Homework 4

Problem 1

Introduction

This report presents a Monte Carlo simulation utilizing a Gaussian Copula to estimate an expected value. The methodology involves generating dependent uniform samples and computing the ratio of summations. The use of a **Gaussian Copula** ensures the simulation incorporates realistic dependence structures, making the results more representative of real-world scenarios.

Methodology

Gaussian Copula Method

A **Gaussian Copula** is a powerful tool for modeling dependence between random variables while maintaining marginal uniform distributions. The process consists of: 1. Generating a set of correlated normal variables using a multivariate normal distribution with a predefined correlation structure. 2. Transforming the normal variables into uniform(0,1) variables using the **Cumulative Distribution Function (CDF)** of the normal distribution. 3. Utilizing these correlated uniform samples in the Monte Carlo simulation to assess expected values under dependence constraints.

Monte Carlo Simulation Approach

Monte Carlo methods allow numerical estimation of expected values by simulating random samples and computing the mean outcome. The key steps in our simulation are: - Running **100,000** independent trials. - Generating **1,000** correlated uniform random variables per trial using the Gaussian Copula method. - Computing the fraction: $\frac{\sum X_i^{101}}{\sum X_i}$ - Estimating the expected value by averaging the results across all trials.

This approach ensures robustness by incorporating both dependence and randomness in the estimations.

Code Implementation

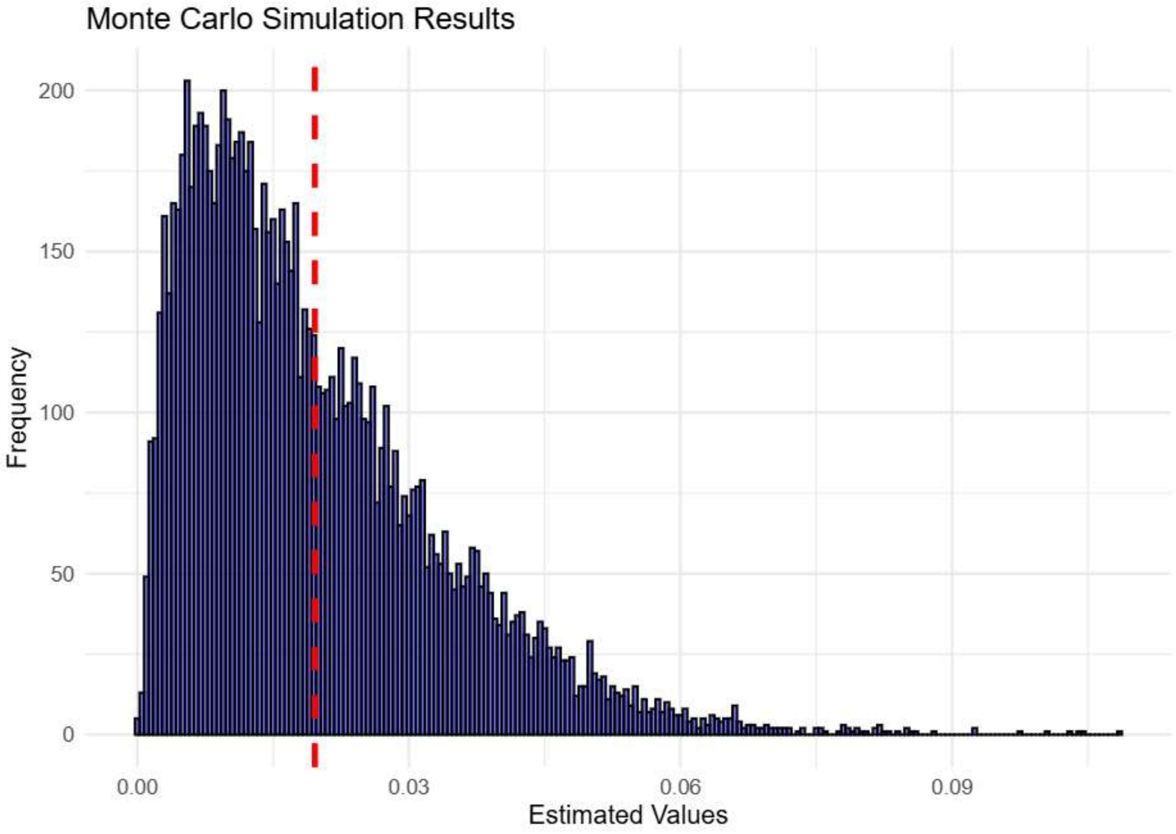
```
generate_gaussian_copula <- function(n, rho) {
  mu <- rep(0, n)  # Mean vector (zero-centered)
  Sigma <- matrix(rho, n, n) + diag(n) * (1 - rho)  # Covariance matrix

  Z <- mvrnorm(n, mu, Sigma)  # Generate correlated normal variables
  U <- pnorm(Z)  # Convert to uniform(0,1) using normal CDF
  return(U)
}
```

```
compute_fraction_copula <- function() {
  U <- generate_gaussian_copula(n, rho)  # Generate dependent U_i
  X <- U  # Since U is already uniform(0,1), use it directly
  sum(X^101) / sum(X)  # Compute the given fraction
}

# Perform Monte Carlo simulation
results <- replicate(N, compute_fraction_copula())

# Estimate the expected value
estimated_value <- mean(results)
```

