Practical No.7

AIM: Practical of Logistics Regression.

Theory:

Logistic regression is the appropriate regression analysis to conduct when the dependent variable is dichotomous (binary). Like all regression analyses, the logistic regression is a predictive analysis. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.

Data Exploration:

```
> library(datasets)
> ir_data<- iris
> head(ir_data)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
> str(ir_data)
'data.frame':          150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
```

So, we have a dataframe with 150 observations of 5 variables. The first 4 variables give information about plant attributes in centimeters and the last one give us the name of plant species. Species are given as Factor variable with 3 levels:

```
> levels(ir_data$Species)
[1] "setosa"     "versicolor" "virginica"
```

We should check whether we have any NA values in our dataset:

```
> sum(is.na(ir_data))
[1] 0
```

So, we are dealing with a complete dataset here. As we want to use Logistic Regression in this post, let's subset the data so that we have to deal with 2 species of plants rather than 3 (because logistic regression will be built on binary outcomes)

```
> ir_data<-ir_data[1:100,]
```

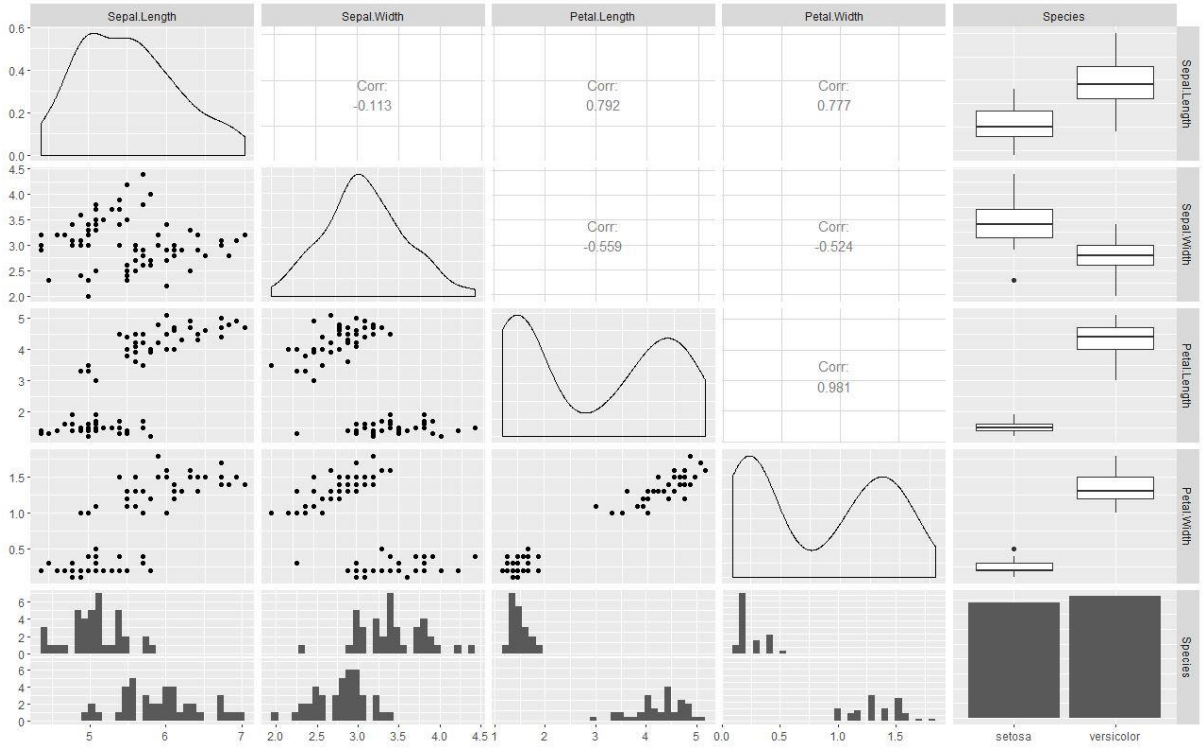
Also we will randomly define our Test and Control groups:

```
> set.seed(100)
```

```
> samp<-sample(1:100,80)

> ir_test<-ir_data[samp,]

> ir_ctrl<-ir_data[-samp,]
```

We will use the test set to create our model and control set to check our model. Now, lets explore the dataset a little bit more with the help of plots:

```
> install.packages("ggplot2")
> library(ggplot2)
> install.packages("GGally")
> library(GGally)

> ggpairs(ir_test)
 plot: [5,1] [==========================================================================>----------] 84% est: 2s `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
 plot: [5,2] [==========================================================================>----------] 88% est: 2s `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
 plot: [5,3] [==========================================================================>----------] 92% est: 1s `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
 plot: [5,4] [==========================================================================>-----] 96% est: 1s `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Model Fitting

Now, we will try to model this data using Logistic Regression. Here, we will keep it simple and will use only a single variable:

```
> y<-ir_test$Species; x<-ir_test$Sepal.Length
> glfit<-glm(y~x, family = 'binomial')
```

The default link function for above model is 'logit', which is what we want. We can use this simple model to get the probability of whether a given plant is 'satosa' or 'versicolor' based on its 'sepal length'. Before jumping to predictions using, let us have a look at the model itself.

