1. **Load the Titanic dataset using read.csv(), explore it with str(), summary(), and head(). Identify numerical/categorical variables and handle missing values by imputing with mean, median, or mode. Convert categorical data into factors, scale numerical features, engineer new features, and save the preprocessed data using write.csv().**

Answer:

- **Survived –** 1 = Survived, 0 = Did not survive
- **Pclass –** Class of the ticket (1st, 2nd, or 3rd)
- **SibSp –** Number of siblings or spouse with them
- **Parch –** Number of parents or children with them
- **Fare –** Ticket price
- **Embarked** – Port of Embarkation (C = Cherbourg, Q = Queenstown, S = Southampton)

Rcode:

```r
# 1. Load dataset
data <- read.csv("titanic.csv")

# 2. Clean missing values
data <- na.omit(data)

# 3. Explore dataset
str(data)
summary(data)
head(data)

# 4. Convert columns to categorical
data$Survived <- as.factor(data$Survived)
data$Pclass <- as.factor(data$Pclass)
data$Sex <- as.factor(data$Sex)
data$Embarked <- as.factor(data$Embarked)

# 5. Add dummy missing values (for demo purpose only)
data[1, "Age"] <- NA
data[2, "Fare"] <- NA

# 6. Impute missing values using mean
print(data$Age[is.na(data$Age)] <- mean(data$Age, na.rm = TRUE))
print(data$Fare[is.na(data$Fare)] <- mean(data$Fare, na.rm = TRUE))

# 7. Scale numeric columns manually (Min-Max normalization)
data$Age <- (data$Age - min(data$Age)) / (max(data$Age) - min(data$Age))
data$Fare <- (data$Fare - min(data$Fare)) / (max(data$Fare) - min(data$Fare))
data$SibSp <- (data$SibSp - min(data$SibSp)) / (max(data$SibSp) - min(data$SibSp))
data$Parch <- (data$Parch - min(data$Parch)) / (max(data$Parch) - min(data$Parch))

# 8. Create a new feature: FamilySize = SibSp + Parch + 1 (including self)
data$FamilySize <- data$SibSp + data$Parch + 1

# 9. Save the cleaned and transformed dataset
write.csv(data, "preprocessed_data.csv", row.names = FALSE)
print("data has been preprocessed")
```

Sample output:

```
> str(data)
'data.frame':   714 obs. of  13 variables:
 $ PassengerId: int  1 2 3 4 5 7 8 9 10 11 ...
 $ Survived   : Factor w/ 2 levels "0","1": 1 2 2 2
 $ Pclass     : Factor w/ 3 levels "1","2","3": 3 1
 $ Name       : chr  "Braund, Mr. Owen Harris" "Cum
nce Briggs Thayer)" "Heikkinen, Miss. Laina" "Futre
May Peel)" ...
 $ Sex        : Factor w/ 2 levels "female","male":
 $ Age        : num  0.368 0.472 0.321 0.435 0.435 ...
 $ SibSp      : num  0.2 0.2 0 0.2 0 0 0.6 0 0.2 0.2 ...
```

```
> summary(data)
  PassengerId    Survived Pclass      Name              Sex
Min.   :  1.0   0:424    1:186   Length:714         female:261
1st Qu.:222.2   1:290    2:173   Class :character   male  :453
Median :445.0            3:355   Mode  :character
Mean   :448.6
3rd Qu.:677.8
Max.   :891.0
```

```
> head(data)
  PassengerId Survived Pclass
1           1        0      3
2           2        1      1
3           3        1      3
4           4        1      1
5           5        0      3
7           7        0      1
```

```
> # 6. Impute missing values using mean
> print(data$Age[is.na(data$Age)] <- mean(data$Age, na.rm = TRUE))
[1] 0.3680562
> print(data$Fare[is.na(data$Fare)] <- mean(data$Fare, na.rm = TRUE))
[1] 0.06761902

> write.csv(data, "preprocessed_data.c
> print("data has been preprocessed")
[1] "data has been preprocessed"
```

2.Use Esquisse to visualize the mtcars dataset: 1.Create a histogram/density plot for mpg (Miles per Gallon). 2. Compare the number of automatic vs. manual cars using a bar plot of am.3.Show the relationship between cyl (Cylinders) and hp (Horsepower) using a box/scatter plot. 4.Visualize mpg vs. wt (Weight) with color for cyl. 5.Create a bar plot showing car counts by gear (Transmission gears). Export and save the R code.

