SAVEETHA SCHOOL OF ENGINEERING

SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES

ITA 0443 - STATISTICS WITH R PROGRAMMING FOR REAL TIME PROBLEM

DAY 4– LAB MANUAL Part 2

Reg No: 192011283

Name: SREEJA YENNAM

LOGISTIC REGRESSION ANALYSIS IN R

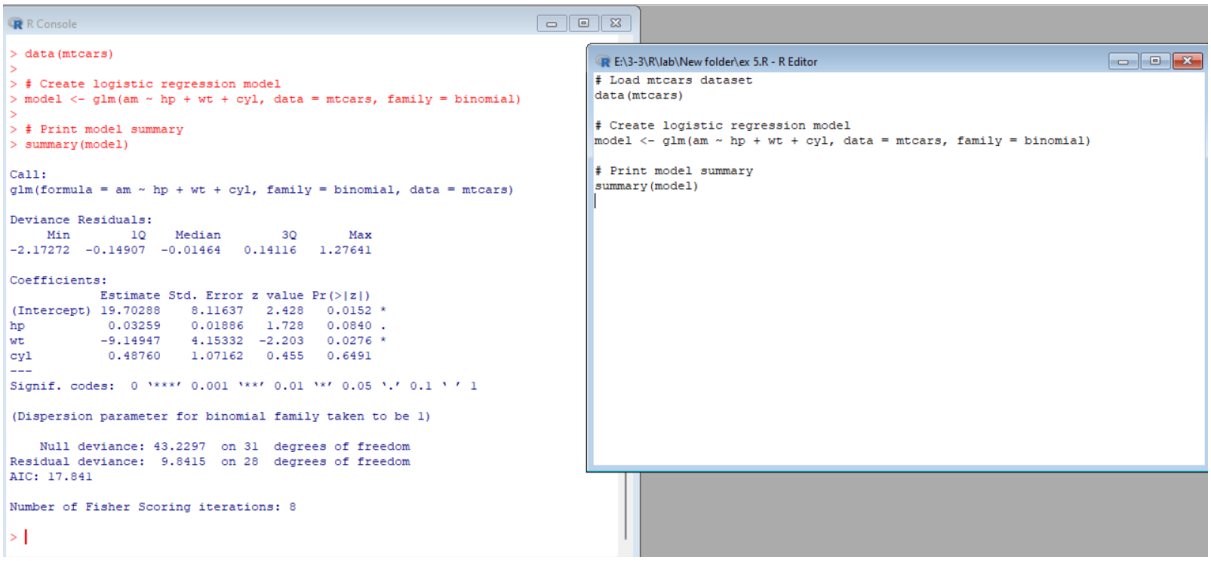
Exercise

5. Create a logistic regression model using the “mtcars” data set with the information given below.

The in-built data set "mtcars" describes different models of a car with their various engine specifications. In "mtcars" data set, the transmission mode (automatic or manual) is described by the column am which is a binary value (0 or 1). Create a logistic regression model between the columns "am" and 3 other columns - hp, wt and cyl.

SOURCE CODE:

```
# Load mtcars dataset
data(mtcars)

# Create logistic regression model
model <- glm(am ~ hp + wt + cyl, data = mtcars, family = binomial)

# Print model summary
summary(model)
```



