**Problem statement**

1. Using the k-nearest-neighbors classification function kknn contained in the R kknn package, suggest a good value of k, and show how well it classifies that data points in the full data set. Don’t forget to scale the data (scale=TRUE in kknn).

# Notes on kknn

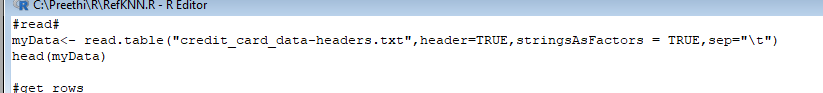
* You need to be a little careful. If you give it the whole data set to find the closest points to i, it’ll use i itself (which is in the data set) as one of the nearest neighbors. A helpful feature of R is the index –i, which means “all indices except i”. For example, data[-i,] is all the data except for the ith data point. For our data file where the first 10 columns are predictors and the 11th column is the response, data[-i,11] is the response for all but the ith data point, and data[-i,1:10] are the predictors for all but the ith data point.

(There are other, easier ways to get around this problem, but I want you to get practice doing some basic data manipulation and extraction, and maybe some looping too.)

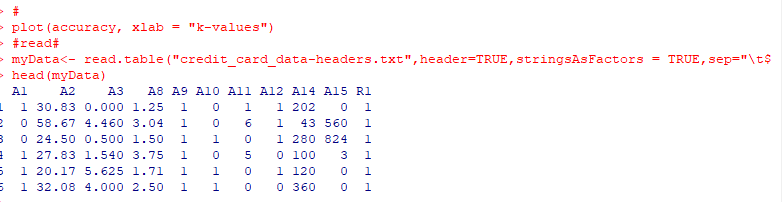
* **Note** that kknn will read the responses as continuous, and return the fraction of the k closest responses that are 1 (rather than the most common response, 1 or 0).

# Ingest data

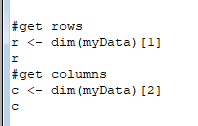
**Code:**

****

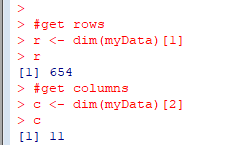
**Results:**

****

**2.get demographics of data**

****

**Results:**

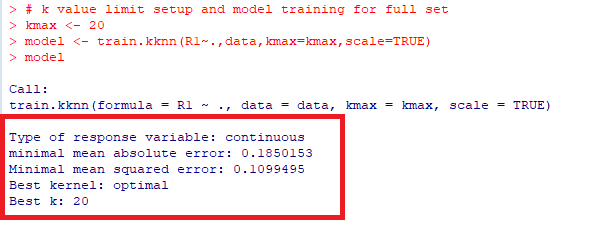
****

# K value and model creation

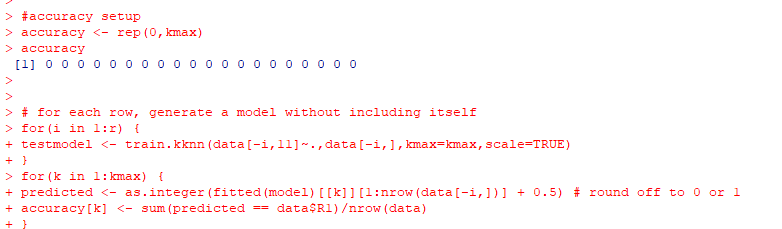
Set a max limit of K value to 20 (I setup 40 initially to run the model for different k values and based on best k accuracy, reduced the K value to 20)

Pre-requisite: scale is set to true

Train a model based on the full data set



# Accuracy and check k values’ accuracy excluding data itself



For each data point j,

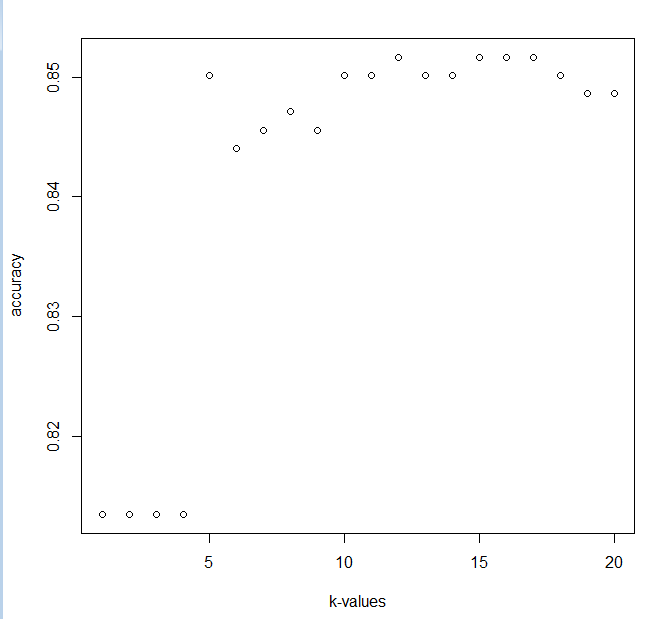
data[-i,] => data except for the itself.

data[-i,11] => response for all but the ith data point

data[-i,1:10] => predictors for all but the ith data point.

# 4.Plot

plot(accuracy, xlab = "k-values")

****