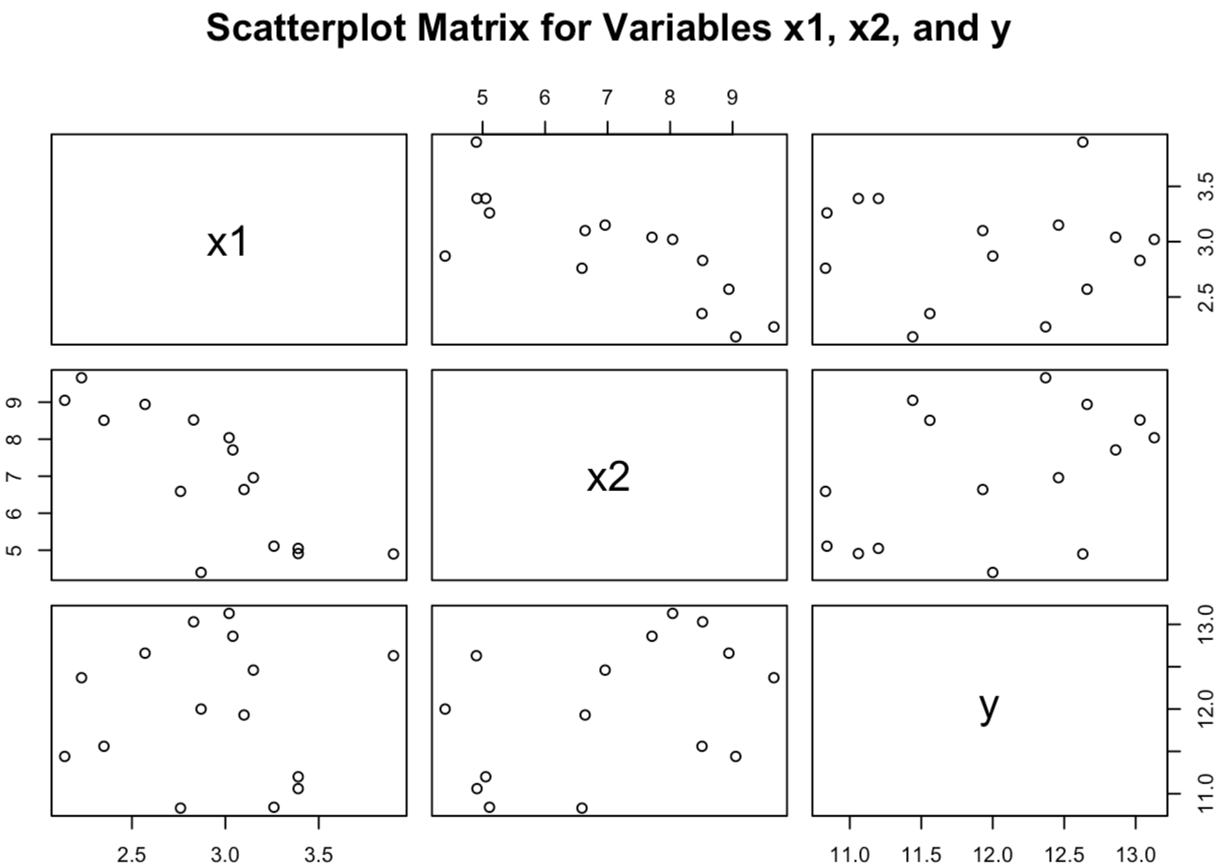
401-hw5

2024-11-04

Problem 1a

```
dat <- data.frame(
     x1=c(2.23,2.57,2.87,3.1,3.39,2.83,3.02,2.14,3.04,3.26,3.39,2.35,
       2.76,3.9,3.15),
     x2=c(9.66,8.94,4.4,6.64,4.91,8.52,8.04,9.05,7.71,5.11,5.05,8.51,
       6.59,4.9,6.96),
     y=c(12.37,12.66,12,11.93,11.06,13.03,13.13,11.44,12.86,10.84,
       11.2,11.56,10.83,12.63,12.46))

# Generate scatterplot matrix
pairs(dat, main = "Scatterplot Matrix for Variables x1, x2, and y")
```



