HW4_Group10

Akshat Saxena, Maitreyee Katre, Rupant Dixit, Sujal Yadav, Tenzin Tsomu

Introduction

1. **Copula technique to compute given integral**

A **copula** is a function that **couples multivariate joint distributions to their one-dimensional margins**. In simpler terms:

> Copula allows us to separate the **dependence structure** of random variables from their **marginal distributions**.

By **Sklar's Theorem**, any joint distribution $F(x_1, ..., x_n)$ with marginals $F_i(x_i)$ can be written as:

$F(x_1, ..., x_n) = C(F_1(x_1), ..., F_n(x_n))$
Where C is the **copula**.

2. LSE and LAD Methods to estimate parameters of simple linear regression

Simple Linear Regression Recap

In simple linear regression, we model the relationship between a predictor x and a response variable y as:

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$$

where ε_i are error terms.

---

1. Least Squares Estimation (LSE)

Goal: Minimize the **sum of squared residuals**:

$$\text{LSE: } \min_{\beta_0,\beta_1} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2$$

Key Points:

- Efficient when errors are **normally distributed**.
- Sensitive to **outliers** — squaring large residuals exaggerates their effect.
- Has a **closed-form solution**.
- Works well when errors are **symmetric** and variance is **constant** (homoscedasticity).

2. Least Absolute Deviations (LAD)

Goal: Minimize the **sum of absolute residuals**:

$$\text{LAD: } \min_{\beta_0,\beta_1} \sum_{i=1}^{n} |y_i - \beta_0 - \beta_1 x_i|$$

Key Points:

- More **robust to outliers**.
- Does not exaggerate large deviations.
- No closed-form solution — requires **numerical optimization**.
- Useful when errors are **skewed** or **heavy-tailed**.

Comparison Table

Feature	LSE (Least Squares)	LAD (Least Absolute Deviations)
Objective	Minimize squared errors	Minimize absolute errors
Outlier Effect	Sensitive	Robust
Solution Form	Closed-form (Normal equations)	Requires numerical optimization
Error Assumption	Normal, symmetric	Skewed or heavy-tailed errors

Methodology

1. Using Copula to compute value of given integral

Step 1: Express Integral as Copula Expectation

Let each $X_i \sim \mathcal{U}(0,1)$. Their joint CDF is:

$$F(x_1,\ldots,x_n) = C(x_1,\ldots,x_n)$$

For independent marginals, the copula is:

$$C(u_1,\ldots,u_n) = u_1 u_2 \cdots u_n$$

So the density of the copula is:

$$c(u_1,\ldots,u_n) = 1$$

The integral becomes an expectation:

$$\mathbb{E}_C\left[\frac{U_1^{101}+\cdots+U_n^{101}}{U_1+\cdots+U_n}\right]$$

R Code and Interpretation

1. Estimate the limit:

$$\lim_{n\to\infty}\int_{[0,1]^n}\frac{x_1^{101}+x_2^{101}+\cdots+x_n^{101}}{x_1+x_2+\cdots+x_n}\,dx_1\cdots dx_n$$

using a **copula-based approach**.

First we make a function to simulate sample from copula

```
simulate_independence <- function(n, n_sim) {
  matrix(runif(n * n_sim), nrow = n_sim, ncol = n)
}
```

then we make a function to compute the given integrand using each sample.

```r
integrand_estimator <- function(u_matrix) {
  apply(u_matrix, 1, function(u) {
    numerator <- sum(u^101)
    denominator <- sum(u)
    numerator / denominator
  })
}
```

And finally the estimation is done as follows:

```r
# Parameters
n <- 1000       # number of variables
n_sim <- 10000  # number of simulations

# --- Estimate using Copula ---
u_indep <- simulate_independence(n, n_sim)
est_indep <- mean(integrand_estimator(u_indep))
```

