# NETAJI SUHBAS UNIVERSITY OF TECHNOLOGY

**AI Hardware and tools**

**Lab File – VI Semester**

TASK-1

## Submitted by –

**Paras Gunjyal (2021UCA1849)**

**Abhi Kansal (2021UCA1863)**

**Shivam Gupta (2021UCA1816)**

**CSAI – 1**

# 1 Explore Basic Data Structure in R.

In R, there are several basic data structures that are commonly used for storing and manipulating data. Here are some of the fundamental data structures in R:

**Vectors:**

A vector is the most basic data structure in R, representing a one-dimensional array of elements. Elements in a vector must be of the same data type. You can create a vector using the c() function.

In [1]:
```
numeric_vector <- c(12, 21, 14, 4, 5)

character_vector <- c("car", "bike", "bus")
```

In [2]:
```
print(numeric_vector)
print(character_vector)
```

```
[1] 12 21 14  4  5
[1] "car"  "bike" "bus"
```

**Matrices:**

A matrix is a two-dimensional data structure in R. It is created using the matrix() function.

In [3]:
```
my_matrix <- matrix(c(1, 2, 3, 4, 5, 6), nrow = 2, ncol = 3)
print(my_matrix)
```

```
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
```

**Arrays:**

An array is a multi-dimensional extension of a matrix. You can create an array using the array() function.

In [4]:
```
# Creating a 3-dimensional array
my_array <- array(c(1:13), dim = c(2, 3, 2))
print(my_array)
```

```
, , 1

     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6

, , 2

     [,1] [,2] [,3]
[1,]    7    9   11
[2,]    8   10   12
```

**Lists:**

A list is a versatile data structure that can store elements of different data types. You can create a list using the list() function.

In [5]:
```
my_list <- list(name = "Paras", age = 25, CCS_score = c(90, 85, 92))
print(my_list)
```

```
$name
[1] "Paras"

$age
[1] 25

$CCS_score
[1] 90 85 92
```

**Data Frames:**

A data frame is a two-dimensional table where each column can be of a different data type. It is created using the data.frame() function.

In [6]:
```
my_data_frame <- data.frame(name = c("Ram", "Raju", "Charlie"),
                            age = c(28, 22, 35),
                            score = c(95, 89, 75))
print(my_data_frame)
```

```
     name age score
1     Ram  28    95
2    Raju  22    89
3 Charlie  35    75
```

**Factors:**

Factors are used to represent categorical data in R. They are created using the factor() function.

In [7]:
```
my_factor <- factor(c("low", "medium", "high", "low", "medium"))
print(my_factor)
```

```
[1] low    medium high   low    medium
Levels: high low medium
```

# 2. Implement Linear Regression in R and Visualize the results.

In [8]:
```
# Read the CSV file
insurance_data <- read.csv("insurance.csv")

# View the structure of the dataset
str(insurance_data)
```

```
'data.frame':	1338 obs. of  7 variables:
 $ age     : int  19 18 28 33 32 31 46 37 37 60 ...
 $ sex     : Factor w/ 2 levels "female","male": 1 2 2 2 2 1 1 1 2 1 ...
 $ bmi     : num  27.9 33.8 33 22.7 28.9 25.7 33.4 27.7 29.8 25.8 ...
 $ children: int  0 1 3 0 0 0 1 3 2 0 ...
 $ smoker  : Factor w/ 2 levels "no","yes": 2 1 1 1 1 1 1 1 1 1 ...
 $ region  : Factor w/ 4 levels "northeast","northwest",..: 4 3 3 2 2 3 3 2 1 2 ...
 $ expenses: num  16885 1726 4449 21984 3867 ...
```

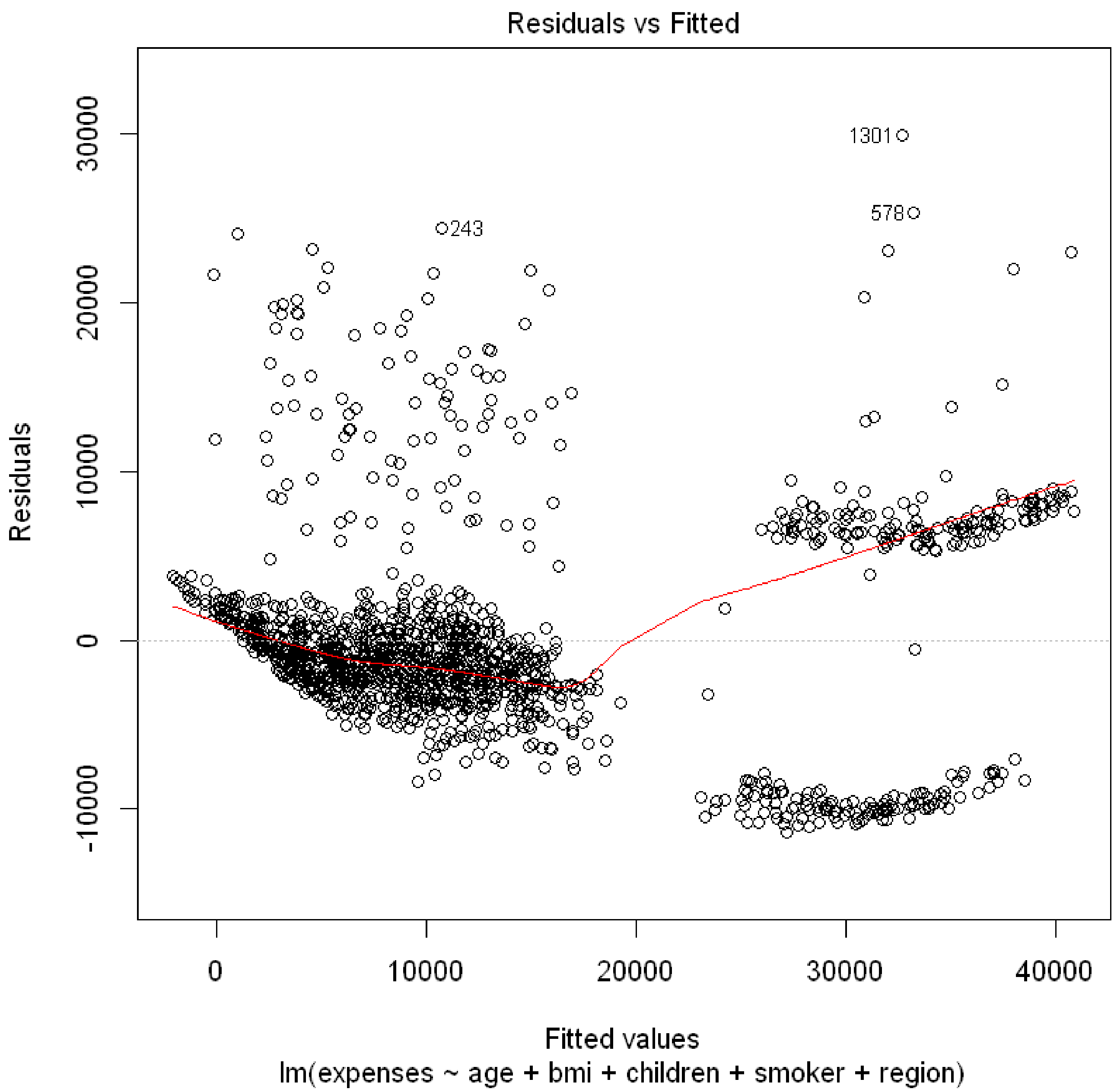
In [9]:
```r
# Fit a linear regression model for predicting expenses
lm_model <- lm(expenses ~ age + bmi + children + smoker + region, data = insurance_data)

# Summary of the linear regression model
summary(lm_model)
```

