**Question 14.1**

**The breast cancer data set breast-cancer-wisconsin.data.txt from http://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/ (description at http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29 ) has missing values.**

**1. Use the mean/mode imputation method to impute values for the missing data.**

**Answer:** Given below is the steps for 14.1.1

**Step 1:** Load the data set and install all the necessary libraries.

Code:

#Clear the environment

*rm(list=ls())*

*install.packages("kableExtra")*

*library(kableExtra)*

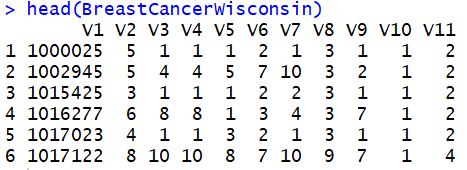
*library(DAAG)*

*library(KKNN)*

*BreastCancerWisconsin <- read.table("breast-cancer-wisconsin.data.txt", stringsAsFactors = FALSE, header = FALSE, sep = ",")*

*head(BreastCancerWisconsin)*

Output:



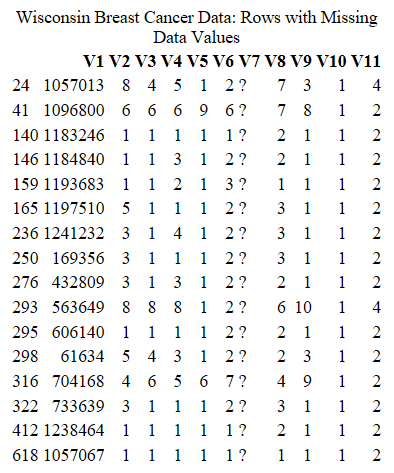
Step 2: To specify which rows we want to Impute we need rows with missing data. Given below is the code to detect rows with missing data.

Code:

*MissingDataValues <- which(BreastCancerWisconsin$V7 == "?")*

*kable(BreastCancerWisconsin[MissingDataValues,], caption="Wisconsin Breast Cancer Data: Rows with Missing Data Values")*

Output:





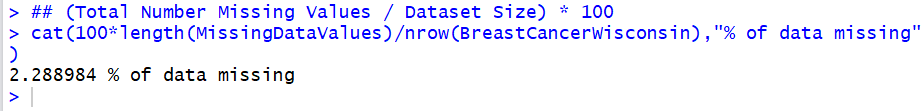
It appears that the data is only missing on the variable v7. So our imputation will occur of V7. Lets now calculate out of the total dataset, how much data is missing.

Code:

*## (Total Number Missing Values / Dataset Size) \* 100*

*cat(100\*length(MissingDataValues)/nrow(BreastCancerWisconsin),"% of data missing")*

Output:



Looking at the output, it shows that 2.28 % of the data is missing from the entire dataset on which we can use data imputation.

Step 3: Setup of datasets. We need clean data and Missing Data.

Code:

*## Clean Dataset with no missing values*

*BreastCancerWisconsin\_Clean <- BreastCancerWisconsin[-MissingDataValues,]*

*head(BreastCancerWisconsin\_Clean)*

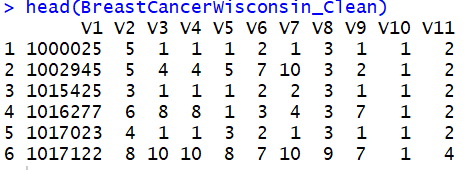
*## Dataset with only missing values*

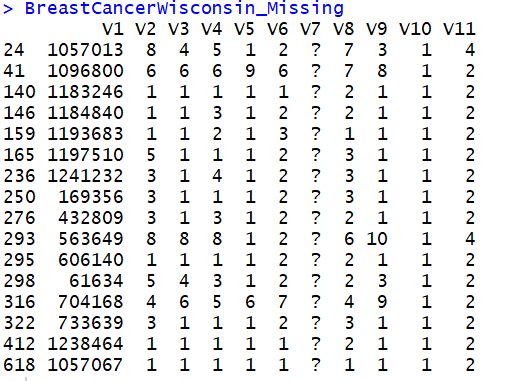
*BreastCancerWisconsin\_Missing <- BreastCancerWisconsin[MissingDataValues,]*

