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().**

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

# Explore data
str(titanic)
summary(titanic)
head(titanic)

# Handle missing values
titanic$Age[is.na(titanic$Age)] <- mean(titanic$Age, na.rm = TRUE)
titanic$Embarked[is.na(titanic$Embarked)] <- "S"

# Convert to factors
titanic$Sex <- as.factor(titanic$Sex)
titanic$Embarked <- as.factor(titanic$Embarked)

# Scale numerical features
titanic$Fare <- scale(titanic$Fare)
titanic$Age <- scale(titanic$Age)

# Engineer new feature: FamilySize
titanic$FamilySize <- titanic$SibSp + titanic$Parch + 1

# Save preprocessed data
write.csv(titanic, "titanic_cleaned.csv", row.names = FALSE)
```

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.**

```
install.packages("esquisse")
library(esquisse)

# Load mtcars
data("mtcars")

# Launch Esquisse GUI
```

```
esquisse::esquisser()
```

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.**

```
library(ggplot2)
library(vioplot)
data("mtcars")

# Boxplot
boxplot(mtcars$mpg, main="MPG Boxplot")

# Histogram
hist(mtcars$hp, main="HP Histogram", col="lightblue")

# Violin plot
vioplot::vioplot(mtcars$wt, names = "Weight")

# Scatter plot with regression line
ggplot(mtcars, aes(x=mpg, y=wt)) +
  geom_point() + geom_smooth(method="lm")

# Correlation heatmap
heatmap(cor(mtcars), main="Correlation Heatmap")
```

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.**

```
library(shiny)
data(iris)

ui <- fluidPage(
  titlePanel("Iris Shiny App"),
  selectInput("num", "Numerical Variable:", choices=names(iris)[1:4]),
  selectInput("cat", "Categorical Variable:", choices=c("Species")),
  verbatimTextOutput("summary"),
  selectInput("x", "X-axis Variable:", choices=names(iris)[1:4]),
  selectInput("y", "Y-axis Variable:", choices=names(iris)[1:4]),
  plotOutput("scatter"),
  plotOutput("boxplot")
)
```

```
server <- function(input, output) {
  output$summary <- renderPrint({
    summary(iris[[input$num]])
  })

  output$scatter <- renderPlot({
    plot(iris[[input$x]], iris[[input$y]], col=iris$Species, pch=19)
  })

  output$boxplot <- renderPlot({
    boxplot(iris[[input$num]] ~ iris[[input$cat]], main="Boxplot")
  })
}

shinyApp(ui = ui, server = server)
```

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.**

```
library(e1071)
data("mtcars")

# IQR method
Q1 <- quantile(mtcars$mpg, 0.25)
Q3 <- quantile(mtcars$mpg, 0.75)
IQR_val <- Q3 - Q1
outliers_iqr <- which(mtcars$mpg < (Q1 - 1.5 * IQR_val) | mtcars$mpg > (Q3 + 1.5 *
IQR_val))

# Z-score method
z_scores <- scale(mtcars$mpg)
outliers_z <- which(abs(z_scores) > 3)

# Print outliers
mtcars[outliers_iqr, ]
mtcars[outliers_z, ]

# Skewness
skewness(mtcars$mpg)
```

6. **Perform correlation analysis on the mtcars dataset to evaluate relationships between mpg and other features using Pearson and Spearman methods. Identify key influencing variables and visualize insights using heatmaps, scatter plots, box plots, bar plots, and clustering dendrograms, ensuring a comprehensive understanding of mpg dependencies and trends.**
data("mtcars")

```
data("mtcars")

# Pearson and Spearman
cor(mtcars$mpg, mtcars$hp, method = "pearson")
cor(mtcars$mpg, mtcars$hp, method = "spearman")

# Correlation heatmap
heatmap(cor(mtcars), main = "Correlation Heatmap")

# Dendrogram clustering
dist_m <- dist(mtcars)
hcl <- hclust(dist_m)
plot(hcl)
```

7. **Analyze sales trends over time using scatter plots and regression models. Fit linear and quadratic regression, compare their R-squared and RMSE values, and interpret results. Identify the best-fitting model and predict sales for the next 6 months (Months: 61-66) to assess future trends and decision-making insights.**

```
set.seed(123)
months <- 1:60
sales <- 100 + 2*months + rnorm(60, 0, 10)
data <- data.frame(months, sales)

# Linear regression
model1 <- lm(sales ~ months, data)
summary(model1)

# Quadratic regression
model2 <- lm(sales ~ months + I(months^2), data)
summary(model2)

# Predict next 6 months
future <- data.frame(months = 61:66)
predict(model2, newdata = future)
```

8. **A medical researcher is studying the relationship between various blood test parameters, such as glucose levels, cholesterol, and blood pressure, to identify potential risk factors for diabetes. Compute the Pearson and Spearman correlation coefficients for a given dataset and analyze the relationships between variables. Explain how correlation matrices assist in feature selection and why highly correlated features might lead to redundancy in predictive models. Using linear algebra, describe how the correlation matrix can be decomposed using eigenvalues and eigenvectors to understand the data structure.**

```
# Simulated data
df <- data.frame(
  glucose = rnorm(100, 100, 15),
  cholesterol = rnorm(100, 200, 25),
  bp = rnorm(100, 120, 10)
)

# Correlation
cor(df, method = "pearson")
cor(df, method = "spearman")

# Eigen decomposition
e <- eigen(cor(df))
e$values
e$vectors
```

9. **Analyze the Heart Disease dataset by exploring its structure, target variable, and missing values. Visualize correlations, split data, and train a Logistic Regression model. Evaluate using accuracy, precision, recall, and F1-score. Apply PCA for dimensionality reduction and assess if performance improves for better heart disease prediction.**

```
heart <- read.csv("heart.csv")

# Train-test split
set.seed(123)
train_idx <- sample(1:nrow(heart), 0.7 * nrow(heart))
train <- heart[train_idx, ]
test <- heart[-train_idx, ]

# Logistic regression
model <- glm(target ~ ., data=train, family="binomial")
pred <- predict(model, newdata=test, type="response")
pred_class <- ifelse(pred > 0.5, 1, 0)

# Evaluation
```

```
conf_matrix <- table(Predicted=pred_class, Actual=test$target)
accuracy <- sum(diag(conf_matrix)) / sum(conf_matrix)
accuracy

# PCA
pca <- prcomp(train[,-ncol(train)], scale. = TRUE)
summary(pca)
```

10. **A real estate company aims to predict house prices based on factors such as square footage, number of bedrooms, and location. Formulate the problem as a multiple linear regression equation and express it in matrix form. Using the normal equation, calculate the regression coefficients to fit the model. Discuss how multicollinearity among predictor variables affects the reliability of the regression model and explain how Principal Component Regression (PCR) can be used to address multicollinearity issues while improving predictive accuracy.**

```
# Sample data
data <- data.frame(
  sqft = c(1000, 1500, 2000),
  bedrooms = c(2, 3, 4),
  location = c(1, 2, 3), # encoded location
  price = c(200000, 300000, 400000)
)

# Design matrix X (with bias term)
X <- as.matrix(cbind(1, data$sqft, data$bedrooms, data$location))
y <- as.matrix(data$price)

# Normal equation
beta <- solve(t(X) %*% X) %*% t(X) %*% y
beta

# PCR to fix multicollinearity
library(pls)
model <- pcr(price ~ ., data=data, scale=TRUE, validation="CV")
summary(model)
```