TCET R-PROGRAMMING AIDS-34

NAME: ujwal sahu BATCH:02

SUB: R-PROGRAMMING PRACTICAL-04 SYBVOC SEM IV

Implement Non-Linear Regression algorithms

AIM:

Implement Non-Linear Regression algorithms

THEORY:

Non-linear regression is a form of regression analysis in which the relationship between the independent variable(s) and the dependent variable is modeled as a non-linear function. Unlike linear regression, which assumes a straight-line relationship, non-linear regression can capture more complex relationships.

PRACTICAL 4 :

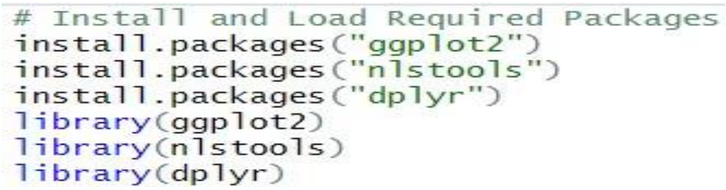
CODE :-

```
library(ggplot2)
library(nlstools)
library(dplyr)

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

str(data)
summary(data)

# Select relevant columns: horsepower (X) and mpg (Y)
data <- data %>% select(horsepower, mpg) %>% na.omit()

# Rename columns for clarity
colnames(data) <- c("X", "Y")
```

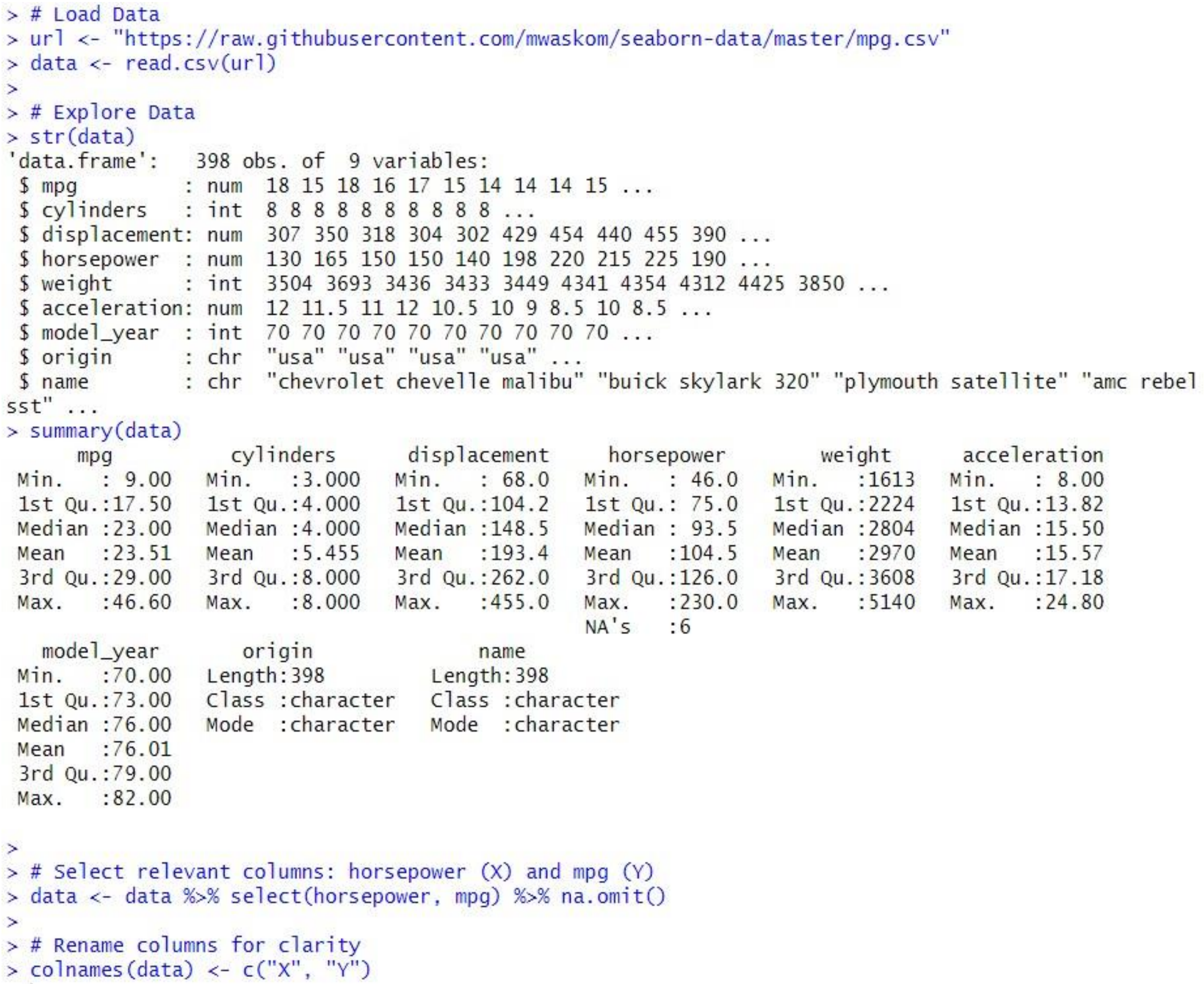
```
# Scatter plot of data
ggplot(data, aes(x = X, y = Y)) +
  geom_point() +
  ggtitle("Scatter Plot of MPG vs Horsepower")

# Define a non-linear function (Example: Exponential Model Y = a * exp(b * X))
nonlinear_model <- nls(Y ~ a * exp(b * X), data = data, start = list(a = 40, b = -0.01))

# Model summary
summary(nonlinear_model)

# Predict values
data$predicted <- predict(nonlinear_model)

# Plot actual vs. predicted values
ggplot(data, aes(x = X)) +
  geom_point(aes(y = Y, color = "Actual")) +
  geom_line(aes(y = predicted, color = "Predicted")) +
  ggtitle("Actual vs Predicted MPG Values") +
  scale_color_manual(values = c("Actual" = "blue", "Predicted" = "red"))

# Residual analysis
residuals <- residuals(nonlinear_model)
hist(residuals, main = "Residuals Distribution", col = "gray")

# Confidence intervals
confint(nonlinear_model)

# Goodness of fit (R-Squared Calculation)
```