which gives us the value as

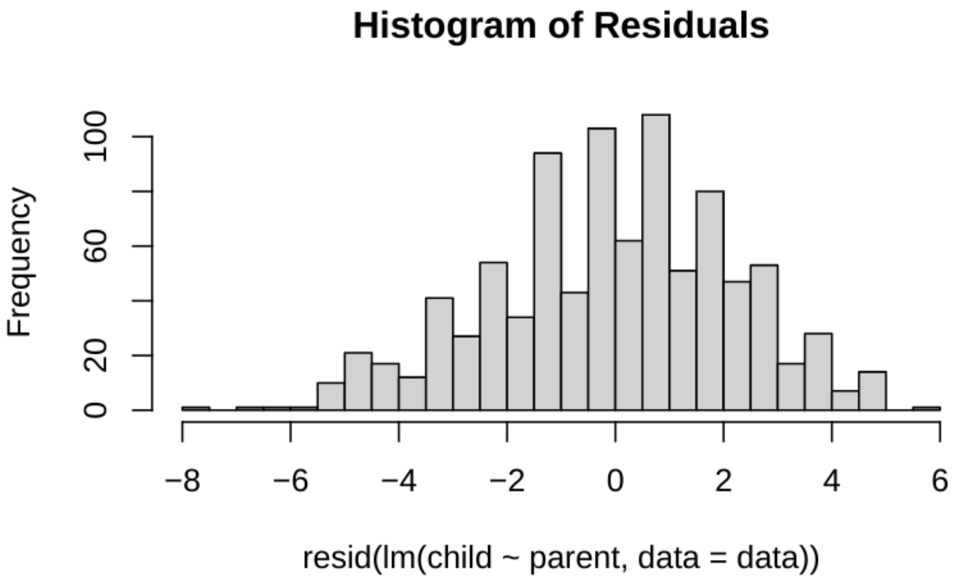
```r
cat("The estimate obtained for given limit is", est_indep)
```

```
The estimate obtained for given limit is 0.01956855
```

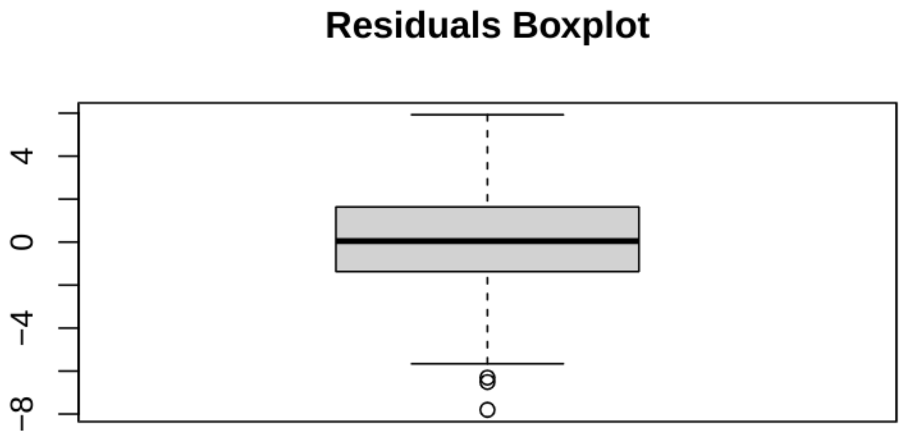
2. Download a bivariate data and fit into a simple linear regression model. For this given data, estimate the unknwon parameters of the aforementioned model using the LSE and the LAD techniques. Compare the performance of the LSE and the LAD estimators.

We have downloaded the dataset from kaggle which depicts relationship between height of children and average of height of their parents, referred to as midparent height, it contains 928 observation. First we outliers present in data, as it can distort the regression results.

```r
# set working directory keeping r, qmd file and csv file in same folder
data = read.csv("galton.csv")

#features of data

# checking for normality
hist(resid(lm( child ~ parent, data = data)), breaks = 30, main = "Histogram of Residuals")
```



```
#looks like normal's density curve

# checking for outliers
boxplot(resid(lm( child ~ parent, data = data)), main = "Residuals Boxplot")
```



Then we use OLS and LAD methods to estimate the parameters as follows

```
#ols estimation
ols_est = lm( child ~ parent, data = data)
ols_parameters = coef(ols_est)
print("LS (Least Squares) Estimates:")
```

```
[1] "LS (Least Squares) Estimates:"
```

```
print(ols_parameters)
```

```
(Intercept)      parent
 23.9415302   0.6462906
```

```
#LAD regression
lad_regression <- function(X, y, tol = 1e-6, max_iter = 100) {
  n <- length(y)
  p <- ncol(X)

  # Add intercept column
  X <- cbind(1, X)
  # Initial estimate using OLS
  beta <- solve(t(X) %*% X) %*% t(X) %*% y

  for (iter in 1:max_iter) {
    # Compute residuals
    residuals <- y - X %*% beta

    # Avoid division by zero: Replace very small residuals with a small constant
    weights <- 1 / (abs(residuals) + 1e-8)

    # Form weighted X and y
    W <- diag(as.vector(weights), n, n)
    XWX <- t(X) %*% W %*% X
    XWy <- t(X) %*% W %*% y

    # Solve weighted least squares problem
    beta_new <- solve(XWX) %*% XWy

    # Check convergence
    if (sum(abs(beta_new - beta)) < tol) {
      break
    }

    beta <- beta_new
  }

  return(beta)
}

lad_est1 = lad_regression(data$parent, data$child)



# verifying LAD regression using the library

#install the following library if it not installed in system already

#install.packages("quantreg")

library(quantreg)
```

```
Loading required package: SparseM
```

```r
lad_model <- rq(child ~ parent, data = data, tau = 0.5)  # tau=0.5 is median regression (LAD
lad_beta <- coef(lad_model)
print("LAD (Least Absolute Deviations) Estimates:")
```