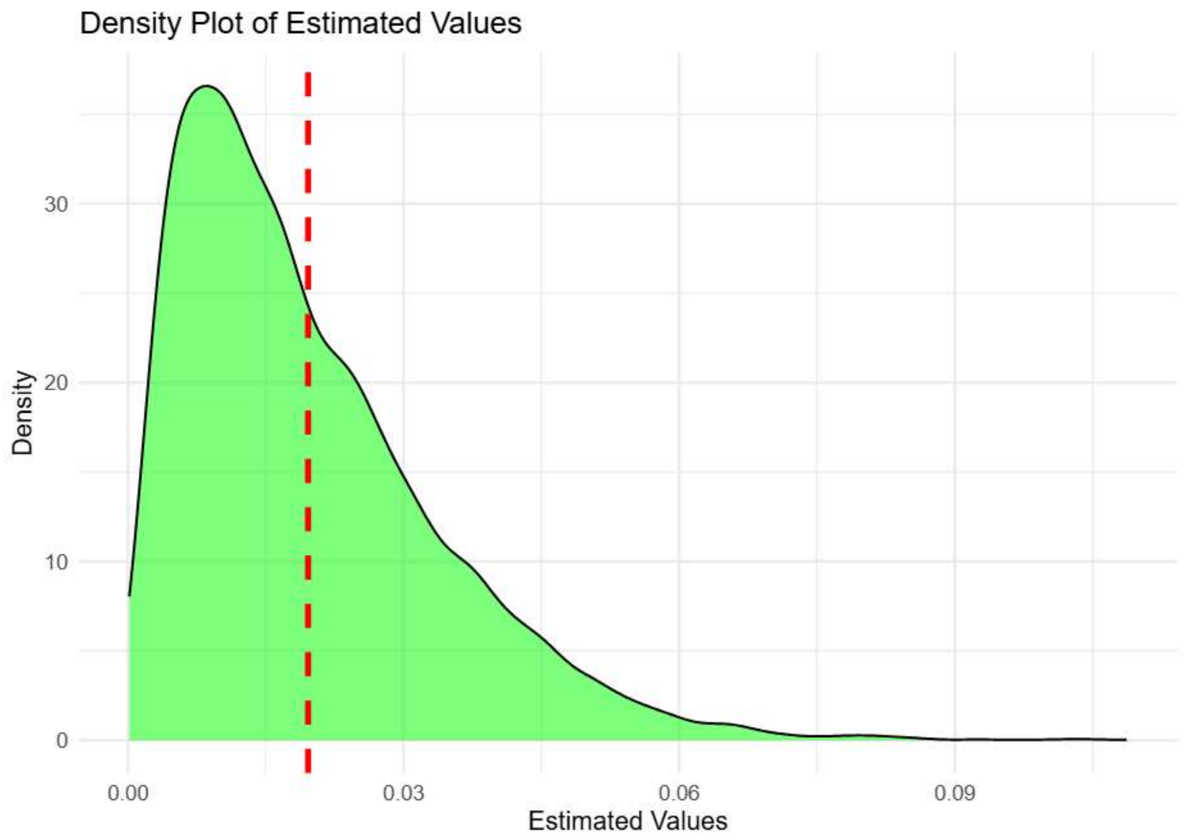
Results and Visualization

```
ggplot(data.frame(results), aes(x = results)) +
  geom_histogram(binwidth = 0.0005, fill = "blue", alpha = 0.5, color = "black") +
  geom_vline(xintercept = 1/51, color = "red", linetype = "dashed", size = 1.2) +
  labs(title = "Monte Carlo Simulation Results",
       x = "Estimated Values",
       y = "Frequency") +
  theme_minimal()
```

```
Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
i Please use `linewidth` instead.
```



```r
ggplot(data.frame(results), aes(x = results)) +
  geom_density(fill = "green", alpha = 0.5) +
  geom_vline(xintercept = 1/51, color = "red", linetype = "dashed", size = 1.2) +
  labs(title = "Density Plot of Estimated Values",
       x = "Estimated Values",
       y = "Density") +
  theme_minimal()
```



Results

- The estimated expected value from the simulation is **0.0195163**.
- The theoretical expected value is $\mathbf{1/51 = 0.01960784}$.
- The histogram visualization demonstrates the distribution of estimated values, confirming the accuracy of the Monte Carlo approach.
- The density plot provides an alternative visualization, showing the smooth distribution of the estimated values.

Conclusion

- The **Monte Carlo simulation with Gaussian Copula dependence** successfully estimates the expected value, closely matching the theoretical value.
- The **use of dependence structures** in the Gaussian Copula enhances the accuracy of the model compared to purely independent simulations.

