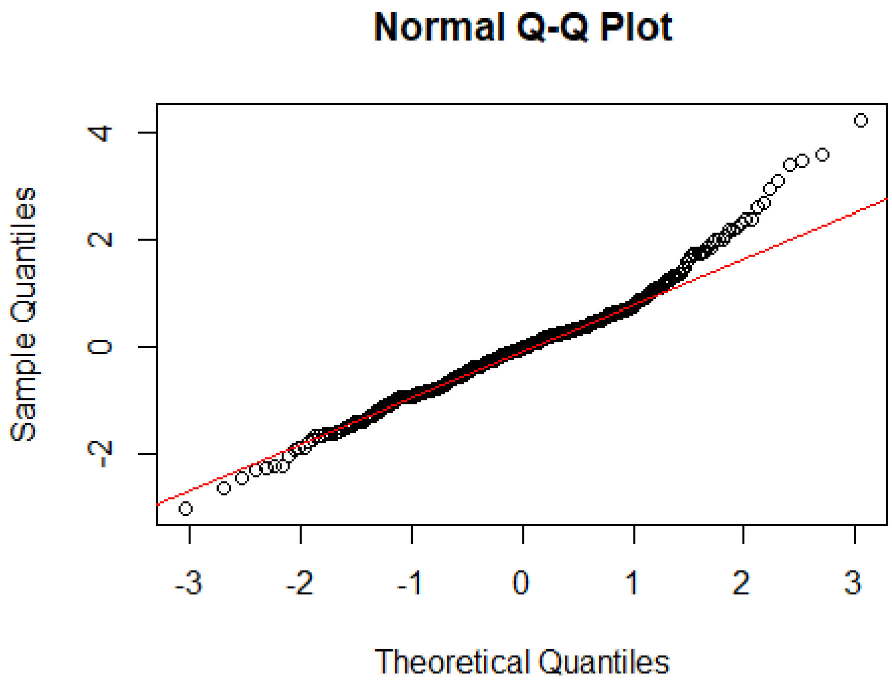


- With the help of residuals plot, we can see that residuals have high value for high predicted values. Hence this is a moderate model. We need to imporve this.

- As per my understanding, this is due to the less number of observations and predictors. We need more data to imporve this model further.

Create a normal probability plot

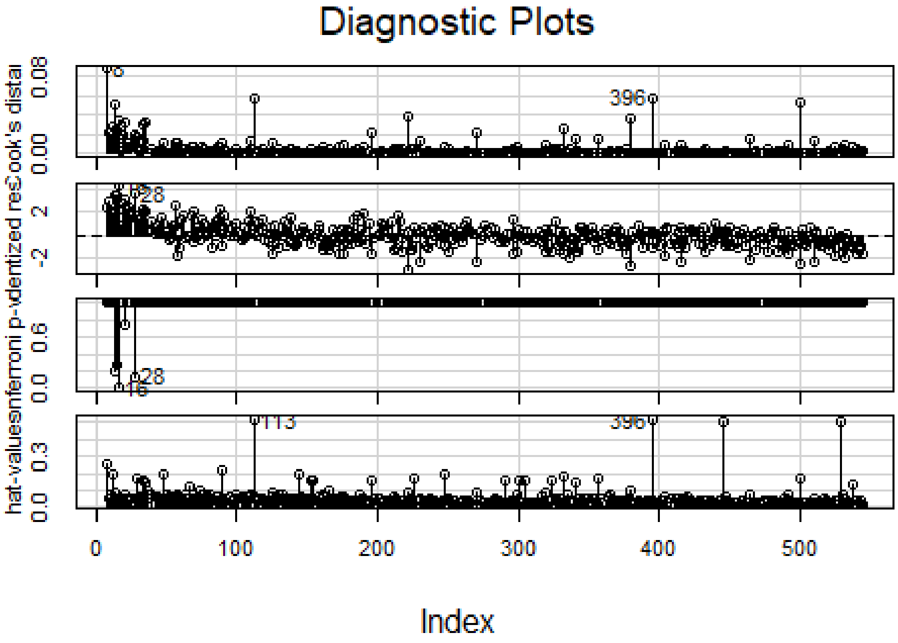
```
qqnorm(rstandard(Initial_model), main="Normal Q-Q Plot")
qqline(rstandard(Initial_model), col="red")
```



```
cooksd <- cooks.distance(Initial_model)
```

Find indices of influential points with Cook’s distance > 1

```
influential_indices <- which(cooksd > 1)

library(car)
influenceIndexPlot(Initial_model)
```

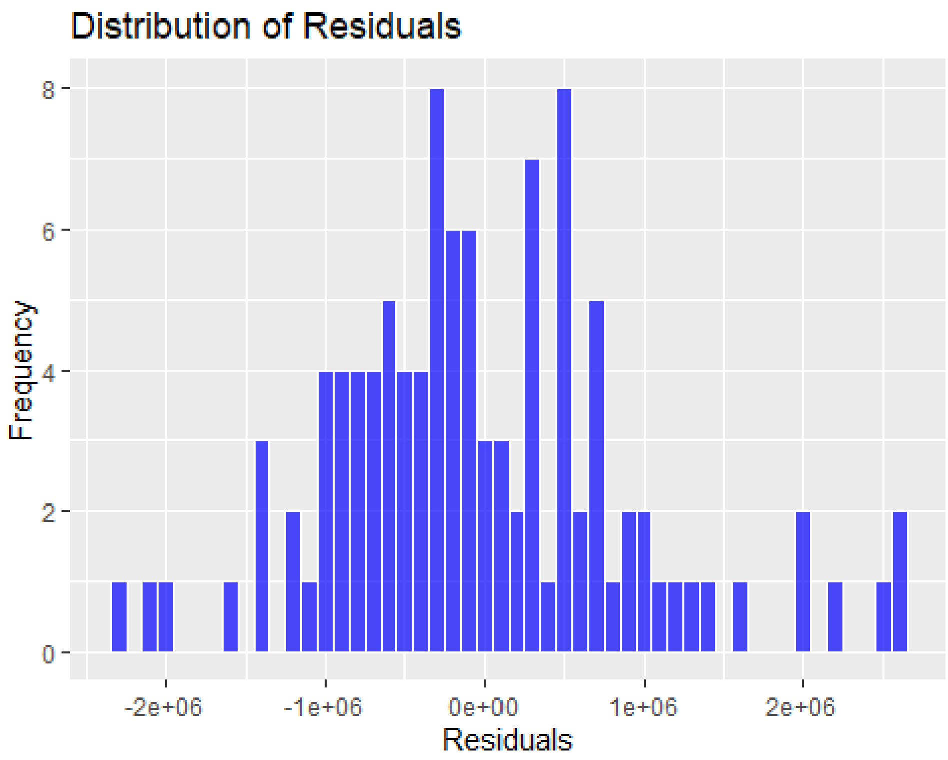


- We have less number of influential points.

```
residuals_df <- data.frame(
  Actual = df_test$price_of_house,
  Predicted = predictions,
  Residuals = df_test$price_of_house - predictions
)
```

Plot histogram or density plot of residuals

```
ggplot(residuals_df, aes(x = Residuals)) +
  geom_histogram(binwidth = 100000, fill = "blue", color = "white", alpha = 0.7) +
  labs(title = "Distribution of Residuals", x = "Residuals", y = "Frequency")
```



```
skewness(residuals_df$Residuals)

## [1] 0.5192377
```

- The residual plot looks like normally distributed. Also skewness is 0.5192377 which is under acceptable range.

Question 10:

Libraries

```
library(Hmisc)

## Warning: package 'Hmisc' was built under R version 4.3.2

##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:dplyr':
##
##     src, summarize

## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
library(psych)
```

```
## Registered S3 method overwritten by 'psych':
##   method         from  
##   plot.residuals rmutil
```

```
## 
## Attaching package: 'psych'
```

```
## The following object is masked from 'package:Hmisc':
## 
##     describe
```

```
## The following object is masked from 'package:car':
## 
##     logit
```

```
## The following objects are masked from 'package:ggplot2':
## 
##     %+%, alpha
```

```
library(GGally)
```

```
## Warning: package 'GGally' was built under R version 4.3.2
```

```
## Registered S3 method overwritten by 'GGally':
##   method from   
##   +.gg   ggplot2
```

```
library(ggplot2)
library(vioplot)
```

```
## Warning: package 'vioplot' was built under R version 4.3.2
```

```
## Loading required package: sm
```

```
## Warning: package 'sm' was built under R version 4.3.2
```

```
## Package 'sm', version 2.2-5.7: type help(sm) for summary information
```

```
## Loading required package: zoo
```

```
## Warning: package 'zoo' was built under R version 4.3.2
```

```
## 
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
## 
##     as.Date, as.Date.numeric
```

```
library(corrplot)
library(REdaS)
```

```
## Warning: package 'REdaS' was built under R version 4.3.2
## Loading required package: grid

library(psych)
library(factoextra)

## Warning: package 'factoextra' was built under R version 4.3.2

## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa

library("FactoMineR")

## Warning: package 'FactoMineR' was built under R version 4.3.2

library(ade4)

## Warning: package 'ade4' was built under R version 4.3.2

##
## Attaching package: 'ade4'

## The following object is masked from 'package:FactoMineR':
##
##     reconst
```

Importing data in R

```
data <- read.csv("D:/Assignments_Depaul/DSC_424_Advance_Data_Analysis/Midterm
Exam/16PF.csv", header = TRUE)
dim(data)

## [1] 49159   163
```

Check NA For All Variables

```
sum(is.na(data))

## [1] 0

library(dplyr)
```

Convert 0 to NA as told in the problem statement

```
data <- data %>%
  mutate_all(~ifelse(. == 0, NA, .))
```

Check NA For All Variables

```
sum(is.na(data))

## [1] 98919
```

```
na_percentages <- colMeans(is.na(data)) * 100
na_percentages
```

```
##        A1        A2        A3        A4        A5        A6        A7        A8 
## 1.4280193 1.4849773 1.2551110 1.4748062 0.9316707 1.0476210 0.6590858 1.4524299 
##        A9       A10        B1        B2        B3        B4        B5        B6 
## 1.5337985 1.1920503 1.4361561 1.1879819 1.0883053 0.9743892 1.5460038 1.5805854 
##        B7        B8        B9       B10       B11       B12       B13        C1 
## 1.4707378 1.0557578 1.4788747 1.3425822 0.6712911 1.2530768 1.0313473 0.9703208 
##        C2        C3        C4        C5        C6        C7        C8        C9 
## 1.1432291 1.6192355 1.1086475 0.6916333 1.4442930 1.4646352 0.6692569 1.0354157 
##       C10        D1        D2        D3        D4        D5        D6        D7 
## 1.2774873 0.8503021 1.0191420 1.2123924 1.3954718 0.7709677 1.5988934 0.9194654 
##        D8        D9       D10        E1        E2        E3        E4        E5 
## 1.1167843 0.7994467 1.2978295 1.0232104 1.4626009 1.4036087 0.9947314 1.4503957 
##        E6        E7        E8        E9       E10        F1        F2        F3 
## 1.1656055 1.3975061 1.1534002 1.3100348 1.2530768 1.4137798 1.2795216 0.8808153 
##        F4        F5        F6        F7        F8        F9       F10        G1 
## 1.0130393 1.1696739 1.4117456 1.3283427 1.3059664 1.3242743 1.4605667 1.1635713 
##        G2        G3        G4        G5        G6        G7        G8        G9 
## 1.4402246 1.3405480 1.4910800 1.3364796 0.8665758 0.7974125 1.4666694 0.9764234 
##       G10        H1        H2        H3        H4        H5        H6        H7 
## 1.1066132 1.3954718 1.3791981 1.2164609 1.3202059 1.3669928 1.2225635 1.3425822 
##        H8        H9       H10        I1        I2        I3        I4        I5 
## 1.3486849 1.4727720 1.4280193 1.1635713 1.3588560 1.4666694 0.9967656 1.4788747 
##        I6        I7        I8        I9       I10        J1        J2        J3 
## 1.4137798 1.2042556 0.6916333 1.5358327 1.2205293 1.1981529 0.8991233 1.3812323 
##        J4        J5        J6        J7        J8        J9       J10        K1 
## 1.0557578 1.0598263 1.3975061 1.0333815 1.3690270 1.3669928 1.3710613 1.5032853 
##        K2        K3        K4        K5        K6        K7        K8        K9 
## 1.0455868 0.9886287 1.3995403 1.5297301 1.5154906 1.5521064 1.4992168 1.1635713 
##       K10        L1        L2        L3        L4        L5        L6        L7 
## 0.8909864 1.0781342 1.4564983 1.2266319 1.1005106 1.5093879 0.8340284 1.0557578 
##        L8        L9       L10        M1        M2        M3        M4        M5 
## 1.3425822 1.1350923 1.0028682 0.8238573 1.6558514 1.3425822 1.4076771 1.1554344 
##        M6        M7        M8        M9       M10        N1        N2        N3 
## 1.4320877 1.5500722 1.4971826 1.4463272 1.4320877 1.3547875 1.1778108 1.3425822 
##        N4        N5        N6        N7        N8        N9       N10        O1 
## 1.4117456 0.9540471 1.0496552 0.8055493 1.0699974 1.3853008 0.7262149 1.3954718 
##        O2        O3        O4        O5        O6        O7        O8        O9 
## 1.3975061 1.4381904 1.1839134 1.1595028 1.5460038 1.4809089 1.3568217 1.2530768 
##       O10        P1        P2        P3        P4        P5        P6        P7 
## 1.5134563 1.2245977 0.9235338 1.0842369 1.4259851 1.3629244 1.2652820 1.1147501 
##        P8        P9       P10 
## 0.9967656 0.7730019 1.1513660
```

