Minor Project

By Siddhartha Sinha

1) What is the Problem Statement?

: - **Breast Cancer Detection.**

Breast Cancer is a tumour in the chest region which turns Malignant. A tumour can be Malignant (cancerous) or Benign (non- cancerous). A tumour can be analysed using its different features and then labelled as Malignant or Benign. A doctor uses his years of knowledge assisted by other medical tests to label the tumour. After analysing and collecting breast tumour attributes and its class we see a distinct pattern that can be used to classify tumour as Cancerous or Non-cancerous. With help of Machine Learning we can train the algorithm to classify tumours.

2)Which dataset is chosen?

: - Dataset for this project is breast_cancer.csv available at Breast Cancer Wisconsin
https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)
It has patient id along with 32 features and its respective diagnosis.

3)Libraries chosen?

caret: Classification and Regression Training
caTools: Moving window statistics
corrplot: Visualisation of a correlation matrix
dplyr: Rules for data manipulation
ggplot2: Data Visualisation such as heatmap ,etc
gridExtra: Miscellaneous Function for grid graphics
pROC: Display and analyse ROC curves
readr: Read Rectangular Text data
MASS: Support Functions and Datasets for venerable and Ripley's MASS

4)How cleaning/EDA was performed?

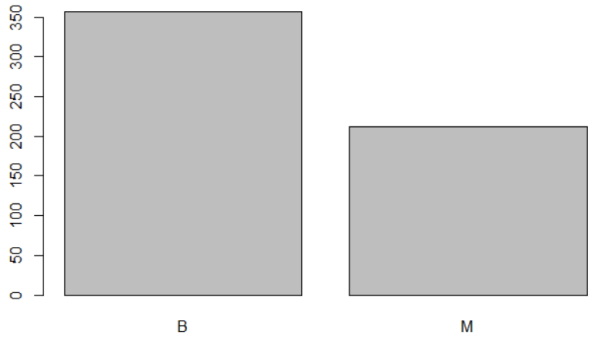
Firstly the dataset was converted into a dataframe. Then the information of dataframe was shown using

```
str(data)
```

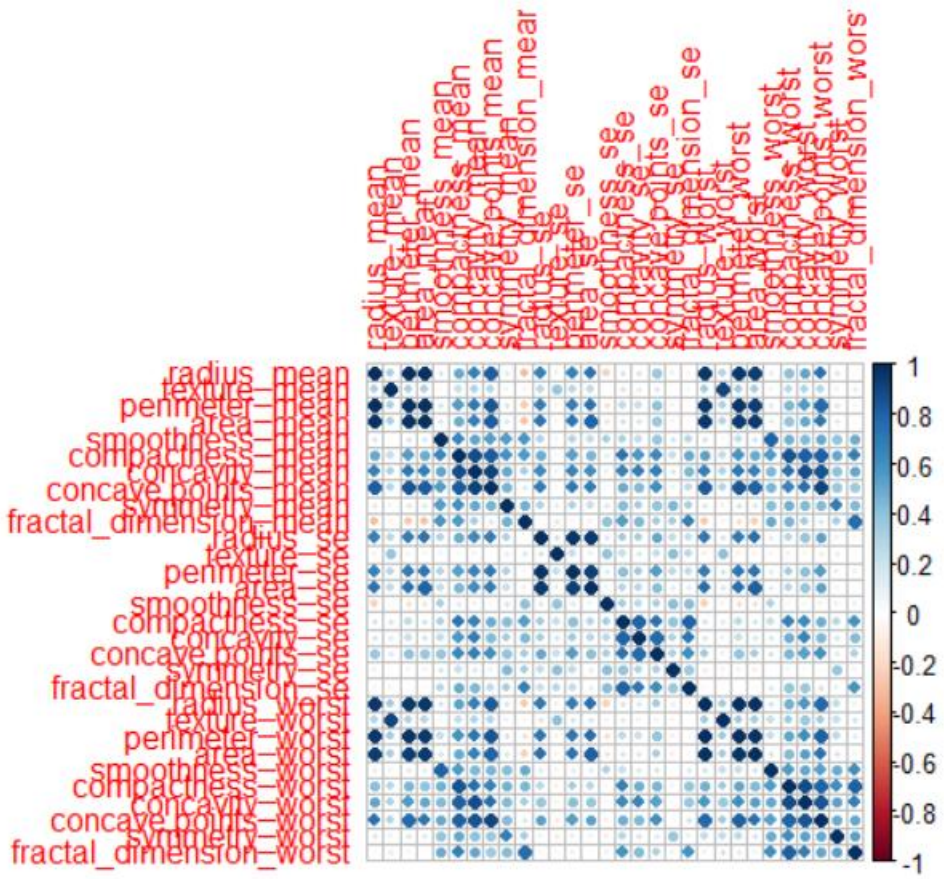
With this we selected the relevant dependent and independent features and also removed the null valued Column using

