Stats Programming

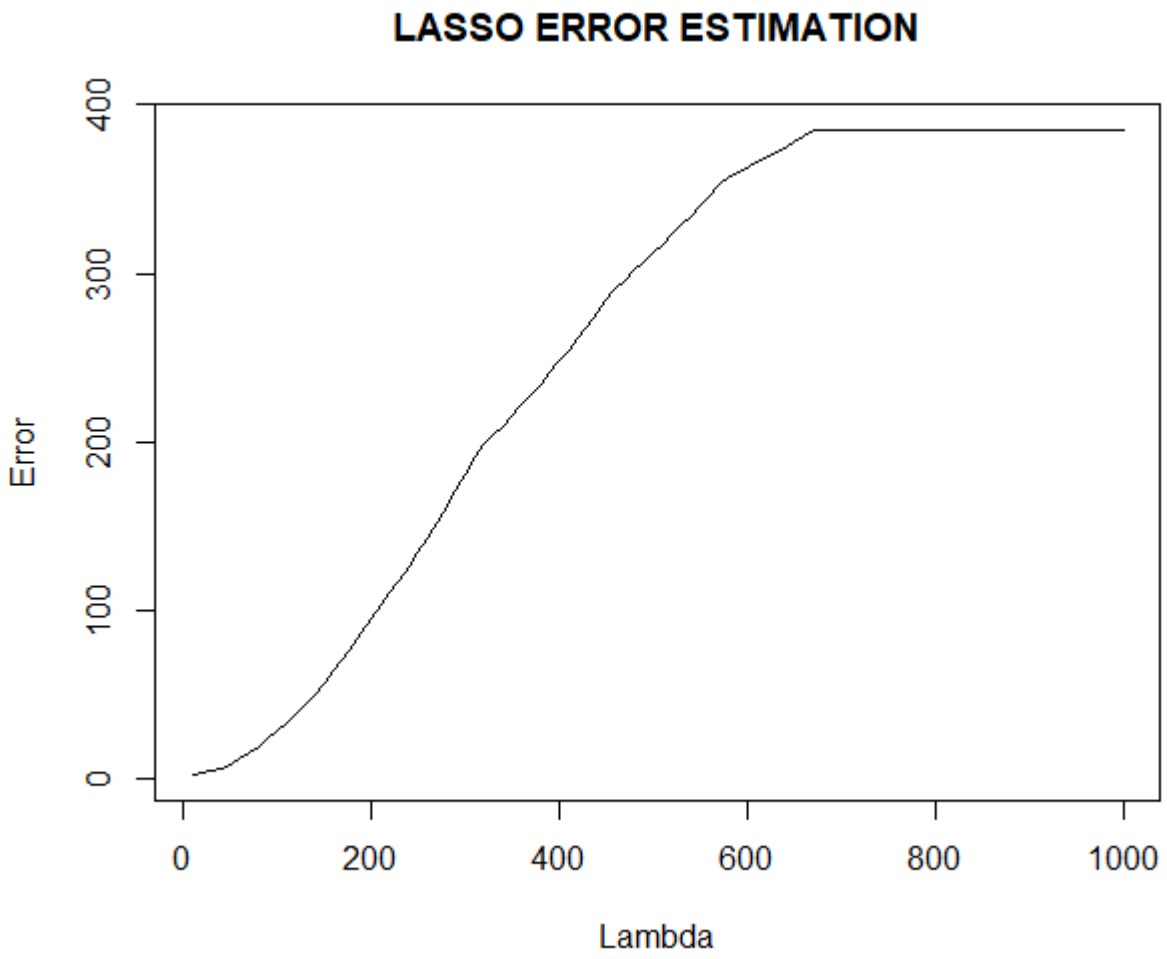
HomeWork-7

Devanshi Patel (UID: **504945601**)

---

2) For Lasso, plot the estimation error over the different values of lambda.

- The following plot is obtained when we compute estimation error between beta_all and beta_true for different values of lambda ranging from 1000 to 10.
- It can be observed from the plot that initially when the value of lambda is high, the error is large.
- As we gradually reduce lambda, it converges and we can see the error getting reduced.



3) Analyze some appropriate real datasets available in R using your package of least squares regression, ridge regression, Lasso regression, logistic regression, and PCA. Submit a pdf or .doc/.docx file with your results (e.g. explanations, tables, plots) showing your results and explaining what you see.