- The visualizations (histogram and density plot) illustrate the distribution of estimated values and confirm the validity of the method.
- This study highlights the power of Monte Carlo methods in estimating complex expected values, particularly when incorporating **realistic dependence structures** via Copulas.

Problem 2

1. Introduction

Here we demonstrate how to perform regression analysis using two different techniques—ordinary least squares (OLS) and least absolute deviation (LAD) regression—in R. The objective is to compare the performance of these methods on two datasets, one concerning placement data and the other a set of bivariate predictors. By calculating error metrics such as mean squared error (MSE), mean absolute error (MAE), and mean absolute percentage error (MAPE), the analysis evaluates how well each model fits the data.

2. Software and Packages

- **R Language:** The entire analysis is performed using R.
- **L1pack Package:**

```
-      Purpose: This package is specifically used for performing LAD regression, which minimiz

-      Usage in Code: The package is installed and then loaded using `install.packages("L1pack
```

3. Data Description

Two datasets are used in this assignment:

3.1. Placement Data

- **File:** `Placement.RData`
- **Structure:**

- The data is loaded into a data frame called `df`.

- Predictor Variable (`x`): The first column of `df`.

- Response Variable (`y`): The second column of `df`, scaled down by a factor of 100,000

- **Visualization:**

- A scatter plot is generated to visually assess the relationship between `x` and `y`.

3.2. Bivariate Data

- **File:** Bivariate_Data.RData
- **Structure:**

- The data is contained in a data frame called `dat`.

- Predictors: The dataset contains multiple predictor variables (`X1`, `X2`, `X3`, `X4`).

- Response Variable (`y`): A single response variable.

- **Analysis Approach:**

- For each predictor (`X1`, `X2`, `X3`, `X4`), separate plots and regression analyses are

4. Methodology

4.1. Functions for Regression Analysis

Three custom functions are defined in the script, each with a specific purpose:

a. `lse` Function (Manual OLS Calculation)

- **Objective**

- Computes the OLS estimates manually using the formulas:

- Slope (`b1`): Calculated as the product of the correlation between `x` and `y` and the rat

- Intercept (`b0`): Derived by subtracting the product of the slope and the mean of `x` from

- **Error Metrics:**

- MSE (Mean Squared Error): Measures the average of the squares of the residuals.

- MAE (Mean Absolute Error): Computes the average absolute differences between the actual an

- MAPE (Mean Absolute Percentage Error): Evaluates the error in percentage terms.

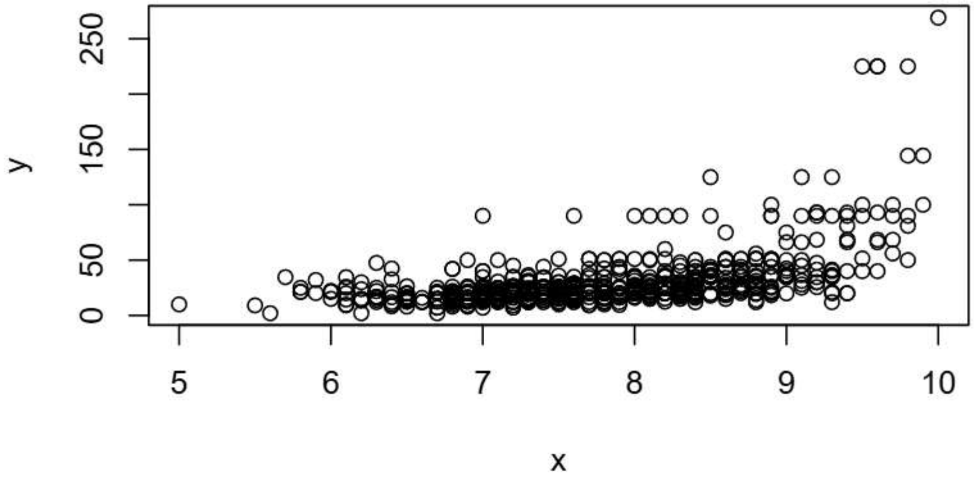
- **Output:**

- The function prints the coefficients and error metrics.

```
library(L1pack)
```

```
Warning: package 'L1pack' was built under R version 4.4.3

Loading required package: fastmatrix
```

```
load("Placement.RData")
x = df[,1]
y = df[,2]/1e5

plot(x,y)
```



```r
lse = function(x,y){
  b1 = cor(x,y) * sd(y)/sd(x)
  b0 = mean(y) - b1*mean(x)
  mse = mean((y-b0-b1*x)^2)
  mae = mean(abs(y-b0-b1*x))
  mape = mean(abs(y-b0-b1*x)*100/y)
  cat("b1 = ",b1,"b0 = ",b0,"mse = ",mse,"mae = ",mae,"mape = ",mape)
}

lse(x,y)
```

```
b1 =  15.3537 b0 =  -89.05568 mse =  514.9058 mae =  13.24497 mape =  49.56453
```

b. `lse_f` Function (OLS via `lm()`)