Calculate the percentage of rows with NA

```
percentage_na_rows <- mean(apply(data, 1, function(row) any(is.na(row)))) * 100
print(percentage_na_rows)
```

```
## [1] 28.02742
```

Creating a function to impute NA values

```r
imputeNA <- function(data) {
  for (col in names(data)) {
    if (is.numeric(data[[col]])) {
      # Calculate rounded mean
      mean_val <- round(mean(data[[col]], na.rm = TRUE))
      # Impute NA with rounded mean for numeric variables
      data[[col]][is.na(data[[col]])] <- mean_val
    } else if (is.factor(data[[col]]) || is.character(data[[col]])) {
      # Calculate mode
      mode_val <- as.character(sort(table(data[[col]]), decreasing = TRUE)[1])
      # Impute NA with mode for categorical or factor variables
      data[[col]][is.na(data[[col]])] <- mode_val
    }
    # If neither numeric nor categorical, do nothing
  }
  return(data)
}

data<-imputeNA(data)

unique_counts <- sapply(data, function(x) length(unique(x)))
unique_counts
```

```
##  A1  A2  A3  A4  A5  A6  A7  A8  A9 A10  B1  B2  B3  B4  B5  B6  B7  B8  B9 B10 
##   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5 
## B11 B12 B13  C1  C2  C3  C4  C5  C6  C7  C8  C9 C10  D1  D2  D3  D4  D5  D6  D7 
##   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5 
##  D8  D9 D10  E1  E2  E3  E4  E5  E6  E7  E8  E9 E10  F1  F2  F3  F4  F5  F6  F7 
##   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5 
##  F8  F9 F10  G1  G2  G3  G4  G5  G6  G7  G8  G9 G10  H1  H2  H3  H4  H5  H6  H7 
##   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5 
##  H8  H9 H10  I1  I2  I3  I4  I5  I6  I7  I8  I9 I10  J1  J2  J3  J4  J5  J6  J7 
##   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5 
##  J8  J9 J10  K1  K2  K3  K4  K5  K6  K7  K8  K9 K10  L1  L2  L3  L4  L5  L6  L7 
##   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5 
##  L8  L9 L10  M1  M2  M3  M4  M5  M6  M7  M8  M9 M10  N1  N2  N3  N4  N5  N6  N7 
##   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5 
##  N8  N9 N10  O1  O2  O3  O4  O5  O6  O7  O8  O9 O10  P1  P2  P3  P4  P5  P6  P7 
##   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5 
##  P8  P9 P10 
##   5   5   5
```

Calculate the percentage of rows with NA after imputaion

```r
percentage_na_rows_1 <- mean(apply(df, 1, function(row) any(is.na(row)))) * 100
print(percentage_na_rows_1)
```

```
## [1] 0
```

Checking the corrplot matrix

```
cor_matrix <- cor(data)

library(caret)
highly_correlated_vars <- findCorrelation(cor_matrix, cutoff = 0.75)
colnames(data[highly_correlated_vars])

## [1] "H3"  "J10"
```

Removing highly correlated columns from data

```
data <- data[, !colnames(data) %in% c("H3", "J10")]
dim(data)

## [1] 49159   161
```

Test KMO Sampling Adequancy

```
library(psych)
KMO(data)

## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = data)
## Overall MSA =  0.97
## MSA for each item = 
##   A1   A2   A3   A4   A5   A6   A7   A8   A9  A10   B1   B2   B3   B4   B5   B6 
## 0.97 0.99 0.96 0.96 0.98 0.98 0.98 0.96 0.97 0.97 0.97 0.95 0.95 0.97 0.96 0.97 
##   B7   B8   B9  B10  B11  B12  B13   C1   C2   C3   C4   C5   C6   C7   C8   C9 
## 0.94 0.96 0.93 0.98 0.95 0.95 0.95 0.96 0.97 0.98 0.98 0.97 0.98 0.98 0.97 0.98 
##  C10   D1   D2   D3   D4   D5   D6   D7   D8   D9  D10   E1   E2   E3   E4   E5 
## 0.98 0.95 0.95 0.97 0.97 0.95 0.98 0.97 0.98 0.96 0.98 0.98 0.97 0.92 0.97 0.96 
##   E6   E7   E8   E9  E10   F1   F2   F3   F4   F5   F6   F7   F8   F9  F10   G1 
## 0.96 0.89 0.97 0.96 0.93 0.95 0.95 0.91 0.96 0.97 0.93 0.96 0.94 0.94 0.96 0.99 
##   G2   G3   G4   G5   G6   G7   G8   G9  G10   H1   H2   H4   H5   H6   H7   H8 
## 0.98 0.98 0.98 0.98 0.99 0.99 0.98 0.98 0.99 0.92 0.93 0.91 0.94 0.94 0.94 0.94 
##   H9  H10   I1   I2   I3   I4   I5   I6   I7   I8   I9  I10   J1   J2   J3   J4 
## 0.94 0.95 0.96 0.97 0.98 0.97 0.97 0.97 0.96 0.96 0.97 0.96 0.97 0.96 0.95 0.92 
##   J5   J6   J7   J8   J9   K1   K2   K3   K4   K5   K6   K7   K8   K9  K10   L1 
## 0.97 0.98 0.95 0.95 0.93 0.97 0.99 0.98 0.98 0.98 0.96 0.96 0.97 0.97 0.98 0.98 
##   L2   L3   L4   L5   L6   L7   L8   L9  L10   M1   M2   M3   M4   M5   M6   M7 
## 0.99 0.98 0.98 0.97 0.97 0.98 0.96 0.98 0.98 0.93 0.97 0.96 0.97 0.96 0.95 0.97 
##   M8   M9  M10   N1   N2   N3   N4   N5   N6   N7   N8   N9  N10   O1   O2   O3 
## 0.95 0.96 0.97 0.98 0.98 0.95 0.97 0.96 0.95 0.97 0.98 0.98 0.97 0.93 0.95 0.94 
##   O4   O5   O6   O7   O8   O9  O10   P1   P2   P3   P4   P5   P6   P7   P8   P9 
## 0.89 0.95 0.90 0.91 0.92 0.93 0.96 0.96 0.96 0.93 0.98 0.99 0.98 0.89 0.96 0.94 
##  P10 
## 0.98
```

- The Kaiser-Meyer-Olkin (KMO) measure evaluates the adequacy of data for factor analysis, with an overall MSA of 0.97 indicating high correlation among variables. Each item's MSA, ideally close to 1, reflects its correlation strength with other variables, suggesting suitability for factor analysis.

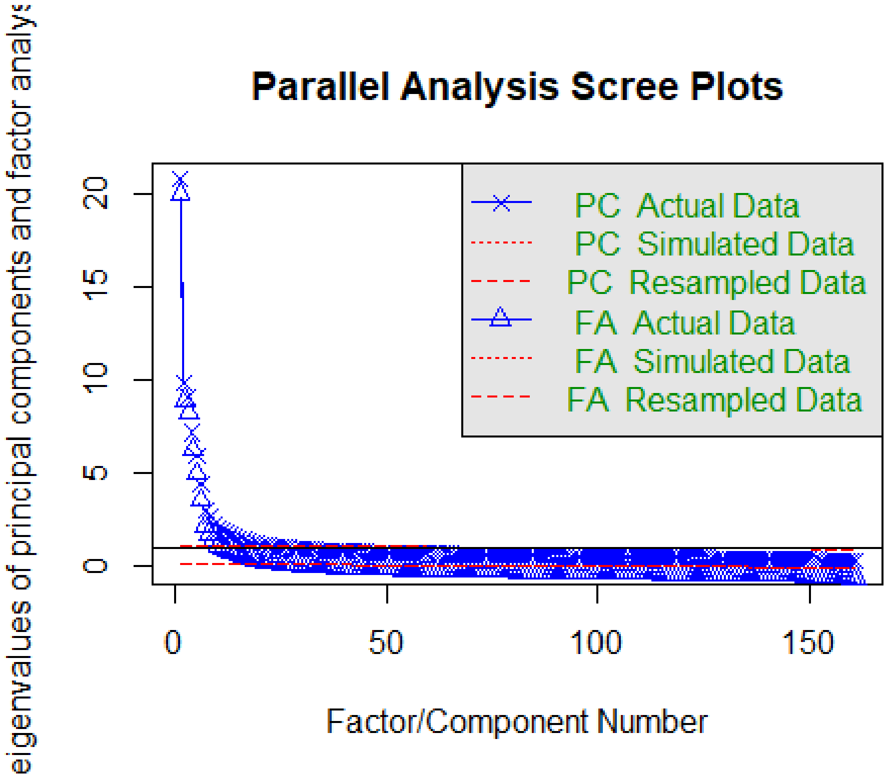
Test Bartlett's test of Sphericity

```
library(REdaS)
bart_spher(data)

##  Bartlett's Test of Sphericity
##
## Call: bart_spher(x = data)
##
##      X2 = 3376359.467
##      df = 12880
## p-value < 2.22e-16
```

- Bartlett's Test of Sphericity checks if variables in your data are related or if they act independently. With a p-value less than 0.05 (2.22e-16), it means there are significant relationships between the variables, suggesting they are not completely independent.

Parallel Analysis (Horn's parallel analysis)

```
comp <- fa.parallel(data)
```



```
## Parallel analysis suggests that the number of factors =  26  and the number of components =  21

comp

## Call: fa.parallel(x = data)
## Parallel analysis suggests that the number of factors =  26  and the number of components =  21 
## 
##  Eigen Values of 
##    Original factors Resampled data Simulated data Original components
## 1             19.98           0.12           0.12               20.75
## 2              8.92           0.11           0.11                9.86
## 3              8.18           0.11           0.11                9.03
## 4              6.26           0.10           0.10                7.16
## 5              4.95           0.10           0.10                5.87
## 6              3.53           0.10           0.10                4.41
## 7              2.11           0.10           0.10                2.95
## 8              1.72           0.10           0.10                2.63
## 9              1.37           0.09           0.09                2.25
## 10             1.10           0.09           0.09                1.94
## 11             0.99           0.09           0.09                1.87
## 12             0.83           0.09           0.09                1.74
## 13             0.77           0.09           0.09                1.64
## 14             0.72           0.08           0.08                1.58
## 15             0.60           0.08           0.08                1.49
## 16             0.50           0.08           0.08                1.39
## 17             0.40           0.08           0.08                1.30
## 18             0.37           0.08           0.08                1.27
## 19             0.34           0.08           0.08                1.22
## 20             0.25           0.07           0.07                1.15
## 21             0.19           0.07           0.07                1.09
## 22             0.17           0.07           0.07                1.07
## 23             0.15           0.07           0.07                1.06
## 24             0.13           0.07           0.07                1.03
## 25             0.10           0.07           0.07                1.01
## 26             0.08           0.07           0.07                0.98
##    Resampled components Simulated components
## 1                  1.11                 1.11
## 2                  1.11                 1.11
## 3                  1.11                 1.11
## 4                  1.10                 1.10
## 5                  1.10                 1.10
## 6                  1.10                 1.10
## 7                  1.10                 1.10
## 8                  1.09                 1.09
## 9                  1.09                 1.09
## 10                 1.09                 1.09
## 11                 1.09                 1.09
## 12                 1.09                 1.09
## 13                 1.09                 1.09
## 14                 1.08                 1.08
## 15                 1.08                 1.08
## 16          1.08          1.08
## 17          1.08          1.08
## 18          1.08          1.08
## 19          1.08          1.08
## 20          1.07          1.07
## 21          1.07          1.07
## 22          1.07          1.07
## 23          1.07          1.07
## 24          1.07          1.07
## 25          1.07          1.07
## 26          1.07          1.07
```