*BreastCancerWisconsin\_Missing*

Output:

Clean Dataset (top 6 rows). No missing values





Step 4: Using mode imputation method to impute missing values

We will first calculate the mode of the clean data with no missing values and then impute it on the copy of the original dataset which has missing values. Since we would need the original dataset in further analysis, we have to create a copy of the original dataset. We will first create a function which will calculate the mode of a all the

Code:

*# calculateing the mode of v7 in the clean data*

*ModeOfV7 <- as.numeric(CalculateMode(BreastCancerWisconsin\_Clean$V7))*

*cat("Mode of V7: ",ModeOfV7)*

*#creating a copy of the original dataset*

*BreastCancerWisconsin\_ModeImpute <- BreastCancerWisconsin*

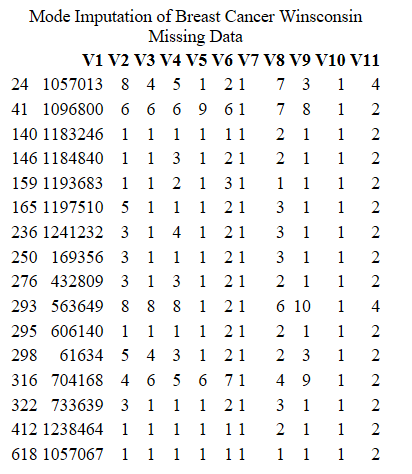
*# impute missing values for V7 with the mode*

*BreastCancerWisconsin\_ModeImpute[MissingDataValues,]$V7 <- as.integer(ModeOfV7)*

*kable(BreastCancerWisconsin\_ModeImpute[MissingDataValues,], caption = "Mode Imputation of Breast Cancer Winsconsin Missing Data")*

Output:

Mode of V7: 1



Analysis: Looking at the above output it is clearly visible that all the missing values have been imputed with the mode.

Step 4: Using mean imputation method to impute missing values

clean data with no missing values and then impute it on the copy of the original dataset which has missing values. Since we would need the original dataset in further analysis, we have to create a copy of the original dataset.

Code:

*# calculating the mode of v7 in the clean data*

*MeanOfV7 <- mean(as.numeric(BreastCancerWisconsin\_Clean$V7))*

*cat("mean: ",MeanOfV7)*

*#creating a copy of the original dataset*

*BreastCancerWisconsin\_MeanImpute <- BreastCancerWisconsin*

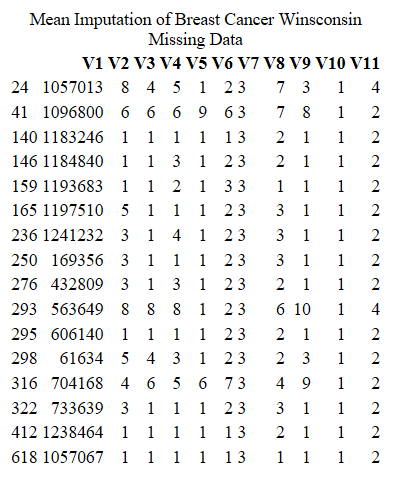
*# impute the missing values with the mean*

*BreastCancerWisconsin\_MeanImpute[MissingDataValues,]$V7 <- as.integer(MeanOfV7)*

*kable(BreastCancerWisconsin\_MeanImpute[MissingDataValues,], caption = "Mean Imputation of Breast Cancer Winsconsin Missing Data")*

Output

mean: 3.544656



Analysis: Looking at the above output it is clearly visible that all the missing values have been imputed with the mean.

**Question 14.1**

**The breast cancer data set breast-cancer-wisconsin.data.txt from http://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/ (description at http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29 ) has missing values.**

1. **Use regression to impute values for the missing data.**

**Answer: Given below are the steps performed for regression**

Step 1: Setting up of datasets.

We will be setting up dataset without missing values and no response variable (assuming v11 in this case).