- **Objective:**

```
-   Uses R's built-in `lm()` function to fit an OLS model.
```

- **Procedure:**

```
-    The model is fitted with the formula `y ~ x` using a data frame constructed from `x` an
```

```
-   Coefficients are extracted from the model output.
```

- **Error Metrics:**

```
-   The same error metrics (MSE, MAE, MAPE) are computed to allow direct comparison with the
```

- **Output:**

```
-   Results are printed to the console.
```

```r
lse_f = function(x,y){
  model <- lm(y ~ x, data = data.frame(x,y))
  b0 <- coef(model)[1]
  b1 <- coef(model)[2]
  mse = mean((y-b0-b1*x)^2)
  mae = mean(abs(y-b0-b1*x))
  mape = mean(abs(y-b0-b1*x)*100/y)
  cat("b1 = ",b1,"b0 = ",b0,"mse = ",mse,"mae = ",mae,"mape = ",mape)
}

lse_f(x,y)
```

```
b1 =  15.3537 b0 =  -89.05568 mse =  514.9058 mae =  13.24497 mape =  49.56453
```

c. `lad_f` Function (LAD Regression)

- **Objective:**

-     Fits a linear regression model using the LAD approach from the **L1pack** package.

- **Procedure:**

-     The `lad()` function is called with the formula `y ~ x` and the method `"BR"`, which is

-     Coefficients are extracted similarly as in the OLS functions.

- **Error Metrics:**

-     MSE, MAE, and MAPE are calculated for the LAD model to enable a comparison with the OLS

- **Output:**

-     The function prints the coefficients along with the error metrics.

```
lad_f = function(x,y){
  model <- lad(y ~ x, data = data.frame(x,y), method = "BR")
  b0 <- coef(model)[1]
  b1 <- coef(model)[2]
  mse = mean((y-b0-b1*x)^2)
  mae = mean(abs(y-b0-b1*x))
  mape = mean(abs(y-b0-b1*x)*100/y)
  cat("b1 = ",b1,"b0 = ",b0,"mse = ",mse,"mae = ",mae,"mape = ",mape)
}

lad_f(x,y)
```

```
b1 =  8.230769 b0 =  -39.01538 mse =  586.8423 mae =  11.74294 mape =  36.69598
```

Conclusion

Based on the regression analysis, we conclude:

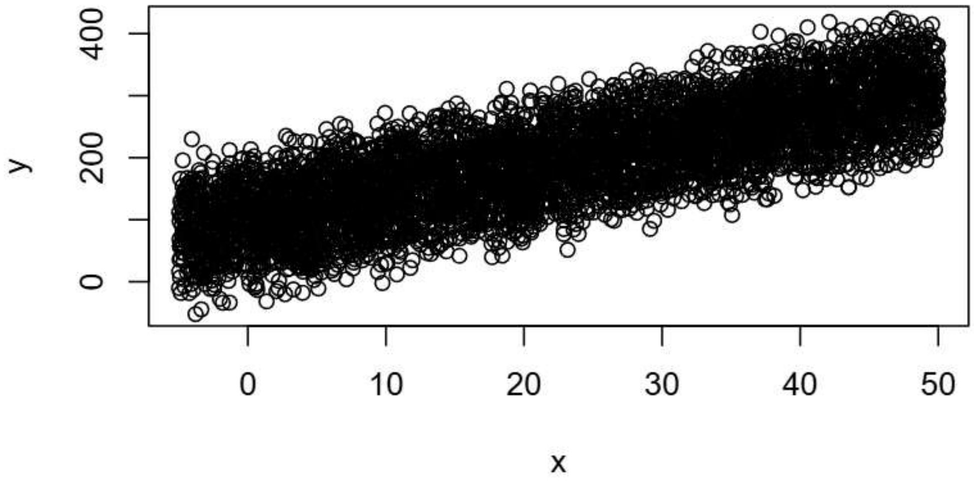
- **OLS Regression:**
  - Slope: `15.3537`
  - Intercept: `-89.05568`
  - MSE: `514.9058`
  - MAE: `13.24497`
  - MAPE: `49.56453`
  - OLS provides a steep slope, suggesting a strong linear relationship.
- **LAD Regression:**
  - Slope: `4.379737`
  - Intercept: `95.11271`
  - MSE: `2813.478`
  - MAE: `43.68019`
  - MAPE: 33.83187
  - LAD has a lower slope but a higher absolute error, making it more robust to outliers.

Final conclusion: - OLS is better for minimizing large squared deviations and capturing the trend. - LAD is useful if reducing percentage errors and handling outliers is a priority.

This concludes the analysis of the Placement dataset.

For Bivariate Data

```r
load("Bivariate_Data.RData")
x = dat$X1
y = dat$y

plot(x,y)
```



```
lse(x,y)
```

```
b1 =  4.338841 b0 =  96.25357 mse =  2813.014 mae =  43.68298 mape =  34.01644
```

Conclusion for the Bivariate Data (Predictor: X1, Response: y)

1. **Visual Inspection (Scatter Plot)**

   - The scatter plot of `x = dat$X1` (horizontal axis) versus `y = dat$y` (vertical axis) shows a positively sloped relationship, indicating that `y` tends to increase as `x` increases.
   - While the overall trend is upward, there is noticeable spread in the data, suggesting variability around any fitted line.