Parallel analysis suggests that the number of factors = 26 and the number of components = 21

———————- PCA_Plot functions

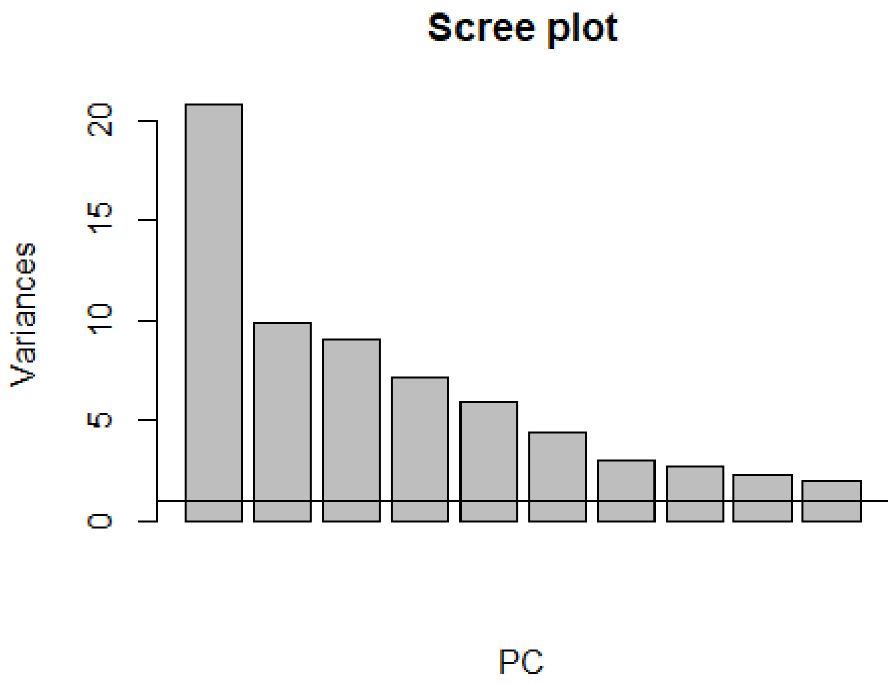
```
PCA_Plot = function(pcaData)
{
  library(ggplot2)

  theta = seq(0,2*pi,length.out = 100)
  circle = data.frame(x = cos(theta), y = sin(theta))
  p = ggplot(circle,aes(x,y)) + geom_path()

  loadings = data.frame(pcaData$rotation, .names = row.names(pcaData$rotation))
  p + geom_text(data=loadings, mapping=aes(x = PC1, y = PC2, label = .names, colour = .names,
fontface="bold")) +
    coord_fixed(ratio=1) + labs(x = "PC1", y = "PC2")
}

PCA_Plot_Secondary = function(pcaData)
{
  library(ggplot2)

  theta = seq(0,2*pi,length.out = 100)
  circle = data.frame(x = cos(theta), y = sin(theta))
  p = ggplot(circle,aes(x,y)) + geom_path()

  loadings = data.frame(pcaData$rotation, .names = row.names(pcaData$rotation))
  p + geom_text(data=loadings, mapping=aes(x = PC3, y = PC4, label = .names, colour = .names,
fontface="bold")) +
    coord_fixed(ratio=1) + labs(x = "PC3", y = "PC4")
}

PCA_Plot_Psyc = function(pcaData)
{
  library(ggplot2)
  theta = seq(0,2*pi,length.out = 100)
  circle = data.frame(x = cos(theta), y = sin(theta))
  p = ggplot(circle,aes(x,y)) + geom_path()

  loadings = as.data.frame(unclass(pcaData$loadings))
  s = rep(0, ncol(loadings))
  for (i in 1:ncol(loadings))
  {
    s[i] = 0
    for (j in 1:nrow(loadings))
      s[i] = s[i] + loadings[j, i]^2
    s[i] = sqrt(s[i])
  }

  for (i in 1:ncol(loadings))
    loadings[, i] = loadings[, i] / s[i]

  loadings$.names = row.names(loadings)

  p + geom_text(data=loadings, mapping=aes(x = PC1, y = PC2, label = .names, colour = .names,
fontface="bold")) +
    coord_fixed(ratio=1) + labs(x = "PC1", y = "PC2")
}

PCA_Plot_Psyc_Secondary = function(pcaData)
{
  library(ggplot2)

  theta = seq(0,2*pi,length.out = 100)
  circle = data.frame(x = cos(theta), y = sin(theta))
  p = ggplot(circle,aes(x,y)) + geom_path()

  loadings = as.data.frame(unclass(pcaData$loadings))
  s = rep(0, ncol(loadings))
  for (i in 1:ncol(loadings))
  {
    s[i] = 0
    for (j in 1:nrow(loadings))
      s[i] = s[i] + loadings[j, i]^2
    s[i] = sqrt(s[i])
  }

  for (i in 1:ncol(loadings))
    loadings[, i] = loadings[, i] / s[i]

  loadings$.names = row.names(loadings)

  print(loadings)
  p + geom_text(data=loadings, mapping=aes(x = PC3, y = PC4, label = .names, colour = .names,
fontface="bold")) +
    coord_fixed(ratio=1) + labs(x = "PC3", y = "PC4")
}
```

Create PCA

```
PCA = prcomp(data, center = T, scale = T)
```

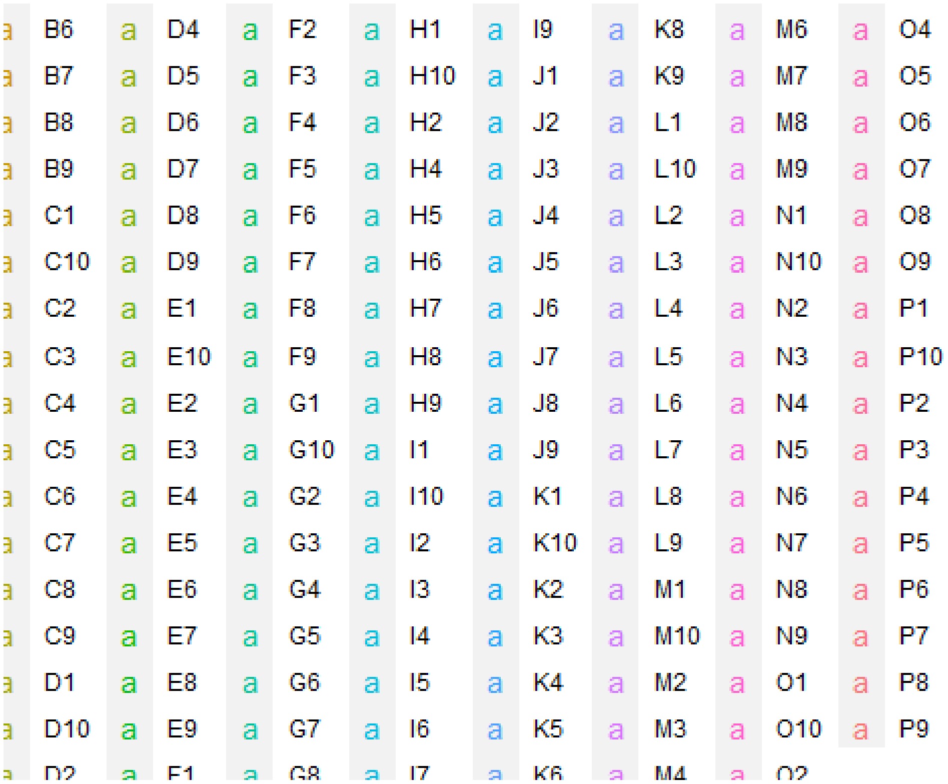
Checking the scree plot

```
plot(PCA, main="Scree plot", xlab="PC")
abline(1,0)
```

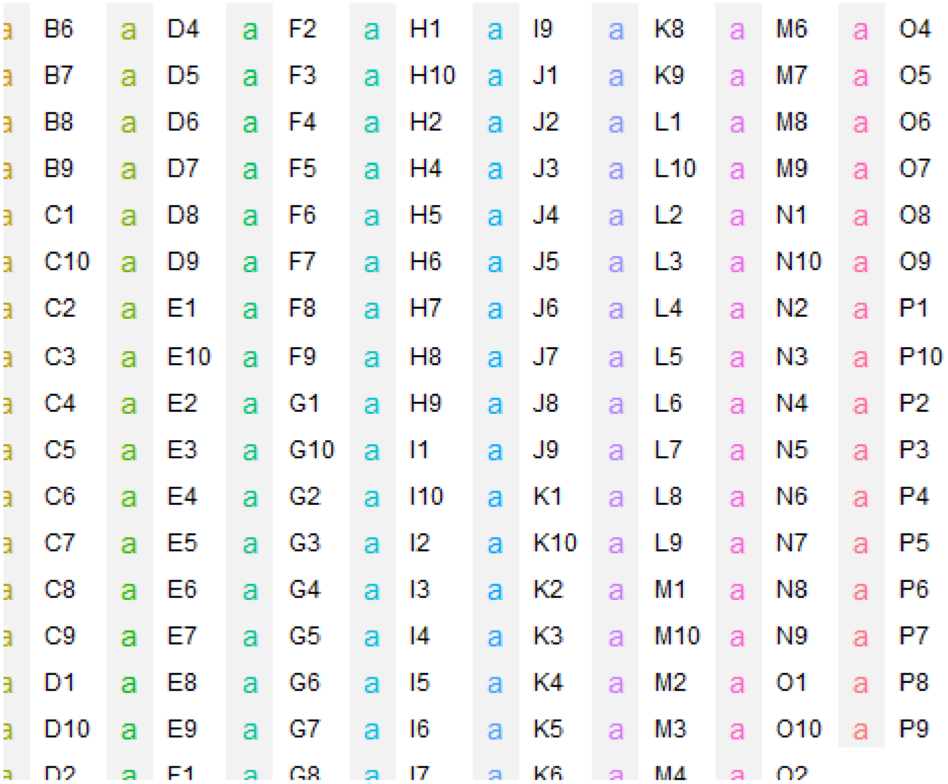


Check PCA visualizations

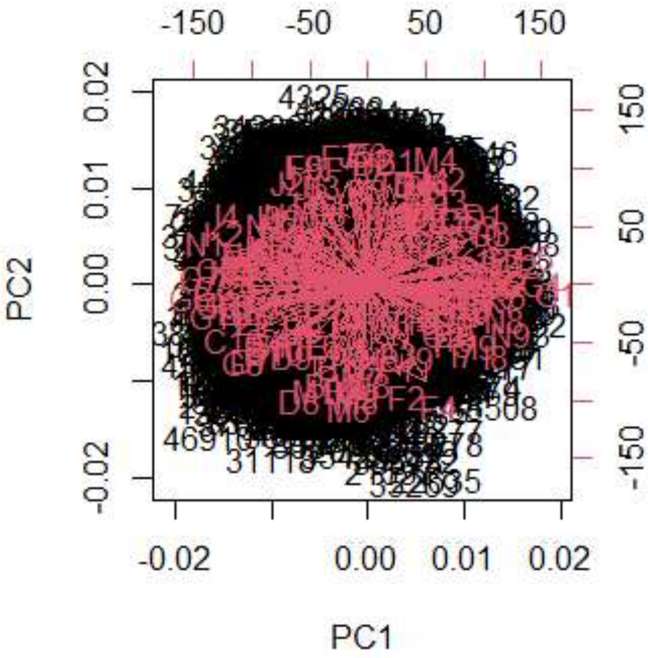
```
PCA_Plot(PCA) #PCA_plot1
```



```
PCA_Plot_Secondary(PCA) #PCA_Plot2
```



```
biplot(PCA) #Biplot
```



Extract the cumulative proportion of variance explained

```
cumulative_variance <- cumsum(PCA$sdev^2) / sum(PCA$sdev^2)
```

Find the number of components needed to account for 80% of the variance

```
num_components <- which(cumulative_variance >= 0.8)[1]
num_components

## [1] 85
```

- 85 components are needed to account for 80% of the variance in the data. The number of components is determined by identifying the smallest number of principal components where the cumulative proportion of variance explained by those components reaches or exceeds 80%.
- This is calculated by summing up the variances explained by each component until the cumulative proportion exceeds the specified threshold (in this case, 80%). The which function in R is then used to find the index of the first component that meets this criterion.