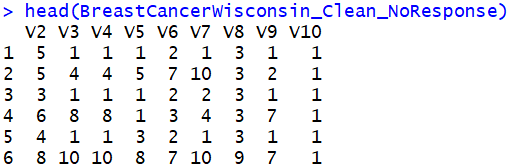
Code:

*BreastCancerWisconsin\_Clean\_NoResponse <- BreastCancerWisconsin\_Clean[,2:10]*

*BreastCancerWisconsin\_Clean\_NoResponse$V7 <- as.integer(BreastCancerWisconsin\_Clean\_NoResponse$V7)*

*head(BreastCancerWisconsin\_Clean\_NoResponse)*

Output:



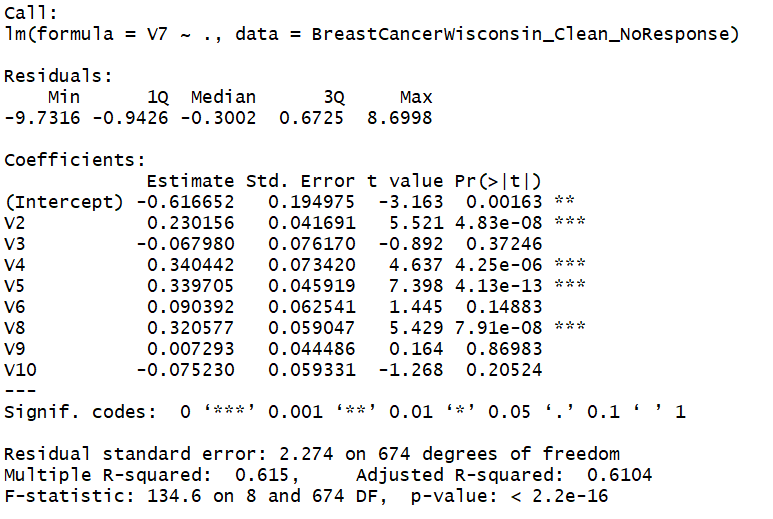
Step 2: Building a linear regression model in the dataset generated in step 2

Code:

*BreastCancerWisconsin\_Model <- lm(V7~., data = BreastCancerWisconsin\_Clean\_NoResponse)*

*summary(BreastCancerWisconsin\_Model)*

Output:



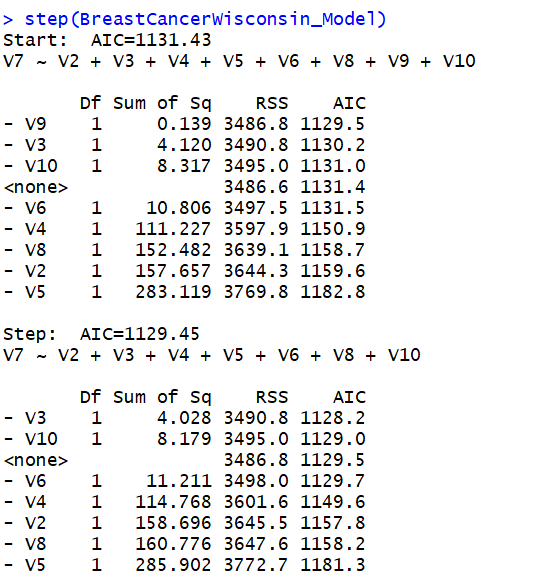
Looking at the data above it appears that variables v2,v4,v5 and v8 are the most significant variables. The R-Squared shoes that the model is 61.5% accurate.

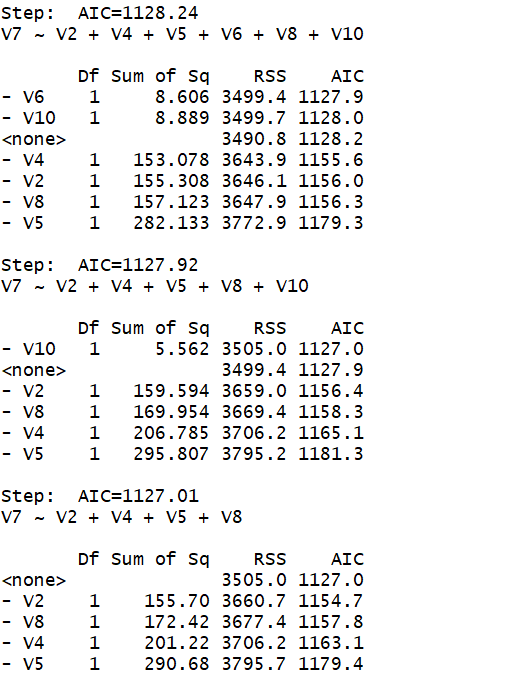
Lets use a stepwise regression method to get the variables with highest impact in accuracy of the mode.