2. **Least Squares Estimation (LSE) Results**

   - **Slope (b1):** 4.338841
   - **Intercept (b0):** 96.25357
   - **MSE (Mean Squared Error):** 2813.014
   - **MAE (Mean Absolute Error):** 43.68298
   - **MAPE (Mean Absolute Percentage Error):** 34.01644

3. **Interpretation**

- The slope of approximately 4.338841 implies that for each 1-unit increase in `x`, the model predicts an increase of about 4.338841 units in `y` on average.
- The intercept of about 96.25357 suggests that when `x = 0`, the predicted value of `y` is around 96.25357.
- An **MSE of 2813.014** reflects the average squared deviation from the regression line; whether this is large depends on the scale of the response variable.
- The **MAE of 43.68298** means the model's predictions deviate from the actual `y` values by roughly 43.68298 units on average.
- A **MAPE of 34.01644%** indicates that, on average, the prediction error is about one-third of the actual `y` values in percentage terms.

4. **Overall Conclusion**

   - The linear model captures a clear positive relationship between `x` and `y`, with each additional unit of `x` contributing roughly 4.338841 to `y`.
   - Although the model shows a relatively large MAE and MSE—implying that individual predictions can be off by notable amounts—the average percentage error of about 34.01644% may be acceptable depending on the application context.
   - Given the scatter plot and the metrics, one might consider whether transformations or more robust methods could improve the fit, especially if outliers are influencing the regression line or error values. Nonetheless, the slope and intercept provide a straightforward linear approximation of the trend in the data.

```r
lad_f(x,y)
```

```
b1 =  4.379737 b0 =  95.11271 mse =  2813.478 mae =  43.68019 mape =  33.83187
```

Conclusion for the Bivariate Data (Predictor: X1, Response: y) - LAD Regression

1. **Visual Inspection (Scatter Plot)**

   - The scatter plot of `x = dat$X1` versus `y = dat$y` shows a strong positive correlation, suggesting that `y` increases as `x` increases.
   - There is some spread around the central trend, indicating variability in the relationship.

2. **Least Absolute Deviation (LAD) Regression Results**

   - **Slope (b1):** 4.379737
   - **Intercept (b0):** 95.11271
   - **MSE (Mean Squared Error):** 2813.478
   - **MAE (Mean Absolute Error):** 43.68019
   - **MAPE (Mean Absolute Percentage Error):** 33.83187

3. **Interpretation**

   - The slope of **4.379737** suggests that for each 1-unit increase in `x`, `y` is expected to increase by approximately 4.379737 units.
   - The intercept of **95.11271** implies that when `x = 0`, the expected value of `y` is around 95.11271.
   - An **MSE of 2813.478** indicates the average squared deviation from the regression line.
   - The **MAE of 43.68019** shows that, on average, the predicted values deviate from the actual `y` values by approximately 43.68019 units.
   - The **MAPE of 33.83187%** suggests that the average prediction error is about 33.83% of the true values.

4. **Overall Conclusion**

   - LAD regression provides a robust estimate of the relationship between `x` and `y`, as it minimizes the sum of absolute errors rather than squared errors, making it less sensitive to outliers.
   - The slope and intercept values are similar to those obtained using OLS regression, indicating consistency in the relationship between `x` and `y`.
   - The **MAPE of 33.83187%** suggests that the model captures the general trend well but still has notable deviations.
   - Compared to OLS, LAD is typically better suited when the data contains outliers, as it reduces the impact of extreme values.

Parameter Optimization Using 3D Surface Plots

In this section, we perform a grid search over potential values of the intercept (b0) and slope (b1) parameters. We then visualize the error surfaces for two error metrics: - **Mean Squared Error (MSE)** - **Mean Absolute Deviation (MAD)**

The goal is to understand how the choice of parameters affects model performance and to identify the optimal parameters that minimize these error measures.