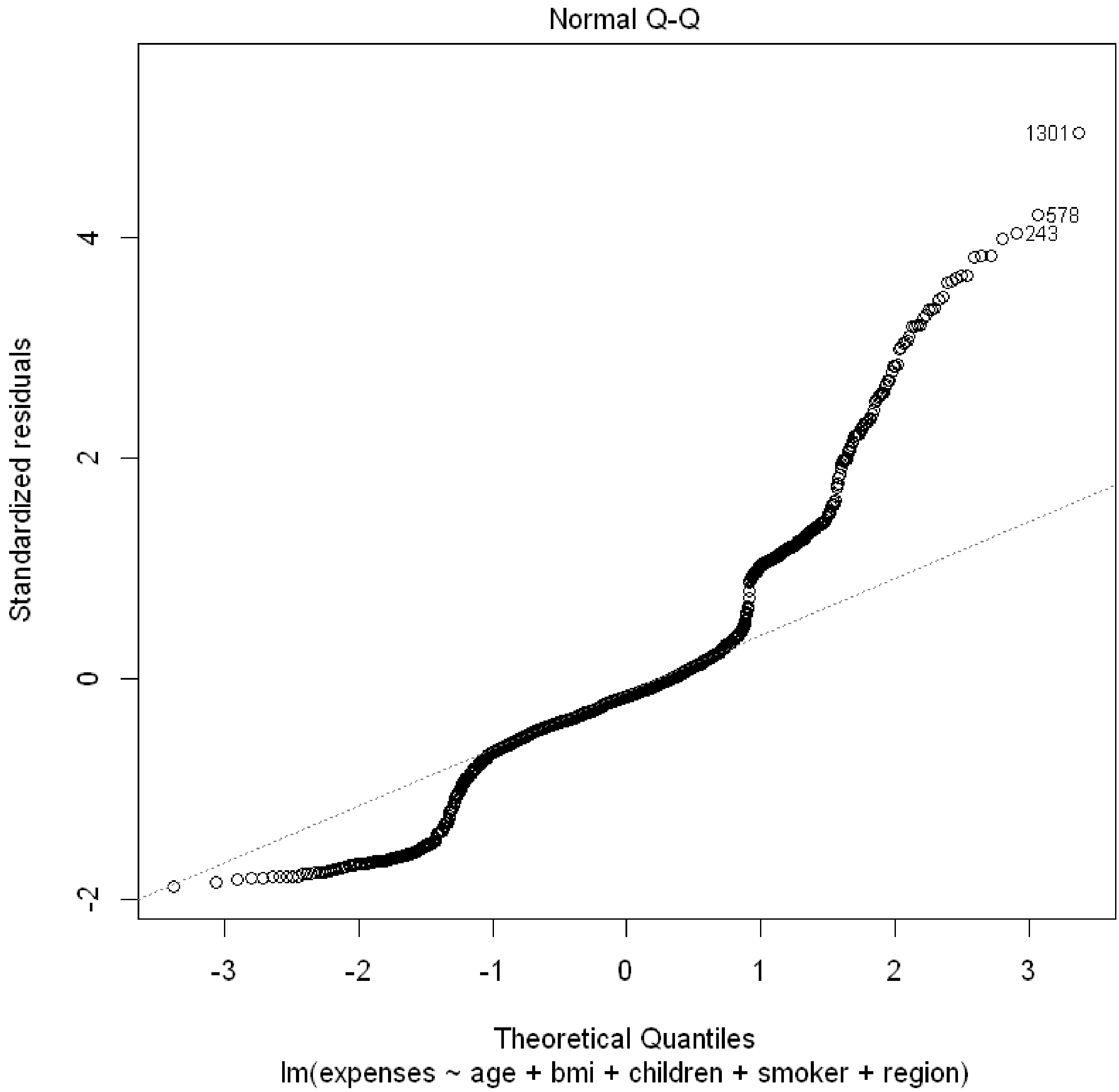
```
Call:
lm(formula = expenses ~ age + bmi + children + smoker + region, 
    data = insurance_data)

Residuals:
     Min       1Q   Median       3Q      Max 
-11365.0  -2839.4   -985.3   1375.5  29924.5 

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)    
(Intercept)     -11993.31     978.75 -12.254  < 2e-16 ***
age                256.96      11.89  21.609  < 2e-16 ***
bmi                338.76      28.56  11.862  < 2e-16 ***
children           474.75     137.74   3.447 0.000585 ***
smokeryes        23835.24     411.84  57.875  < 2e-16 ***
regionnorthwest   -352.01     476.11  -0.739 0.459825    
regionsoutheast  -1034.93     478.53  -2.163 0.030738 *  
regionsouthwest   -958.63     477.76  -2.007 0.045003 *  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6060 on 1330 degrees of freedom
Multiple R-squared:  0.7509,	Adjusted R-squared:  0.7496 
F-statistic: 572.7 on 7 and 1330 DF,  p-value: < 2.2e-16
```

