3) **Tasks**:

* Perform Linear regression and KNN classification on the zipcode data for predicting digits: 2 and 3.
* Compare the classification performance of both the methods
* Show the error for k=1,3,5,7,9,11,13,15

Dataset to be used: zipcode data from ElemStatLearn library

**Answers**:

* The zipcode dataset has total 257 columns out of which the first columns contains the actual digits 0 to 9 and the variables V2 to V257 contains pixel values for the 16by16 image of that handwritten digits.
* We have to consider only 2’s and 3’s so sub-setting the data as below. I have used the Dataframe function for sub-setting instead of the subset() function since both are similar:

zipTr <- as.data.frame(zip.train)

zipTr <- zipTr[zipTr$V1 ==2 | zipTr$V1 == 3,]

zipTest <- as.data.frame(zip.test)

zipTest <- zipTest[zipTest$V1 ==2 | zipTest$V1 == 3,]

* Error rate Formula:
  + I have calculated the error rate by **subtracting** the **predicted output** (obtained from the model) from the **actual output** in variable **V1**. And then **dividing it by the total number of observations** (nRow).

* + For Linear Regression:

**Error.rate.lr=** **0.04120879** for predicting the zip.test response

* + For KNN Classification:

**Error.rate.knn = 0.024** for K=1 for zip.test response

* Additionally, I have used “system(…….model formula…)” function to compute the time taken by the models for predicting the rsponse and can see that there is not much of the difference in the times taken by KNN and Linear Regression.

**“*So we can conclude that the performance of the KNN classification is better than that of Linear Regression.”***

* ***Error Rate for different values of K=1,3,5,7,9,11,13,15:***

Here for different values of K, I have calculated the response for Test Data and Train Data then calculated the error rate by subtracting the predicted data from the actual output available in zipTest$V1 and zipTr$V1

The vector **error.rate.knn.test and**  **error.rate.knn.tr** contains error rate for K=1,3,5,7,9,11,13,15.

> print(error.rate.knn.test)

[1] 0.02472527 0.03021978 0.03021978 0.03021978 0.03571429 0.03571429 0.03296703 0.03846154

> print(error.rate.knn.tr)

[1] 0.000000000 0.004319654 0.005759539 0.005759539 0.007919366 0.007919366 0.007919366 0.009359251

As we can see the Training error is very much less than that of the test one.

After plotting the error rate for values of K we can see that the error rate is increasing with K.