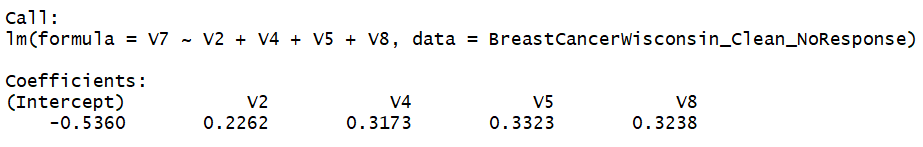
Code:

*step(BreastCancerWisconsin\_Model)*

Output:







We find the significant variables with stepwise regression method are V2, V4, V5 and V8 as seen in the original regression step. Lets refine our model with the significant variables.

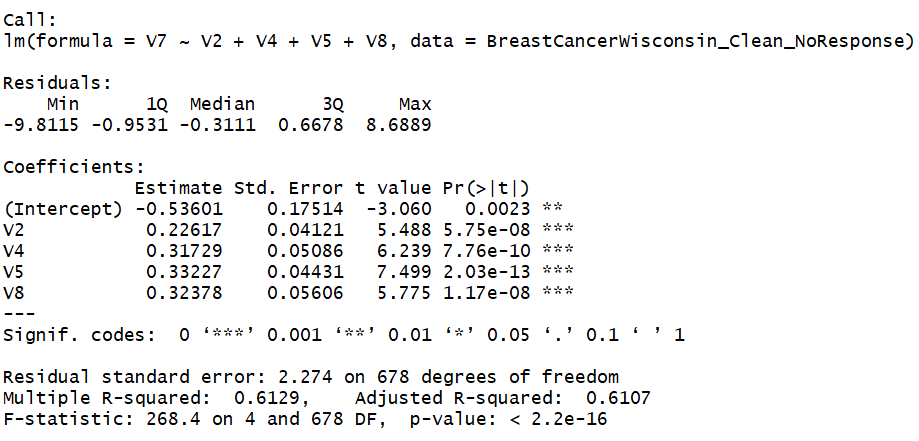
Code:

*# build a linear regression model with only significant predictors*

*BreastCancerWisconsin\_Model2 <- lm(V7~V2+V4+V5+V8, data = BreastCancerWisconsin\_Clean\_NoResponse)*

*summary(BreastCancerWisconsin\_Model2)*

Output:

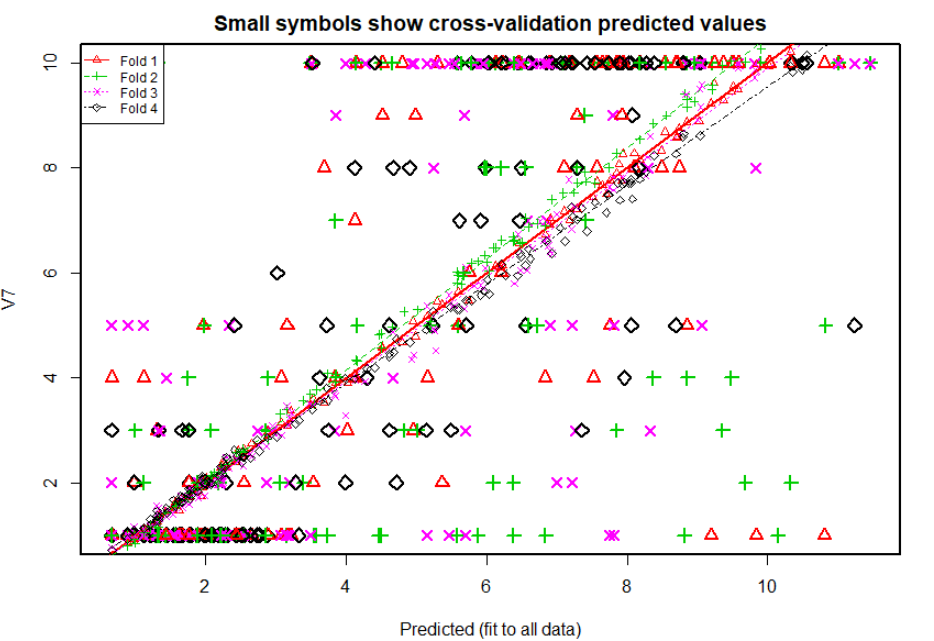


Lets cross validate the data to determine the quality of the model. Lets perform 4 fold crossvalidation

Code:

*BreastCancerWisconsin\_Model\_CV <- cv.lm(BreastCancerWisconsin\_Clean\_NoResponse, BreastCancerWisconsin\_Model2, m=4)*

Output:



Step 4: Now lets calculate the R-Squared of the cross validated model.

Code:

*# Calculation of R-Squared*

*n = length(BreastCancerWisconsin\_Clean$V7)*

*avg = mean(as.numeric(BreastCancerWisconsin\_Clean$V7))*

*SSE<-0*

*SSR<-0*

*SST<-0*

*for(i in 1:n){*

*SST = SST + (as.numeric(BreastCancerWisconsin\_Clean$V7[i]) - avg)^2*

*SSE = SSE + (as.numeric(BreastCancerWisconsin\_Clean$V7[i]) - as.numeric(BreastCancerWisconsin\_Model\_CV$cvpred[i]))^2*

*SSR = SSR + (as.numeric(BreastCancerWisconsin\_Model\_CV$cvpred) - avg)^2*

*}*

*SSE*

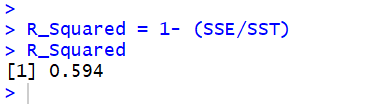
*SST*

*SSR*

*R\_Squared = 1- (SSE/SST)*

*R\_Squared*

Output:



The R-Squared is found to be 59.4 %

Step 5: Lets impute the missing values with the values that the cross-validated linear regression model predicts for V7.

Code:

# predicted values

V7\_predicted <- predict(BreastCancerWisconsin\_Model2, newdata = BreastCancerWisconsin\_Missing)

# impute the missing values with predicted values

BreastCancerWisconsin\_Reg\_Impute <- BreastCancerWisconsin

BreastCancerWisconsin\_Reg\_Impute[MissingDataValues,]$V7 <- as.integer(V7\_predicted)

# make sure the data is within the original range of 1 to 10

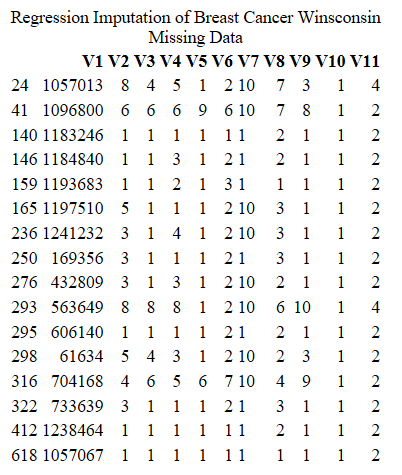
BreastCancerWisconsin\_Reg\_Impute$V7[BreastCancerWisconsin\_Reg\_Impute$V7 > 10] <- 10

BreastCancerWisconsin\_Reg\_Impute$V7[BreastCancerWisconsin\_Reg\_Impute$V7 < 1] <- 1

# resulting imputed values

kable(BreastCancerWisconsin\_Reg\_Impute[MissingDataValues,], caption = "Regression Imputation of Breast Cancer Winsconsin Missing Data")

Output:



The data has been imputed with the predicted regression response.