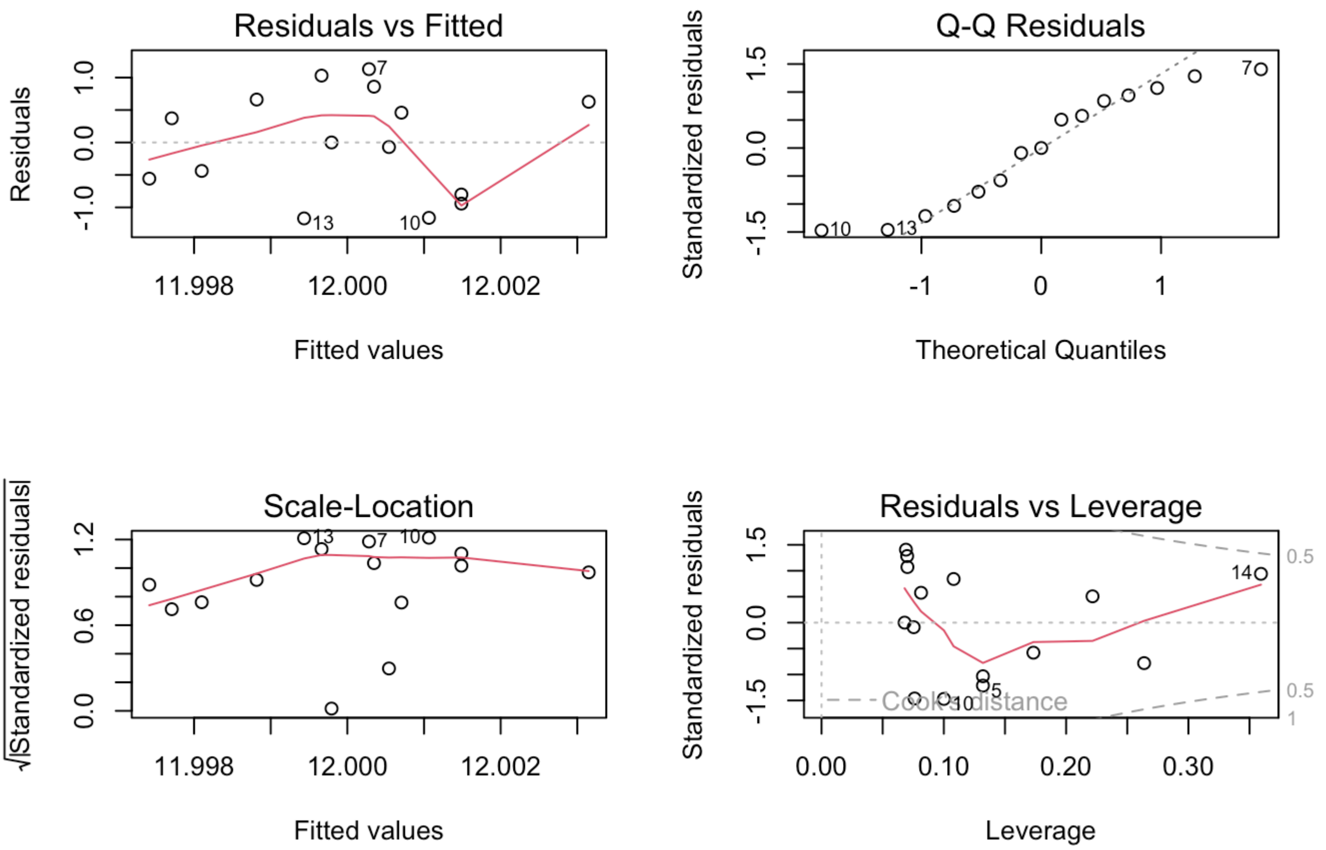
Comments: There appears to be a negative correlation between x1 and x2, with x1 ranging from approximately 2.5 to 3.5 and x2 spanning from about 5 to 9, while y extends from around 11.0 to 13.0. The relationships between the dependent variable y and the predictors show moderate to weak correlations, and neither relationship appears strongly linear. The data points are relatively evenly distributed within their ranges without any obvious outliers or clustering, suggesting that while linear modeling might be appropriate, it may not capture all the complexity in these relationships.

Problem 1b

```
# Linear regression of y on x1
model_x1 <- lm(y ~ x1, data = dat)
summary(model_x1)
```

```
##
## Call:
## lm(formula = y ~ x1, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.16944 -0.67945  0.00021  0.64402  1.12972
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11.990446   1.383341   8.668  9.2e-07 ***
## x1           0.003257   0.465866   0.007    0.995
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8324 on 13 degrees of freedom
## Multiple R-squared:  3.76e-06,   Adjusted R-squared:  -0.07692
## F-statistic: 4.888e-05 on 1 and 13 DF,  p-value: 0.9945
```

```
# Residual diagnostics
par(mfrow = c(2, 2))
plot(model_x1)
```



- **Null Hypothesis (H_0)**: The coefficient of x_1 is zero ($\beta_1 = 0$), meaning x_1 has no effect on y.
- **Alternative Hypothesis (H_a)**: The coefficient of x_1 is not zero ($\beta_1 \neq 0$), meaning x_1 has a significant effect on y.