Question 2

Eigenvalue method

```
eigenvalues <- PCA$sdev^2
num_components_eigenvalue <- sum(eigenvalues > 1)
num_components_eigenvalue

## [1] 25
```

- With the help of eigen values, we will take 25 components which have eigen values > 1.

Knee of the scree plot method

```
scree_values <- PCA$sdev^2
variance_explained <- scree_values / sum(scree_values)
num_components_scree_0.05 <- which.max(diff(variance_explained) < 0.05) + 1
num_components_scree_0.01 <- which.max(diff(variance_explained) < 0.05) + 1
num_components_scree_0.05

## [1] 2

num_components_scree_0.01

## [1] 2
```

- If we are using the knee of the scree plot, we can choose 2 components only. But those components explains around 19% of the varience only.

#——————————————————-

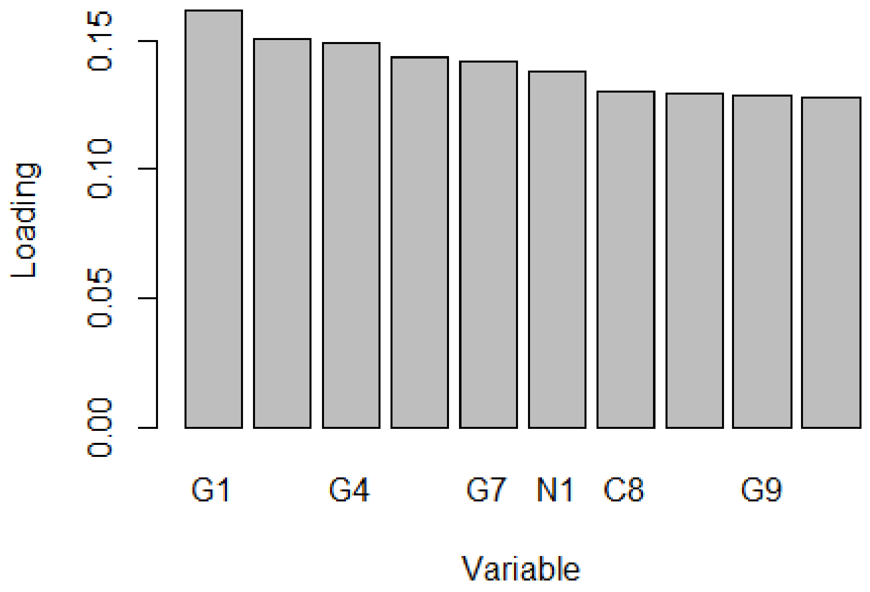
# Question 3:

Get the loadings of the top 10 variables for the first component

```
top_loadings <- abs(PCA$rotation[, 1])  # Absolute values of loadings for first component
top_loadings <- sort(top_loadings, decreasing = TRUE)[1:10]  # Top 10 loadings
top_variables <- names(top_loadings)
top_variables

##  [1] "G1" "G6" "G4" "G5" "G7" "N1" "C8" "G2" "G9" "C7"
```

Plot the top variables

```
barplot(top_loadings, names.arg = top_variables, xlab = "Variable", ylab = "Loading")
```



# Question 3:

i)

- # We will choose the eigen value method to choose the number of components.
- # With the eigenvalue method, 25 components are chosen which have eigenvalues greater than 1.

- # We choose this method because variation explained method is giving us 85 components which are explaining 80% of the variation in the data and knee of scree plot is giving 2 variables with 19% of the variation which is too less.

Extract the loadings of each principal component

```r
loadings <- PCA$rotation
```

Define a function to interpret each component

```r
interpret_component <- function(component_number, top_n = 5) {        # Taking top 5 variables

  component_loadings <- loadings[, component_number]

  sorted_loadings <- sort(abs(component_loadings), decreasing = TRUE)

  top_variable_names <- names(sorted_loadings)[1:top_n]

  interpretation <- paste("Component", component_number, "is primarily influenced by the following variables:")
  for (variable_name in top_variable_names) {
    interpretation <- paste(interpretation, variable_name, sep = " ")
  }

  return(interpretation)
}
```

Interpret the first 25 components

```r
for (i in 1:25) {
  cat(interpret_component(i), "\n\n")
}
```

```
## Component 1 is primarily influenced by the following variables: G1 G6 G4 G5 G7
##
## Component 2 is primarily influenced by the following variables: B6 J5 F7 M4 B1
##
## Component 3 is primarily influenced by the following variables: E6 L3 C6 E4 B10
##
## Component 4 is primarily influenced by the following variables: A3 M1 H1 F2 H6
##
## Component 5 is primarily influenced by the following variables: O3 O4 D5 O1 D1
##
## Component 6 is primarily influenced by the following variables: C4 K5 H10 K1 G8
##
## Component 7 is primarily influenced by the following variables: K7 K9 K6 K4 K1
##
## Component 8 is primarily influenced by the following variables: O10 P7 O8 H8 O2
##
```

```
## Component 9 is primarily influenced by the following variables: N3 N7 N6 N10 P8
##
## Component 10 is primarily influenced by the following variables: A1 I2 I10 I4 I7
##
## Component 11 is primarily influenced by the following variables: I6 I9 J3 A7 I10
##
## Component 12 is primarily influenced by the following variables: O6 O7 O8 O9 O3
##
## Component 13 is primarily influenced by the following variables: B11 B9 F8 H4 E10
##
## Component 14 is primarily influenced by the following variables: F3 H1 G2 N3 O8
##
## Component 15 is primarily influenced by the following variables: J1 H6 P7 H7 P3
##
## Component 16 is primarily influenced by the following variables: N5 N3 B5 B12 B2
##
## Component 17 is primarily influenced by the following variables: B4 O4 O5 D1 D2
##
## Component 18 is primarily influenced by the following variables: O4 D5 F3 P2 O5
##
## Component 19 is primarily influenced by the following variables: E10 A8 H4 M3 E8
##
## Component 20 is primarily influenced by the following variables: E9 M5 A4 C1 H4
##
## Component 21 is primarily influenced by the following variables: F3 H4 E3 P9 E5
##
## Component 22 is primarily influenced by the following variables: J9 J4 M6 J3 M8
##
## Component 23 is primarily influenced by the following variables: D9 H2 D2 A10 H4
##
## Component 24 is primarily influenced by the following variables: L8 P9 L5 I1 H2
##
## Component 25 is primarily influenced by the following variables: F3 B7 M5 E7 J9
```

Question 3:
ii)

- PC1: This component explains approximately 12.89% of the total variance in the dataset, capturing a substantial portion of the overall variation.
- PC2: Accounting for around 19.02% of the variance, PC2 contributes significantly to understanding additional patterns not captured by PC1.
- PC3: With approximately 24.63% of the variance explained, PC3 further expands on the variability present in the data, potentially capturing more nuanced relationships.
- PC4: Explaining about 29.08% of the variance, PC4 continues to contribute significantly to understanding the data’s structure.

- PC5: Capturing around 32.72% of the variance, PC5 adds to the understanding of unique patterns and relationships in the data.
- PC6: Explaining about 35.46% of the variance, PC6 contributes notably to the overall variability captured by the model.
- PC7: With approximately 37.29% of the variance explained, PC7 continues to enrich our understanding of the data’s structure.
- PC8: Accounting for around 38.93% of the variance, PC8 adds further insights into the variability present in the dataset.
- PC9: Explaining about 40.33% of the variance, PC9 contributes significantly to understanding additional patterns beyond the previous components.
- PC10: Capturing approximately 41.54% of the variance, PC10 continues to provide valuable information about the data’s structure.
- PC11 captures additional unique patterns in the data, explaining approximately 42.70% of the total variance beyond what the previous components have accounted for.
- PC12 further contributes to explaining the variability in the dataset, accounting for approximately 44.80% of the total variance.
- PC13 continues to capture distinct patterns, explaining approximately 45.79% of the total variance.
- PC14 adds to the understanding of the dataset by explaining approximately 46.71% of the total variance.
- PC15 provides insight into additional underlying structures, explaining approximately 47.57% of the total variance.
- PC16 uncovers further patterns in the data, explaining approximately 48.38% of the total variance.
- PC17 continues the trend of revealing unique aspects, explaining approximately 49.27% of the total variance.
- PC18 contributes to understanding the dataset by explaining approximately 49.97% of the total variance.
- PC19 captures additional variation in the data, explaining approximately 50.64% of the total variance.
- PC20 provides further insights into the underlying structure, explaining approximately 51.34% of the total variance.
- PC21 continues to reveal unique patterns, explaining approximately 51.98% of the total variance.

- PC22 adds to the understanding of the dataset by explaining approximately 52.64% of the total variance.

- PC23 uncovers additional variation in the data, explaining approximately 53.27% of the total variance.

- PC24 contributes to understanding the dataset by explaining approximately 53.87% of the total variance.

- PC25 provides further insights into the underlying structure, explaining approximately 54.39% of the total variance.

Calculating equation of first component

```
loadings_first_component <- PCA$rotation[1,]
variable_names <- colnames(data)
equation <- "PC1 = "

for (i in seq_along(loadings_first_component)) {
  # Append each term to the equation string
  equation <- paste(equation, paste(loadings_first_component[i], "*", variable_names[i]), sep = " + ")
}
print(equation)
```

```
## [1] "PC1 =  + 0.107688837512609 * A1 + -0.000607634358017455 * A2 + 0.0717719266179847 *
A3 + -0.104288909915405 * A4 + 0.0361707877257911 * A5 + -0.0377196528950324 * A6 +
0.0108627405502496 * A7 + -0.0949412631864418 * A8 + 0.0742239562877308 * A9 + -
0.202099188492801 * A10 + 0.134190824071809 * B1 + -0.0440628815146651 * B2 +
0.0583842488242584 * B3 + -0.0125127655564609 * B4 + 0.0258549766627675 * B5 + -
0.082774611595253 * B6 + 0.0197545936886563 * B7 + -0.06615985238417 * B8 +
0.090148382094934 * B9 + -0.0831653291469757 * B10 + 0.0341752619477459 * B11 +
0.027465466150087 * B12 + -0.115583127702119 * B13 + 0.0369037672722159 * C1 + -
0.0186301481933703 * C2 + 0.0458493302425082 * C3 + -0.0536567313906488 * C4 +
0.00626231638061165 * C5 + -0.138474511065638 * C6 + 0.00813740484688702 * C7 + -
0.0293024187032727 * C8 + 0.0132597698718794 * C9 + -0.0206882834086204 * C10 +
0.0253930781344029 * D1 + -0.0266849240677188 * D2 + 0.0299298775960761 * D3 + -
0.0330785191278043 * D4 + 0.0232053360060502 * D5 + -0.109001417272977 * D6 + -
0.00744055306777567 * D7 + 0.00526716342822014 * D8 + 0.167566780936231 * D9 + -
0.00499994108811575 * D10 + -0.0345310312771531 * E1 + 0.0436322197732453 * E2 +
0.0344586643534838 * E3 + -0.07464926431246 * E4 + -0.0148308699522505 * E5 +
0.0173304210981651 * E6 + 0.0623276110200467 * E7 + -0.0265983223814058 * E8 + -
0.0325463689286593 * E9 + -0.0102225363394566 * E10 + -0.0430668943595973 * F1 +
0.046650786848225 * F2 + 0.0363525902207215 * F3 + 0.0129763537649023 * F4 +
0.0593034776859585 * F5 + -0.00852878187425003 * F6 + -0.0511123648951801 * F7 + -
0.0142356438181979 * F8 + 0.0122071665311602 * F9 + 0.0889436155965487 * F10 +
0.0495503355154718 * G1 + 0.0205705001045581 * G2 + -0.000252522322346653 * G3 + -
0.0473884919864824 * G4 + -0.04471103366004 * G5 + -0.00713374521976899 * G6 + -
0.102983510101151 * G7 + 0.00905197854065297 * G8 + 0.00351666000748918 * G9 +
0.132922024482152 * G10 + 7.87871979465952e-05 * H1 + -0.0452895221633688 * H2 + -
0.121535385651749 * H4 + -0.0592903955366528 * H5 + -0.0183661785037103 * H6 + -
0.114483601182519 * H7 + 0.044247286997945 * H8 + 0.0177113885615819 * H9 +
0.0479810294533258 * H10 + 0.0207156233961283 * I1 + 0.0510543636023547 * I2 + -
0.00261263664171243 * I3 + -0.00338225368848512 * I4 + -0.126094042095063 * I5 + -
0.0018658179770743 * I6 + 0.029368333604155 * I7 + 0.104063256303854 * I8 +
0.0333547434745203 * I9 + -0.0362380143811021 * I10 + -0.0221775207886494 * J1 + -
0.0443165118262443 * J2 + -0.0276167022101894 * J3 + -0.0304943957753827 * J4 + -
0.0260631444243015 * J5 + -0.0265074122823339 * J6 + 0.0992938199235987 * J7 +
0.0137499795398283 * J8 + 0.00304779774376655 * J9 + 0.00616726376684408 * K1 + -
0.0118389335848942 * K2 + -0.0671177757988993 * K3 + -0.0564937965780474 * K4 + -
0.167245148214775 * K5 + 0.0412848840549026 * K6 + 0.0320885474808324 * K7 +
0.0225769257023241 * K8 + 0.0255680441062135 * K9 + 0.0680148160697296 * K10 +
0.111627201580665 * L1 + -0.00926535505586133 * L2 + -0.0159699089755806 * L3 + -
0.0831364779021479 * L4 + -0.0507479751767364 * L5 + -0.115335683785823 * L6 +
0.143979335499705 * L7 + 0.201953829364046 * L8 + 0.0182134565729764 * L9 +
0.177001867941051 * L10 + -0.156978626258852 * M1 + -0.314903929746294 * M2 + -
0.214936911951467 * M3 + 0.0385273157715941 * M4 + 0.201232969024521 * M5 +
0.146675990022069 * M6 + -0.0375304179677273 * M7 + -0.0833523469773029 * M8 + -
0.0222794261571607 * M9 + -0.137409408642863 * M10 + -0.0573389571745327 * N1 +
0.0105016761904649 * N2 + 0.0431237199273686 * N3 + 0.252237272241084 * N4 +
0.0477283244505034 * N5 + -0.0759909752701043 * N6 + -0.122398729741589 * N7 + -
0.0958042382841144 * N8 + 0.00640965310133097 * N9 + 0.126458533601053 * N10 + -
0.118256919990697 * O1 + -0.0990494820415183 * O2 + -0.0161346838278886 * O3 +
0.0918516672968629 * O4 + 0.0274410180270775 * O5 + -0.0219994010334594 * O6 + -
0.0507228628419817 * O7 + 0.0334856025596856 * O8 + 0.0190071188519293 * O9 + -
0.115365905984597 * O10 + 0.0369193513528551 * P1 + -0.0281681480766451 * P2 +
0.00416515271667506 * P3 + -0.000921787008894993 * P4 + 0.0202823279788101 * P5 + -
0.00268909563624996 * P6 + 0.00405948203126332 * P7 + -0.0128736833495133 * P8 +
0.00101108701966617 * P9 + 0.112262435240244 * P10"
```