Answer:

Esquisse is a simple and easy-to-use tool in R that helps you create beautiful data visualizations without writing much code.

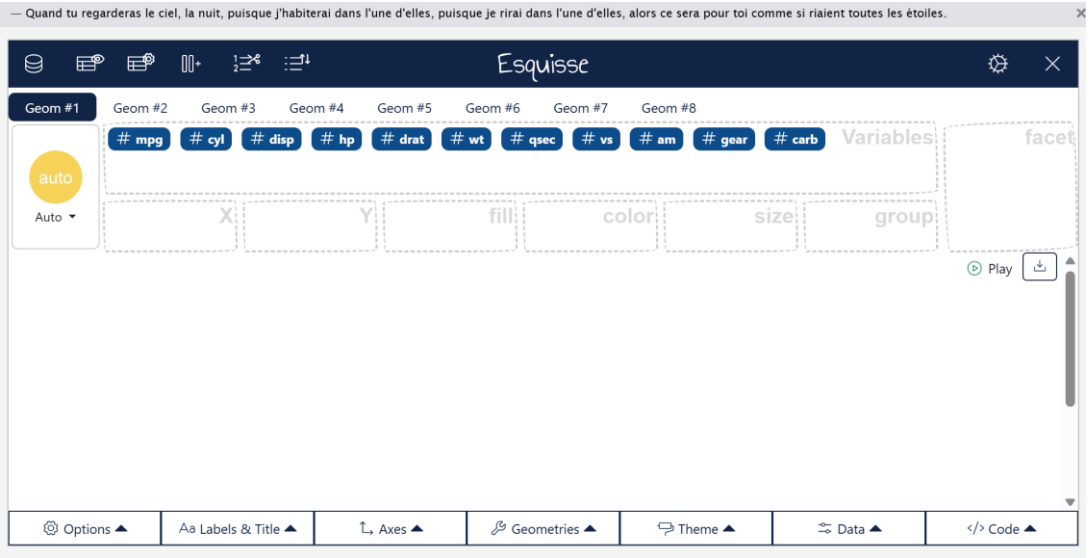
In the most basic terms:

- **Esquisse** is like a **drag-and-drop** tool for making plots and charts in R.
- It lets you **choose the data**, **pick the type of plot** you want (like bar charts, histograms, or scatter plots), and **customize** the appearance of the plot.
- You don't need to know complicated coding for visualizations. Just interact with the interface, and it generates the code for you!

Rcode:

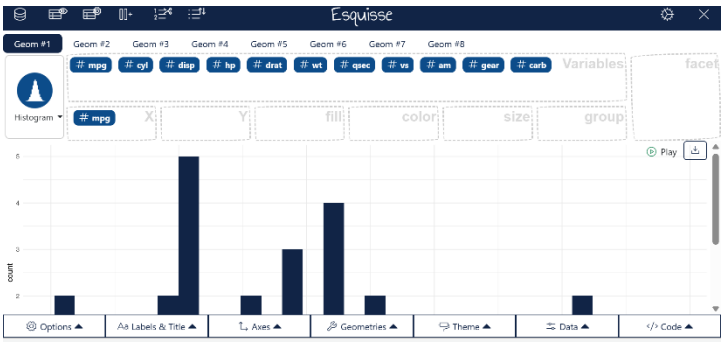
```r
# Load the libraries
library(esquisse)
library(ggplot2)

# Load the mtcars dataset
data(mtcars)

# Open Esquisse for visualization
esquisser(mtcars)
```

