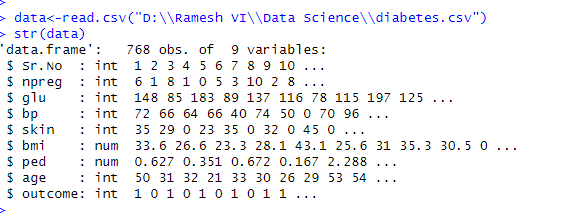
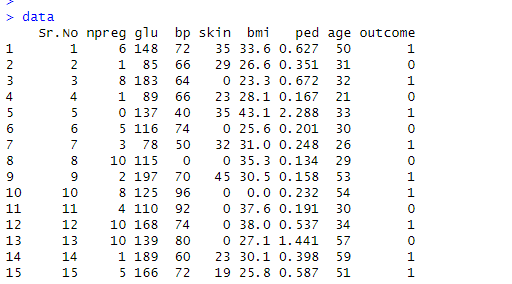
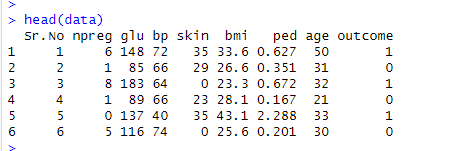
* **Logistic Regression**

1. **Load the csv file**

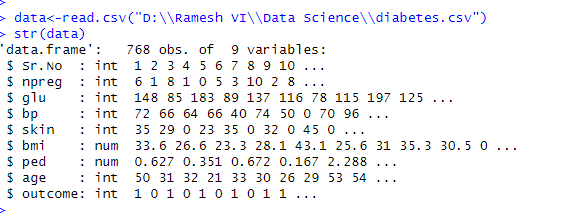
****

****

1. **Display the original dataset**

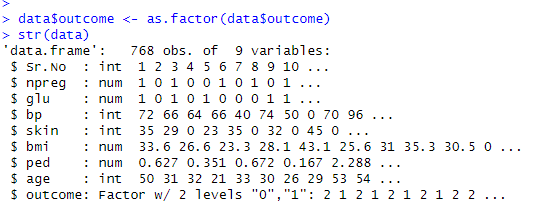
****

1. **Structure of dataset**

**>**

1. **Convert the 'outcome' variable to a factor**

**data$outcome <- as.factor(data$outcome)**

****

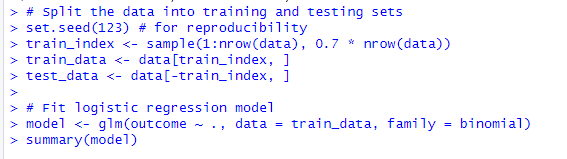
1. **# Split the data into training and testing sets**

**set.seed(123) # for reproducibility**

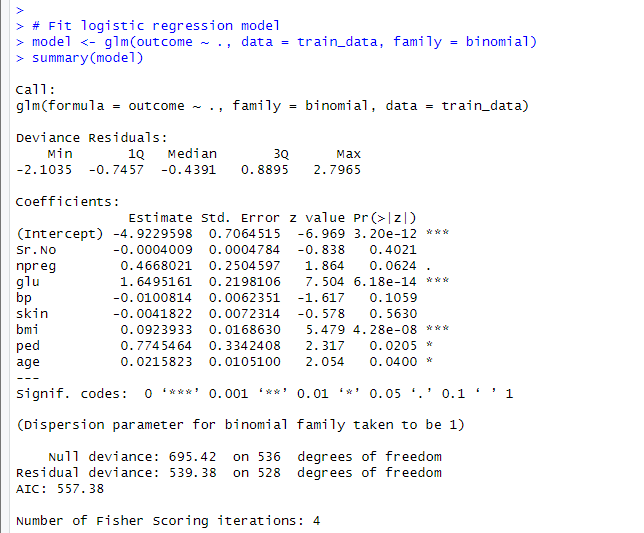
**train\_index <- sample(1:nrow(data), 0.7 \* nrow(data))**