PC1 = 0.107688837512609 * A1 - 0.000607634358017455 * A2 + 0.0717719266179847 * A3 - 0.104288909915405 * A4 + 0.0361707877257911 * A5 - 0.0377196528950324 * A6 + 0.0108627405502496 * A7 - 0.0949412631864418 * A8 + 0.0742239562877308 * A9 - 0.202099188492801 * A10 + 0.134190824071809 * B1 - 0.0440628815146651 * B2 + 0.0583842488242584 * B3 - 0.0125127655564609 * B4 + 0.0258549766627675 * B5 - 0.082774611595253 * B6 + 0.0197545936886563 * B7 - 0.06615985238417 * B8 + 0.090148382094934 * B9 - 0.0831653291469757 * B10 + 0.0341752619477459 * B11 + 0.027465466150087 * B12 - 0.115583127702119 * B13 + 0.0369037672722159 * C1 - 0.0186301481933703 * C2 + 0.0458493302425082 * C3 - 0.0536567313906488 * C4 + 0.00626231638061165 * C5 - 0.138474511065638 * C6 + 0.00813740484688702 * C7 - 0.0293024187032727 * C8 + 0.0132597698718794 * C9 - 0.0206882834086204 * C10 + 0.0253930781344029 * D1 - 0.0266849240677188 * D2 + 0.0299298775960761 * D3 - 0.0330785191278043 * D4 + 0.0232053360060502 * D5 - 0.109001417272977 * D6 - 0.00744055306777567 * D7 + 0.00526716342822014 * D8 + 0.167566780936231 * D9 - 0.00499994108811575 * D10 - 0.0345310312771531 * E1 + 0.0436322197732453 * E2 + 0.0344586643534838 * E3 - 0.07464926431246 * E4 - 0.0148308699522505 * E5 + 0.0173304210981651 * E6 + 0.0623276110200467 * E7 - 0.0265983223814058 * E8 - 0.0325463689286593 * E9 - 0.0102225363394566 * E10 - 0.0430668943595973 * F1 + 0.046650786848225 * F2 + 0.0363525902207215 * F3 + 0.0129763537649023 * F4 + 0.0593034776859585 * F5 - 0.00852878187425003 * F6 - 0.0511123648951801 * F7 - 0.0142356438181979 * F8 + 0.0122071665311602 * F9 + 0.0889436155965487 * F10 + 0.0495503355154718 * G1 + 0.0205705001045581 * G2 - 0.000252522322346653 * G3 - 0.0473884919864824 * G4 - 0.04471103366004 * G5 - 0.00713374521976899 * G6 - 0.102983510101151 * G7 +

0.00905197854065297 * G8 + 0.00351666000748918 * G9 +
0.132922024482152 * G10 + 7.87871979465952e-05 * H1 -
0.0452895221633688 * H2 - 0.121535385651749 * H4 -
0.0592903955366528 * H5 - 0.0183661785037103 * H6 -
0.114483601182519 * H7 + 0.044247286997945 * H8 +
0.0177113885615819 * H9 + 0.0479810294533258 * H10 +
0.0207156233961283 * I1 + 0.0510543636023547 * I2 -
0.00261263664171243 * I3 - 0.00338225368848512 * I4 -
0.126094042095063 * I5 - 0.0018658179770743 * I6 +
0.029368333604155 * I7 + 0.104063256303854 * I8 +
0.0333547434745203 * I9 - 0.0362380143811021 * I10 -
0.0221775207886494 * J1 - 0.0443165118262443 * J2 -
0.0276167022101894 * J3 - 0.0304943957753827 * J4 -
0.0260631444243015 * J5 - 0.0265074122823339 * J6 +
0.0992938199235987 * J7 + 0.0137499795398283 * J8 +
0.00304779774376655 * J9 + 0.00616726376684408 * K1 -
0.0118389335848942 * K2 - 0.0671177757988993 * K3 -
0.0564937965780474 * K4 - 0.167245148214775 * K5 +
0.0412848840549026 * K6 + 0.0320885474808324 * K7 +
0.0225769257023241 * K8 + 0.0255680441062135 * K9 +
0.0680148160697296 * K10 + 0.111627201580665 * L1 -
0.00926535505586133 * L2 - 0.0159699089755806 * L3 -
0.0831364779021479 * L4 - 0.0507479751767364 * L5 -
0.115335683785823 * L6 + 0.143979335499705 * L7 +
0.201953829364046 * L8 + 0.0182134565729764 * L9 +
0.177001867941051 * L10 - 0.156978626258852 * M1 -
0.314903929746294 * M2 - 0.214936911951467 * M3 +
0.0385273157715941 * M4 + 0.201232969024521 * M5 +
0.146675990022069 * M6 - 0.0375304179677273 * M7 -
0.0833523469773029 * M8 - 0.0222794261571607 * M9 -
0.137409408642863 * M10 - 0.0573389571745327 * N1 +
0.0105016761904649 * N2 + 0.0431237199273686 * N3 +
0.252237272241084 * N4 + 0.0477283244505034 * N5 -
0.0759909752701043 * N6 - 0.122398729741589 * N7 -
0.0958042382841144 * N8 + 0.00640965310133097 * N9 +

0.126458533601053 * N10 - 0.118256919990697 * O1 - 0.0990494820415183 * O2 - 0.0161346838278886 * O3 + 0.0918516672968629 * O4 + 0.0274410180270775 * O5 - 0.0219994010334594 * O6 - 0.0507228628419817 * O7 + 0.0334856025596856 * O8 + 0.0190071188519293 * O9 - 0.115365905984597 * O10 + 0.0369193513528551 * P1 - 0.0281681480766451 * P2 + 0.00416515271667506 * P3 - 0.000921787008894993 * P4 + 0.0202823279788101 * P5 - 0.00268909563624996 * P6 + 0.00405948203126332 * P7 - 0.0128736833495133 * P8 + 0.00101108701966617 * P9 + 0.112262435240244 * P10

Question 4:

Extract component scores for first 25 elements

```
component_scores <- PCA$x[, 1:25]
```

Get the five-number summary

```
summary_component_scores <- apply(component_scores, 2, summary)
summary_component_scores
```

```
##                   PC1           PC2           PC3           PC4           PC5
## Min.    -1.779320e+01 -1.439748e+01 -1.478512e+01 -1.110289e+01 -1.107223e+01
## 1st Qu. -2.950171e+00 -2.188120e+00 -1.958731e+00 -1.712896e+00 -1.524834e+00
## Median   1.278522e-01 -1.979380e-01  3.647284e-02 -7.523147e-02  3.891840e-02
## Mean    -1.649512e-16  3.504188e-17  5.886843e-16  3.176245e-16 -9.064714e-17
## 3rd Qu.  3.162454e+00  1.992682e+00  1.962562e+00  1.600530e+00  1.530749e+00
## Max.     1.619957e+01  1.364631e+01  1.294157e+01  1.732589e+01  1.161567e+01
##                   PC6           PC7           PC8           PC9          PC10
## Min.    -1.626798e+01 -8.685520e+00 -8.171756e+00 -7.754561e+00 -7.379787e+00
## 1st Qu. -1.207044e+00 -1.126025e+00 -1.079946e+00 -9.587800e-01 -8.680200e-01
## Median   6.167251e-02 -3.132096e-02 -2.669309e-02  1.688170e-03  1.732168e-02
## Mean     1.029905e-16  7.329281e-17 -1.810359e-16  4.081878e-16 -1.659744e-16
## 3rd Qu.  1.278679e+00  1.075237e+00  1.034350e+00  9.734263e-01  8.915730e-01
## Max.     1.798814e+01  8.626801e+00  7.888475e+00  6.931532e+00  7.099293e+00
##                  PC11          PC12          PC13          PC14          PC15
## Min.    -7.527767e+00 -6.223872e+00 -6.875376e+00 -9.096488e+00 -5.902834e+00
## 1st Qu. -8.816025e-01 -8.716936e-01 -8.384954e-01 -8.134754e-01 -7.643258e-01
## Median  -3.796025e-02 -1.873282e-02 -8.217513e-03 -2.620702e-02  2.656322e-02
## Mean     1.460329e-16 -2.859581e-16  8.083033e-17 -1.587626e-16  1.221809e-16
## 3rd Qu.  8.576892e-01  8.482024e-01  8.309225e-01  7.884796e-01  7.925273e-01
## Max.     7.925337e+00  6.202242e+00  6.575711e+00  6.845803e+00  6.431436e+00
##                  PC16          PC17          PC18          PC19          PC20
## Min.    -6.065441e+00 -6.588935e+00 -5.559624e+00 -4.960736e+00 -5.207750e+00
## 1st Qu. -7.620555e-01 -7.332883e-01 -7.249945e-01 -7.114920e-01 -6.918997e-01
## Median  -1.591907e-02 -2.196463e-02 -4.377204e-05  1.190373e-02 -6.000709e-03
## Mean    -9.514339e-17 -6.374305e-17  9.983664e-17  3.593041e-17 -1.231226e-16
## 3rd Qu.  7.537479e-01  7.266368e-01  7.204348e-01  7.124852e-01  6.897956e-01
## Max.     5.785975e+00  5.547510e+00  5.734297e+00  6.023398e+00  5.308008e+00
##                  PC21          PC22          PC23          PC24          PC25
## Min.    -6.621833e+00 -5.592240e+00 -5.200801e+00 -4.737915e+00 -4.748267e+00
## 1st Qu. -6.681870e-01 -6.474869e-01 -6.568038e-01 -6.562682e-01 -6.684284e-01
## Median  -6.986536e-03 -1.600384e-03  2.648964e-03  9.550553e-06 -2.282875e-02
## Mean     8.065224e-17  1.670584e-17  2.103828e-16 -3.300439e-17 -2.190917e-16
## 3rd Qu.  6.743047e-01  6.604934e-01  6.490948e-01  6.503533e-01  6.448628e-01
## Max.     4.944333e+00  5.553091e+00  5.783060e+00  4.788818e+00  5.201023e+00
```