Output:

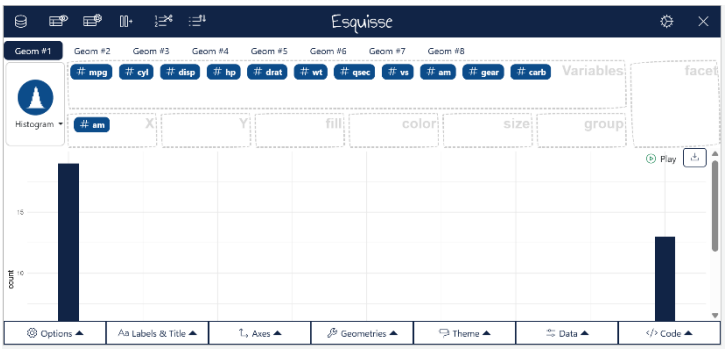


After opening Esquisse, follow these steps:

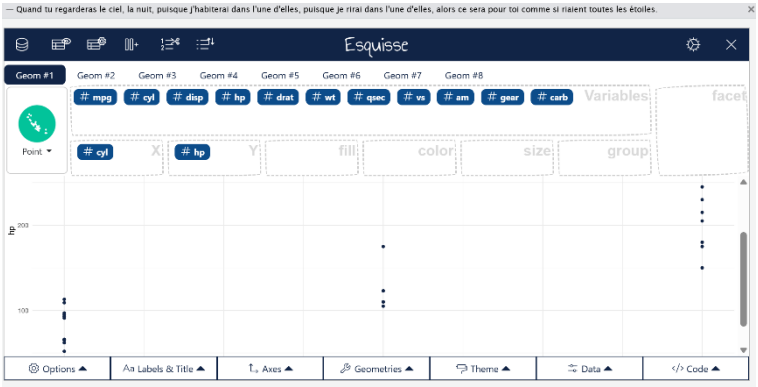
1. **Histogram/Density plot for mpg (Miles per Gallon):**
    - drag **mpg** for the **X-axis** and choose the **Histogram** or **Density Plot** option.



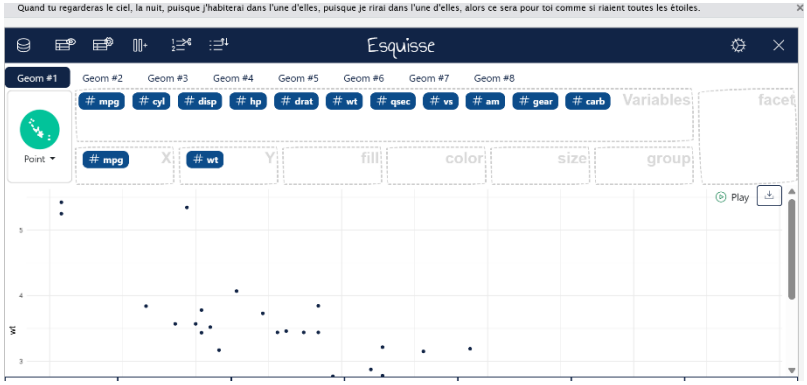
2. **Bar plot for am (Automatic vs. Manual):**
    - drag **am** (automatic vs. manual) for the **X-axis** and choose a **Bar Plot**.
      if barplot not enabling do same histogram



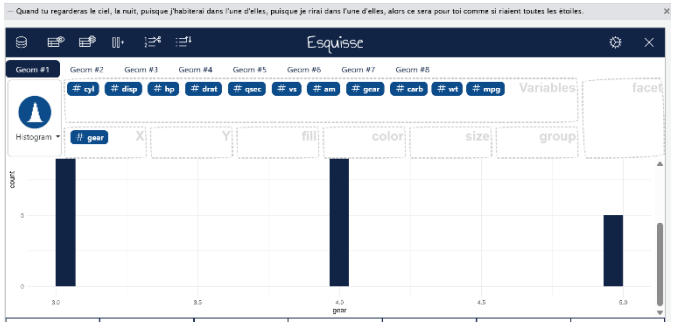
3. **Scatter plot for cyl (Cylinders) vs hp (Horsepower):**
    - For a **Scatter plot**, drag **cyl** for the **X-axis** and **hp** for the **Y-axis**.