In [10]:
```r
anova(lm_model)
```

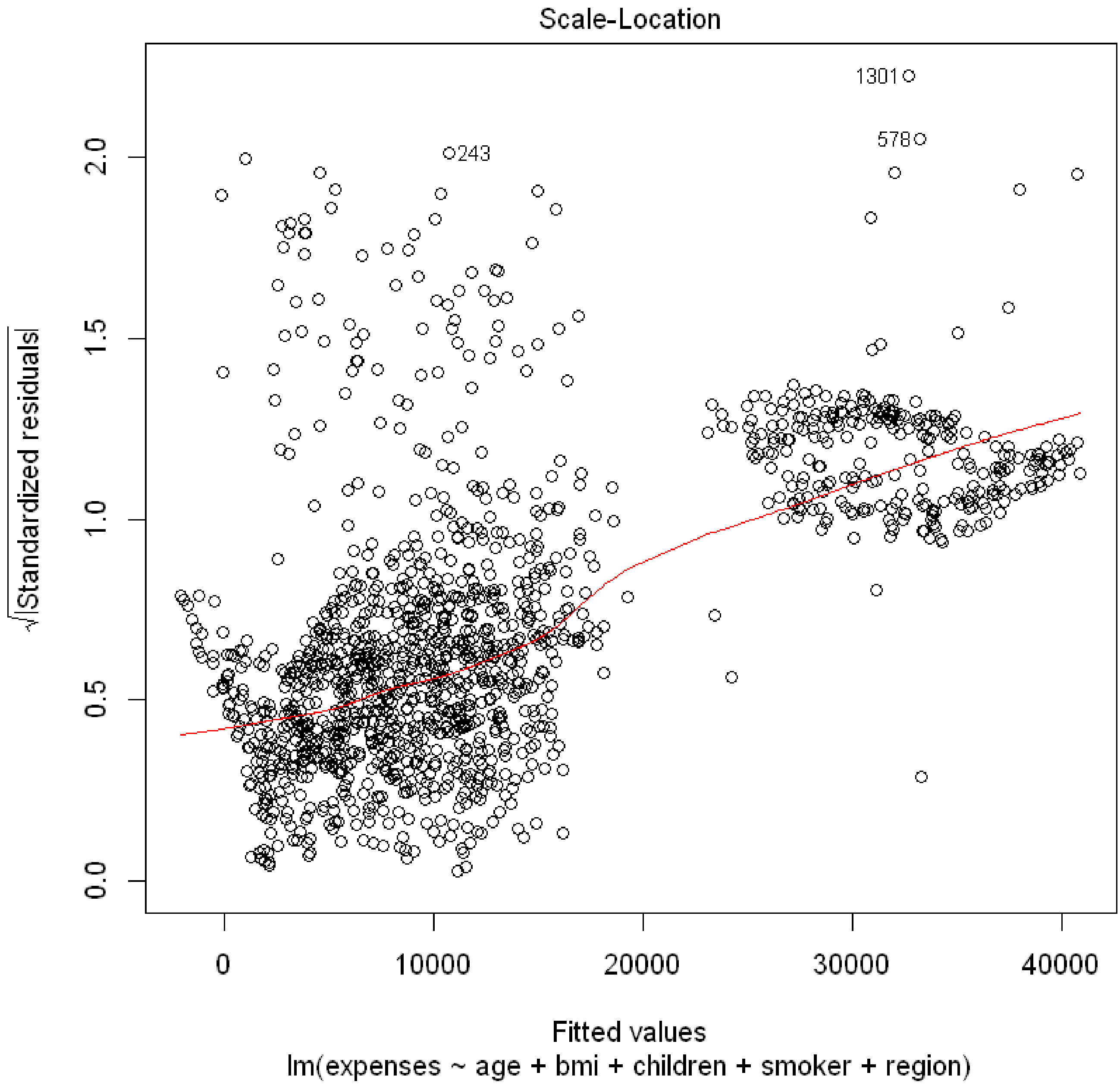
| | Df | Sum Sq | Mean Sq | F value | Pr(>F) |
|---|---|---|---|---|---|
| **age** | 1 | 17530192069 | 17530192069 | 477.356520 | 1.135857e-90 |
| **bmi** | 1 | 5460652936 | 5460652936 | 148.696505 | 1.688995e-32 |
| **children** | 1 | 571896996 | 571896996 | 15.573062 | 8.352503e-05 |
| **smoker** | 1 | 123436032578 | 123436032578 | 3361.229288 | 0.000000e+00 |
| **region** | 3 | 233220248 | 77740083 | 2.116904 | 9.627598e-02 |
| **Residuals** | 1330 | 48842226838 | 36723479 | NA | NA |

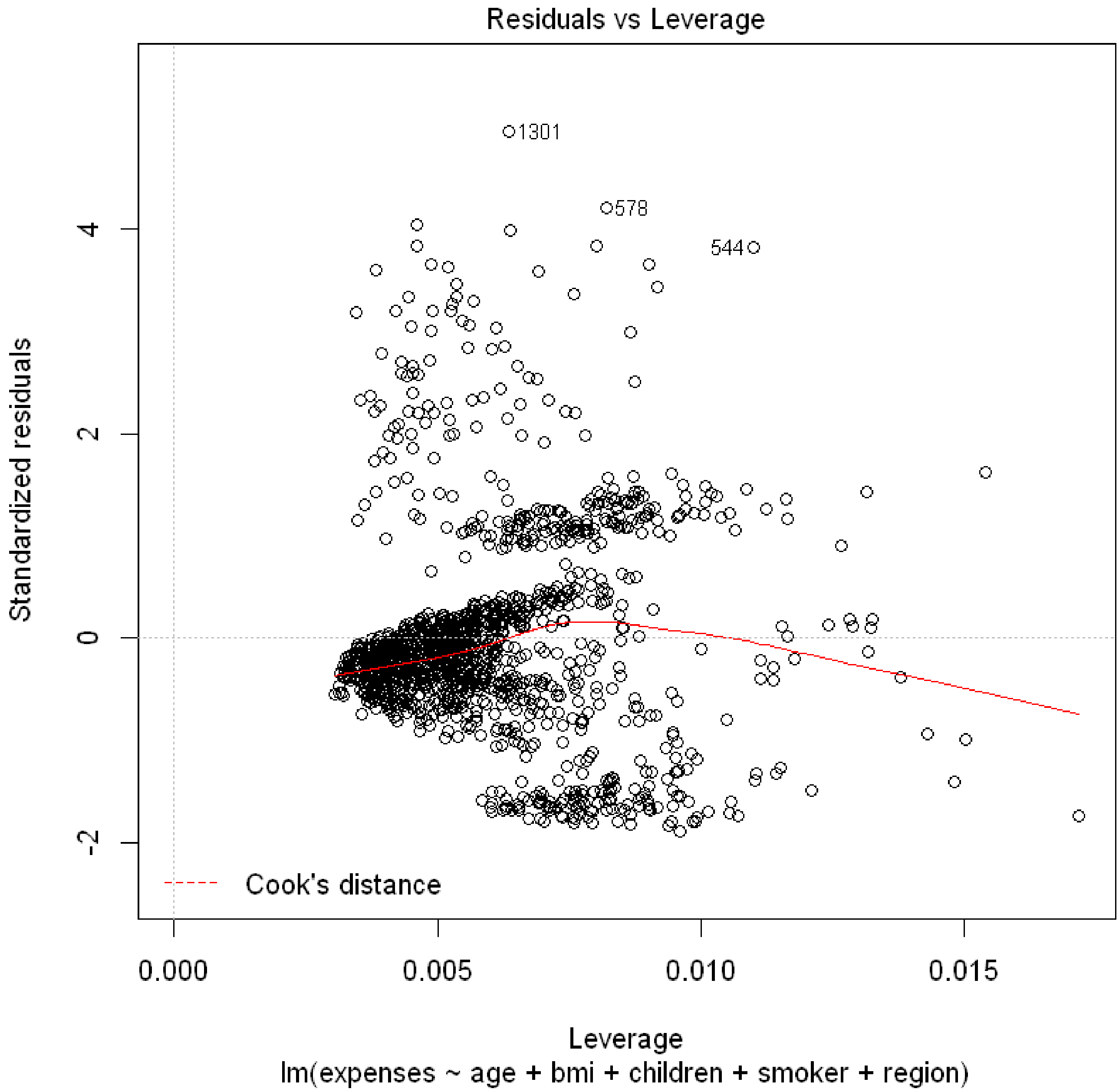
In [11]:
```r
plot(lm_model)
```


Residuals vs Fitted
Residuals
Fitted values
lm(expenses ~ age + bmi + children + smoker + region)
30000
20000
10000
0
-10000
0
10000
20000
30000
40000
1301
578
243