POISSON REGRESSION ANALYSIS IN R

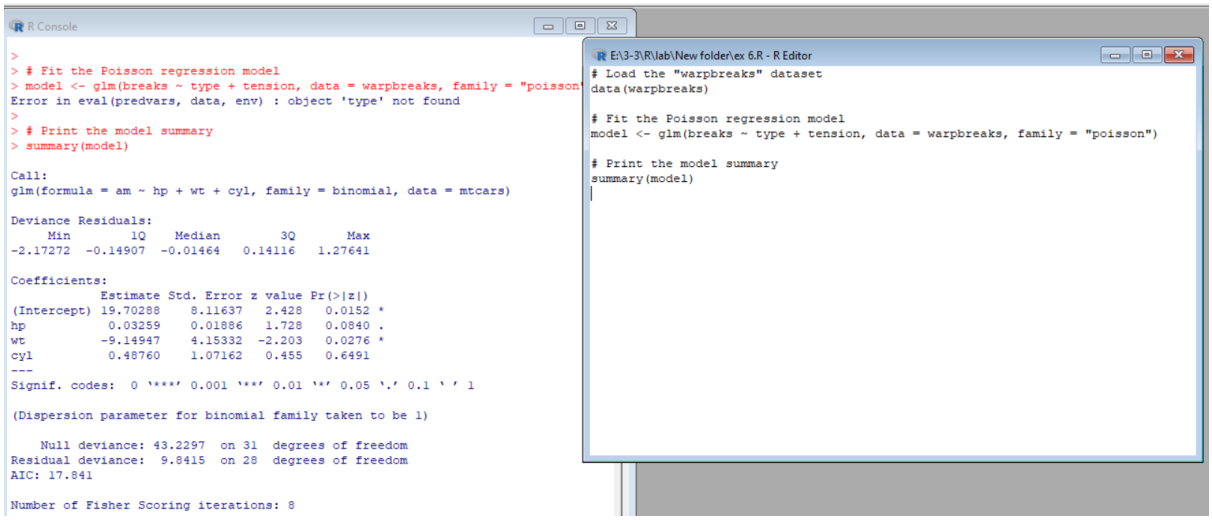
Exercise :

6. Create a Poisson regression model using the in-built data set “warpbreaks” with information given below.

In-built data set "warpbreaks” describes the effect of wool type (A or B) and tension (low, medium or high) on the number of warp breaks per loom. Consider "breaks" as the response variable which is a count of number of breaks. The wool "type" and "tension" are taken as predictor variables.

SOURCE CODE:

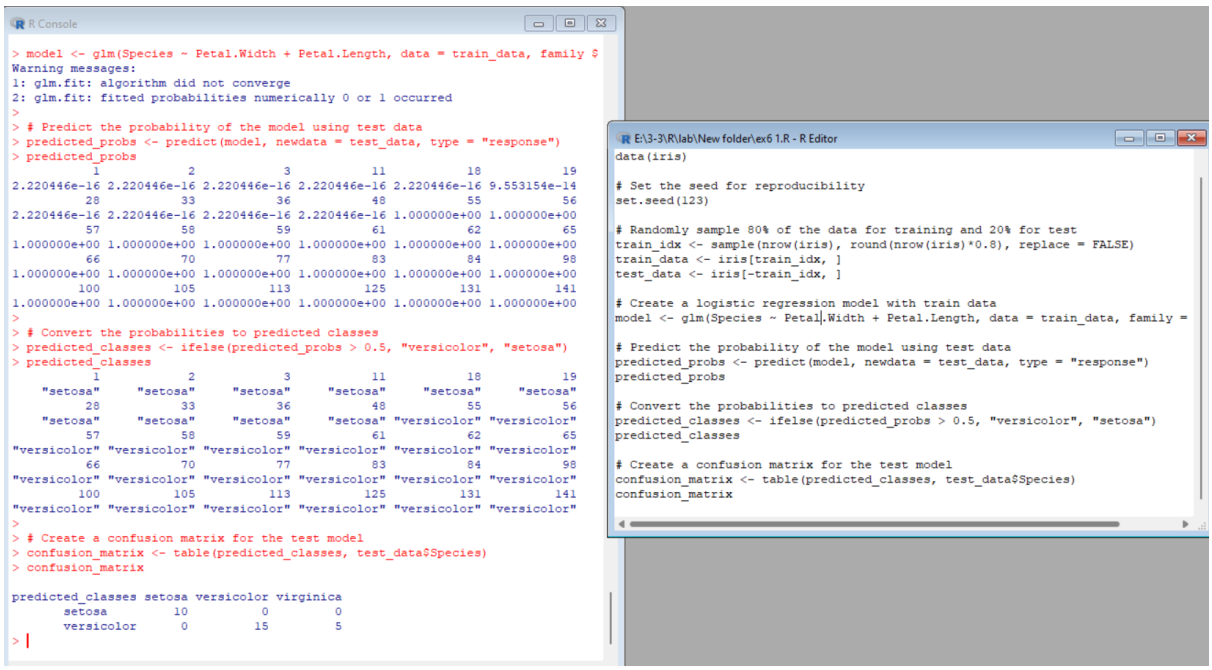
```
# Load the "warpbreaks" dataset
data(warpbreaks)

# Fit the Poisson regression model
model <- glm(breaks ~ type + tension, data = warpbreaks, family = "poisson")

# Print the model summary
summary(model)
```