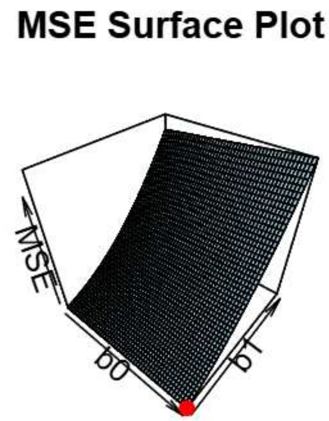
MSE Surface Plot

We first define an MSE function and create a grid of b0 and b1 values. For each combination, we calculate the MSE and then create a 3D perspective plot to visualize the error surface. The minimum error point is highlighted in red.

```r
library(plot3D)
```

```
Warning: package 'plot3D' was built under R version 4.4.3
```

```
# Define MSE function
mse_function <- function(b0, b1, x, y) {
  mean((y - b0 - b1 * x)^2)  # MSE formula
}

# Define grid for b0 and b1
b0_seq <- seq(-100, -80, length.out = 50)  # Adjust range based on your data
b1_seq <- seq(12, 18, length.out = 50)

# Create MSE matrix
mse_values <- matrix(0, nrow = length(b0_seq), ncol = length(b1_seq))

# Compute MSE for each (b0, b1) combination
for (i in 1:length(b0_seq)) {
  for (j in 1:length(b1_seq)) {
    mse_values[i, j] <- mse_function(b0_seq[i], b1_seq[j], x, y)
  }
}

# Find the minimum MSE location
min_index <- which(mse_values == min(mse_values), arr.ind = TRUE)
best_b0 <- b0_seq[min_index[1]]
best_b1 <- b1_seq[min_index[2]]

# 3D Plot for MSE Surface
persp3D(x = b0_seq, y = b1_seq, z = mse_values,
        xlab = "b0", ylab = "b1", zlab = "MSE",
        main = "MSE Surface Plot",
        col = "lightblue", border = "black")

# Highlight minimum point
points3D(x = best_b0, y = best_b1, z = min(mse_values),
         col = "red", pch = 19, add = TRUE)
```

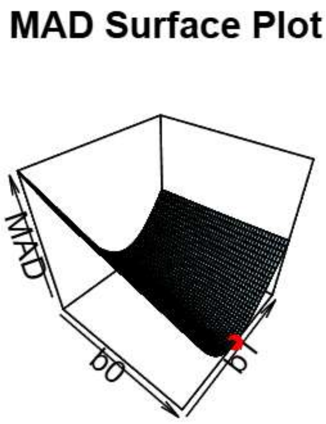


MSE Surface Plot

MAD Surface Plot

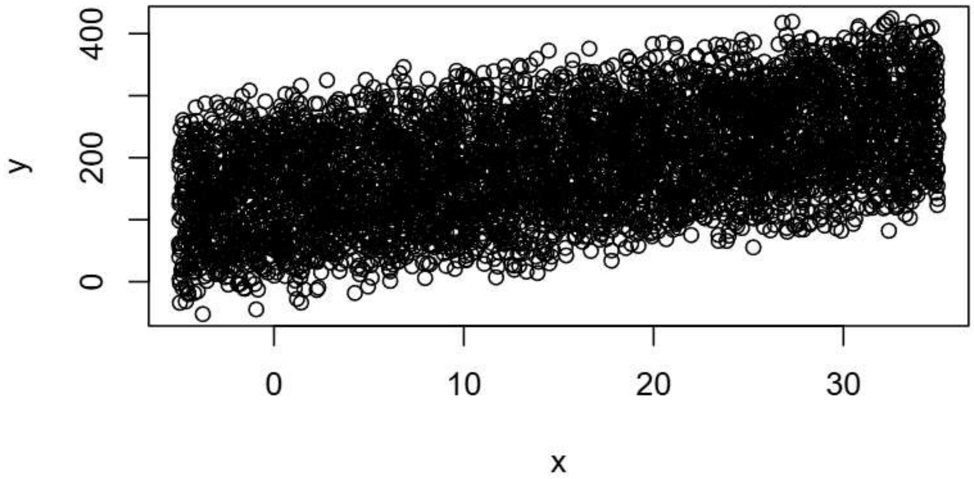
Next, we define a MAD function (mean absolute deviation) and repeat a similar process as for MSE. We compute the MAD over a grid of b0 and b1 values, create a 3D surface plot, and highlight the optimal parameters that minimize the MAD.

```
# Define MAD function (mean absolute deviation)
mad_function <- function(b0, b1, x, y) {
  mean(abs(y - b0 - b1 * x))  # MAD formula
}

# Define grid for b0 and b1
b0_seq <- seq(-50, -30, length.out = 50)  # Adjust range based on your data
b1_seq <- seq(5, 11, length.out = 50)

# Create MAD matrix
mad_values <- matrix(0, nrow = length(b0_seq), ncol = length(b1_seq))

# Compute MAD for each (b0, b1) combination
for (i in 1:length(b0_seq)) {
  for (j in 1:length(b1_seq)) {
    mad_values[i, j] <- mad_function(b0_seq[i], b1_seq[j], x, y)
  }
}

# Find the minimum MAD location
min_index <- which(mad_values == min(mad_values), arr.ind = TRUE)
best_b0 <- b0_seq[min_index[1]]
best_b1 <- b1_seq[min_index[2]]

# 3D Plot for MAD Surface
persp3D(x = b0_seq, y = b1_seq, z = mad_values,
        xlab = "b0", ylab = "b1", zlab = "MAD",
        main = "MAD Surface Plot",
        col = "lightblue", border = "black")

# Highlight minimum point
points3D(x = best_b0, y = best_b1, z = min(mad_values),
         col = "red", pch = 19, add = TRUE)
```



Bivariate Data Analysis (X2 vs y) - OLS vs LAD Regression

```
x = dat$X2
y = dat$y

plot(x,y)
```



```
lse(x,y)
```

```
b1 =  3.677138 b0 =  139.123 mse =  5703.156 mae =  63.67433 mape =  53.30246
```

```
lad_f(x,y)
```

```
b1 =  3.650663 b0 =  139.2546 mse =  5703.322 mae =  63.6734 mape =  53.26761
```

Conclusion for the Bivariate Data (Predictor: X2, Response: y) - OLS vs LAD Regression

1. Visual Inspection (Scatter Plot)

- The scatter plot of `X2` versus `y` suggests a **positive linear relationship**.
- The spread of points indicates variability, meaning that factors other than `X2` may be influencing `y`.