Normal Q-Q
1301
578
243
4
2
0
-2
Standardized residuals
-3
-2
-1
0
1
2
3
Theoretical Quantiles
lm(expenses ~ age + bmi + children + smoker + region)


Scale-Location
1301
578
243
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
0
10000
20000
30000
40000
Fitted values
lm(expenses ~ age + bmi + children + smoker + region)




# 3. Implement Logistic Regression in R and Visualize the results.

In [12]:

```
# Read the CSV file
heart_data <- read.csv("heart.csv")

# View the structure of the dataset
str(heart_data)
```

```
'data.frame':	303 obs. of  14 variables:
 $ age     : int  63 37 41 56 57 57 56 44 52 57 ...
 $ sex     : int  1 1 0 1 0 1 0 1 1 1 ...
 $ cp      : int  3 2 1 1 0 0 1 1 2 2 ...
 $ trestbps: int  145 130 130 120 120 140 140 120 172 150 ...
 $ chol    : int  233 250 204 236 354 192 294 263 199 168 ...
 $ fbs     : int  1 0 0 0 0 0 0 0 1 0 ...
 $ restecg : int  0 1 0 1 1 1 0 1 1 1 ...
 $ thalach : int  150 187 172 178 163 148 153 173 162 174 ...
 $ exang   : int  0 0 0 0 1 0 0 0 0 0 ...
 $ oldpeak : num  2.3 3.5 1.4 0.8 0.6 0.4 1.3 0 0.5 1.6 ...
 $ slope   : int  0 0 2 2 2 1 1 2 2 2 ...
 $ ca      : int  0 0 0 0 0 0 0 0 0 0 ...
 $ thal    : int  1 2 2 2 2 1 2 3 3 2 ...
 $ target  : int  1 1 1 1 1 1 1 1 1 1 ...
```

In [13]:

```r
# Fit logistic regression model
logit_model <- glm(target ~ age + sex + cp + trestbps + chol + fbs + restecg + thalach + exan

# Summary of the logistic regression model
summary(logit_model)
```

```
Call:
glm(formula = target ~ age + sex + cp + trestbps + chol + fbs + 
    restecg + thalach + exang + oldpeak + slope + ca + thal, 
    family = "binomial", data = heart_data)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-2.5849  -0.3872   0.1551   0.5863   2.6249  

Coefficients:
             Estimate Std. Error z value Pr(>|z|)    
(Intercept)  3.450472   2.571479   1.342 0.179653    
age         -0.004908   0.023175  -0.212 0.832266    
sex         -1.758181   0.468774  -3.751 0.000176 ***
cp           0.859851   0.185397   4.638 3.52e-06 ***
trestbps    -0.019477   0.010339  -1.884 0.059582 .  
chol        -0.004630   0.003782  -1.224 0.220873    
fbs          0.034888   0.529465   0.066 0.947464    
restecg      0.466282   0.348269   1.339 0.180618    
thalach      0.023211   0.010460   2.219 0.026485 *  
exang       -0.979981   0.409784  -2.391 0.016782 *  
oldpeak     -0.540274   0.213849  -2.526 0.011523 *  
slope        0.579288   0.349807   1.656 0.097717 .  
ca          -0.773349   0.190885  -4.051 5.09e-05 ***
thal        -0.900432   0.290098  -3.104 0.001910 ** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 417.64  on 302  degrees of freedom
Residual deviance: 211.44  on 289  degrees of freedom
AIC: 239.44

Number of Fisher Scoring iterations: 6
```

In [16]:

```r
# Visualize the results
# Let's create a plot for the probability of having heart disease based on age
new_data <- data.frame(age = seq(min(heart_data$age), max(heart_data$age), length.out = 100))
new_data$target_prob <- predict(logit_model, newdata = new_data, type = "response")
```

```
Error in eval(predvars, data, env): object 'sex' not found
Traceback:

1. predict(logit_model, newdata = new_data, type = "response")
2. predict.glm(logit_model, newdata = new_data, type = "response")
3. predict.lm(object, newdata, se.fit, scale = 1, type = if (type ==
 .     "link") "response" else type, terms = terms, na.action = na.action)
4. model.frame(Terms, newdata, na.action = na.action, xlev = object$xlevels)
5. model.frame.default(Terms, newdata, na.action = na.action, xlev = object$xlevels)
6. eval(predvars, data, env)
7. eval(predvars, data, env)
```

