IRIS DATASET

The data were collected from 3 Iris flower types (or 3 classes), 50 instances were chosen from each type.
Below is the description of dataset:

Column 1: sepal length in cm
Column 2: sepal with in cm
Column 3: petal length in cm
Column 4: petal width in cm
Column 5: classes (setosa, versicolor and virginica)

EXPLORATION OF DATASET

head() method is used to return top 6 rows of a data frame or series.
The summary() function is used to print a statistical summary of a data set.
For extracting the structure of data set, str() function is used.

Code:

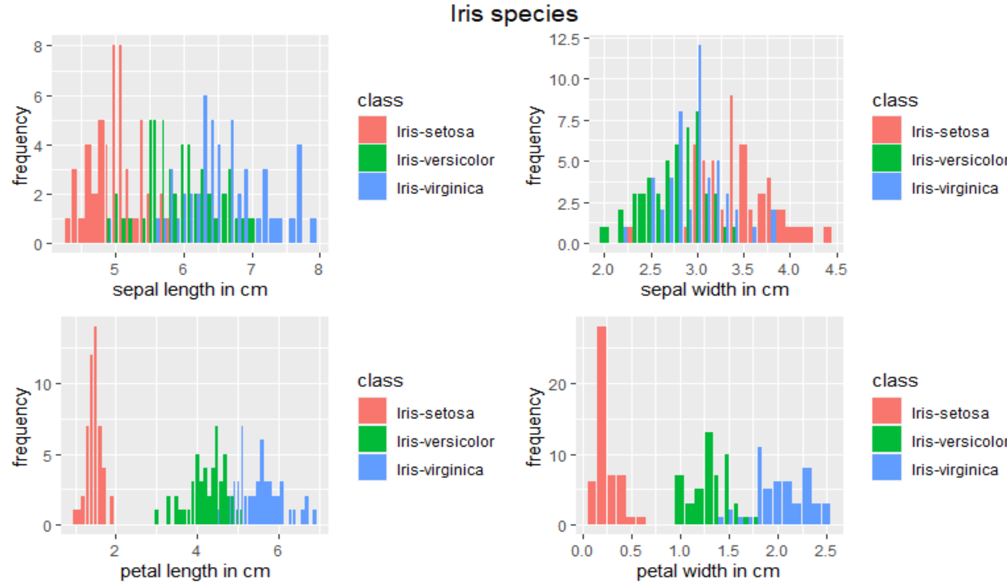
head(iris)
summary(iris)
str(iris)

Output:

```
> head(iris)    #view top 6 rows of dataset
  sepal.length sepal.width petal.length petal.width       class
1          5.1         3.5          1.4         0.2 Iris-setosa
2          4.9         3.0          1.4         0.2 Iris-setosa
3          4.7         3.2          1.3         0.2 Iris-setosa
4          4.6         3.1          1.5         0.2 Iris-setosa
5          5.0         3.6          1.4         0.2 Iris-setosa
6          5.4         3.9          1.7         0.4 Iris-setosa
> View(iris)
> head(iris)    #top 6 rows of dataset
  sepal.length sepal.width petal.length petal.width       class
1          5.1         3.5          1.4         0.2 Iris-setosa
2          4.9         3.0          1.4         0.2 Iris-setosa
3          4.7         3.2          1.3         0.2 Iris-setosa
4          4.6         3.1          1.5         0.2 Iris-setosa
5          5.0         3.6          1.4         0.2 Iris-setosa
6          5.4         3.9          1.7         0.4 Iris-setosa
> summary(iris) #statistical summary of dataset
  sepal.length    sepal.width     petal.length    petal.width       class          
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100   Length:150        
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300   Class :character  
 Median :5.800   Median :3.000   Median :4.350   Median :1.300   Mode  :character  
 Mean   :5.843   Mean   :3.054   Mean   :3.759   Mean   :1.199                     
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800                     
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500                     
> str(iris)     #structure of dataset
'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 ...
 $ class       : chr  "Iris-setosa" "Iris-setosa" "Iris-setosa" "Iris-setosa" ...
```

DATASET VISUALIZATION

To see how the model performed on the data, 'ggplot2' package was used to build graphs on data.



As the histograms, 'Iris -setosa' is well separated from the other two, and 'versicolor' and 'virginica' were overlapped each other at some points. It's clear under petal features.

DATA ANALYTIC TECHNIQUES

1. **LINEAR REGRESSION**

Linear regression is a statistical model used to predict the relationship between independent and dependent variables.
For applying linear regression, first convert the target values to numeric values. Then lm() is used to fit linear model.