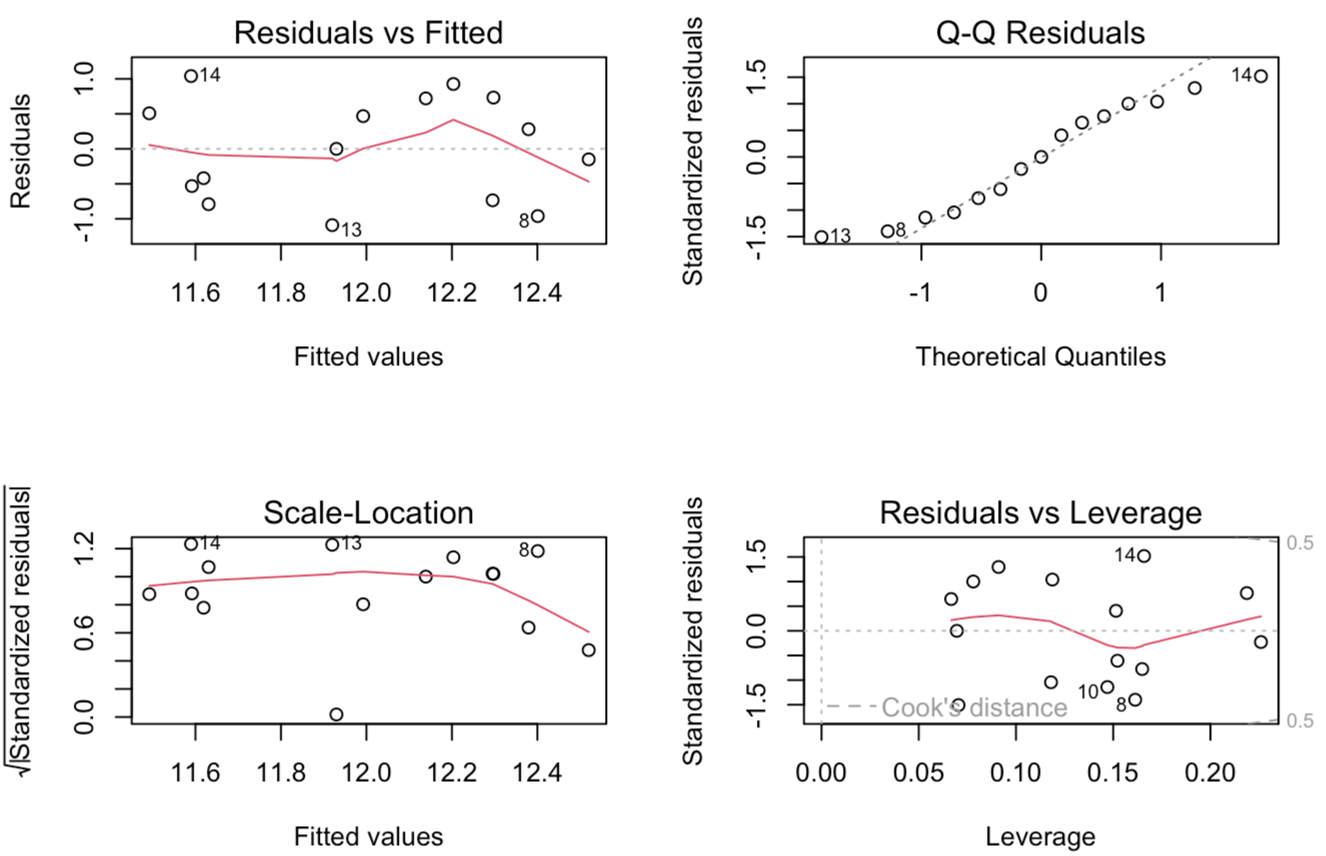
Based on the regression results, the p-value for x_1 (0.995) is far greater than the significance level of 0.05. Therefore, we fail to reject the null hypothesis (H_0). This implies that there is insufficient evidence to conclude that x_1 has a significant effect on y. The overall model is also not significant, with an F-statistic p-value of 0.9945, further indicating that x_1 does not explain the variability in y.