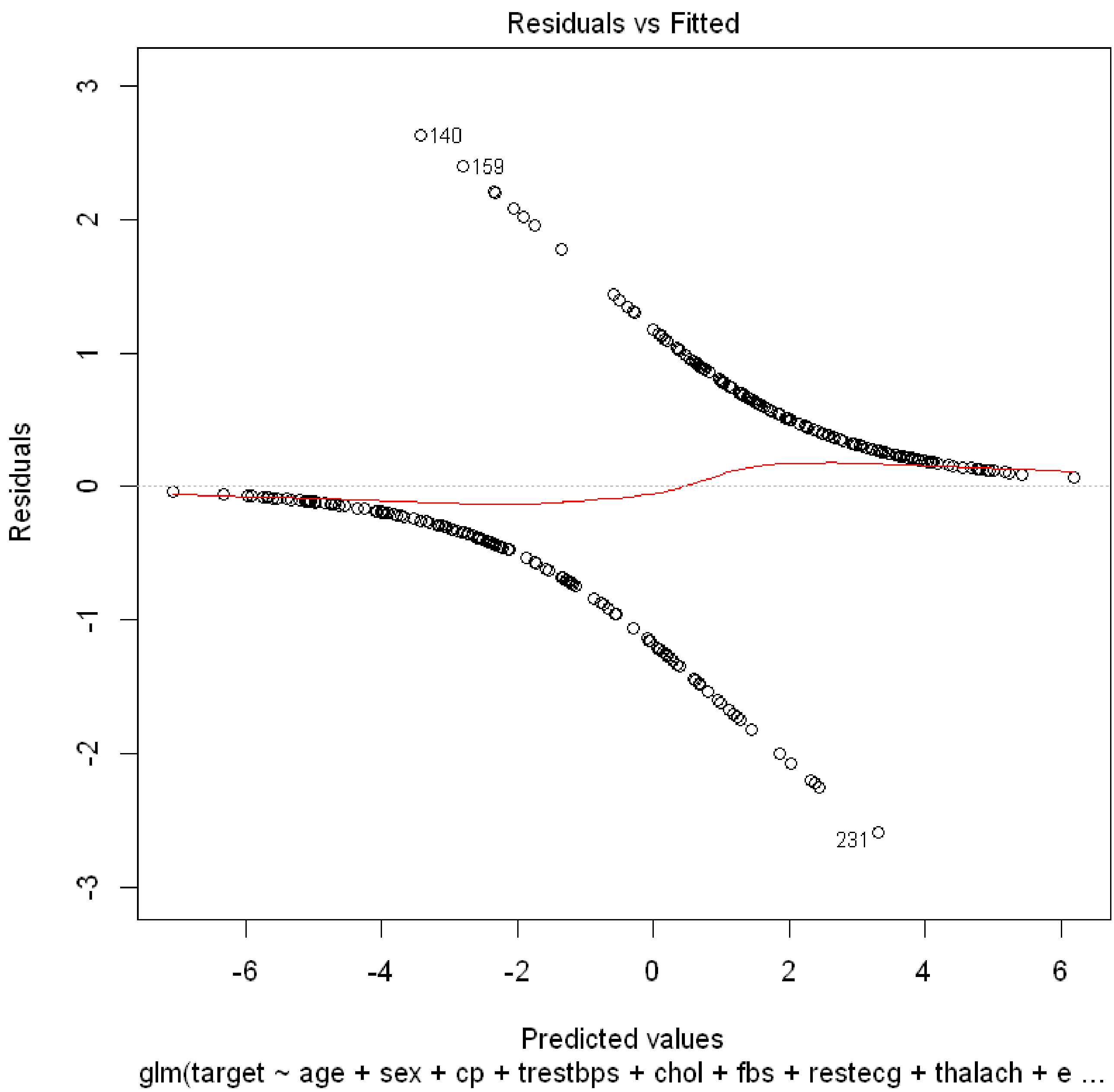
In [15]:
```
colnames(heart_data)
```

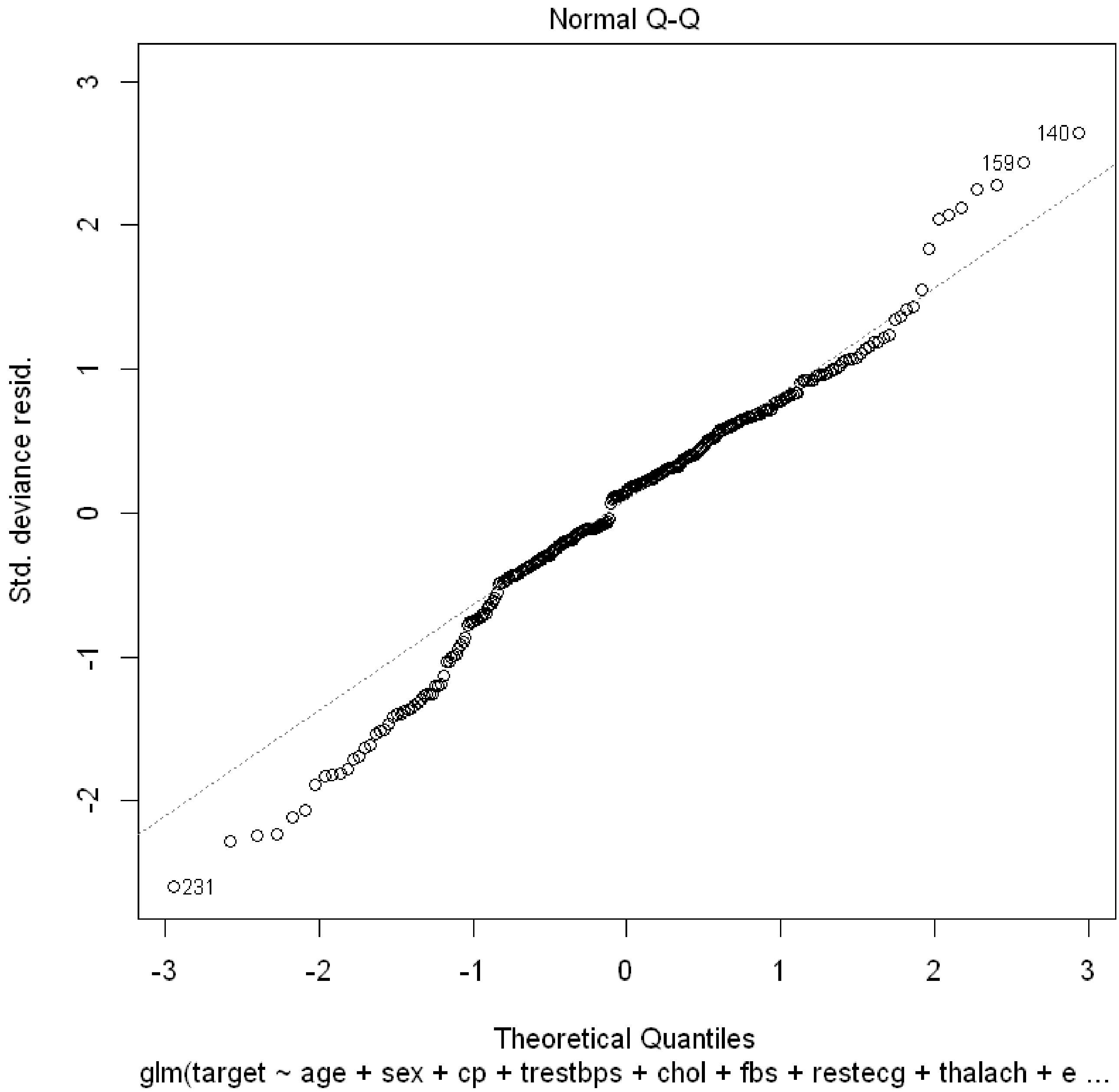
1. 'age'
2. 'sex'
3. 'cp'
4. 'trestbps'
5. 'chol'
6. 'fbs'
7. 'restecg'
8. 'thalach'
9. 'exang'
10. 'oldpeak'
11. 'slope'
12. 'ca'
13. 'thal'
14. 'target'