```
> summary(glfit)

Call:
glm(formula = y ~ x, family = "binomial")

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.94538  -0.50121   0.04079   0.45923   2.26238

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -25.386      5.517  -4.601 4.20e-06 ***
x              4.675      1.017   4.596 4.31e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 110.854  on 79  degrees of freedom
Residual deviance:  56.716  on 78  degrees of freedom
AIC: 60.716

Number of Fisher Scoring iterations: 6
```

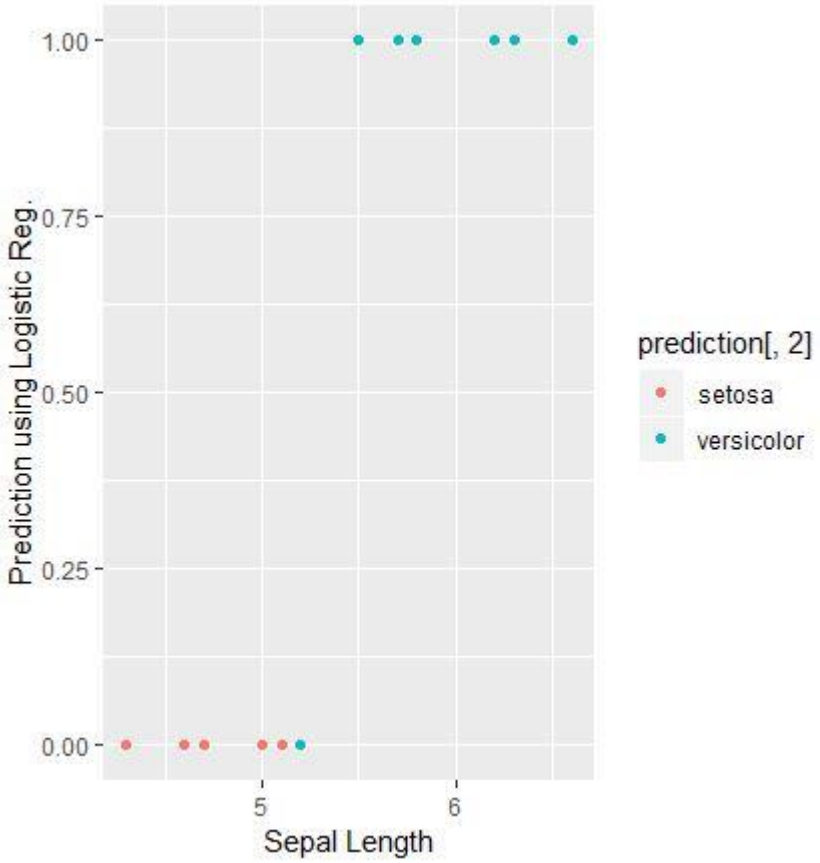
We can see that the P-Values indicate highly significant results for this model. Although, we should check any model deeply, but right now we will move to the prediction part.
###Checking Model's Predictions Let us use our Control set which we defined earlier to predict using this model:

```
> newdata<- data.frame(x=ir_ctrl$Sepal.Length)
> predicted_val<-predict(glfit, newdata, type="response")
> prediction<-data.frame(ir_ctrl$Sepal.Length, ir_ctrl$Species,predicted_val)
> prediction
   ir_ctrl.Sepal.Length ir_ctrl.Species predicted_val
1                   5.1         setosa   0.176005274
2                   4.7         setosa   0.031871367
3                   4.6         setosa   0.020210042
4                   5.0         setosa   0.118037011
5                   4.6         setosa   0.020210042
6                   4.3         setosa   0.005048194
7                   4.6         setosa   0.020210042
```

8	5.2	setosa	0.254235573
9	5.2	setosa	0.254235573
10	5.0	setosa	0.118037011
11	5.0	setosa	0.118037011
12	6.6	versicolor	0.995801728
13	5.2	versicolor	0.254235573
14	5.8	versicolor	0.849266756
15	6.2	versicolor	0.973373695
16	6.6	versicolor	0.995801728
17	5.5	versicolor	0.580872616
18	6.3	versicolor	0.983149322
19	5.7	versicolor	0.779260130
20	5.7	versicolor	0.779260130

We can see in the table above that what probability our model is predicting for a given plant to be 'versicolor' based on its 'sepal length'. Let's visualize this result using a simple plot. Let's say that we will consider any plant to be 'versicolor' if its probability for the same is more than 0.5:

```
> qplot(prediction[,1], round(prediction[,3]), col=prediction[,2], xlab = 'Sepal Length', ylab
+       = 'Prediction using Logistic Reg.')
```



So, from the above plot, we can see that our simple model is doing a fairly good prediction for plant species. We can also see a blue dot in the bottom cluster. This blue dot is showing that although correct specie of this plant is 'versicolor' but our model is predicting it as 'setosa'.

All Command

```
library(datasets)
ir_data<- iris
head(ir_data)
str(ir_data)
levels(ir_data$Species)
sum(is.na(ir_data))
ir_data<-ir_data[1:100,]
set.seed(100)
samp<-sample(1:100,80)
ir_test<-ir_data[samp,]
ir_ctrl<-ir_data[-samp,]
install.packages("ggplot2")
library(ggplot2)
install.packages("GGally")
library(GGally)
ggpairs(ir_test)
y<-ir_test$Species; x<-ir_test$Sepal.Length
glfit<-glm(y~x, family = 'binomial')
summary(glfit)
newdata<- data.frame(x=ir_ctrl$Sepal.Length)
predicted_val<-predict(glfit, newdata, type="response")
prediction<-data.frame(ir_ctrl$Sepal.Length, ir_ctrl$Species,predicted_val)
prediction

qplot(prediction[,1], round(prediction[,3]), col=prediction[,2], xlab = 'Sepal Length', ylab
    = 'Prediction using Logistic Reg.')
```