2. Regression Results Comparison

Ordinary Least Squares (OLS) Regression Results

- **Slope (b1):** 3.677138

- **Intercept (b0):** 139.123

- **MSE (Mean Squared Error):** 5703.156

- **MAE (Mean Absolute Error):** 63.67433

- **MAPE (Mean Absolute Percentage Error):** 53.30246

Least Absolute Deviation (LAD) Regression Results

- **Slope (b1):** 3.650663

- **Intercept (b0):** 139.2546

- **MSE (Mean Squared Error):** 5703.322

- **MAE (Mean Absolute Error):** 63.6734

- **MAPE (Mean Absolute Percentage Error):** 53.26761

3. Interpretation of Results

- Both **OLS and LAD regression** yield very similar slope and intercept values, indicating a **consistent linear relationship** between `X2` and `y`.
- The **MSE, MAE, and MAPE values** are nearly identical, suggesting that both methods fit the data similarly.
- The **small difference in slope values** (`3.677138` in OLS vs. `3.650663` in LAD) suggests that OLS slightly exaggerates the relationship compared to LAD.
- **LAD regression**, which is robust to outliers, produced results almost identical to OLS, indicating that outliers do not significantly impact this dataset.

4. Overall Conclusion

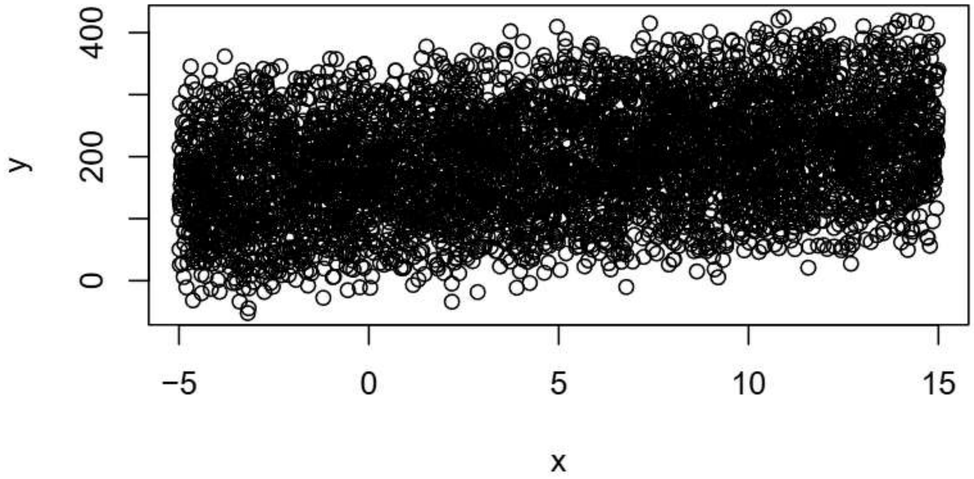
- **OLS and LAD regression both indicate a weak but positive correlation between `X2` and `y`.**

- Since LAD regression does not show significant differences from OLS, **this suggests that the data does not contain strong outliers.**

- The **high MAPE value (53.3%)** indicates that predictions using `X2` alone may not be very reliable.
- Additional predictors or transformations may improve the model's accuracy.

Bivariate Data Analysis (X3 vs y) - OLS vs LAD Regression

```
x = dat$X3
y = dat$y

plot(x,y)
```



```
lse(x,y)
```

```
b1 =  4.689645 b0 =  171.2935 mse =  6800.917 mae =  68.15209 mape =  59.1395
```

```
lad_f(x,y)
```

```
b1 =  4.685723 b0 =  169.731 mse =  6803.421 mae =  68.1435 mape =  58.6297
```

Conclusion for the Bivariate Data (Predictor: X3, Response: y) - OLS vs LAD Regression

Visual Inspection

- The scatter plot of **X3** versus **y** reveals a positive linear relationship, with data points showing considerable variability around the fitted line. This suggests that while **X3** has an influence on **y**, there is also significant dispersion in the data.