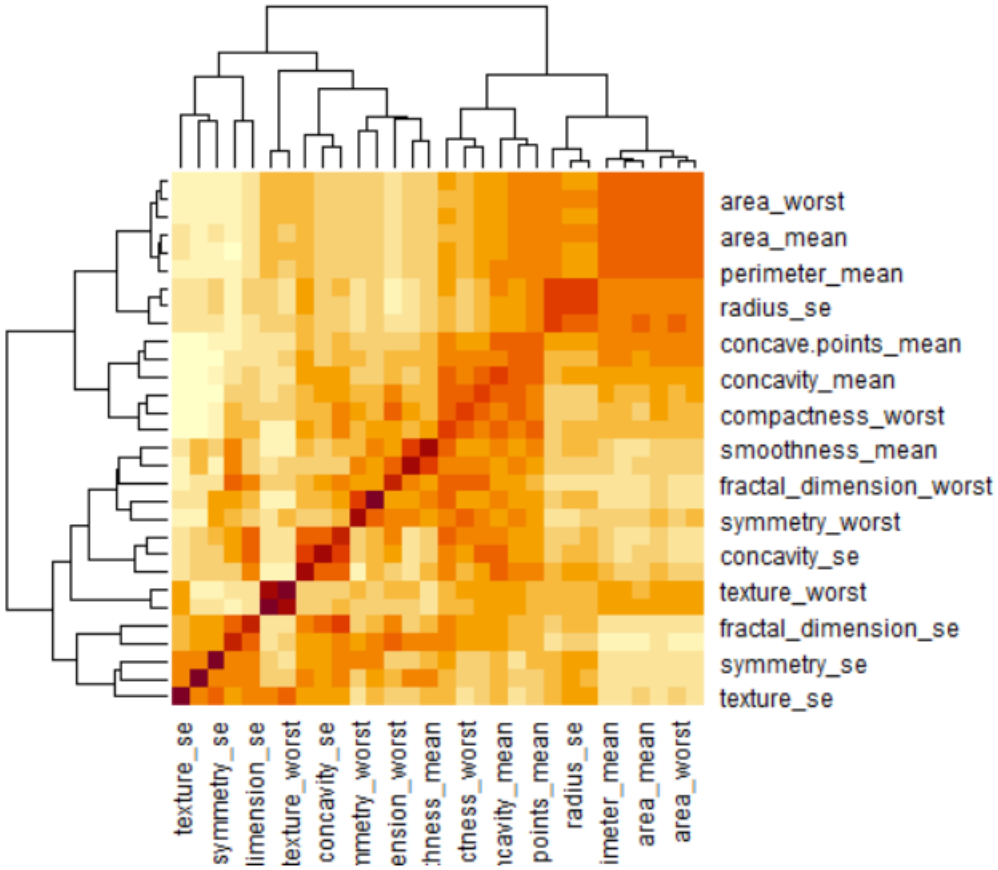
```
data$diagnosis <- as.factor(data$diagnosis)
data[,33] <- NULL
```

Then to find the amount of Malignant and Benign cases in the dataset
prop.table(table(data$diagnosis))
plot(data$diagnosis) was used, from that we deduced that the dataset had 62.74% Benign cases and 37.25% Malignant cases.