**Question 14.1**

**The breast cancer data set breast-cancer-wisconsin.data.txt from http://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/ (description at http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29 ) has missing values.**

**3. Use regression with perturbation to impute values for the missing data.**

**Answer:** Using regression with perturbation to impute values for the missing data.

Step 1: Find the perturbation values.

Code:

*set.seed(1)*

*# perturbation using normal distribution of the predicted values*

*perturbation\_values <- rnorm(nrow(BreastCancerWisconsin\_Missing), V7\_predicted, sd(V7\_predicted))*

*cat("perturbation values: ", perturbation\_values)*

Output:



Step 2: Lets impute the missing values.

Code:

*# imputing the missing values*

*BreastCancerWisconsin\_Petrubation <- BreastCancerWisconsin*

*BreastCancerWisconsin\_Petrubation[*MissingDataValues*,]$V7 <- perturbation\_values*

*BreastCancerWisconsin\_Petrubation[*MissingDataValues*,]$V7 <- as.integer(BreastCancerWisconsin\_Petrubation[MissingDataValues,]$V7)*

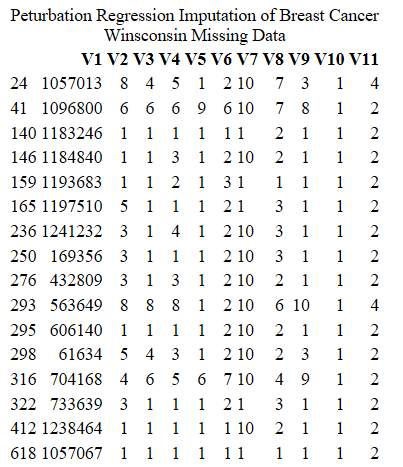
*# Checking if the data is within the range*

*BreastCancerWisconsin\_Petrubation$V7[BreastCancerWisconsin\_Petrubation$V7 > 10] <- 10*

*BreastCancerWisconsin\_Petrubation$V7[BreastCancerWisconsin\_Petrubation$V7 < 1] <- 1*

*kable(BreastCancerWisconsin\_Petrubation[MissingDataValues,], caption = "Peturbation Regression Imputation of Breast Cancer Winsconsin Missing Data")*

Output:



**Question 14.1**

**The breast cancer data set breast-cancer-wisconsin.data.txt from http://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/ (description at http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29 ) has missing values.**

**4. (Optional) Compare the results and quality of classification models (e.g., SVM, KNN) build using**

**(1) the data sets from questions 1,2,3;**

**(2) the data that remains after data points with missing values are removed; and**

**(3) the data set when a binary variable is introduced to indicate missing values.**

**Answer:**

Step 1: Lets split the data into Training dataset and Testing dataset.

Code:

*# set seed*

*set.seed(1)*

*# training and validation sets*

*training <- sample(nrow(BreastCancerWisconsin), size = floor(nrow(BreastCancerWisconsin) \* 0.7))*

*testing <- setdiff(1:nrow(BreastCancerWisconsin),training)*

Step 2: Lets apply K- Nearest neighbor on the 3 imputed datasets.

Mode Imputed Data

Code:

*results\_knn\_mode <- c()*

*for (k in 1:15) {*

*BreastCancerWisconsin\_ModeImpute\_model <- kknn(V11~V2+V3+V4+V5+V6+V7+V8+V9+V10, BreastCancerWisconsin\_ModeImpute[training,], BreastCancerWisconsin\_ModeImpute[testing,], k=k)*

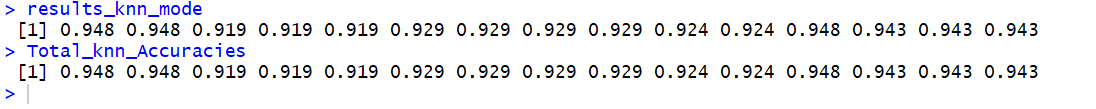
*pred <- as.integer(fitted(BreastCancerWisconsin\_ModeImpute\_model)+0.5)*

*results\_knn\_mode[k] <- sum(pred == BreastCancerWisconsin\_ModeImpute[testing,]$V11) / nrow(BreastCancerWisconsin\_ModeImpute[testing,])*

*}*

*results\_knn\_mode*

Output:



The vector results\_knn\_mode shows accuracy above 90% for all the values of K and highest accuracy of 94.8% for k = 1,2,12 when applied for values of k = 1 to 15.