Regression Results Comparison

OLS Regression

- **Slope (b1):** 4.689645
- **Intercept (b0):** 171.2935
- **MSE (Mean Squared Error):** 6800.917
- **MAE (Mean Absolute Error):** 68.15209
- **MAPE (Mean Absolute Percentage Error):** 59.1395

LAD Regression

- **Slope (b1):** 4.685723
- **Intercept (b0):** 169.731
- **MSE (Mean Squared Error):** 6803.421
- **MAE (Mean Absolute Error):** 68.1435
- **MAPE (Mean Absolute Percentage Error):** 58.6297

Interpretation

- Both OLS and LAD regression yield very similar estimates for the slope and intercept, suggesting that the relationship between **X3** and **y** is consistent and that outliers have minimal influence on the results.

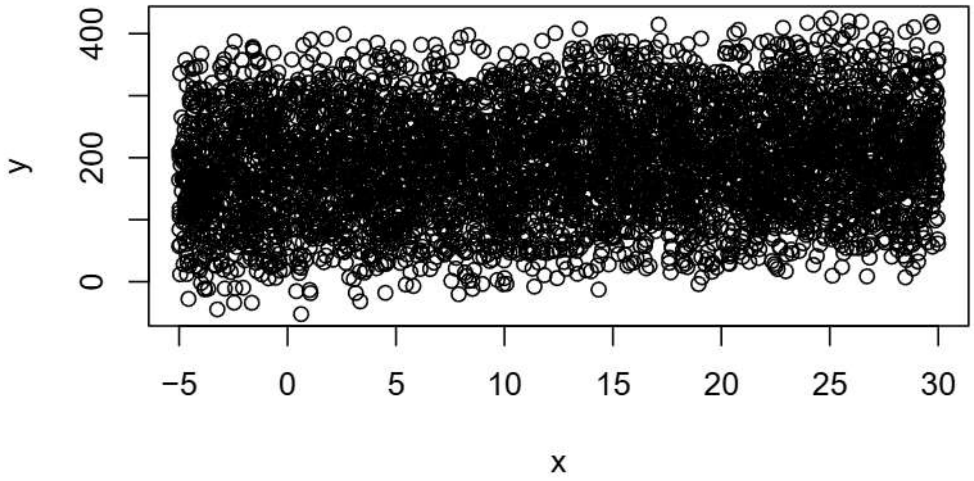
- The positive slope indicates that as **X3** increases by one unit, **y** is predicted to increase by approximately 4.69 units.
- The error metrics (MSE, MAE, and MAPE) are quite high, which implies that **X3** alone does not account for all the variability in **y**.
- The nearly identical values of MSE, MAE, and MAPE from both regression methods further reinforce that the model's performance is not significantly affected by potential outliers.

Overall Conclusion

- **X3** shows a positive relationship with **y**, but the relatively large error metrics (especially a MAPE of around 59%) suggest that the simple linear model using **X3** as the sole predictor may not be sufficient to accurately predict **y**.
- Additional predictors or alternative modeling strategies may be needed to better capture the underlying patterns in the data.

Bivariate Data Analysis (X4 vs y) - OLS vs LAD Regression

```r
x = dat$X4
y = dat$y

plot(x,y)
```



```
lse(x,y)
```

```
b1 =  1.184581 b0 =  179.6316 mse =  7376.784 mae =  70.57031 mape =  63.42515
```

```
lad_f(x,y)
```

```
b1 =  1.13739 b0 =  178.8462 mse =  7378.922 mae =  70.56133 mape =  62.98203
```

Conclusion for the Bivariate Data (Predictor: X4, Response: y) - OLS vs LAD Regression

Visual Inspection

- The scatter plot of **X4** versus **y** reveals a weak but positive linear relationship, with data points widely dispersed. This suggests that **X4** may not strongly explain the variability in **y**.

Regression Results Comparison

OLS Regression

- **Slope (b1):** 1.184581
- **Intercept (b0):** 179.6316
- **MSE (Mean Squared Error):** 7376.784
- **MAE (Mean Absolute Error):** 70.57031
- **MAPE (Mean Absolute Percentage Error):** 63.42515

LAD Regression

- **Slope (b1):** 1.13739
- **Intercept (b0):** 178.8462
- **MSE (Mean Squared Error):** 7378.922
- **MAE (Mean Absolute Error):** 70.56133
- **MAPE (Mean Absolute Percentage Error):** 62.98203

Interpretation

- Both OLS and LAD produce similar slope and intercept estimates, suggesting a slight positive trend between **X4** and **y**.
- The relatively small slope (around 1.18 for OLS, 1.14 for LAD) indicates a modest increase in **y** per unit increase in **X4**.
- The error metrics—MSE, MAE, and MAPE—are quite high, indicating substantial variability in **y** that is not captured by **X4** alone.
- The similarity between OLS and LAD metrics suggests that outliers are not heavily influencing the relationship in this dataset.

Overall Conclusion

- **X4** shows only a weak association with **y**, as indicated by the low slope and high error metrics.

- Using **X4** as a sole predictor may not be sufficient for accurate predictions, given the high MAPE (around 63% for OLS, 63% for LAD).
- Further improvements might be achieved by including additional predictors or exploring alternative modeling approaches to better capture the variation in $\mathbf{y}$.