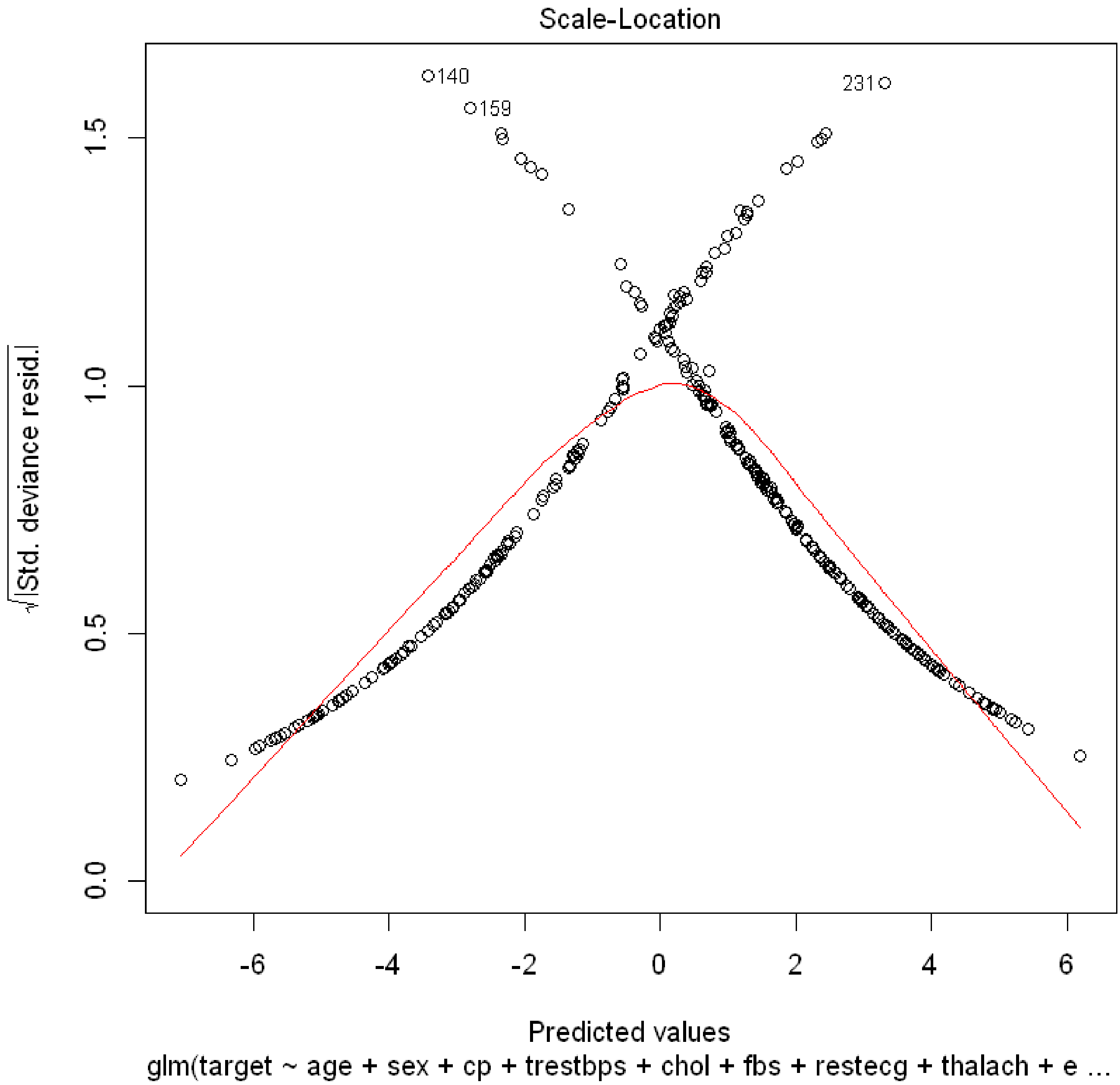
In [20]:
```
anova(logit_model)
```

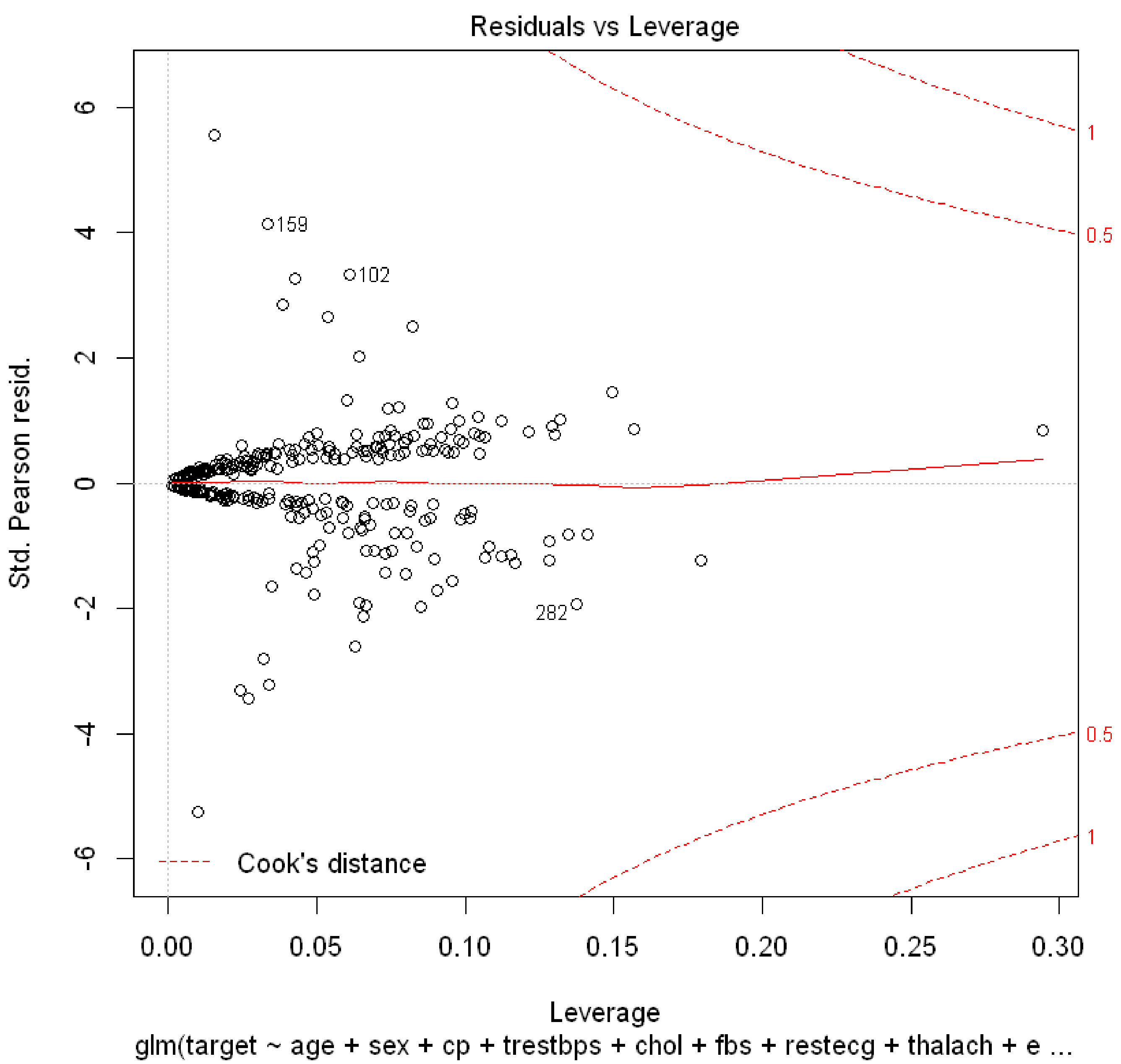
| | Df | Deviance | Resid. Df | Resid. Dev |
|---|---|---|---|---|
| **NULL** | NA | NA | 302 | 417.6381 |
| **age** | 1 | 15.7766919 | 301 | 401.8614 |
| **sex** | 1 | 31.2872127 | 300 | 370.5742 |
| **cp** | 1 | 59.7682306 | 299 | 310.8059 |
| **trestbps** | 1 | 6.7281625 | 298 | 304.0778 |
| **chol** | 1 | 1.9478502 | 297 | 302.1299 |
| **fbs** | 1 | 0.1098086 | 296 | 302.0201 |
| **restecg** | 1 | 1.7155609 | 295 | 300.3045 |
| **thalach** | 1 | 27.4338356 | 294 | 272.8707 |
| **exang** | 1 | 7.9650327 | 293 | 264.9057 |
| **oldpeak** | 1 | 22.2413718 | 292 | 242.6643 |
| **slope** | 1 | 1.0720758 | 291 | 241.5922 |
| **ca** | 1 | 20.2312203 | 290 | 221.3610 |
| **thal** | 1 | 9.9250308 | 289 | 211.4360 |

In [21]:
```
plot(logit_model)
```




Normal Q-Q
Std. deviance resid.
Theoretical Quantiles
glm(target ~ age + sex + cp + trestbps + chol + fbs + restecg + thalach + e ...
140
159
231
-3
-2
-1
0
1
2
3


Scale-Location
140
159
231
1.5
1.0
0.5
0.0
√|Std. deviance resid.|
-6
-4
-2
0
2
4
6
Predicted values
glm(target ~ age + sex + cp + trestbps + chol + fbs + restecg + thalach + e ...




