Statistical Programming - Project 3 Report - GROUP 4

Students Name:

STUDENT NUMBER	NAME and SURNAME
221805001	OKTAY CAN SEVİMGİN

Data File: MultRegData.txt

◇ 1. Linear Regression via Matrix Operations

In this part, we created an R function called `my_regression()` that performs linear regression using matrix operations (without using `lm()`).

☑ *Function Inputs:*

- `Y`: dependent variable vector
- `X`: independent variable matrix or dataframe

📌 *Used Matrix Formulas (Plain Format):*

- **Beta Hat (Estimated Coefficients):**
  `beta_hat = solve(t(X) %*% X) %*% t(X) %*% Y`
- **Estimated Y Values:**
  `Y_hat = X %*% beta_hat`
- **Residuals:**
  `e_hat = Y - Y_hat`
- **TSS (Total Sum of Squares):**
  `TSS = sum((Y - mean(Y))^2)`
- **RMSS (Regression Model Sum of Squares):**
  `RMSS = sum((Y_hat - mean(Y))^2)`

- **RSS (Residual Sum of Squares):**
  `RSS = sum((Y - Y_hat)^2)`
- **R-Squared (R^2):**
  `R_squared = 1 - (RSS / TSS)`

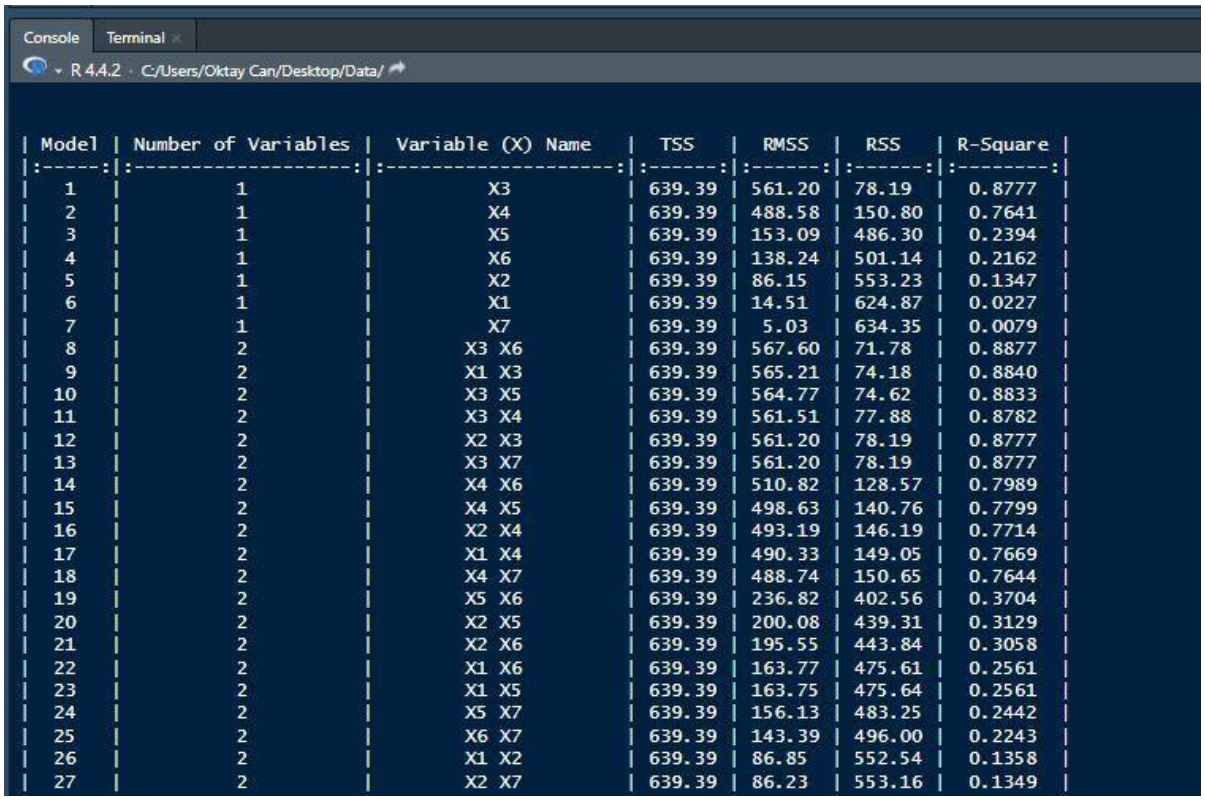
All matrix operations (e.g. transpose `t()`, inverse `solve()`, and multiplication %*%) were coded manually from scratch, not using built-in functions like `lm()`.

◇ 2. Model Selection Based on R-Squared

We created another function `model_selection()` that:

- Tries all combinations of X variables (from 1 to 7 at a time),
- Applies `my_regression()` to each combination,
- Calculates and stores TSS, RMSS, RSS, and R^2,
- Sorts models by number of variables and descending R-squared.

Output Table Example :



Model	Number of Variables	Variable (X) Name	TSS	RMSS	RSS	R-Square
1	1	X3	639.39	561.20	78.19	0.8777
2	1	X4	639.39	488.58	150.80	0.7641
3	1	X5	639.39	153.09	486.30	0.2394
4	1	X6	639.39	138.24	501.14	0.2162
5	1	X2	639.39	86.15	553.23	0.1347
6	1	X1	639.39	14.51	624.87	0.0227
7	1	X7	639.39	5.03	634.35	0.0079
8	2	X3 X6	639.39	567.60	71.78	0.8877
9	2	X1 X3	639.39	565.21	74.18	0.8840
10	2	X3 X5	639.39	564.77	74.62	0.8833
11	2	X3 X4	639.39	561.51	77.88	0.8782
12	2	X2 X3	639.39	561.20	78.19	0.8777
13	2	X3 X7	639.39	561.20	78.19	0.8777
14	2	X4 X6	639.39	510.82	128.57	0.7989
15	2	X4 X5	639.39	498.63	140.76	0.7799
16	2	X2 X4	639.39	493.19	146.19	0.7714
17	2	X1 X4	639.39	490.33	149.05	0.7669
18	2	X4 X7	639.39	488.74	150.65	0.7644
19	2	X5 X6	639.39	236.82	402.56	0.3704
20	2	X2 X5	639.39	200.08	439.31	0.3129
21	2	X2 X6	639.39	195.55	443.84	0.3058
22	2	X1 X6	639.39	163.77	475.61	0.2561
23	2	X1 X5	639.39	163.75	475.64	0.2561
24	2	X5 X7	639.39	156.13	483.25	0.2442
25	2	X6 X7	639.39	143.39	496.00	0.2243
26	2	X1 X2	639.39	86.85	552.54	0.1358
27	2	X2 X7	639.39	86.23	553.16	0.1349

🗀 Dataset Info:

- File: `MultRegData.txt`
- Variables: `IndNo, Y, X1, X2, X3, X4, X5, X6, X7`
- We used `Y` as dependent and `X1` to `X7` as independent variables.

📌 Conclusion:

- We successfully implemented matrix-based regression manually in R.
- We calculated all statistical metrics (TSS, RSS, R^2) without using built-in functions.
- We generated and compared **all possible models** to find the one with the highest explanatory power (R^2).
- This approach deepened our understanding of **linear regression fundamentals** and **matrix operations**.