NAME: ujwal sahu BATCH:02

SUB: R-PROGRAMMING PRACTICAL-03 SYBVOC SEM IV

Implement Linear Regression Algorithms

AIM:

Implement Linear Regression algorithms

THEORY:

Linear regression is a statistical method used to model the relationship between a dependent variable (target) and one or more independent variables (predictors). The goal is to find the linear equation that best predicts the dependent variable based on the independent variables.

PRACTICAL 3 :

CODE :-

```
install.packages("caret")

library(tidyverse)

library(caret)

url <- "https://raw.githubusercontent.com/mwaskom/seaborn-data/master/mpg.csv"

data <- read.csv(url)

write.csv(data, "mpg_data.csv", row.names = FALSE)

head(data)

str(data)

summary(data)

data <- na.omit(data)

data <- data %>% select(mpg, horsepower, weight)

model <- lm(mpg ~ horsepower + weight, data = data)

summary(model)

cat("R-squared:", summary(model)$r.squared, "\n")

par(mfrow = c(2, 2))

plot(model)

new_data <- data.frame(horsepower = c(100, 150), weight = c(3000, 3500))
```

predictions <- predict(model, new_data)

print(predictions)

saveRDS(model, "linear_model.rds")

loaded_model <- readRDS("linear_model.rds")

new_predictions <- predict(loaded_model, new_data)