For the analysis, I am using Iris dataset. It consists of four measurements (length and width of the petals and sepals) of one hundred and fifty Iris flowers from three species: setosa, versicolor, virginica

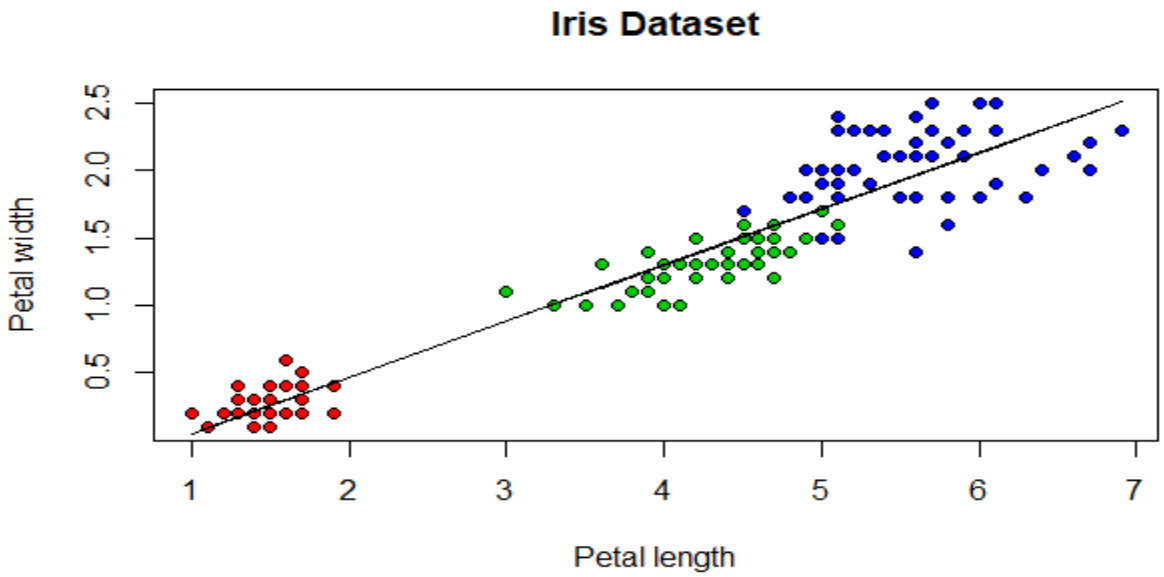
Linear Regression

The petal length and petal width are highly correlated over all species so we'll be running a linear regression on them.

```
> head(iris)
  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
```

```
> X = matrix(data = iris$Petal.Length, nrow = nrow(iris))
> Y = matrix(data = iris$Petal.Width, nrow = nrow(iris))
> list = myLinearModel(X,Y)
> list$Beta
            [,1]
[1,] -0.3630755
[2,]  0.4157554
```

```
> plot(X, Y, pch=21, bg=c("red","green3","blue")[unclass(iris$Species)], main
="Iris Dataset", xlab="Petal length", ylab="Petal width")
> lines(X,Yhat,type="l")
```



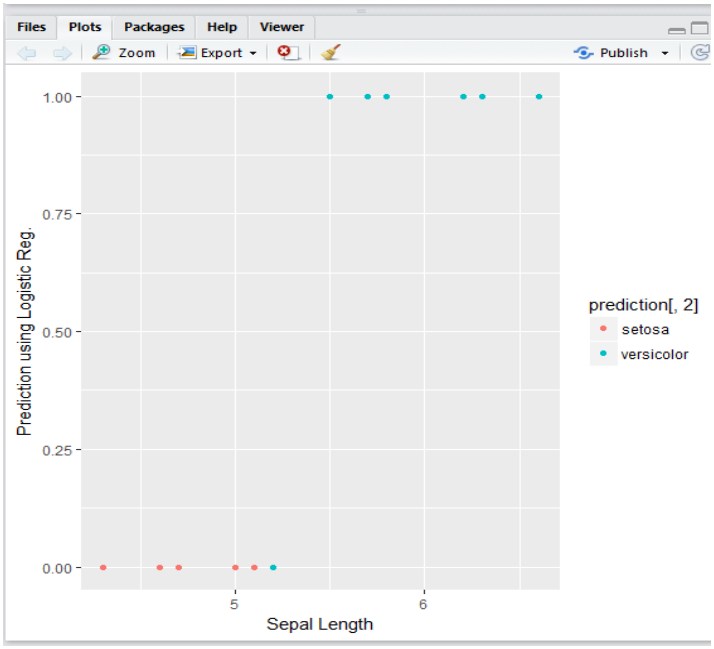
Logistic Regression

To analyze the dataset for logistic regression, I am considering setosa and versicolor species to make the results binary.

The training and test set is in the ratio of 4:1 for first 100 values. After fitting the logistic regression model, we get the following results:

```
> 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
```