r_squared <- 1 - (sum(residuals^2) / sum((data$Y - mean(data$Y))^2))

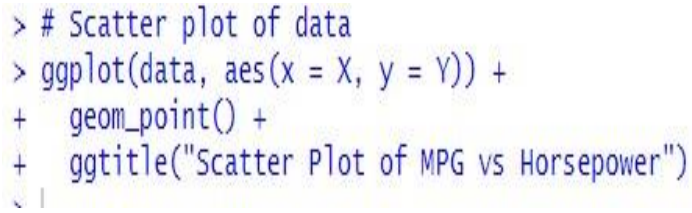
print(paste("R-squared:", round(r_squared, 4)))



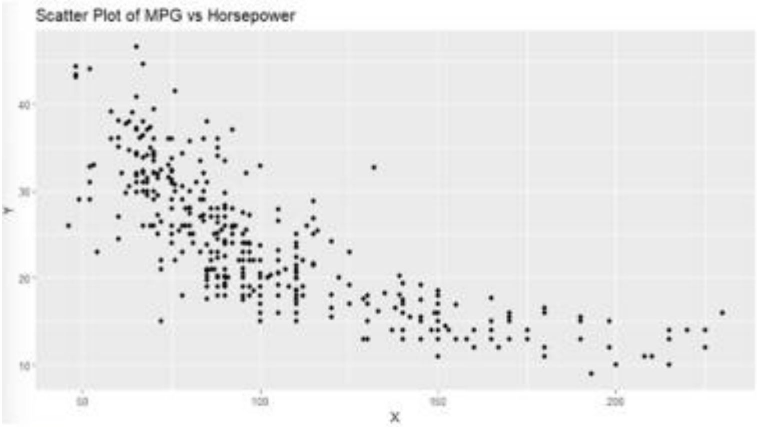
```
# Install and Load Required Packages
install.packages("ggplot2")
install.packages("nlstools")
install.packages("dplyr")
library(ggplot2)
library(nlstools)
library(dplyr)
```

