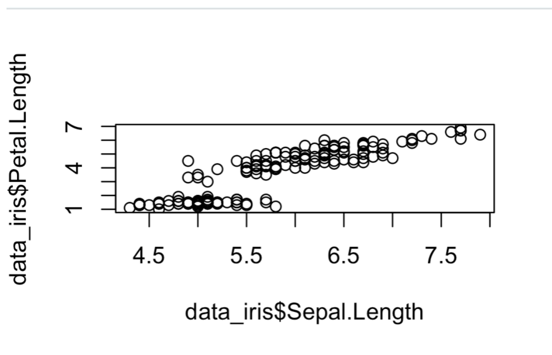
```r
# Data Science --- R --- Homework questions #2
# Student name: Mihir Sachdeva --- student ID: 14244197
# --- Excercise 6.7 --- Question 3


# calling iris and setting it as dataframe for 'data_iris'
data_iris <- as.data.frame(iris)

#viewing the iris data
View(data_iris)

# summary for the iris dataframe
summary(data_iris)

###################### Answers to question 3 part A ###########################

# Using the gsub function to create the groups by updating the binary variables
data_iris$Group <- gsub("setosa", "0", data_iris$Species)

data_iris$Group <- gsub("versicolor", "0", data_iris$Group)

data_iris$Group <- gsub("virginica", "1", data_iris$Group)

# changing the data type of the varaibles for the group created above
data_iris$Group <- as.numeric(data_iris$Group)

# summary for the iris dataframe
summary(data_iris)

#loading the library for scatter plots
library(ggplot2)

# checking the behavior of the group with the sepal length using scatter plot,
# violin and line
ggplot(data = data_iris,aes(x=Sepal.Length,y=Group))+
  geom_point()

ggplot(data = data_iris,aes(x=Sepal.Length,y=Group))+
  geom_violin()

ggplot(data = data_iris,aes(x=Sepal.Length,y=Group))+
  geom_line()

# _____ Analysis for 3A _____ #
# combined the Setosa and Versicolor into group “0” and labelled the Virginica as “1”.
# Created a new variable called data_iris$Group with the 0 or 1 labels.
# The max Sepal.length is 7.9 for this particular run and the max Petal.Length is 6.9
# which is pretty close to the competitive.
# On plotting the ggplot as a scatter plot we can see that it is a S shaped linear
# curve which also represents a A sigmoid function is a mathematical function
# having a characteristic "S"-shaped curve or sigmoid curve.
# The violin and line plots are just visual statistics that are of no use for
# any of our analysis. I have just used them to see what insights the outputs
# could potentially provide us.

###################### Answers to question 3 part B ###########################
# using the desc function from the class
desc_func <- function (x){
```

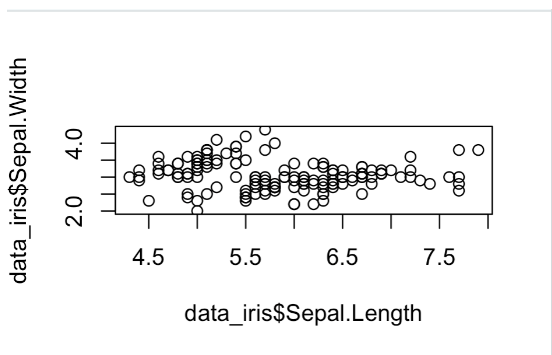
```
  min  <- try(min(x, na.rm=TRUE))
  mean <- try(mean(x, na.rm=TRUE))
  sd   <- try(sd(x, na.rm=TRUE))
  max  <- try(max(x, na.rm=TRUE))
  return(c(min, mean, sd, max))
} #Closing the i-loop

# using the Z score for standardization
s_function <- function(var){
  s_score <- (var - mean(var))/sd(var)
  return(s_score)
} #closing the standard function

# using the desc function from the class on the data_iris group
desc_func(s_function(var = data_iris$Group))

#recreating UDF - t score
s_function <- function(var){
  s_score  <- (var - mean(var))/sd(var)
  t_score  <- s_score*10 + 50
  return(t_score)
} #closing the function

#creating UDF - in order to normalize with min and max
n_function <- function(var){
  data_iris_norm <- (var - min(var))/(max(var) - min(var))
  return(data_iris_norm)
} #closing the norm func loop

data_iris$Sepal.Length_norm  <- n_function(var = data_iris$Sepal.Length)
data_iris$Sepal.Width_norm   <- n_function(var = data_iris$Sepal.Width)
data_iris$Petal.Length_norm  <- n_function(var = data_iris$Petal.Length)
data_iris$Petal.Width_norm   <- n_function(var = data_iris$Petal.Width)

#random sampling - training and testing
training_idx <- sample(1:nrow(data_iris), size = 0.8*nrow(data_iris))

data_iris_train <- data_iris[training_idx, ]