4. **Visualize mpg vs. wt (Weight) with color for cyl (Cylinders):**
    - drag **mpg** for the **X-axis**, **wt** for the **Y-axis**, and **cyl** for the **color**.

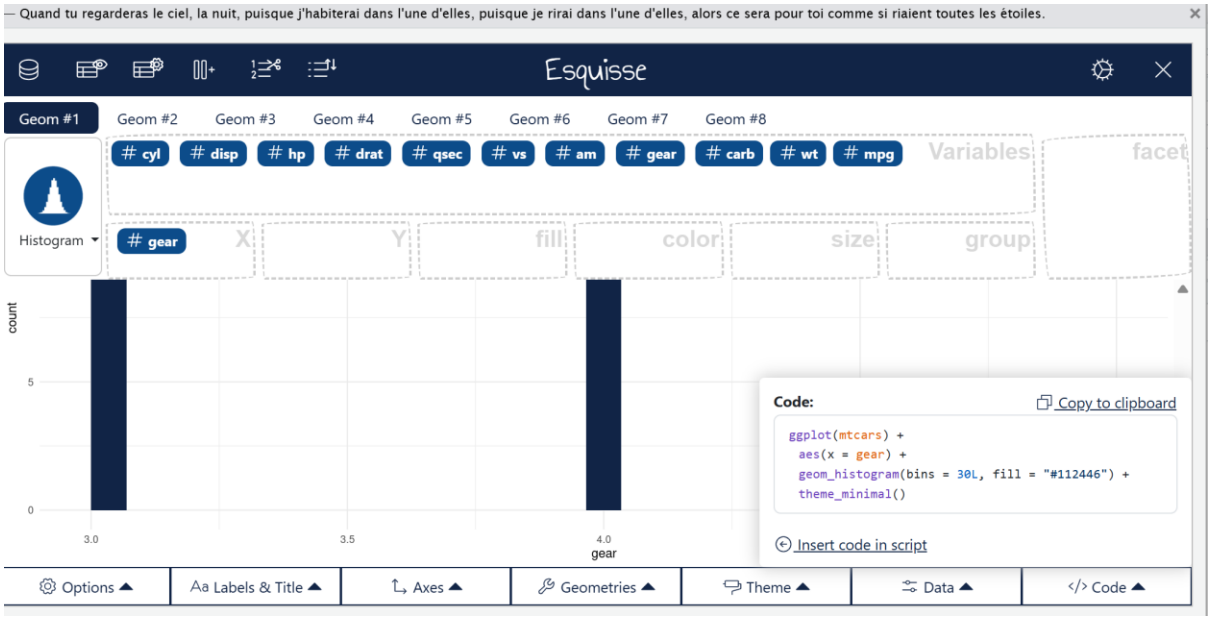


5. **Bar plot showing car counts by gear (Transmission gears):**
    - drag **gear** for the **X-axis** and choose a **Bar Plot** to show the count of each gear type. **if barplot not enabling do same histogram**



Export the R code:

Once you have finished visualizing your data in Esquisse, click the **code** button in Esquisse and select **insert code to scrit**. Paste this code in an R script file to save it.

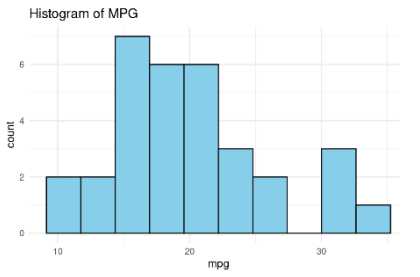


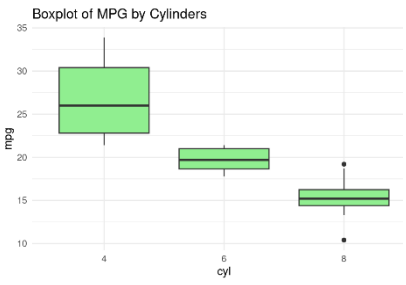
Do this for every step

3.Load a mtcars dataset, identify numerical and categorical variables, and visualize distributions using box plots, histograms, and violin plots. Use scatter plots with trend lines to analyze relationships. Create facets, bar plots, and heatmaps to explore patterns and correlations. Provide insights on distributions, outliers, and variable relationships.