- The five-number summary of the component scores provides insights into the distribution of scores for each component.
- If the range between the minimum and maximum scores is wide, it indicates a significant variation in the corresponding personality trait among the respondents. On the other hand, if the interquartile range (IQR) between Q1 and Q3 is small, it suggests that most respondents have similar scores for that particular trait.
- The median provides information about the central tendency of the scores, while the quartiles give insights into the spread of scores around the median.
- PC1, PC2, PC4, PC7, PC10, PC11, PC14, PC16, PC17, PC19, PC21, PC23, and PC25 have relatively wide score distributions based on the large differences between their minimum and maximum scores and their IQRs.
- PC6, PC8, PC9, PC12, PC13, PC15, PC18, PC20, and PC22 have relatively similar score distributions compared to the other components due to smaller differences between their minimum and maximum scores and their IQRs.

```
library(stats)
```

Perform factor analysis

```
factor_analysis_result <- factanal(data, factors = 25)

print(factor_analysis_result)
```

```
## 
## Call:
## factanal(x = data, factors = 25)
## 
## Uniquenesses:
##    A1    A2    A3    A4    A5    A6    A7    A8    A9   A10    B1    B2    B3 
## 0.462 0.482 0.538 0.528 0.440 0.500 0.603 0.728 0.495 0.763 0.633 0.574 0.602 
##    B4    B5    B6    B7    B8    B9   B10   B11   B12   B13    C1    C2    C3 
## 0.576 0.599 0.650 0.657 0.753 0.617 0.531 0.611 0.553 0.665 0.710 0.452 0.656 
##    C4    C5    C6    C7    C8    C9   C10    D1    D2    D3    D4    D5    D6 
## 0.590 0.457 0.481 0.379 0.356 0.473 0.469 0.288 0.523 0.655 0.702 0.403 0.693 
##    D7    D8    D9   D10    E1    E2    E3    E4    E5    E6    E7    E8    E9 
## 0.455 0.653 0.596 0.574 0.381 0.292 0.421 0.397 0.526 0.457 0.602 0.430 0.763 
##   E10    F1    F2    F3    F4    F5    F6    F7    F8    F9   F10    G1    G2 
## 0.715 0.584 0.390 0.749 0.352 0.714 0.298 0.436 0.674 0.292 0.622 0.359 0.388 
##    G3    G4    G5    G6    G7    G8    G9   G10    H1    H2    H4    H5    H6 
## 0.546 0.334 0.350 0.365 0.508 0.532 0.456 0.462 0.522 0.638 0.837 0.707 0.652 
##    H7    H8    H9   H10    I1    I2    I3    I4    I5    I6    I7    I8    I9 
## 0.763 0.654 0.645 0.805 0.424 0.561 0.676 0.338 0.755 0.534 0.440 0.320 0.440 
##   I10    J1    J2    J3    J4    J5    J6    J7    J8    J9    K1    K2    K3 
## 0.618 0.567 0.430 0.549 0.310 0.681 0.685 0.549 0.515 0.667 0.397 0.534 0.432 
##    K4    K5    K6    K7    K8    K9   K10    L1    L2    L3    L4    L5    L6 
## 0.532 0.500 0.288 0.251 0.794 0.439 0.591 0.516 0.517 0.465 0.500 0.585 0.658 
##    L7    L8    L9   L10    M1    M2    M3    M4    M5    M6    M7    M8    M9 
## 0.609 0.628 0.579 0.557 0.606 0.603 0.599 0.591 0.723 0.508 0.694 0.514 0.625 
##   M10    N1    N2    N3    N4    N5    N6    N7    N8    N9   N10    O1    O2 
## 0.618 0.421 0.589 0.420 0.404 0.737 0.509 0.584 0.458 0.523 0.704 0.666 0.503 
##    O3    O4    O5    O6    O7    O8    O9   O10    P1    P2    P3    P4    P5 
## 0.571 0.555 0.613 0.692 0.632 0.371 0.592 0.578 0.260 0.338 0.593 0.646 0.665 
##    P6    P7    P8    P9   P10 
## 0.737 0.620 0.408 0.456 0.619 
## 
## Loadings:
##     Factor1 Factor2 Factor3 Factor4 Factor5 Factor6 Factor7 Factor8 Factor9
## A1            0.253                   0.191   0.610                        
## A2            0.324                   0.136   0.493  -0.150  -0.154        
## A3    0.146                           0.171   0.570                        
## A4            0.151  -0.157           0.163   0.561  -0.111  -0.145        
## A5   -0.109   0.314                   0.115   0.516          -0.103        
## A6   -0.159   0.309                   0.115   0.501          -0.111        
## A7            0.123           0.123   0.107   0.527          -0.148        
## A8                    0.170          -0.156  -0.337   0.178   0.110        
## A9           -0.144   0.222          -0.136  -0.488   0.189   0.213        
## A10                   0.167                  -0.279                        
## B1                   -0.433                   0.132   0.118           0.101
## B2                   -0.325                                           0.153
## B3                   -0.347                           0.142                
## B4   -0.180          -0.352                           0.139                
## B5   -0.190          -0.285                                                
## B6                   -0.331  -0.144                           0.107   0.197
## B7   -0.101  -0.116  -0.198   0.128                   0.155          -0.114
## B8   -0.124          -0.213   0.112                                        
## B9    0.127           0.298   0.181                                        
## B10   0.439           0.298                                           0.154
## B11   0.277  -0.150   0.131          -0.109                                
## B12   0.134           0.473                                                
## B13   0.131           0.424                                                
## C1   -0.432                                                                
## C2   -0.594   0.156                   0.143   0.105          -0.111        
## C3   -0.470   0.107                           0.138                        
## C4   -0.504                                                  -0.136        
## C5  -0.444                                          -0.105
## C6   0.566         0.112 -0.102                       0.116  0.192
## C7   0.674 -0.130        -0.108                0.134  0.127
## C8   0.636 -0.176               -0.121                0.123
## C9   0.639 -0.117                                            0.152
## C10  0.654 -0.110  0.116
## D1  -0.230  0.244                      0.109
## D2          0.200                                            0.122
## D3  -0.209  0.185               0.293                       0.243
## D4  -0.235  0.104 -0.163         0.110                       0.178
## D5  -0.199  0.168                      0.103
## D6  -0.216  0.111 -0.153                                     0.236
## D7   0.314 -0.266  0.123
## D8   0.212         0.324  0.166                             -0.137
## D9   0.241 -0.164  0.102
## D10  0.478 -0.143  0.105
## E1  -0.142  0.591  0.112         0.138  0.119 -0.200         0.189
## E2  -0.103  0.530  0.120                0.118 -0.244
## E3          0.224                       0.137                0.221
## E4          0.444  0.152                      -0.282         0.137
## E5          0.254 -0.110                0.247                0.162
## E6          0.297  0.106 -0.169                              0.508
## E7                 0.268
## E8   0.149 -0.422                              0.288
## E9   0.149         0.227
## E10        -0.124                              0.109
## F1                 0.172  0.540
## F2                        0.701         0.115        -0.115 -0.132
## F3                 0.272  0.264         0.124
## F4                 0.159  0.727         0.178        -0.141
## F5  -0.117  0.144  0.156  0.360         0.179        -0.112
## F6                       -0.720                       0.107  0.150
## F7                       -0.604                       0.103  0.333
## F8   0.154               -0.224                       0.114
## F9                       -0.730                       0.126  0.142
## F10                      -0.352                       0.154  0.349
## G1  -0.304  0.566                0.190  0.256 -0.184 -0.194
## G2  -0.151  0.669                0.137  0.172 -0.161
## G3  -0.122  0.472                0.181        -0.109         0.167
## G4  -0.217  0.664                0.187  0.271 -0.114 -0.103
## G5  -0.165  0.672                0.229  0.231 -0.117
## G6   0.340 -0.628               -0.184 -0.121  0.119
## G7   0.346 -0.498               -0.129 -0.126  0.180  0.115
## G8   0.144 -0.371  0.331        -0.203 -0.116  0.181
## G9   0.216 -0.604               -0.184         0.194
## G10  0.196 -0.529               -0.191         0.236
## H1                -0.412                0.188
## H2                -0.387
## H4   0.177
## H5   0.234                       0.192  0.209
## H6                               0.116  0.250        -0.102
## H7                                    -0.141
## H8                 0.366
## H9                 0.415
## H10                0.235        -0.110 -0.150                0.200
## I1    0.255                     -0.112 -0.109         0.313
## I2    0.249                                    0.101  0.486  0.195
## I3    0.264 -0.129              -0.132         0.125  0.378  0.142
## I4    0.240 -0.110        -0.115 -0.184 -0.121  0.133  0.665
## I5    0.166                                    0.121  0.353  0.124
## I6    0.154         0.130               -0.177         0.491  0.191
## I7                        0.140  0.182  0.158        -0.657
## I8   -0.127  0.114         0.123  0.235  0.201 -0.106 -0.679
## I9   -0.119  0.110         0.121  0.126  0.239        -0.621
## I10                        0.124         0.164        -0.504
## J1    0.164        -0.144 -0.131                0.152  0.123  0.513
## J2    0.200 -0.117 -0.295 -0.142                0.209         0.166
## J3    0.152        -0.186 -0.123                              0.348
## J4    0.208        -0.129                       0.148         0.199
## J5                 -0.201 -0.247                0.116         0.385
## J6                        -0.277        -0.106         0.146  0.375
## J7           0.149        -0.215                              0.570
## J8                  0.102  0.550                             -0.177
## J9   -0.128  0.109  0.277
## K1          -0.210  0.114        -0.663         0.196  0.145
## K2    0.157 -0.351        -0.128 -0.381 -0.112  0.206  0.158  0.140
## K3          -0.566               -0.275         0.256
## K4    0.331 -0.147               -0.486         0.172  0.126  0.118
## K5    0.129 -0.254  0.162        -0.507         0.264
## K6           0.213                0.756  0.203        -0.149
## K7           0.180                0.792  0.183        -0.136
## K8                                0.400
## K9           0.163                0.657  0.202        -0.131
## K10          0.239 -0.100         0.521  0.122
## L1    0.596 -0.132
## L2    0.608 -0.134                                     0.105
## L3    0.642                              0.145
## L4    0.626                                            0.103
## L5    0.561
## L6    0.423                              0.211        -0.113
## L7    0.587
## L8   -0.498                                                   0.149
## L9   -0.530                                                   0.107
## L10  -0.596                                                   0.141
## M1                 -0.324                       0.108         0.118
## M2   -0.170  0.142 -0.227                0.120                0.337
## M3   -0.110        -0.362                0.129                0.183
## M4   -0.105  0.226 -0.382         0.105  0.144                0.205
## M5   -0.144  0.135        -0.195                              0.260
## M6                  0.670
## M7                  0.500
## M8                  0.674
## M9                    0.538   0.119
## M10                   0.555   0.102
## N1    0.257  -0.252                  -0.209  -0.166   0.544   0.164
## N2    0.109  -0.186                  -0.127  -0.138   0.492   0.114   0.122
## N3           -0.143  -0.141                           0.712
## N4    0.132  -0.313                  -0.122           0.630
## N5           -0.104  -0.135                           0.456
## N6           -0.171                                   0.650
## N7           -0.178                  -0.149           0.549   0.106
## N8   -0.108   0.366           0.115   0.100   0.262  -0.306  -0.189
## N9   -0.211   0.269   0.109   0.162   0.106   0.285  -0.244  -0.197
## N10   0.154   0.138                   0.109   0.153  -0.393  -0.106
## O1    0.266                   0.169
## O2   -0.187           0.122   0.187
## O3                            0.302                   0.110
## O4                            0.132                                   0.110
## O5   -0.141                   0.150
## O6                                                                    0.105
## O7   -0.111                  -0.135                                   0.145
## O8    0.186                  -0.102
## O9    0.153                                                           0.104
## O10   0.298                  -0.148
## P1    0.448                                  -0.104           0.157
## P2    0.381           0.121                                   0.148   0.145
## P3    0.203                                  -0.147           0.156
## P4    0.213                                  -0.177           0.170
## P5    0.470           0.134
## P6    0.269           0.134                                   0.141
## P7    0.173           0.110                  -0.116
## P8   -0.352                                   0.114          -0.147
## P9   -0.132                                   0.218          -0.269
## P10           0.127           0.136           0.354          -0.245
##     Factor10 Factor11 Factor12 Factor13 Factor14 Factor15 Factor16 Factor17
## A1
## A2              0.118                               0.127    0.133
## A3                                                  0.214
## A4
## A5                                                           0.277
## A6              0.118                                        0.148
## A7
## A8              0.137
## A9
## A10                                        0.234   -0.156
## B1              0.202                      0.136
## B2     0.127    0.246                      0.181   -0.129
## B3              0.474
## B4     0.134    0.410
## B5     0.147    0.314
## B6     0.193    0.125    0.107             0.184   -0.167
## B7              0.386   -0.130   -0.137
## B8              0.352
## B9  -0.117
## B10 -0.161  -0.113   0.102                                   0.105
## B11 -0.122
## B12                                                 -0.124
## B13
## C1                                   0.120
## C2           0.209                   0.107
## C3   0.156   0.182
## C4                   0.111  -0.172                   0.109
## C5                          -0.546
## C6                           0.275
## C7
## C8          -0.132
## C9
## C10 -0.149                           0.113
## D1   0.723   0.149
## D2   0.581   0.155                   0.179
## D3   0.223   0.111           0.137
## D4   0.243   0.112                   0.117  -0.113
## D5   0.657   0.188  -0.117           0.115
## D6   0.259   0.195
## D7  -0.522           0.104           0.150
## D8  -0.230
## D9  -0.479                           0.111
## D10 -0.260           0.116  -0.136
## E1   0.127                                           0.141
## E2
## E3                   0.110                           0.630
## E4
## E5                                                   0.523
## E6                   0.135                           0.186
## E7                                                  -0.515
## E8
## E9                                                  -0.300
## E10                                                 -0.169
## F1           0.158
## F2           0.182
## F3
## F4
## F5                                           0.106
## F6                   0.124
## F7                   0.135           0.148
## F8                   0.130   0.159   0.103  -0.114   0.250
## F9                   0.131
## F10  0.114                           0.186
## G1
## G2
## G3   0.217                           0.157
## G4                                                   0.101
## G5   0.135
## G6  -0.133
## G7
## G8  -0.177                                                -0.171
## G9  -0.125
## G10 -0.237
## H1                                                 0.386
## H2                                                 0.310    0.154
## H4                                                        -0.124
## H5                                                 0.346
## H6                                                 0.480
## H7                                                -0.403
## H8                                        0.156   -0.383
## H9                                                -0.233   -0.161   -0.155
## H10                                              -0.104
## I1                               0.165    0.186
## I2             0.113                      0.207
## I3             0.105
## I4                               0.106
## I5                                        0.150
## I6                                        0.114            -0.106
## I7
## I8
## I9             0.110
## I10            0.166                               0.102
## J1                      0.134                                        0.161
## J2                      0.144                                        0.537
## J3                      0.140                      0.168             0.418
## J4                      0.149                                        0.708
## J5   0.133    0.104
## J6                      0.104
## J7                      0.117
## J8             0.195   -0.119
## J9                                                        -0.135   -0.395
## K1
## K2                                                        -0.109
## K3  -0.136                                                -0.171
## K4
## K5  -0.125
## K6
## K7                                                 0.105
## K8
## K9                                                 0.166
## K10                                      0.106
## L1  -0.119    0.100                                                 0.104
## L2                                        0.173
## L3                               0.148             0.178
## L4             0.144             0.107
## L5             0.105
## L6  -0.119
## L7
## L8
## L9                             -0.300
## L10  0.110    0.174
## M1                                               0.447             0.109
## M2   0.118    0.271
## M3            0.275                               0.184
## M4   0.147    0.224
## M5                     0.155                      0.164
## M6
## M7
## M8
## M9                                      0.102   -0.116            -0.115
## M10
## N1
## N2            0.161
## N7            0.108
## N8                                      0.145
## N9
## N10                                     0.148
## O1            0.382   -0.200             0.102
## O2            0.202   -0.511                              -0.126   -0.107
## O3            0.287   -0.418             0.139
## O4   0.153    0.541   -0.257
## O5   0.160    0.493   -0.205
## O6                     0.482
## O7                     0.513
## O8                     0.737
## O9                     0.579
## O10                    0.434             0.201             0.147    0.132
## P1                              0.656    0.164
## P2   0.106                      0.625    0.121
## P3                              0.147    0.505
## P4   0.131    0.159             0.207    0.343
## P5                              0.130    0.137
## P6                              0.209    0.217
## P7                                       0.544
## P8                             -0.631
## P9                             -0.145   -0.118
## P10           0.163            -0.153   -0.149    0.120
##     Factor18 Factor19 Factor20 Factor21 Factor22 Factor23 Factor24 Factor25
## A1                                      0.113
## A2
## A3
## A4                                     -0.146
## A5                                      0.183
## A6                                      0.189
## A7
## A8
## A9                                      0.200
## A10                              0.134
## B1  -0.120                                     0.117
## B2                -0.332
## B3
## B4                -0.193
## B5                -0.362
## B6                                                   0.126
## B7                                     0.128
## B8
## B9          0.420
## B10                0.277
## B11         0.463
## B12                0.390
## B13                0.322
## C1                               0.104 -0.159
## C2         -0.154                             -0.111
## C3
## C4                               0.109
## C5
## C6                                     0.111
## C7                                     0.201  0.105  0.105
## C8          0.276                              0.170
## C9                                     0.120  0.111  0.109
## C10
## D1
## D2
## D3
## D4
## D5
## D6
## D7          0.132
## D8                               0.185         0.119
## D9          0.148
## D10                              0.114
## E1   0.213                                     0.148
## E2   0.519                                     0.107
## E3   0.117
## E4   0.473
## E5
## E6   0.195
## E7                               0.115
## E8  -0.502
## E9                 0.138               -0.109
## E10 -0.408
## F1                                                   0.203
## F2
## F3         -0.138         0.121                       0.134
## F4
## F5
## F6                                                   0.318
## F7
## F8   0.174  0.196
## F9                                                    0.303
## F10
## G1
## G2   0.182
## G3   0.137 -0.141
## G4
## G5
## G6
## G7                                             0.113
## G8          0.123                0.127
## G9
## G10         0.184
## H1  -0.112  0.120 -0.262
## H2          0.129                                     0.136
## H4  -0.179                                     0.209
## H9                 0.136                       0.112
## H10         0.122                0.132
## I1                       -0.539
## I2
## I3
## I4                                             0.149
## I5
## I6                               0.203                0.124
## I7
## I8                               0.106        -0.160
## I9
## I10                                            0.101
## J1
## J2
## J3
## J4
## J5
## J6                                                    0.104
## J7
## J8                                                    0.217
## J9                               0.103
## K1
## K2                                             0.132
## K3                                      0.125
## K4
## K5                               0.116
## K6
## K7
## K8
## K9
## K10
## L1                                               -0.109
## L2
## L3
## L4                                                        -0.146
## L5                                                        -0.144
## L6                                               -0.164
## L7
## L8                               0.113    0.106             0.193
## L9
## L10
## M1
## M2                                                0.191
## M3                                                0.210
## M4
## M5                                                0.219
## M6
## M7
## M8
## M9
## M10                                       0.120
## N1
## N2
## N3
## N4  -0.176
## N5
## N6
## N7
## N8   0.253                                        0.179
## N9   0.114    0.108                               0.229   -0.102
## N10
## O1                                               -0.116
## O2                                       0.122
## O3
## O4
## O5
## O6
## O7                                                0.100
## O8
## O9
## O10
## P1
## P2
## P3                              -0.121
## P4
## P5                                       0.116
## P6                                               -0.125
## P7
## P8
## P9                               0.580
## P10                              0.180    0.146
##
```

```
##                Factor1 Factor2 Factor3 Factor4 Factor5 Factor6 Factor7 Factor8
## SS loadings     10.584   7.133   5.801   4.597   4.481   4.412   4.181   4.113
## Proportion Var   0.066   0.044   0.036   0.029   0.028   0.027   0.026   0.026
## Cumulative Var   0.066   0.110   0.146   0.175   0.202   0.230   0.256   0.281
##                Factor9 Factor10 Factor11 Factor12 Factor13 Factor14 Factor15
## SS loadings      3.053    2.976    2.866    2.740    2.286    1.836    1.825
## Proportion Var   0.019    0.018    0.018    0.017    0.014    0.011    0.011
## Cumulative Var   0.300    0.319    0.337    0.354    0.368    0.379    0.391
##                Factor16 Factor17 Factor18 Factor19 Factor20 Factor21 Factor22
## SS loadings       1.763    1.477    1.465    1.093    0.994    0.912    0.764
## Proportion Var    0.011    0.009    0.009    0.007    0.006    0.006    0.005
## Cumulative Var    0.402    0.411    0.420    0.427    0.433    0.438    0.443
##                Factor23 Factor24 Factor25
## SS loadings       0.699    0.616    0.612
## Proportion Var    0.004    0.004    0.004
## Cumulative Var    0.448    0.451    0.455
## 
## Test of the hypothesis that 25 factors are sufficient.
## The chi square statistic is 172301.2 on 9155 degrees of freedom.
## The p-value is 0
```