data_iris_test  <- data_iris[-training_idx, ]

#Building a logistic regression for data_iris
print(data_iris)

my_logit <- glm(Group~Petal.Length+Petal.Width+Sepal.Length+Sepal.Width,
                data = data_iris_train, family = "binomial")

#summary for the iris dataframe - logistic regression model
summary(my_logit)

# logistic model coefficients - saving as vector
logit_coeff <- my_logit$coefficients

# it is important to create the exponential extraction of our coefficients to
# be able to take business insights and make a decision accordingly
exponential_output <- exp(logit_coeff)

# printing exponential_output
print(exponential_output)
```

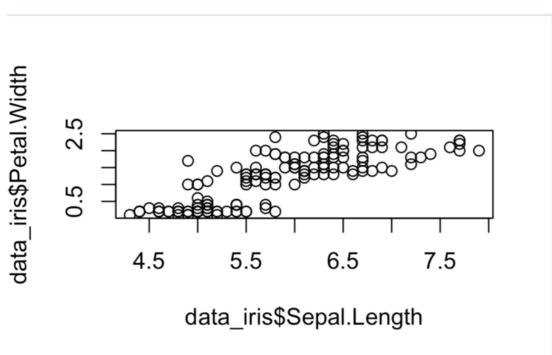
```
# _____ Analysis for 3B _____ #
# The p values for the coefficients are very good outputs and this shows that are
# train and test data is very suitable for our regression model.However, this
# data set is also close to an over fit due to the near 0 values of the
# Petal.length and the Petal.Width. These observations are key for further
# investigation of the data.
# The AIC for this test was 21.824 is low as per the expectation. It is known that
# the lower AIC score of the model means the lesser the significance of the
# model is in predicting the accuracy of the species.

###################### Answers to question 3 part C ##########################

# spotting the variables and declaring values for species prediction
# creating dataframe previously created variables
data_iris_prediction <- data.frame(Sepal.Width = 5, Petal.Length = 10,
                                   Petal.Width = 7, Sepal.Length = 9)

# predicting species
pred_val <- predict(my_logit, newdata = data_iris_prediction, type = "response")

# printing the output for species prediction
print(pred_val)

# _____ Analysis for 3C _____ #
# Calculated the probability of the new plant being a Virginica for the following parameters:
# Sepal.Width =5 Petal.Length =10 Petal.Width =7 Sepal.Length=9
# We cannot fully comment of the output of this prediction whether it is a true
# positive or a false negative.


###################### Answers to question 4 part A ##########################
# installing packages for rpart
install.packages("rpart")

# loading the library for rpart
library(rpart)

# printing the output for kyphosis
print(kyphosis)

# kyphosis - saving as a data frame
k_data <- as.data.frame(kyphosis)

summary(k_data)

# changing present value from kyphosis variable to 1
k_data$Kyphosis <- gsub("present", "1", k_data$Kyphosis)

# changing absent value from kyphosis variable to 0
k_data$Kyphosis <- gsub("absent", "0", k_data$Kyphosis)

# updating the data type of the kyphosis variable
k_data$Kyphosis <- as.numeric(k_data$Kyphosis)

# _____ Analysis for 4A _____ #
# Changing the present and absent values to 1 and 0 respectively
# This is to prepare the data frame for our training and testing
```

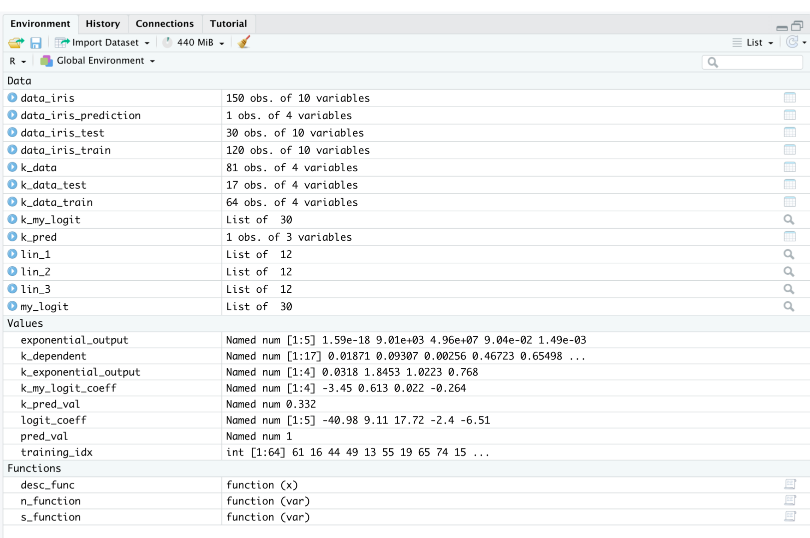
```
# converted the kyphosis$Kyphosis variable to numeric

###################### Answers to question 4 part B ##########################

#random sampling - training and testing

training_idx <- sample(1:nrow(k_data), size = 0.8*nrow(k_data))

k_data_train <- k_data[training_idx, ]
k_data_test  <- k_data[-training_idx, ]

#Building a logistic regression for kyphosis
k_my_logit <- glm(Kyphosis~Number+Age+Start,
            data = k_data_train, family = "binomial")

#summary for k_my_logit - logistic regression model
summary(k_my_logit)