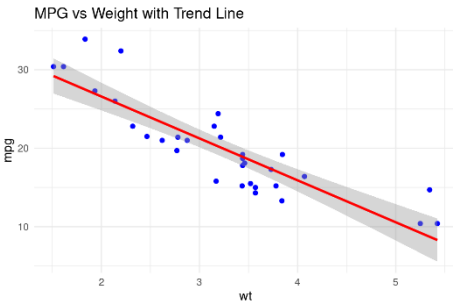
```r
# 1. Load required packages
library(ggplot2)

# 2. Load mtcars dataset
data <- mtcars
# Convert suitable variables to factors (categorical)
data$cyl <- as.factor(data$cyl)
data$gear <- as.factor(data$gear)
data$am <- as.factor(data$am)
data$vs <- as.factor(data$vs)
data$carb <- as.factor(data$carb)

# 3. Visualizations
# Histogram of mpg
ggplot(data, aes(x = mpg)) +
  geom_histogram(fill = "skyblue", bins = 10, color = "black") +
  labs(title = "Histogram of MPG") +
  theme_minimal()

# Box plot of mpg by cyl
ggplot(data, aes(x = cyl, y = mpg)) +
  geom_boxplot(fill = "lightgreen") +
  labs(title = "Boxplot of MPG by Cylinders") +
  theme_minimal()

# Violin plot of hp by gear
ggplot(data, aes(x = gear, y = hp)) +
  geom_violin(fill = "orange") +
  labs(title = "Violin Plot of HP by Gear") +
  theme_minimal()
```

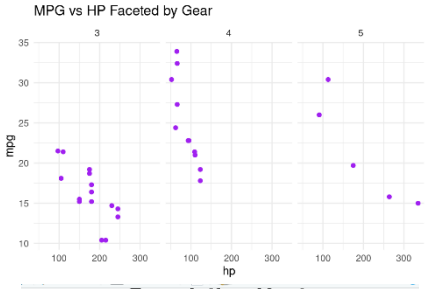
outputs:

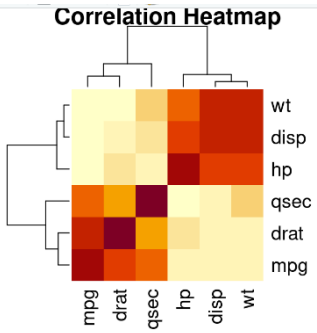




```r
# Scatter plot with trend line: mpg vs wt
ggplot(data, aes(x = wt, y = mpg)) +
  geom_point(color = "blue") +
  geom_smooth(method = "lm", color = "red") +
  labs(title = "MPG vs Weight with Trend Line") +
  theme_minimal()

# Facet scatter plot: mpg vs hp by gear
ggplot(data, aes(x = hp, y = mpg)) +
  geom_point(color = "purple") +
  facet_wrap(~ gear) +
  labs(title = "MPG vs HP Faceted by Gear") +
  theme_minimal()

# Bar plot: count of cars by gear
ggplot(data, aes(x = gear)) +
  geom_bar(fill = "tomato") +
  labs(title = "Car Count by Gear") +
  theme_minimal()
# 4. Heatmap: Correlation Between Numerical Variables (Only Numeric Columns)
numeric_data <- Filter(is.numeric, data)  # Keep only numeric columns
cor_matrix <- cor(numeric_data)  # Calculate correlation matrix
heatmap(cor_matrix, main = "Correlation Heatmap")


# 5. Outlier Detection using IQR (for mpg)
Q1 <- quantile(data$mpg, 0.25)
Q3 <- quantile(data$mpg, 0.75)
IQR <- Q3 - Q1
lower_bound <- Q1 - 1.5 * IQR
upper_bound <- Q3 + 1.5 * IQR
outliers <- data$mpg[data$mpg < lower_bound | data$mpg > upper_bound]
print("Outliers in mpg using IQR method:")
print(outliers)
```