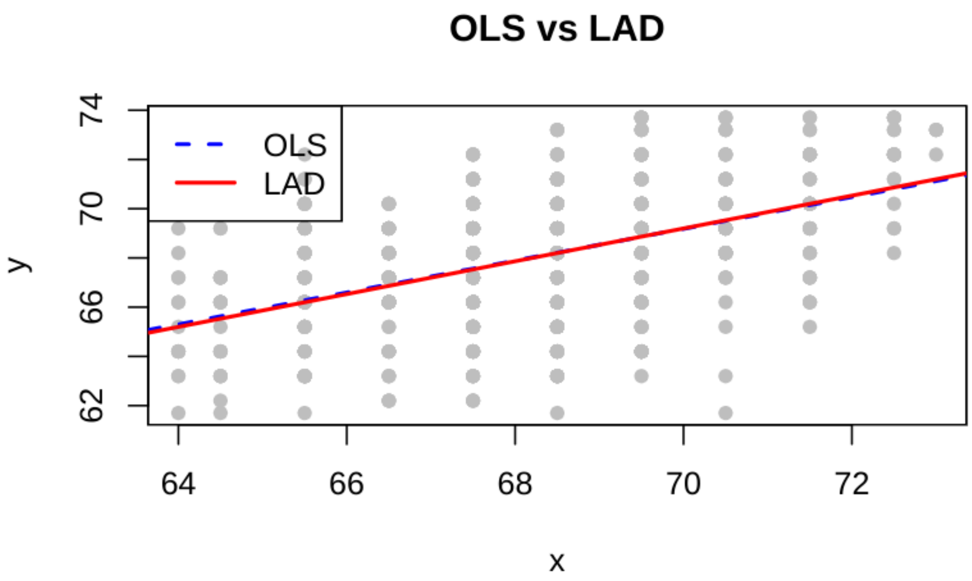
```
[1] "LAD (Least Absolute Deviations) Estimates:"
```

```r
print(lad_beta)
```

```
(Intercept)      parent 
 22.5333333   0.6666667 
```

After this we compare both kind of estimation methods as follows

```r
#by plotting both the models on the same graph
plot(data$parent, data$child, pch = 16, col = "gray", main = "OLS vs LAD", xlab = "x", ylab =
abline(ols_est, col = "blue", lwd = 2, lty = 2)   # OLS (dashed blue line)
abline(lad_est1, col = "red", lwd = 2)            # LAD (solid red line)
legend("topleft", legend = c("OLS", "LAD"), col = c("blue", "red"), lty = c(2, 1), lwd = 2)
```



```r
# comparing mean square error (mse) and mean absolute error (mae)

res_ols <- resid(ols_est)
res_lad <- resid(lad_model)

mae <- function(res) mean(abs(res))
mse <- function(res) mean(res^2)

cat("OLS  - MAE:", mae(res_ols), ", MSE:", mse(res_ols), "\n")
```

```
OLS  - MAE: 1.797718 , MSE: 5.000294
```

```r
cat("LAD  - MAE:", mae(res_lad), ", MSE:", mse(res_lad), "\n")
```

```
LAD  - MAE: 1.796157 , MSE: 5.001886
```

Conclusion

- **Evaluating limit using Copula**:
  - We get the value as

    ```
    The estimate obtained for given limit is 0.01965747
    ```

- **Estimation of Parameters and comparison of techniques**:
  - OLS/LSE parameters are as follows

  ```
  (Intercept)      parent
   23.9415302   0.6462906
  ```

  - LAD parameters are as follows

    ```
    (Intercept)      parent
     22.5333333   0.6666667
    ```

  - On comparing OLS and LAD we get the following result

    ```
    OLS  - MAE: 1.797718 , MSE: 5.000294
    ```

  ```
  LAD  - MAE: 1.796157 , MSE: 5.001886
  ```

- Here we can see that there is not much difference in MAE and MSE values of LAD and LSE techniques as there are only a few outliers present in the data, and the slight difference can be explained by the method of computation. As **OLS will have lower MSE** (it minimizes squared error) and **LAD will often have lower MAE**, especially with **outliers.**