Extracted cumulative variance for the first 25 factors

```
cumulative_variance_factor_analysis <- c(0.066, 0.110, 0.146, 0.175, 0.202, 0.230, 0.256, 0.281, 0.300, 0.319, 
                                         0.337, 0.354, 0.368, 0.379, 0.391, 0.402, 0.411, 0.420, 0.427, 0.433,
                                         0.438, 0.443, 0.448, 0.451, 0.455)
```

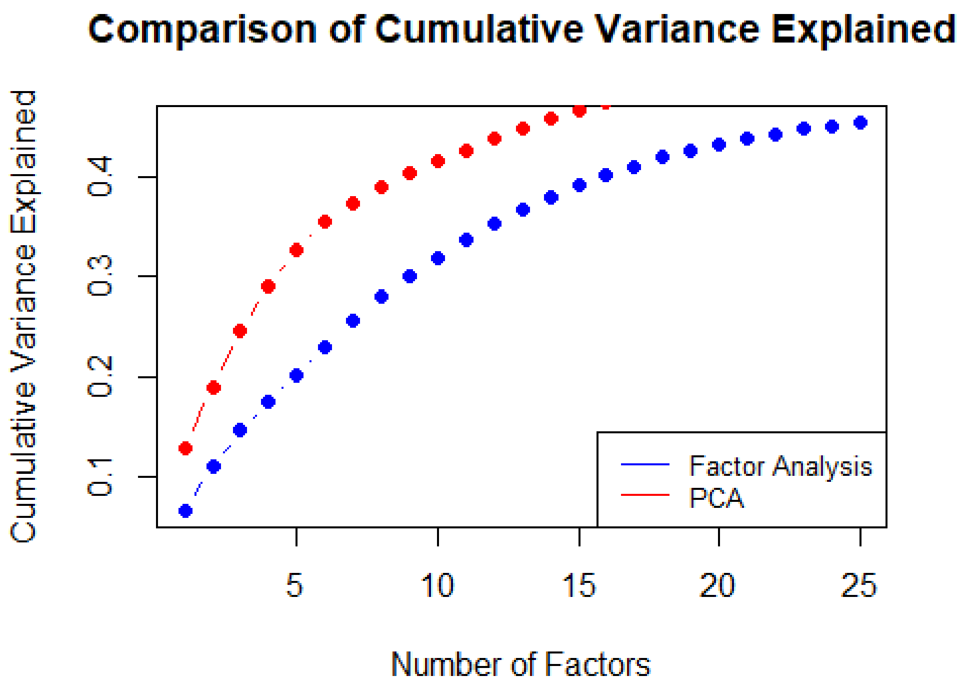
PCA loadings for first 25 selected components

```
cumulative_variance_first_25 <- cumulative_variance[1:25]


cumulative_variance_factor_analysis

##  [1] 0.066 0.110 0.146 0.175 0.202 0.230 0.256 0.281 0.300 0.319 0.337 0.354
## [13] 0.368 0.379 0.391 0.402 0.411 0.420 0.427 0.433 0.438 0.443 0.448 0.451
## [25] 0.455

cumulative_variance_first_25

##  [1] 0.1289027 0.1901729 0.2462770 0.2907731 0.3272241 0.3546038 0.3729398
##  [8] 0.3892965 0.4033017 0.4153824 0.4270187 0.4378507 0.4480324 0.4578583
## [15] 0.4670877 0.4757453 0.4838016 0.4916664 0.4992659 0.5063861 0.5131762
## [22] 0.5198166 0.5263771 0.5327499 0.5390109
```