4.Develop a Shiny app using the iris dataset with three features: (1) Select a numerical and categorical variable to display summary statistics. (2) Choose two numerical variables for a scatter plot, colored by a categorical variable. (3) Generate a box plot for a selected numerical and categorical variable.

Shiny is an R package that lets you build interactive web applications using R. It allows users to create data visualizations and dashboards easily without needing HTML, CSS, or JavaScript.

UI (User Interface):

- **fluidPage()**: Creates the app's layout.
- **3 dropdowns**: Allow the user to choose:
  - **X-axis** for scatter plot.
  - **Y-axis** for scatter plot.
  - **Box plot** variable.
- **Summary output**: Shows basic stats of the iris dataset.

- **Plot outputs**: Displays scatter plot and box plot based on user choices.

Server (Backend):

- **Summary**: Shows basic statistics of the iris dataset.
- **Scatter plot**: Plots the X and Y variables chosen by the user.
- **Box plot**: Plots the variable selected for the box plot.

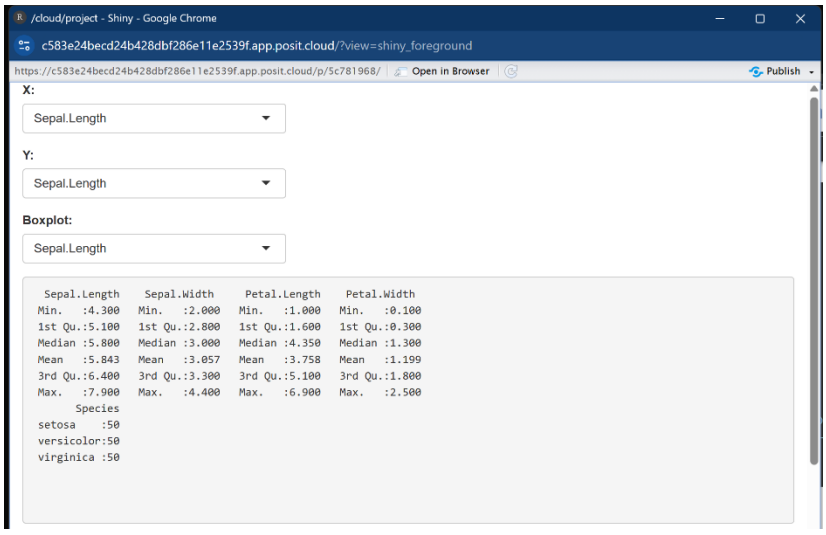
Run the App:

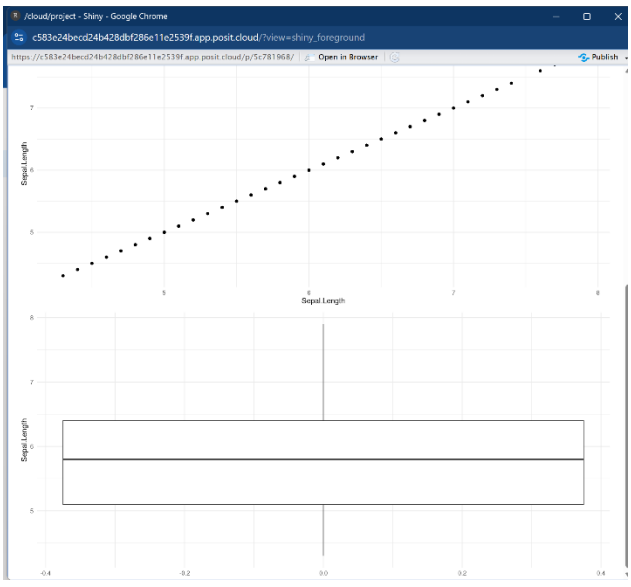
- **shinyApp(ui, server)**: Starts the app, showing the user interface and plots based on the server-side code.