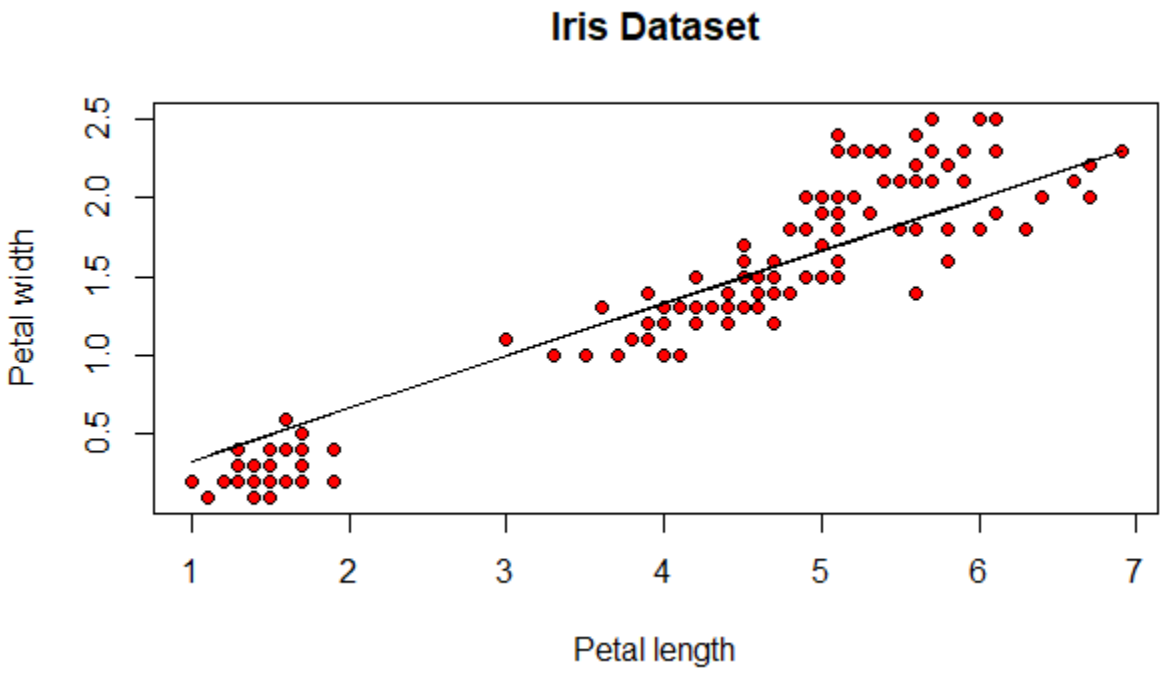
The predicted_val gives the probability that a plant is versicolor or setosa. As can be seen in below plot, they are classified correctly except for one result which is acceptable.



Lasso Regression

The below plot is obtained when we run Lasso Regression and plot it for lambda=100. It takes lesser time to run and the results are similar to ridge regression.

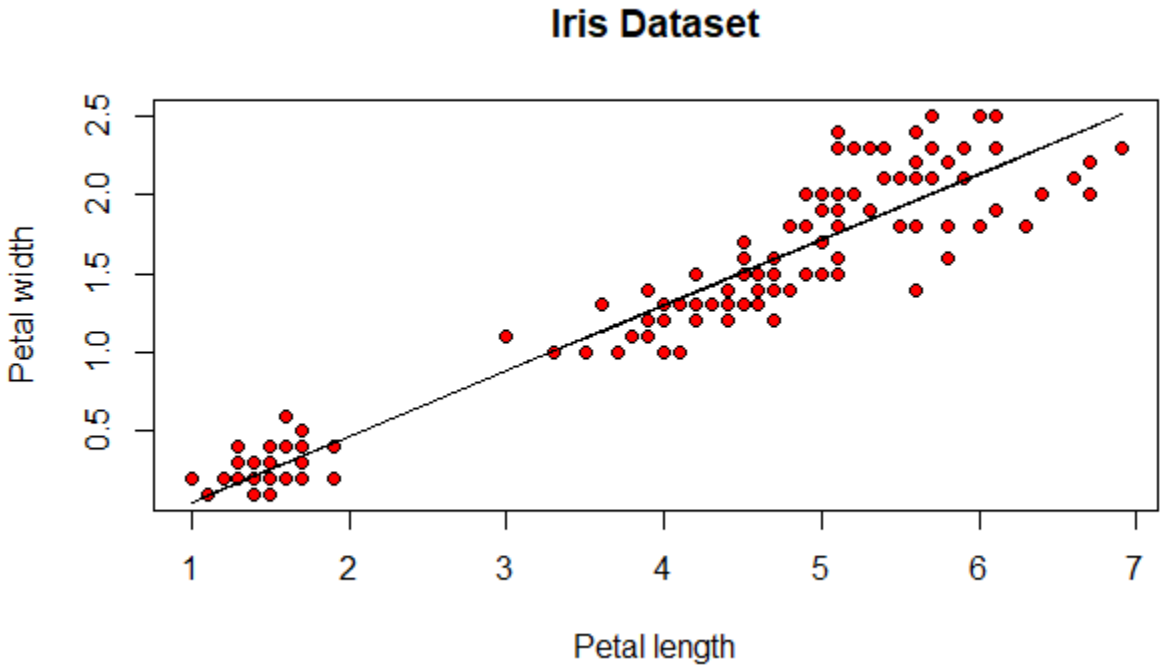
```
> Yhat = X %*% beta_all[,100]
> plot(X, Y, pch=21, bg=c("red"), main="Iris Dataset", xlab="Petal length", y
lab="Petal width")
> lines(X,Yhat, type="l")
```



Ridge Regression

This regression returns an efficient best-fit model however, it takes a longer time to run.

```
> plot(x,Y, bg=c("red"), main="Iris Dataset", xlab="Petal length", ylab="Peta
l width",pch=21)
> lines(x,Yhat)
```



PCA

From the figures it is clear that the variance of last 2 components is small, hence it is possible to remove these columns for dimensionality reduction. The model can be trained on just first two columns, resulting in less computation with no significant change in efficiency.