Problem 1c

```
# Linear regression of y on x2
model_x2 <- lm(y ~ x2, data = dat)
summary(model_x2)
```

```
##
## Call:
## lm(formula = y ~ x2, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.08999 -0.63345  0.00023  0.61458  1.04033
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  10.6319     0.8109  13.111 7.18e-09 ***
## x2            0.1955     0.1125   1.737    0.106
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7499 on 13 degrees of freedom
## Multiple R-squared:  0.1884, Adjusted R-squared:  0.126
## F-statistic: 3.018 on 1 and 13 DF,  p-value: 0.106
```

```
# Residual diagnostics
par(mfrow = c(2, 2))
plot(model_x2)
```



In this regression analysis for Problem 1c, we can set up the hypotheses as follows:

- **Null Hypothesis (H_0)**: The coefficient of x_2 is zero ($\beta_1 = 0$), meaning x_2 has no effect on y.

- **Alternative Hypothesis (H_a)**: The coefficient of x_2 is not zero ($\beta_1 \neq 0$), meaning x_2 has a significant effect on y.

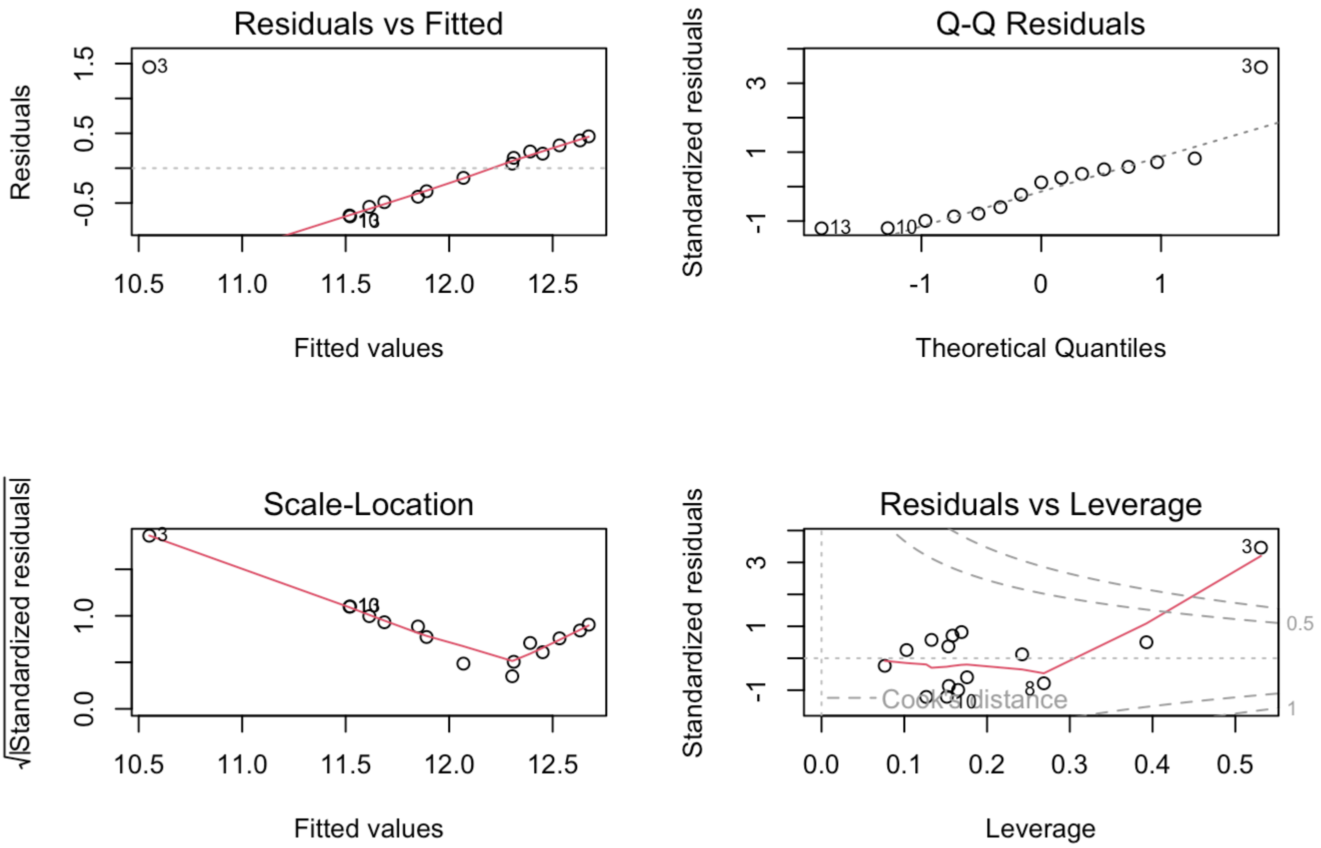
The p-value for x_2 is 0.106, which is greater than the significance level of 0.05, indicating that we fail to reject the null hypothesis (H_0). This suggests that there is insufficient evidence to conclude that x_2 has a significant effect on y. The overall model's F-statistic has a p-value of 0.106, which also indicates that the model is not statistically significant.