Code:

```
#Copying iris dataset into a variable dataset
dataset<- iris
#Convert target variables to numeric values
dataset$class = factor(iris$class,
                       levels = c('Iris-setosa', 'Iris-versicolor', 'Iris-virginica'),
                       labels = c(1, 2, 3))
dataset$class = as.numeric(dataset$class)
#Applying linear model function
linear_iris<- lm(dataset$class ~ sepal.length+sepal.width + petal.length + petal.width,
                 data = dataset )
 summary(linear_iris)
```

Output:

```
> summary(linear_iris)

Call:
lm(formula = dataset$class ~ sepal.length + sepal.width + petal.length +
    petal.width, data = dataset)

Residuals:
     Min       1Q   Median       3Q      Max
-0.59046 -0.15230  0.01338  0.10332  0.55061

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.19208    0.20470   5.824 3.57e-08 ***
sepal.length -0.10974    0.05776  -1.900 0.059418 .
sepal.width  -0.04424    0.05996  -0.738 0.461832
petal.length  0.22700    0.05699   3.983 0.000107 ***
petal.width   0.60989    0.09447   6.456 1.52e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2191 on 145 degrees of freedom
Multiple R-squared:  0.9304,    Adjusted R-squared:  0.9285
F-statistic: 484.8 on 4 and 145 DF,  p-value: < 2.2e-16
```

If the Pr(>|t|) is low, the coefficients are significant. If the Pr(>|t|) is high, the coefficients are not significant. It is clearer that more stars beside the Pr(>|t|) Value, the more significant the variable. Other variables can be eliminated for better result.

Code:

```
#lm() produces a few statistics on the residuals
linear_iris_model <- lm(class~ petal.length + petal.width, data = dataset)
#R-squared shows the accuracy
summary(linear_iris_model)
```

Output:

```
> summary(linear_iris_model)

Call:
lm(formula = class ~ petal.length + petal.width, data = dataset)

Residuals:
     Min       1Q   Median       3Q      Max
-0.56418 -0.13943  0.01386  0.09458  0.58840

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.57394    0.05428  10.573  < 2e-16 ***
petal.length  0.17912    0.03861   4.639 7.66e-06 ***
petal.width   0.62803    0.08926   7.036 6.98e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2248 on 147 degrees of freedom
Multiple R-squared:  0.9257,    Adjusted R-squared:  0.9247
F-statistic: 915.8 on 2 and 147 DF,  p-value: < 2.2e-16
```

As the summary from the table, the multiple R-squared value is 0.9257 which is closer to 1 indicating that the model is better at explaining the data. So, the accuracy of Linear regression model was 92.57% which was good to predict type of Iris plant with petal features.

CONFIDENCE AND PREDICTION INTERVAL:

Code:

```
#Confidence interval
confint(linear_iris_model,level = .95)
```

Ouput:

```
> confint(linear_iris_model,level = .95)
                 2.5 %    97.5 %
(Intercept)  0.4666642 0.6812147
petal.length 0.1028225 0.2554200
petal.width  0.4516322 0.8044361
```

Here, the 95% confidence interval for petal length is (0.10, 0.25) and petal width is (0.45,0.80).

Code:

```
# new input variables
petal.length <- 1.3
petal.width <- 0.3
new_values <- data.frame(petal.length+petal.width)
#the predict() function provides a 95%confidence interval.
confidence_interval <- predict(linear_iris_model, new_values, level=.95, interval="confidence")
confidence_interval
```

Output:

```
> confidence_interval
        fit       lwr      upr
1 0.9952073 0.9290565 1.061358
```

Here, the 95% confidence interval is (0.92,1.06). The expected class of Iris is 1(that is Setosa).

Code:

```
#Compute 95% prediction interval
prediction_interval <- predict(linear_iris_model, new_values, level=.95, interval="prediction")
prediction_interval
```

Output:

```
> prediction_interval
        fit       lwr      upr
1 0.9952073 0.5460186 1.444396
```

Here, the 95% prediction interval is (0.54, 1.44). The fit value is 0.99 (~ 1) resembles Iris-setosa class.

2. RIDGE REGRESSION

For ridge regression, first define matrix of predictor variables. Then 'glmnet()' package is used to fit ridge regression. In that alpha = 0 represents ridge regression and alpha =1 represents Lasso regression

Code:

```
iris_matrix <- data.matrix(dataset[, c('sepal.length', 'sepal.width', 'petal.length', 'petal.width')])
iris_model <- glmnet(iris_matrix, dataset$class, alpha = 0)
#View summary of model
summary(iris_model)
```

Output:

```
> summary(iris_model)
          Length Class     Mode
a0        100    -none-    numeric
beta      400    dgCMatrix S4
df        100    -none-    numeric
dim         2    -none-    numeric
lambda    100    -none-    numeric
dev.ratio 100    -none-    numeric
nulldev     1    -none-    numeric
npasses     1    -none-    numeric
jerr        1    -none-    numeric
offset      1    -none-    logical
call        4    -none-    call
nobs        1    -none-    numeric
```

Then perform k-fold cross-validation to find optimal lambda value that minimizes error.