Mean Imputed Data

Code:

*results\_knn\_mean <- c()*

*for (k in 1:15) {*

*BreastCancerWisconsin\_MeanImpute\_model <- kknn(V11~V2+V3+V4+V5+V6+V7+V8+V9+V10, BreastCancerWisconsin\_MeanImpute[training,], BreastCancerWisconsin\_MeanImpute[testing,], k=k)*

*pred <- as.integer(fitted(BreastCancerWisconsin\_MeanImpute\_model)+0.5)*

*results\_knn\_mean[k] <- sum(pred == BreastCancerWisconsin\_MeanImpute[testing,]$V11) / nrow(BreastCancerWisconsin\_MeanImpute[testing,])*

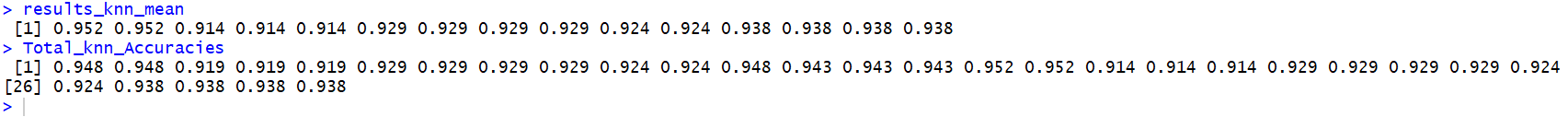
*Total\_knn\_Accuracies[k+15] <- sum(pred == BreastCancerWisconsin\_MeanImpute[testing,]$V11) / nrow(BreastCancerWisconsin\_MeanImpute[testing,])*

*}*

*results\_knn\_mode*

*Total\_knn\_Accuracies*

Output



The vector results\_knn\_mean shows accuracy above 90% for all the values of K and highest accuracy of 95.2% for k = 1 and 2 when applied for values of k = 1 to 15.

Regression Imputed Data

Code:

*results\_knn\_regression <- c()*

*for (k in 1:15) {*

*BreastCancerWisconsin\_RegImpute\_model <- kknn(V11~V2+V3+V4+V5+V6+V7+V8+V9+V10, BreastCancerWisconsin\_Reg\_Impute[training,], BreastCancerWisconsin\_Reg\_Impute[testing,], k=k)*

*pred <- as.integer(fitted(BreastCancerWisconsin\_MeanImpute\_model)+0.5)*

*results\_knn\_regression[k] <- sum(pred == BreastCancerWisconsin\_Reg\_Impute[testing,]$V11) / nrow(BreastCancerWisconsin\_Reg\_Impute[testing,])*

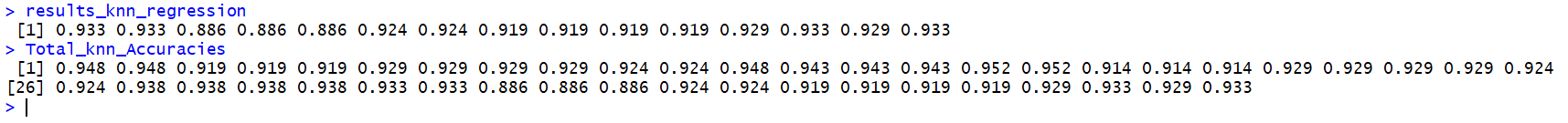
*Total\_knn\_Accuracies[k+30] <- sum(pred == BreastCancerWisconsin\_Reg\_Impute[testing,]$V11) / nrow(BreastCancerWisconsin\_Reg\_Impute[testing,])*

*}*

*results\_knn\_regression*

*Total\_knn\_Accuracies*

Output



The vector results\_knn\_regression shows accuracy above 90% for all the values of K and highest accuracy of 93.3% for k = 1 , 2 and 15 when applied for values of k = 1 to 15.