Problem 1d

```
# Linear regression of y on both x1 and x2
model_x1_x2 <- lm(y ~ x1 + x2, data = dat)
summary(model_x1_x2)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.69127 -0.44813  0.06541  0.28281  1.44873
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.8610     2.5440   1.518   0.1550
## x1            1.5339     0.5566   2.756   0.0174 *
## x2            0.5200     0.1492   3.485   0.0045 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6108 on 12 degrees of freedom
## Multiple R-squared:  0.503,  Adjusted R-squared:  0.4202
## F-statistic: 6.073 on 2 and 12 DF,  p-value: 0.01507
```

```
# Residual diagnostics
par(mfrow = c(2, 2))
plot(model_x1_x2)
```



In this regression analysis for Problem 1d, we are testing the significance of both x_1 and x_2 in predicting y. The hypotheses are as follows:

- **Null Hypothesis (H_0)**: The coefficients of x_1 and x_2 are zero ($\beta_1 = 0$ and $\beta_2 = 0$), meaning neither x_1 nor x_2 has an effect on y.
- **Alternative Hypothesis (H_a)**: At least one of the coefficients (β_1 or β_2) is not zero, indicating that at least one predictor has a significant effect on y.

The overall model is significant, with an F-statistic p-value of 0.01507, which is below the 0.05 threshold. This indicates that the model with both x_1 and x_2 provides a statistically significant fit for predicting y. Looking at the individual predictors, x_1 has a p-value of 0.0174, and x_2 has a p-value of 0.0045, both of which are significant at the 0.05 level. This suggests that both x_1 and x_2 contribute meaningfully to the model.

Problem 1e

Using a significance level of 0.05, forward selection would fail to identify either x_1 or x_2 as significant predictors when considered individually, meaning neither variable would enter the model. This is problematic because we would miss the significant combined effect found when both variables are included together (model d). In contrast, backward selection, starting with the full model containing both variables, would retain both predictors since their joint effect is significant. This illustrates a key limitation of forward selection - it can miss important variable combinations by only considering one variable at a time - while demonstrating an advantage of backward selection in capturing joint effects that aren't apparent when variables are considered in isolation.

Problem 4a

```
# Load required libraries
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
# (a) Read and process customer data
customer <- read.csv("/Users/homura/Desktop/customer2.csv")
customer$logtarg <- log(customer$target + 1)

# Print basic statistics for customer data
cat("\nCustomer Data Summary:\n")
```

```
##
## Customer Data Summary:
```

```
summary(customer)
```

```
##        id               train            target          logtarg
##  Min.   :     957   Min.   :0.0000   Min.   :  0.000   Min.   :0.0000
##  1st Qu.: 4448960   1st Qu.:0.0000   1st Qu.:  0.000   1st Qu.:0.0000
##  Median : 8090750   Median :0.0000   Median :  0.000   Median :0.0000
##  Mean   : 8563488   Mean   :0.3308   Mean   :  3.241   Mean   :0.2529
##  3rd Qu.:13378724   3rd Qu.:1.0000   3rd Qu.:  0.000   3rd Qu.:0.0000
##  Max.   :16456238   Max.   :1.0000   Max.   :739.480   Max.   :6.6073
```