Code:

```
iris_cv <- cv.glmnet(iris_matrix, dataset$class, alpha = 0)
iris_lambda <- iris_cv$lambda.min
iris_lambda
```

Output:

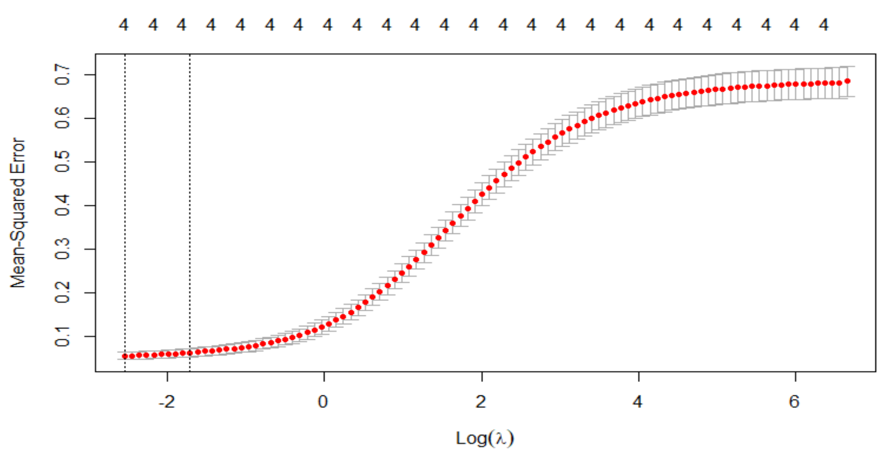
```
> iris_lambda
[1] 0.07809494
```

Plot the graph of Mean squared error by lambda value to visualize it clearly.

Code:

```
plot(iris_cv)
```

Output:



Rebuild the model and check the coefficient.

Code:

```
iris_coef <- glmnet(iris_matrix, dataset$class, alpha = 0, lambda = iris_lambda)
coef(iris_coef)
```

Output:

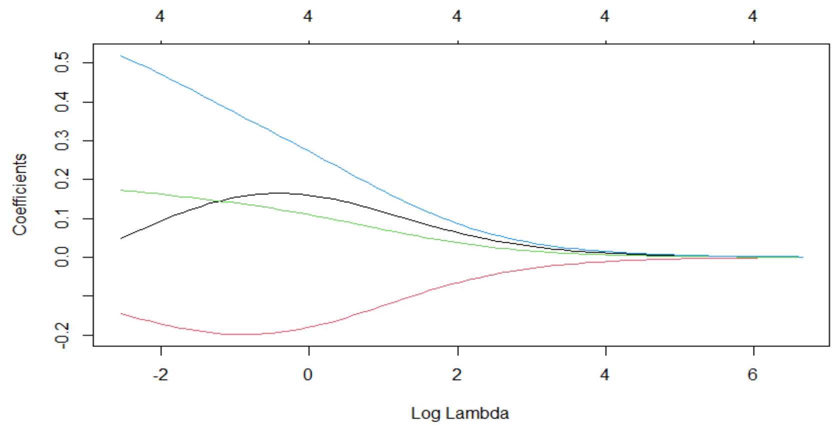
```
> coef(iris_coef)
5 x 1 sparse Matrix of class "dgCMatrix"
                      s0
(Intercept)   0.89303435
sepal.length  0.04813332
sepal.width  -0.14633344
petal.length  0.17264750
petal.width   0.52031522
```

Visualize the coefficient value against increasing lambda value.

Code:

```
# Ridge trace plot
plot(iris_model, xvar = "lambda")
```

Output:



The next step is to predict with best fitted model to compute accuracy.

Code:

```
#Use fitted best model to make predictions
iris_predicted <- predict(iris_model, s = iris_lambda, newx = iris_matrix)
#Find sum of squared total and sum of squared error.
sq_total <- sum((dataset$class - mean(dataset$class))^2)
sq_error <- sum((iris_predicted - dataset$class)^2)
#R-Squared
r_sq <- 1 - (sq_error/sq_total)
r_sq
```

Output:

```
> r_sq
[1] 0.9235568
```

The r squared value is 0.9235, So, the accuracy of Ridge regression model was 92% which was good to predict type of Iris plant.

CONCLUSION

Overall, with accuracy over 92.57%, Linear regression model performed well on Iris data that distinguished 3 classes separately. In other words, this model can be used to predict 3 types of Iris plant.