In [22]:

```
# Fit a simpler logistic regression model (null model)
null_model <- glm(target ~ 1, data = heart_data, family = "binomial")

# Fit the full logistic regression model
full_model <- glm(target ~ age + sex + cp + trestbps + chol + fbs + restecg + thalach + exang

# Compare the models using anova
anova_result <- anova(null_model, full_model, test = "Chi")
print(anova_result)
```

```
Analysis of Deviance Table

Model 1: target ~ 1
Model 2: target ~ age + sex + cp + trestbps + chol + fbs + restecg + thalach +
    exang + oldpeak + slope + ca + thal
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1       302     417.64
2       289     211.44 13    206.2 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

# 4. Implement any Machine learning Algorithm along with feature selection and data visualization on any dataset of your choice.

In [36]:
```
# Load necessary libraries
library(randomForest)
library(caret)
library(ggplot2)
```

In [39]:
```
titanic <-read.csv("Titanic.csv")
```

In [43]:
```
titanic$Survived <- factor(titanic$Survived)
titanic$Age[is.na(titanic$Age)] <-meantitanic$Age()
titanic$Fare[is.na(titanic$Fare)]<- mean(titanic$Fare)
titanic$Embarked[is.na(titanic$Embarked)] <-"Unknown"
```

In [29]:
```
# Fit a Random Forest model
rf_model <- randomForest(Species ~ ., data = train_data, ntree = 500)
```

```
# Load required libraries
library(randomForest)
library(caret)
library(ggplot2)

# Load the Titanic dataset
titanic <- read.csv("titanic.csv")

# Explore the dataset
head(titanic)
summary(titanic)

# Convert Survived to factor
titanic$Survived <- factor(titanic$Survived)

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

# Data Visualization
# Barplot of Survived by Sex
ggplot(titanic, aes(x = Sex, fill = Survived)) +
  geom_bar(position = "dodge") +
  labs(x = "Sex", y = "Count", fill = "Survived", title = "Barplot of Survived by Sex") +
  theme_minimal()

# Barplot of Survived by Pclass
ggplot(titanic, aes(x = factor(Pclass), fill = Survived)) +
  geom_bar(position = "dodge") +
  labs(x = "Pclass", y = "Count", fill = "Survived", title = "Barplot of Survived by Pclass") +
  theme_minimal()

# Perform feature selection using caret package (wrapper method)
set.seed(123)
ctrl <- rfeControl(functions = rfFuncs, method = "cv", number = 10)
```

```r
feature_selection <- rfe(titanic[, -c(1, 4, 9)], titanic$Survived, sizes = c(1:8), rfeControl = ctrl)

# Get selected features
selected_features <- feature_selection$optVariables
selected_features

# Subset the data with selected features
titanic_subset <- titanic[, c("Survived", selected_features)]

# Split data into training and testing sets
set.seed(456)
train_index <- createDataPartition(titanic_subset$Survived, p = 0.7, list = FALSE)
train_data <- titanic_subset[train_index, ]
test_data <- titanic_subset[-train_index, ]

# Train Random Forest model
rf_model <- randomForest(Survived ~ ., data = train_data, ntree = 500)

# Predict on test data
predictions <- predict(rf_model, newdata = test_data)

# Model evaluation
confusion_matrix <- table(test_data$Survived, predictions)
accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)
accuracy

# Data Visualization of model performance
# Confusion Matrix
# Data Visualization of model performance
# Confusion Matrix
confusion_matrix <- as.data.frame(confusion_matrix)
colnames(confusion_matrix) <- c("Actual", "Predicted", "Frequency")

confusion_matrix_plot <- ggplot(confusion_matrix, aes(x = Actual, y = Predicted, fill = Frequency)) +
  geom_tile(color = "white") +
  geom_text(aes(label = sprintf("%1.0f", Frequency)), vjust = 1) +
```

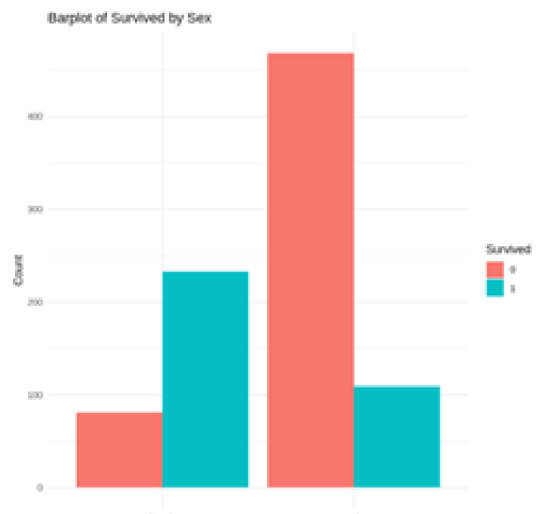

```r
# Confusion Matrix
confusion_matrix <- as.data.frame(confusion_matrix)
colnames(confusion_matrix) <- c("Actual", "Predicted", "Frequency")

confusion_matrix_plot <- ggplot(confusion_matrix, aes(x = Actual, y = Predicted, fill = Frequency)) +
  geom_tile(color = "white") +
  geom_text(aes(label = sprintf("%1.0f", Frequency)), vjust = 1) +
```

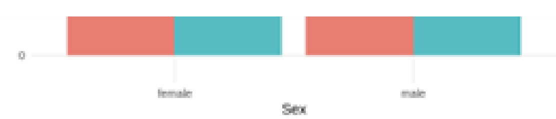