Plotting

```
plot(1:25, cumulative_variance_factor_analysis, type="b", col="blue", pch=16,
     xlab="Number of Factors", ylab="Cumulative Variance Explained",
     main="Comparison of Cumulative Variance Explained")
```

```
lines(1:25, cumulative_variance_first_25, type="b", col="red", pch=16)
legend("bottomright", legend=c("Factor Analysis", "PCA"), col=c("blue", "red"), lty=1:1, cex=0.8)
```



Extract loadings from PCA

```
loadings_pca <- PCA$rotation[, 1:25]
head(loadings_pca)

##           PC1           PC2        PC3         PC4           PC5          PC6
## A1 0.10768884 -0.0006076344 0.07177193 -0.10428891  0.0361707877 -0.037719653
## A2 0.11937137  0.0010261761 0.10390163 -0.09822410  0.0206400264 -0.077550738
## A3 0.05567695 -0.0111939745 0.09561995 -0.17535296 -0.0003834672 -0.025152854
## A4 0.07937222 -0.0057066211 0.09985948 -0.13320403 -0.0221891592  0.008396858
## A5 0.11580845  0.0026518796 0.08507859 -0.08201028  0.0130091681 -0.092612132
## A6 0.11511517  0.0018846994 0.04260325 -0.09522134  0.0232314233 -0.088005426
##            PC7          PC8          PC9        PC10       PC11          PC12
## A1  0.01086274 -0.094941263  0.074223956 -0.20209919 0.13419082 -0.0440628815
## A2  0.07532253 -0.059517904 -0.008385399 -0.05852804 0.06468151  0.0062213192
## A3 -0.02641261 -0.126505002  0.044782992 -0.12356030 0.11736405 -0.0258109205
## A4  0.04168444 -0.004933295 -0.028032904 -0.13231079 0.12609381 -0.0265423960
## A5  0.08315361 -0.031379091  0.097290589 -0.14434065 0.09462559  0.0279172779
## A6  0.03622940 -0.045738156  0.070279498 -0.16445240 0.10254301  0.0008964866
##          PC13        PC14       PC15        PC16         PC17        PC18
## A1 0.05838425 -0.01251277 0.02585498 -0.08277461  0.019754594 -0.06615985
## A2 0.01025281  0.01129166 0.05912800 -0.02639748 -0.035850227 -0.05243588
## A3 0.06710513  0.07112347 0.10844413 -0.03658630  0.028577897 -0.02142427
## A4 0.01701779  0.05939206 0.12530907  0.04417441 -0.138311924 -0.10111845
## A5 0.07150349 -0.01563692 -0.07010748 -0.11755482  0.003949923 -0.00017013
## A6 0.10768005 -0.07163898 -0.03801706 -0.06089082  0.043166584 -0.04109107
##            PC19         PC20        PC21        PC22        PC23        PC24
## A1  0.090148382 -0.083165329 0.034175262  0.02746547 -0.11558313  0.03690377
## A2 -0.027276159 -0.001197372 0.012029971 -0.04002820 -0.04265873  0.03152778
## A3  0.106575546 -0.037785022 0.063644699  0.04415853 -0.01737077  0.06273636
## A4  0.081381425 -0.198449348 0.005023004  0.05484413  0.02062132  0.01368610
## A5  0.004829001 -0.025265431 0.068203665 -0.05774556 -0.14303866 -0.01817162
## A6 -0.015935787 -0.016772763 0.022635523 -0.01289308 -0.14543542  0.05272298
##            PC25
## A1 -0.018630148
## A2  0.002244189
## A3 -0.013687551
## A4  0.064597331
## A5  0.021835494
## A6  0.021383526
```

Extract loadings from Factor Analysis

```
loadings_FA <- loadings(factor_analysis_result)
head(loadings_FA)
```

```
##        Factor1    Factor2      Factor3    Factor4   Factor5   Factor6
## A1 -0.07420233 0.25288159 -0.014260298 0.06299586 0.1908701 0.6096830
## A2 -0.05317612 0.32350792 -0.046701625 0.09040337 0.1362955 0.4932417
## A3  0.14592264 0.06710945 -0.059287381 0.05059188 0.1707258 0.5704934
## A4  0.04917590 0.15058717 -0.157054671 0.07166480 0.1628174 0.5609988
## A5 -0.10888469 0.31435536  0.026080751 0.09219644 0.1153279 0.5159475
## A6 -0.15866190 0.30893092 -0.003647801 0.08063729 0.1147075 0.5011238
##         Factor7     Factor8     Factor9     Factor10    Factor11      Factor12
## A1 -0.016823738 -0.05848148 0.003629389  0.070510251  0.05955511 -0.0411244768
## A2 -0.149608761 -0.15350198 0.097136013  0.063848688  0.11761818 -0.0074158204
## A3 -0.004355633 -0.09685426 0.018465752 -0.004571956  0.07563064  0.0134922572
## A4 -0.110963465 -0.14469933 0.001357191 -0.006005745 -0.00235004  0.0396923231
## A5 -0.061724211 -0.10333071 0.095496236  0.057635103  0.05129225  0.0007984753
## A6  0.007851732 -0.11116131 0.002443158  0.036017192  0.11782069 -0.0172585335
##       Factor13     Factor14   Factor15    Factor16      Factor17     Factor18
## A1 -0.02867998 -0.098246527 0.06177394 0.080983087 -0.0296428667  0.004772529
## A2 -0.05327863  0.001555005 0.12713350 0.132967517 -0.0079595365  0.093091485
## A3 -0.03299469 -0.073872481 0.21378578 0.007299391  0.0281289918  0.003863827
## A4 -0.02979541  0.050113282 0.03568606 0.022074524  0.0325246867  0.049328816
## A5 -0.06373070 -0.077880105 0.07607760 0.277252881  0.0066973617  0.022225136
## A6 -0.08235709 -0.065663757 0.06302593 0.147641028  0.0001819166 -0.032712860
##       Factor19     Factor20     Factor21     Factor22    Factor23    Factor24
## A1 -0.03415554 -0.068916818 -0.018659414  0.113048995  0.03804565  0.01715064
## A2  0.04106695  0.062703613  0.077399879  0.029068572  0.07805354 -0.01258403
## A3 -0.05310685  0.006724337  0.013717088  0.004569046  0.04784082  0.02359567
## A4  0.07680547  0.038976202  0.007264379 -0.145964020 -0.04233038  0.01303271
## A5 -0.03301355 -0.014170049  0.042511618  0.182590735 -0.02396402 -0.01885120
## A6 -0.02395434 -0.044777620  0.046198179  0.189327867  0.03030218  0.01392809
##        Factor25
## A1 -0.011643610
## A2  0.024684680
## A3 -0.025716677
## A4  0.035587092
## A5 -0.000737003
## A6 -0.034360580
```

- The varience explained by factor analysis is 0.455 and 0.5390109 for the first 25 components in PCA.
- By looking at the above plot, we can see the variance explained by PCA is having higher value than varience explained by factor analysis.
- The loading from factor analysis are having less values than loadings in PCA.
- PCA loadings are optimized to maximize variance along the extracted components.
- Factor Analysis explicitly models underlying latent constructs or factors, and the loadings represent the relationships between the observed variables and these factors.

Calculate absolute differences between PCA and FA loadings

```
loadings_diff <- abs(loadings_pca - loadings_FA)
```

Identify variables with the largest differences

```
max_diff <- apply(loadings_diff, 1, max, na.rm = TRUE)
```

Identify variables with the largest differences

```
max_diff <- apply(loadings_diff, 1, max, na.rm = TRUE)
print(max_diff)
```

```
##        A1        A2        A3        A4        A5        A6        A7        A8 
## 0.6474026 0.5707925 0.5956463 0.5526019 0.6085597 0.5891293 0.5590366 0.3598306 
##        A9       A10        B1        B2        B3        B4        B5        B6 
## 0.3973935 0.3720549 0.4121392 0.2996560 0.4941408 0.3676288 0.4171796 0.3517162 
##        B7        B8        B9       B10       B11       B12       B13        C1 
## 0.4377643 0.3714124 0.5778978 0.5191845 0.5978143 0.4025846 0.3392836 0.5059841 
##        C2        C3        C4        C5        C6        C7        C8        C9 
## 0.7098377 0.5643264 0.5893895 0.6432820 0.6653780 0.8011082 0.7669267 0.7509237 
##       C10        D1        D2        D3        D4        D5        D6        D7 
## 0.7696388 0.6314164 0.4253407 0.3309555 0.2765649 0.5799412 0.2782892 0.4569210 
##        D8        D9       D10        E1        E2        E3        E4        E5 
## 0.2968487 0.3808042 0.5571839 0.5596150 0.5263869 0.7869894 0.4604450 0.6660540 
##        E6        E7        E8        E9       E10        F1        F2        F3 
## 0.4301553 0.6598232 0.5138957 0.4696718 0.3245097 0.5892581 0.8502467 0.3993198 
##        F4        F5        F6        F7        F8        F9       F10        G1 
## 0.8384219 0.4047385 0.8165161 0.7159516 0.3826190 0.8336062 0.4367824 0.5758614 
##        G2        G3        G4        G5        G6        G7        G8        G9 
## 0.6443894 0.4058005 0.6613718 0.6370051 0.6120302 0.5086240 0.3859878 0.5803523 
##       G10        H1        H2        H4        H5        H6        H7        H8 
## 0.4902168 0.3564085 0.4053361 0.3161782 0.2247249 0.2885577 0.2234658 0.3658689
##        H9       H10        I1        I2        I3        I4        I5        I6
## 0.4076227 0.2914192 0.7019249 0.5209505 0.4246638 0.7411690 0.3399360 0.5485732
##        I7        I8        I9       I10        J1        J2        J3        J4
## 0.8067623 0.7987271 0.6909607 0.5460837 0.4891573 0.6130592 0.4770320 0.7545197
##        J5        J6        J7        J8        J9        K1        K2        K3
## 0.4062541 0.4628575 0.5159859 0.6424219 0.4869137 0.6862000 0.4160282 0.5415182
##        K4        K5        K6        K7        K8        K9       K10        L1
## 0.5030776 0.5065652 0.7418510 0.7723090 0.3891835 0.6243274 0.4866846 0.6924058
##        L2        L3        L4        L5        L6        L7        L8        L9
## 0.7214700 0.7244787 0.7140295 0.6491670 0.4617689 0.6764670 0.5604740 0.6087762
##       L10        M1        M2        M3        M4        M5        M6        M7
## 0.6897028 0.3367536 0.3927336 0.3435903 0.3880735 0.2843185 0.6355053 0.4721426
##        M8        M9       M10        N1        N2        N3        N4        N5
## 0.6239212 0.5139986 0.5392026 0.5992354 0.5706776 0.7933371 0.7546024 0.5266110
##        N6        N7        N8        N9       N10        O1        O2        O3
## 0.7587234 0.5584795 0.4182411 0.3806946 0.4257666 0.3961987 0.6040847 0.5964190
##        O4        O5        O6        O7        O8        O9       O10        P1
## 0.5659759 0.4969518 0.7986235 0.8108697 0.9864497 0.8095799 0.4549261 0.7505572
##        P2        P3        P4        P5        P6        P7        P8        P9
## 0.7517700 0.5454853 0.3769218 0.5665991 0.3278966 0.6234052 0.7327139 0.7666677
##       P10
## 0.4868869
```

Above values are the difference between PCA loadings and Factor analysis loadings

```
max_diff_var <- names(which(max_diff == max(max_diff, na.rm = TRUE)))
```

Display the variables with the largest differences

```
for (var in max_diff_var) {
  max_diff_value <- max_diff[var]
  cat("Variable", var, "has the largest difference of", max_diff_value, "between PCA and FA.\n")
}

## Variable O8 has the largest difference of 0.9864497 between PCA and FA.
```