Rcode:

```
library(shiny)
library(ggplot2)

# Read the iris dataset
data(iris)

ui <- fluidPage(
  selectInput("x", "X:", names(iris)),
  selectInput("y", "Y:", names(iris)),
  selectInput("box", "Boxplot:", names(iris)),
  verbatimTextOutput("sum"),
  plotOutput("scat"),
  plotOutput("boxp")
)

server <- function(input, output) {
  output$sum <- renderPrint({ summary(iris) })
  output$scat <- renderPlot({ ggplot(iris, aes_string(x = input$x, y = input$y)) + geom_point() + theme_minimal() })
  output$boxp <- renderPlot({ ggplot(iris, aes_string(y = input$box)) + geom_boxplot() + theme_minimal() })
}

shinyApp(ui, server)
```

Output:





5.Analyze outliers in the mtcars dataset using IQR and Z-score methods. Compute probabilities of selecting outlier cars based on mpg, hp, and wt. Explore conditional probabilities, independence, and expected outliers. Examine distribution shapes, skewness, correlations, and compare different outlier detection methods to assess consistency and insights.

Rcode:

```r
# Load libraries
library(ggplot2)
library(e1071)

# Load data
data <- mtcars

# IQR Outliers for mpg
q1 <- quantile(data$mpg, 0.25)
q3 <- quantile(data$mpg, 0.75)
iqr <- q3 - q1
outliers_iqr <- data$mpg[data$mpg < (q1 - 1.5 * iqr) | data$mpg > (q3 + 1.5 * iqr)]
print(outliers_iqr)

# Z-score Outliers for mpg
mean_mpg <- mean(data$mpg)
sd_mpg <- sd(data$mpg)
z_scores <- (data$mpg - mean_mpg) / sd_mpg
outliers_z <- data$mpg[abs(z_scores) > 2]
print(outliers_z)

# Probability of being an outlier
prob_iqr <- length(outliers_iqr) / nrow(data)
prob_z <- length(outliers_z) / nrow(data)
print(prob_iqr)
print(prob_z)

# Conditional Probability: P(hp > 150 | mpg < 20)
cond <- sum(data$hp > 150 & data$mpg < 20) / sum(data$mpg < 20)
print(cond)

# Skewness
skew_mpg <- skewness(data$mpg)
skew_hp <- skewness(data$hp)
skew_wt <- skewness(data$wt)
print(skew_mpg)
print(skew_hp)
print(skew_wt)
```

```r
# Correlations
cor_mpg_hp <- cor(data$mpg, data$hp)
cor_mpg_wt <- cor(data$mpg, data$wt)
print(cor_mpg_hp)
print(cor_mpg_wt)

# ggplot2: Boxplot of mpg
ggplot(data, aes(y = mpg)) +
  geom_boxplot(fill = "lightblue") +
  ggtitle("Boxplot of MPG")

# ggplot2: Histogram of mpg
ggplot(data, aes(x = mpg)) +
  geom_histogram(fill = "lightgreen", bins = 10, color = "black") +
  ggtitle("Histogram of MPG")
```

Output:

```
> # Print results
> print(outliers_iqr)
[1] 33.9
> print(outliers_z)
[1] 32.4 33.9
> print(prob_iqr)
[1] 0.03125
> print(prob_z)
[1] 0.0625
> print(cond)
[1] 0.7222222
> print(skew_mpg)
[1] 0.610655
> print(skew_hp)
[1] 0.7260237
> print(skew_wt)
[1] 0.4231465
> print(cor_mpg_hp)
[1] -0.7761684
> print(cor_mpg_wt)
[1] -0.8676594
```