Problem 4b

```
# Read orders data
orders_data <- read.csv("/Users/homura/Desktop/orders.csv")

# Remove duplicate rows based on 'id', 'orddate', and 'ordnum' (if these uniquely identify an
order)
orders_data <- orders_data %>%
  distinct(id, orddate, ordnum, .keep_all = TRUE)

# Create a new variable 't' for time (years) since the transaction as of 2014-11-25
orders_data <- orders_data %>%
  mutate(t = as.numeric(as.Date("2014-11-25", format="%Y-%m-%d") - as.Date(orddate, format="%
d%b%Y")) / 365.25)

# Print basic descriptive statistics
summary(orders_data)
```

```
##        id               orddate              ordnum           category
##  Min.   :     957   Length:102555      Min.   :   1018   Min.   : 1.00
##  1st Qu.: 3887413   Class :character   1st Qu.: 365248   1st Qu.:14.00
##  Median : 6109373   Mode  :character   Median : 690438   Median :20.00
##  Mean   : 6678104                      Mean   : 669318   Mean   :32.64
##  3rd Qu.: 8689962                      3rd Qu.: 982118   3rd Qu.:37.00
##  Max.   :16456238                      Max.   :1256189   Max.   :99.00
##       qty              price               t
##  Min.   :  0.000   Min.   :   0.00   Min.   :0.002738
##  1st Qu.:  1.000   1st Qu.:   6.95   1st Qu.:1.322382
##  Median :  1.000   Median :   9.95   Median :2.956879
##  Mean   :  1.038   Mean   :  14.00   Mean   :3.086623
##  3rd Qu.:  1.000   3rd Qu.:  15.24   3rd Qu.:4.711841
##  Max.   :100.000   Max.   :5010.66   Max.   :7.058179
```

Problem 4c

```
# Aggregate the transaction file to create the RFM table
RFM_table <- orders_data %>%
  group_by(id) %>%
  summarise(
    tof = max(t),                     # Time on file: years since the first order
    r = min(t),                       # Recency: years since the most recent order
    f = n_distinct(ordnum),           # Frequency: number of unique orders
    m = sum(price * qty)              # Monetary: total amount spent
  )

# Print basic summary statistics for the RFM table
summary(RFM_table)
```

```
##        id                 tof                   r                   f          
##  Min.   :     957   Min.   :0.002738   Min.   :0.002738   Min.   :  1.000  
##  1st Qu.: 4448960   1st Qu.:1.338809   1st Qu.:0.303901   1st Qu.:  2.000  
##  Median : 8090750   Median :3.800137   Median :0.851472   Median :  4.000  
##  Mean   : 8563488   Mean   :3.681231   Mean   :1.439085   Mean   :  6.111  
##  3rd Qu.:13378724   3rd Qu.:6.036961   3rd Qu.:2.031485   3rd Qu.:  8.000  
##  Max.   :16456238   Max.   :7.058179   Max.   :7.058179   Max.   :160.000  
##        m          
##  Min.   :    0.00  
##  1st Qu.:   18.95  
##  Median :   45.80  
##  Mean   :   88.64  
##  3rd Qu.:  103.75  
##  Max.   :26564.51
```

Problem 4d

```
# Join the customer and RFM tables
merged_data <- customer %>%
  inner_join(RFM_table, by = "id")

# Regress 'logtarg' on 'log(tof)', 'log(r)', 'log(f)', and 'log(m + 1)' using only training da
ta
train_data <- merged_data %>% filter(train == 1)
model <- lm(logtarg ~ log(tof) + log(r) + log(f) + log(m + 1), data = train_data)

# Show a summary of the fitted model
summary(model)
```

```
## 
## Call:
## lm(formula = logtarg ~ log(tof) + log(r) + log(f) + log(m + 1), 
##     data = train_data)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -0.9643 -0.3745 -0.2178 -0.0539  5.5507 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  0.12108    0.05385   2.249  0.02458 *  
## log(tof)    -0.06006    0.02063  -2.912  0.00361 ** 
## log(r)      -0.07702    0.01298  -5.935 3.12e-09 ***
## log(f)       0.18231    0.02787   6.541 6.65e-11 ***
## log(m + 1)  -0.01707    0.02010  -0.849  0.39574    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.9151 on 5546 degrees of freedom
## Multiple R-squared:  0.05224,    Adjusted R-squared:  0.05156 
## F-statistic: 76.42 on 4 and 5546 DF,  p-value: < 2.2e-16
```

Problem 4e

```
# (e) Compute MSE on test set
# Apply the model from the training part to the test set
test_data <- merged_data %>% filter(train == 0)

# Predict 'logtarg' for the test set using the fitted model
test_data$predicted_logtarg <- predict(model, newdata = test_data)

# Compute the Mean Squared Error (MSE) on the test set
mse <- mean((test_data$logtarg - test_data$predicted_logtarg)^2)
print(paste("Mean Squared Error on the test set:", mse))
```

```
## [1] "Mean Squared Error on the test set: 0.80997002836259"
```