```r
  labs(x = "Actual", y = "Predicted", fill = "Frequency", title = "Confusion Matrix") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

print(confusion_matrix_plot)
```

| | PassengerId | Survived | Pclass | Name | Sex | Age | SibSp | Parch | Ticket | Fare | Cabin | Embarked |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | <int> | <int> | <int> | <chr> | <chr> | <dbl> | <int> | <int> | <chr> | <dbl> | <chr> | <chr> |
| 1 | 1 | 0 | 3 | Braund, Mr. Owen Harris | male | 22 | 1 | 0 | A/5 21171 | 7.2500 | | S |
| 2 | 2 | 1 | 1 | Cumings, Mrs. John Bradley (Florence Briggs Thayer) | female | 38 | 1 | 0 | PC 17599 | 71.2833 | C85 | C |
| 3 | 3 | 1 | 3 | Heikkinen, Miss. Laina | female | 26 | 0 | 0 | STON/O2. 3101282 | 7.9250 | | S |
| 4 | 4 | 1 | 1 | Futrelle, Mrs. Jacques Heath (Lily May Peel) | female | 35 | 1 | 0 | 113803 | 53.1000 | C123 | S |
| 5 | 5 | 0 | 3 | Allen, Mr. William Henry | male | 35 | 0 | 0 | 373450 | 8.0500 | | S |
| 6 | 6 | 0 | 3 | Moran, Mr. James | male | NA | 0 | 0 | 330877 | 8.4583 | | Q |

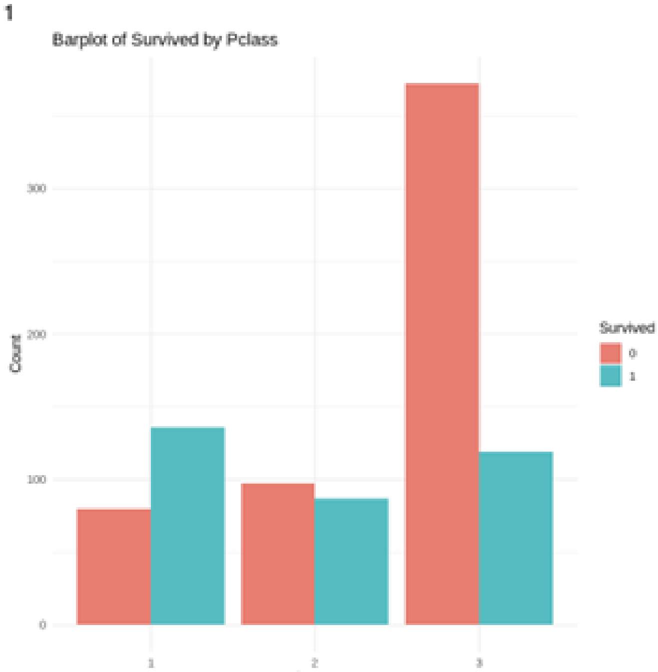
```
  PassengerId       Survived          Pclass           Name          
 Min.   :  1.0   Min.   :0.0000   Min.   :1.000   Length:891        
 1st Qu.:223.5   1st Qu.:0.0000   1st Qu.:2.000   Class :character  
 Median :446.0   Median :0.0000   Median :3.000   Mode  :character  
 Mean   :446.0   Mean   :0.3838   Mean   :2.309                     
 3rd Qu.:668.5   3rd Qu.:1.0000   3rd Qu.:3.000                     
 Max.   :891.0   Max.   :1.0000   Max.   :3.000                     
```

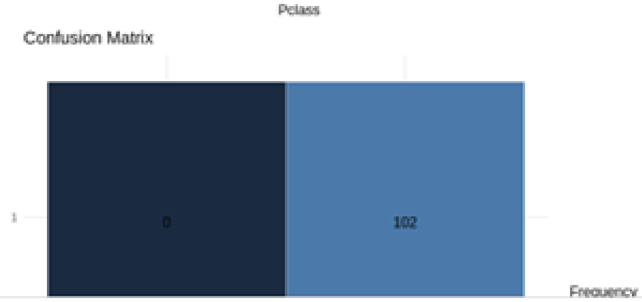
```
      Sex                 Age            SibSp           Parch       
 Length:891         Min.   : 0.42   Min.   :0.000   Min.   :0.0000  
 Class :character   1st Qu.:20.12   1st Qu.:0.000   1st Qu.:0.0000  
 Mode  :character   Median :28.00   Median :0.000   Median :0.0000  
                    Mean   :29.70   Mean   :0.523   Mean   :0.3816  
                    3rd Qu.:38.00   3rd Qu.:1.000   3rd Qu.:0.0000  
                    Max.   :80.00   Max.   :8.000   Max.   :6.0000  
                    NA's   :177                                     
    Ticket               Fare           Cabin             Embarked        
 Length:891         Min.   :  0.00   Length:891         Length:891        
 Class :character   1st Qu.:  7.91   Class :character   Class :character  
 Mode  :character   Median : 14.45   Mode  :character   Mode  :character  
                    Mean   : 32.20                                        
                    3rd Qu.: 31.00                                        
                    Max.   :512.33                                        
```

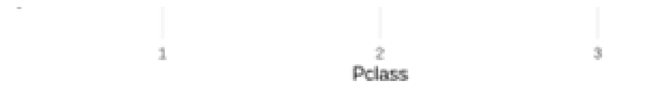


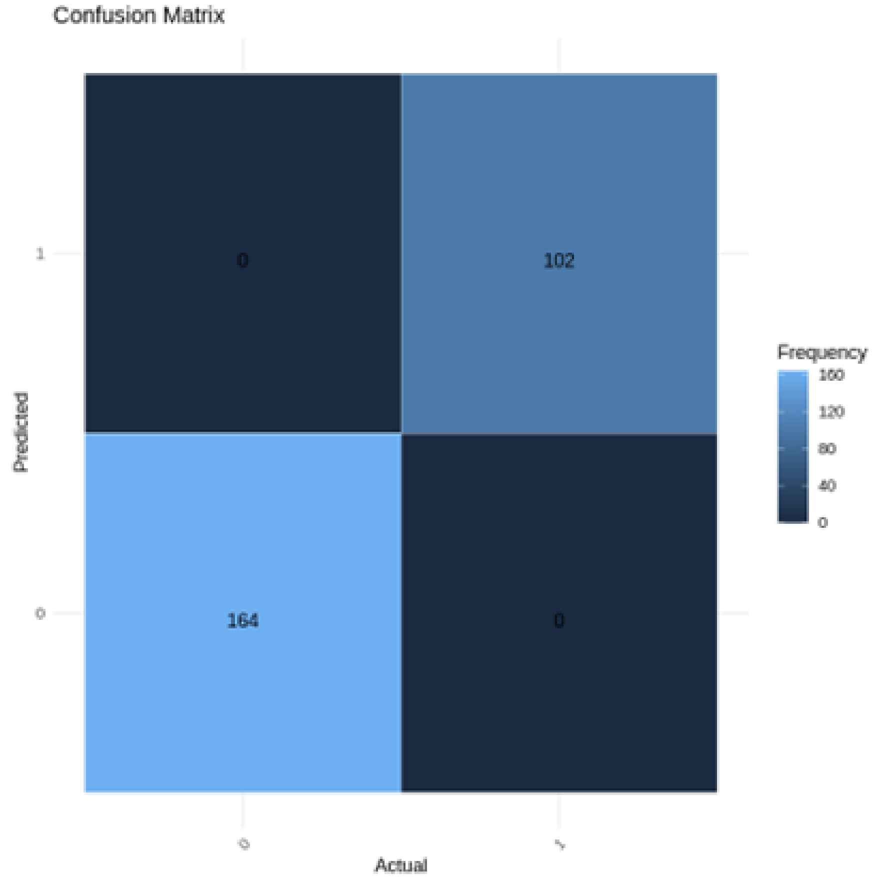




'Survived'
1




1
2
3
Pclass


Confusion Matrix
Predicted
Actual
1
0
0
1
0
102
164
0
Frequency
160
120
80
40
0