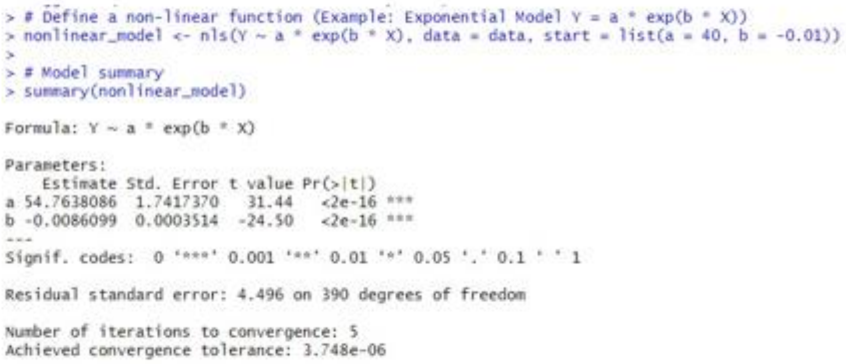


```
> # Load Data
> url <- "https://raw.githubusercontent.com/mwaskom/seaborn-data/master/mpg.csv"
> data <- read.csv(url)
>
> # Explore Data
> str(data)
'data.frame':   398 obs. of  9 variables:
 $ mpg         : num  18 15 18 16 17 15 14 14 14 15 ...
 $ cylinders   : int  8 8 8 8 8 8 8 8 8 8 ...
 $ 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 satellite" "amc rebel
sst" ...
> summary(data)
      mpg          cylinders      displacement     horsepower        weight      acceleration  
 Min.   : 9.00   Min.   :3.000   Min.   : 68.0   Min.   : 46.0   Min.   :1613   Min.   : 8.00  
 1st Qu.:17.50   1st Qu.:4.000   1st Qu.:104.2   1st Qu.: 75.0   1st Qu.:2224   1st Qu.:13.82  
 Median :23.00   Median :4.000   Median :148.5   Median : 93.5   Median :2804   Median :15.50  
 Mean   :23.51   Mean   :5.455   Mean   :193.4   Mean   :104.5   Mean   :2970   Mean   :15.57  
 3rd Qu.:29.00   3rd Qu.:8.000   3rd Qu.:262.0   3rd Qu.:126.0   3rd Qu.:3608   3rd Qu.:17.18  
 Max.   :46.60   Max.   :8.000   Max.   :455.0   Max.   :230.0   Max.   :5140   Max.   :24.80  
                                                 NA's   :6                                     
   model_year       origin              name          
 Min.   :70.00   Length:398         Length:398        
 1st Qu.:73.00   Class :character   Class :character  
 Median :76.00   Mode  :character   Mode  :character  
 Mean   :76.01                                        
 3rd Qu.:79.00                                        
 Max.   :82.00                                        

>
> # Select relevant columns: horsepower (X) and mpg (Y)
> data <- data %>% select(horsepower, mpg) %>% na.omit()
>
> # Rename columns for clarity
> colnames(data) <- c("X", "Y")
```

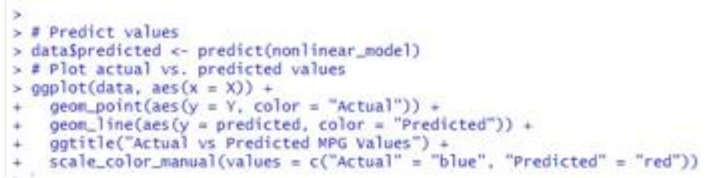


```
> # Scatter plot of data
> ggplot(data, aes(x = X, y = Y)) +
+   geom_point() +
+   ggtitle("Scatter Plot of MPG vs Horsepower")
```