1.Randomly Sample the iris dataset such as 80% data for training and 20% for test and create Logistics regression with train data, use species as target and petals width and length as feature variables , Predict the probability of the model using test data, Create Confusion matrix for above test model

SOURCE CODE:

```
# Load the iris dataset
data(iris)

# Set the seed for reproducibility
set.seed(123)

# Randomly sample 80% of the data for training and 20% for test
train_idx <- sample(nrow(iris), round(nrow(iris)*0.8), replace = FALSE)
train_data <- iris[train_idx, ]
test_data <- iris[-train_idx, ]

# Create a logistic regression model with train data
model <- glm(Species ~ Petal.Width + Petal.Length, data = train_data, family = "binomial")

# Predict the probability of the model using test data
predicted_probs <- predict(model, newdata = test_data, type = "response")
predicted_probs

# Convert the probabilities to predicted classes
predicted_classes <- ifelse(predicted_probs > 0.5, "versicolor", "setosa")
predicted_classes

# Create a confusion matrix for the test model
confusion_matrix <- table(predicted_classes, test_data$Species)
confusion_matrix
```


R Console
> model <- glm(Species ~ Petal.Width + Petal.Length, data = train_data, family $
Warning messages:
1: glm.fit: algorithm did not converge
2: glm.fit: fitted probabilities numerically 0 or 1 occurred
>
> # Predict the probability of the model using test data
> predicted_probs <- predict(model, newdata = test_data, type = "response")
> predicted_probs
1 2 3 11 18 19
2.220446e-16 2.220446e-16 2.220446e-16 2.220446e-16 2.220446e-16 9.553154e-14
28 33 36 48 55 56
2.220446e-16 2.220446e-16 2.220446e-16 2.220446e-16 1.000000e+00 1.000000e+00
57 58 59 61 62 65
1.000000e+00 1.000000e+00 1.000000e+00 1.000000e+00 1.000000e+00 1.000000e+00
66 70 77 83 84 98
1.000000e+00 1.000000e+00 1.000000e+00 1.000000e+00 1.000000e+00 1.000000e+00
100 105 113 125 131 141
1.000000e+00 1.000000e+00 1.000000e+00 1.000000e+00 1.000000e+00 1.000000e+00
>
> # Convert the probabilities to predicted classes
> predicted_classes <- ifelse(predicted_probs > 0.5, "versicolor", "setosa")
> predicted_classes
1 2 3 11 18 19
"setosa" "setosa" "setosa" "setosa" "setosa" "setosa"
28 33 36 48 55 56
"setosa" "setosa" "setosa" "setosa" "versicolor" "versicolor"
57 58 59 61 62 65
"versicolor" "versicolor" "versicolor" "versicolor" "versicolor" "versicolor"
66 70 77 83 84 98
"versicolor" "versicolor" "versicolor" "versicolor" "versicolor" "versicolor"
100 105 113 125 131 141
"versicolor" "versicolor" "versicolor" "versicolor" "versicolor" "versicolor"
>
> # Create a confusion matrix for the test model
> confusion_matrix <- table(predicted_classes, test_data$Species)
> confusion_matrix
predicted_classes setosa versicolor virginica
setosa 10 0 0
versicolor 0 15 5
>
E:\3-3\R\lab\New folder\ex6 1.R - R Editor
data(iris)
# Set the seed for reproducibility
set.seed(123)
# Randomly sample 80% of the data for training and 20% for test
train_idx <- sample(nrow(iris), round(nrow(iris)*0.8), replace = FALSE)
train_data <- iris[train_idx,]
test_data <- iris[-train_idx,]
# Create a logistic regression model with train data
model <- glm(Species ~ Petal.Width + Petal.Length, data = train_data, family =
# Predict the probability of the model using test data
predicted_probs <- predict(model, newdata = test_data, type = "response")
predicted_probs
# Convert the probabilities to predicted classes
predicted_classes <- ifelse(predicted_probs > 0.5, "versicolor", "setosa")
predicted_classes
# Create a confusion matrix for the test model
confusion_matrix <- table(predicted_classes, test_data$Species)
confusion_matrix