# logistic model coefficients - saving as vector
k_my_logit_coeff <- k_my_logit$coefficients

# it is important to create the exponential extraction of our coefficients to
# be able to take business insights and make a decision accordingly
k_exponential_output <- exp(k_my_logit_coeff)

# printing k_exponential_output
print(k_exponential_output)

# predicting group
k_dependent <- predict(k_my_logit, newdata =  k_data_test, type = "response")

print(round(k_dependent, digits = 0))

# _____ Analysis for 4B _____ #
# The p value for start coefficients is that this value is highly significant
# for our data. This has been a growth from the previously created models in 3.
# The AIC of 51.2 is really 5 and this reflects that on the totality this model
# is less significant for our anaylsis and business decision making.

###################### Answers to question 4 part C ##########################

# spotting the variables and declaring values for k data
# creating dataframe for the defined variables
k_pred <- data.frame(Age = 50, Start = 10, Number = 7)

# kyphosis prediction for the above data frame
k_pred_val <- predict(k_my_logit, newdata = k_pred,
                type = "response")

# printing the results
print(k_pred_val)

# _____ Analysis for 4C _____ #
# The probability of kyphosis being present in the resultant is 33.12%
# This is a decent score however, anything below a 50% is not a suitable
# choice for business decisions.
```

```
###################### Answers to question 5 Q1 part #############################
lin_1 <- lm(Sepal.Length~Sepal.Width, data = data_iris)

#summary for linear regression 1
summary(lin_1)

lin_2 <- lm(Sepal.Length~Petal.Length, data = data_iris)

#summary for linear regression 2
summary(lin_2)

lin_3 <- lm(Sepal.Length~Petal.Width, data = data_iris)

#summary for linear regression 3
summary(lin_3)

# _____ Analysis for 5 part 1 _____ #
# Adjusted r-squared for lin1 - 0.007 high significance
# Adjusted r-squared for lin2 - 0.758 high significance
# Adjusted r-squared  for lin3 - 0.667 high significance

############################ Answers to question 5 ################################

#script for plots

plot(x=data_iris$Sepal.Length, y=data_iris$Sepal.Width, type= "p")
# heteroscedastic

plot(x=data_iris$Sepal.Length, y=data_iris$Petal.Length, type= "p")
# homoscedastic

plot(x=data_iris$Sepal.Length, y=data_iris$Petal.Width, type= "p")
# heteroscedastic

# _____ Analysis for 5 _____ #
# Sepal.Length by Sepal.Width
# Sepal.Length by Petal.Length
# Sepal.Length by Petal.Width
```









Data	
data_iris	150 obs. of 10 variables
data_iris_prediction	1 obs. of 4 variables
data_iris_test	30 obs. of 10 variables
data_iris_train	120 obs. of 10 variables
k_data	81 obs. of 4 variables
k_data_test	17 obs. of 4 variables
k_data_train	64 obs. of 4 variables
k_my_logit	List of 30
k_pred	1 obs. of 3 variables
lin_1	List of 12
lin_2	List of 12
lin_3	List of 12
my_logit	List of 30
Values	
exponential_output	Named num [1:5] 1.59e-18 9.01e+03 4.96e+07 9.04e-02 1.49e-03
k_dependent	Named num [1:17] 0.01871 0.09307 0.00256 0.46723 0.65498 ...
k_exponential_output	Named num [1:4] 0.0318 1.8453 1.0223 0.768
k_my_logit_coeff	Named num [1:4] -3.45 0.613 0.022 -0.264
k_pred_val	Named num 0.332
logit_coeff	Named num [1:5] -40.98 9.11 17.72 -2.4 -6.51
pred_val	Named num 1
training_idx	int [1:64] 61 16 44 49 13 55 19 65 74 15 ...
Functions	
desc_func	function (x)
n_function	function (var)
s_function	function (var)