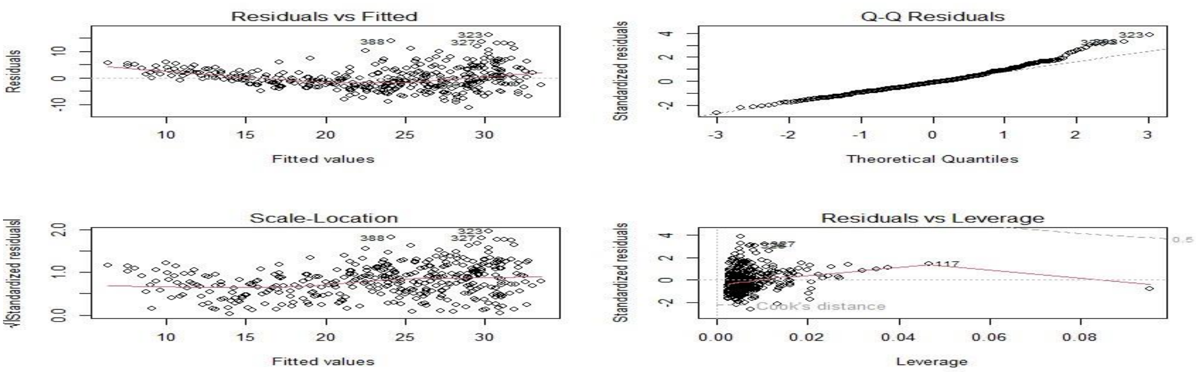
print(new_predictions)

```
install.packages("caret")
install.packages("tidyverse")
library(tidyverse)
library(caret)
```

```
> # Load the dataset from the provided URL
> url <- "https://raw.githubusercontent.com/mwaskom/seaborn-data/master/mpg.csv"
> data <- read.csv(url)
>
> # Save the dataset as a CSV file locally
> write.csv(data, "mpg_data.csv", row.names = FALSE)
>
> # Display the first few rows of the data
> head(data)
  mpg cylinders displacement horsepower weight acceleration model_year origin
1  18         8          307        130   3504         12.0         70    usa
2  15         8          350        165   3693         11.5         70    usa
3  18         8          318        150   3436         11.0         70    usa
4  16         8          304        150   3433         12.0         70    usa
5  17         8          302        140   3449         10.5         70    usa
6  15         8          429        198   4341         10.0         70    usa
                       name
1 chevrolet chevelle malibu
2         buick skylark 320
3        plymouth satellite
4             amc rebel sst
5               ford torino
6          ford galaxie 500
>
> # Display the structure of the dataset (column types, data types, etc.)
> str(data)
'data.frame':	398 obs. of  9 variables:
 $ mpg         : num  18 15 18 16 17 15 14 14 14 15 ...
 $ cylinders   : int  8 8 8 8 8 8 8 8 8 8 ...
 $ displacement: num  307 350 318 304 302 429 454 440 455 390 ...
 $ horsepower  : num  130 165 150 150 140 198 220 215 225 190 ...
 $ weight      : int  3504 3693 3436 3433 3449 4341 4354 4312 4425 3850 ...
 $ acceleration: num  12 11.5 11 12 10.5 10 9 8.5 10 8.5 ...
 $ model_year  : int  70 70 70 70 70 70 70 70 70 70 ...
 $ origin      : chr  "usa" "usa" "usa" "usa" ...
 $ name        : chr  "chevrolet chevelle malibu" "buick skylark 320" "plymouth satel
lite" "amc rebel sst" ...
```

```
> # Summary statistics of the dataset
> summary(data)
      mpg          cylinders      displacement     horsepower
 Min.   : 9.00   Min.   :3.000   Min.   : 68.0   Min.   : 46.0
 1st Qu.:17.50   1st Qu.:4.000   1st Qu.:104.2   1st Qu.: 75.0
 Median :23.00   Median :4.000   Median :148.5   Median : 93.5
 Mean   :23.51   Mean   :5.455   Mean   :193.4   Mean   :104.5
 3rd Qu.:29.00   3rd Qu.:8.000   3rd Qu.:262.0   3rd Qu.:126.0
 Max.   :46.60   Max.   :8.000   Max.   :455.0   Max.   :230.0
                                                 NA's   :6
     weight      acceleration     model_year       origin
 Min.   :1613   Min.   : 8.00   Min.   :70.00   Length:398
 1st Qu.:2224   1st Qu.:13.82   1st Qu.:73.00   Class :character
 Median :2804   Median :15.50   Median :76.00   Mode  :character
 Mean   :2970   Mean   :15.57   Mean   :76.01
 3rd Qu.:3608   3rd Qu.:17.18   3rd Qu.:79.00
 Max.   :5140   Max.   :24.80   Max.   :82.00

     name
 Length:398
 Class :character
 Mode  :character

> # Remove rows with missing values
> data <- na.omit(data)
>
> # Select relevant columns: mpg, horsepower, and weight
> data <- data %>% select(mpg, horsepower, weight)
>
> # Build a linear model to predict mpg based on horsepower and weight
> model <- lm(mpg ~ horsepower + weight, data = data)
>
> # Display the summary of the model
> summary(model)

Call:
lm(formula = mpg ~ horsepower + weight, data = data)

Residuals:
     Min       1Q   Median       3Q      Max
-11.0762  -2.7340  -0.3312   2.1752  16.2601

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) 45.6402108  0.7931958  57.540  < 2e-16 ***
horsepower  -0.0473029  0.0110851  -4.267 2.49e-05 ***
weight      -0.0057942  0.0005023 -11.535  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.24 on 389 degrees of freedom
Multiple R-squared:  0.7064,    Adjusted R-squared:  0.7049
F-statistic: 467.9 on 2 and 389 DF,  p-value: < 2.2e-16
```

```
> # Display the R-squared value of the model
> cat("R-squared:", summary(model)$r.squared, "\n")
R-squared: 0.7063753
>
> # Create diagnostic plots for the linear model
> par(mfrow = c(2, 2))
> plot(model)
```




```
> # Create new data for prediction with horsepower and weight values
> new_data <- data.frame(horsepower = c(100, 150), weight = c(3000, 3500))
>
> # Make predictions using the linear model
> predictions <- predict(model, new_data)
>
> # Print the predictions
> print(predictions)
       1        2
23.52745 18.26523
>
> # Save the model as an RDS file for future use
> saveRDS(model, "linear_model.rds")
>
> # Load the saved model from the RDS file
> loaded_model <- readRDS("linear_model.rds")
>
> # Make predictions with the loaded model
> new_predictions <- predict(loaded_model, new_data)
>
> # Print the predictions from the loaded model
> print(new_predictions)
       1        2
23.52745 18.26523
```

Conclusion : The key takeaways from this implementation are:

- Linear regression assumes a linear relationship between the dependent and independent variables.
- The model can be easily fitted and evaluated using R.
- Visualization helps in understanding the relationship and the fit of the model.

For Faculty Use

Correction Parameters	Formative Assessment [40%]	Timely completion of Practical [40%]	Attendance / Learning Attitude [20%]	
Marks Obtained				