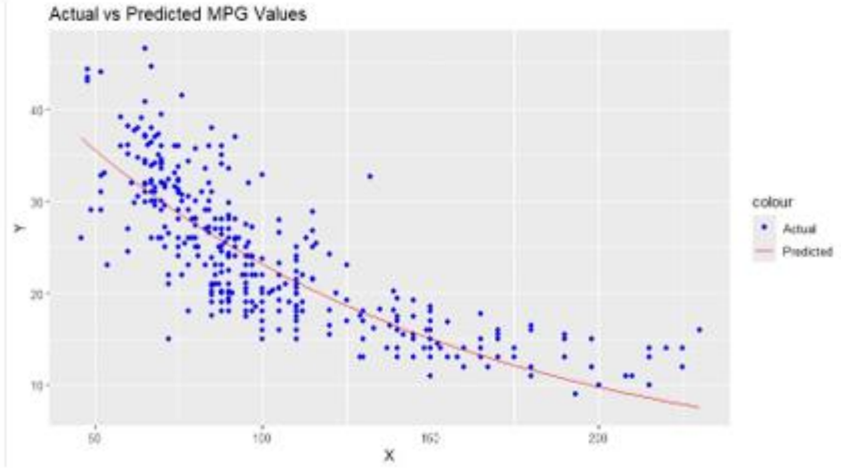


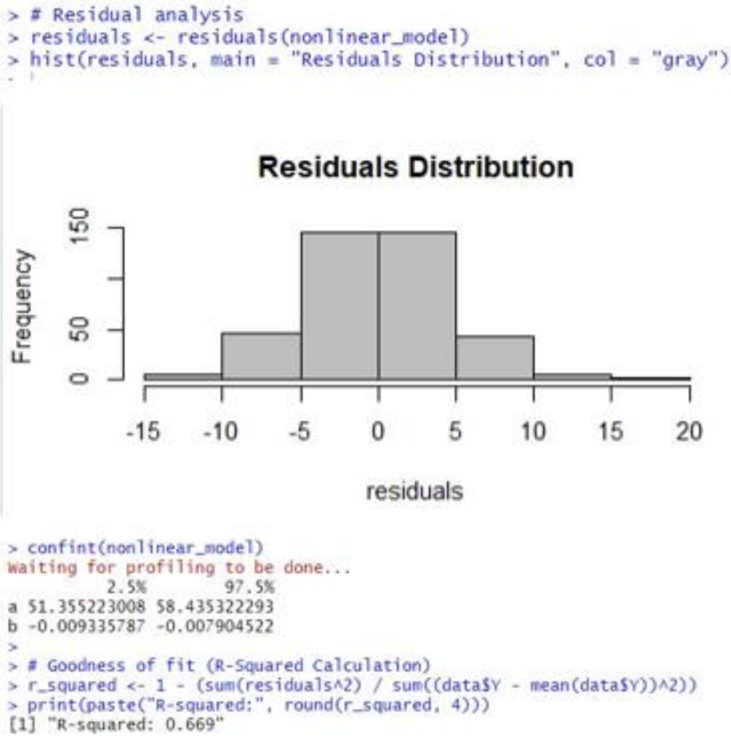


```
> # Define a non-linear function (Example: Exponential Model Y = a * exp(b * X))
> nonlinear_model <- nls(Y ~ a * exp(b * X), data = data, start = list(a = 40, b = -0.01))
>
> # Model summary
> summary(nonlinear_model)

Formula: Y ~ a * exp(b * X)

Parameters:
    Estimate Std. Error t value Pr(>|t|)
a 54.7638086  1.7417370   31.44   <2e-16 ***
b -0.0086099  0.0003514  -24.50   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.496 on 390 degrees of freedom

Number of iterations to convergence: 5
Achieved convergence tolerance: 3.748e-06
```



```
>
> # Predict values
> data$predicted <- predict(nonlinear_model)
> # Plot actual vs. predicted values
> ggplot(data, aes(x = X)) +
+   geom_point(aes(y = Y, color = "Actual")) +
+   geom_line(aes(y = predicted, color = "Predicted")) +
+   ggtitle("Actual vs Predicted MPG Values") +
+   scale_color_manual(values = c("Actual" = "blue", "Predicted" = "red"))
```





```
> # Residual analysis
> residuals <- residuals(nonlinear_model)
> hist(residuals, main = "Residuals Distribution", col = "gray")
```

```
> confint(nonlinear_model)
Waiting for profiling to be done...
          2.5%       97.5%
a 51.355223008 58.435322293
b -0.009335787 -0.007904522
>
> # Goodness of fit (R-Squared Calculation)
> r_squared <- 1 - (sum(residuals^2) / sum((data$Y - mean(data$Y))^2))
> print(paste("R-squared:", round(r_squared, 4)))
[1] "R-squared: 0.669"
```

Conclusion :

Key takeaways from this implementation include:

- Non-linear regression can model complex relationships that linear regression cannot.
- The function in R allows for flexible modeling of non-linear relationships.
- Visualization is crucial for understanding the fit of the model to the data.

For Faculty Use

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