**train\_data <- data[train\_index, ]**

**test\_data <- data[-train\_index, ]**

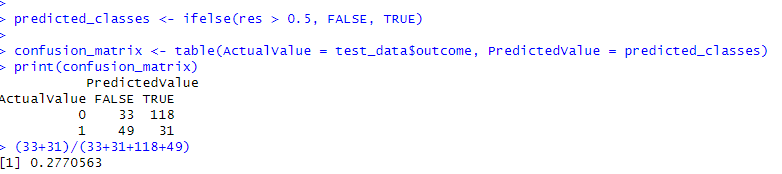
****

1. **Fit logistic regression model**

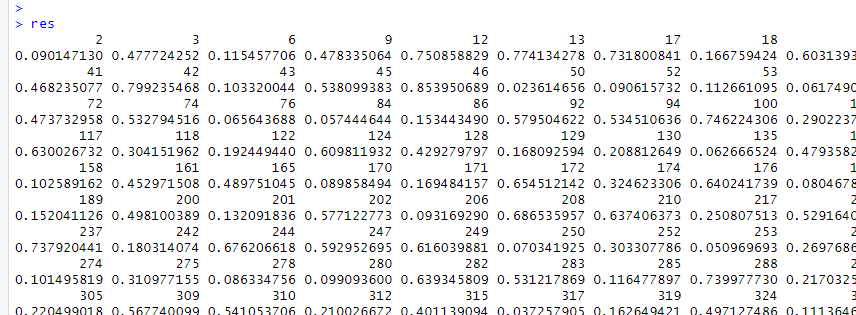
****

1. **Create confusion matrix for training dataset and finding accuracy**

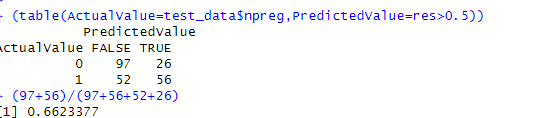
****

****

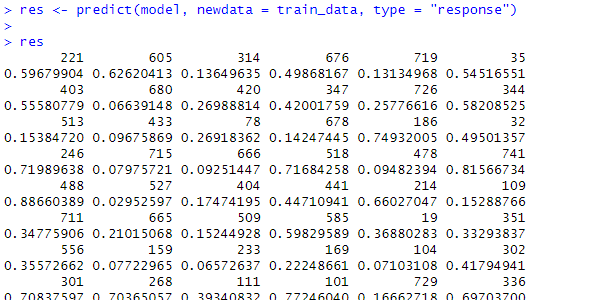
1. **Res**

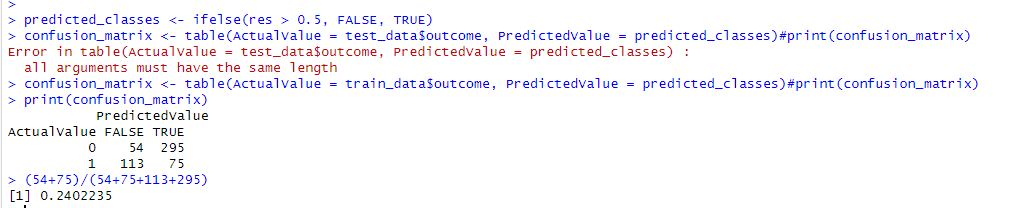
****

1. **Create table and find the accuracy**

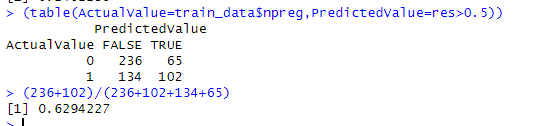
****

1. **Predict on train\_data and find the accuracy**

****

****

1. **Create table on train data and find the accuracy**

****

* **Logistic Regression using python**

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import classification\_report, confusion\_matrix

# Load the CSV file into a DataFrame

data = pd.read\_csv("D:\\Ramesh VI\\Data Science\\diabetes.csv")

# Display the first few rows of the dataset

print(data.head())

# Separate the predictors (X) and the target variable (y)

X = data.drop(columns=["outcome"]) # Assuming "outcome" is the target variable

y = data["outcome"]

# Split the data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

# Fit logistic regression model

model = LogisticRegression()

model.fit(X\_train, y\_train)

# Predict on the testing data

y\_pred = model.predict(X\_test)

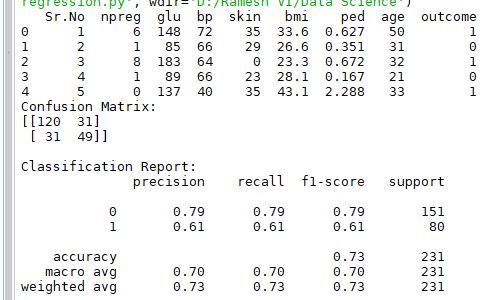
# Evaluate the model

print("Confusion Matrix:")

print(confusion\_matrix(y\_test, y\_pred))

print("\nClassification Report:")

print(classification\_report(y\_test, y\_pred))

****