Regression with perturbation imputed data

Code:

*results\_knn\_peturbation <- c()*

*for (k in 1:15) {*

*BreastCancerWisconsin\_Petrubation\_model <- kknn(V11~V2+V3+V4+V5+V6+V7+V8+V9+V10, BreastCancerWisconsin\_Petrubation[training,], BreastCancerWisconsin\_Petrubation[testing,], k=k)*

*pred <- as.integer(fitted(BreastCancerWisconsin\_Petrubation\_model)+0.5)*

*results\_knn\_peturbation[k] <- sum(pred == BreastCancerWisconsin\_Petrubation[testing,]$V11) / nrow(BreastCancerWisconsin\_Petrubation[testing,])*

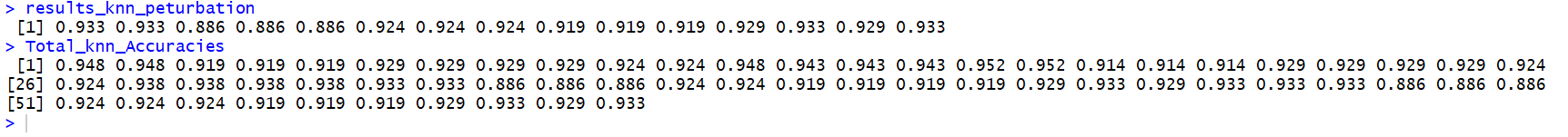
*Total\_knn\_Accuracies[k+45] <- sum(pred == BreastCancerWisconsin\_Petrubation[testing,]$V11) / nrow(BreastCancerWisconsin\_Petrubation[testing,])*

*}*

*results\_knn\_peturbation*

*Total\_knn\_Accuracies*

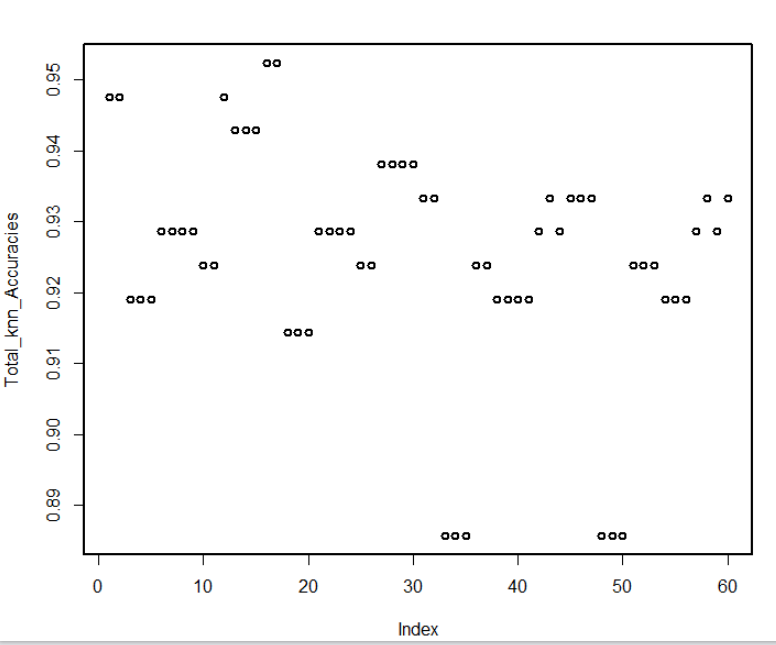
Output:



The vector results\_knn\_regression shows accuracy above 90% for all the values of K and highest accuracy of 93.3% for k = 1 , 2 and 15 when applied for values of k = 1 to 15.

The vector Total\_knn\_Accuracies was used in all the classification models above to store all the accuracies. This vector has all the 60 accuracies generated by the 4 KNN models above.

Lets plot the accuracies and see how it looks like.



When we look at the plot above except for few cases, accuracies for the k-nearest neighbor classification for different data sets are similar. This also confirms that the methods of imputation does not have a huge impact on our accuracies for our dataset.

Question 15.1 Describe a situation or problem from your job, everyday life, current events, etc., for which optimization would be appropriate. What data would you need?