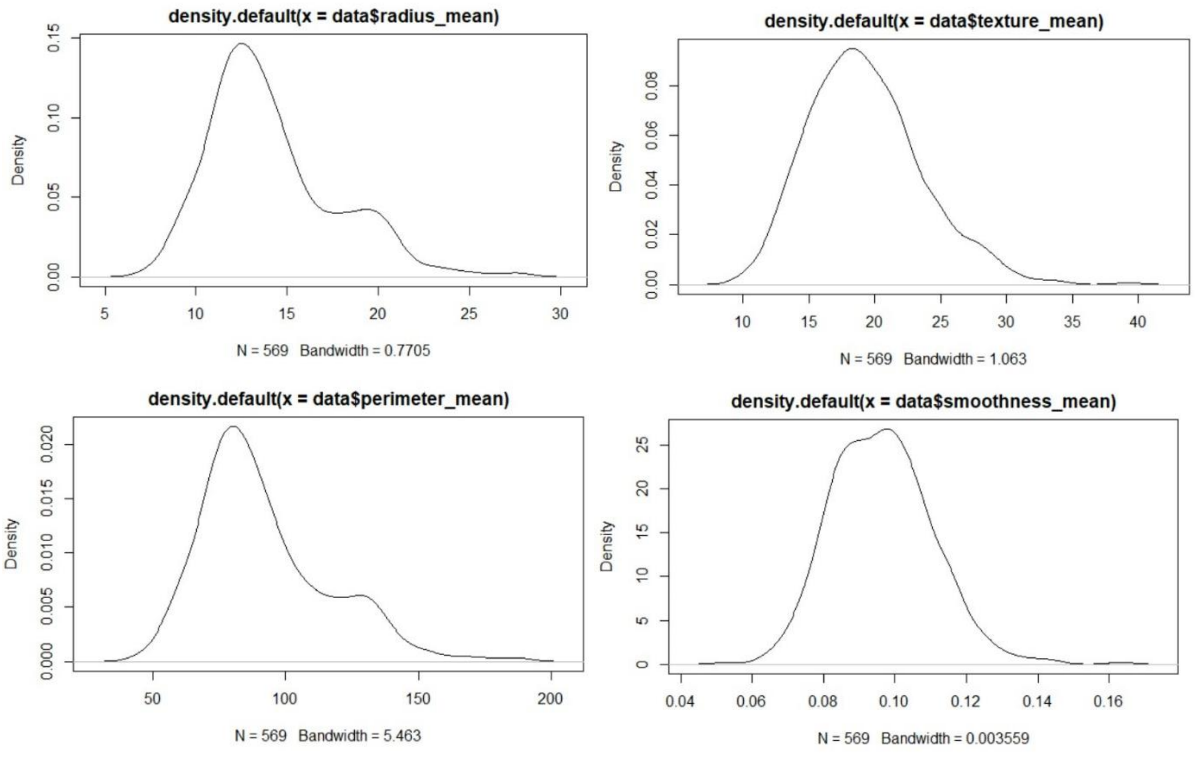


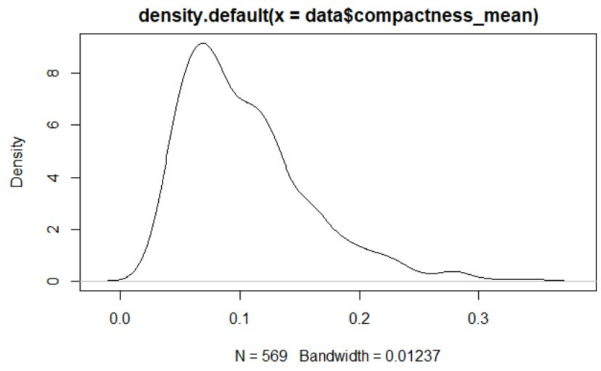
To see the correlation between the features we plotted the correlation matrix and heatmap.

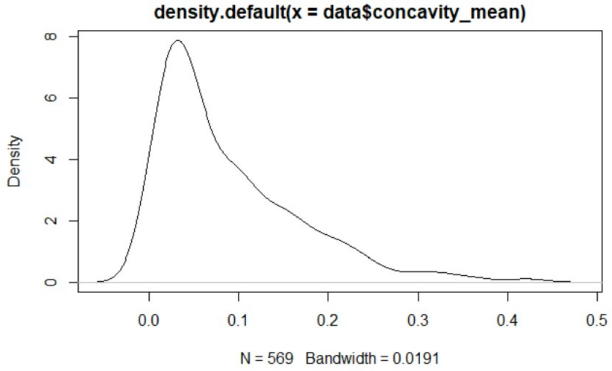


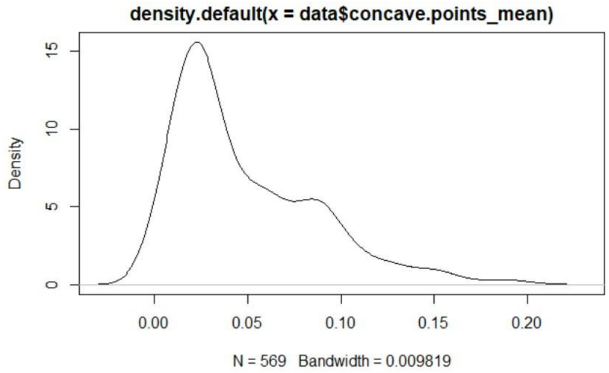


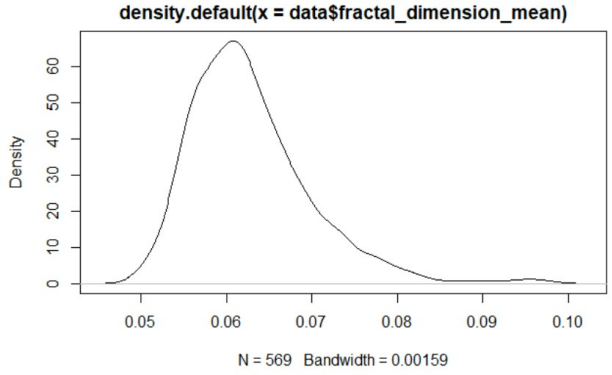
Then in order to properly fit the features we needed to know the density of the individual features.

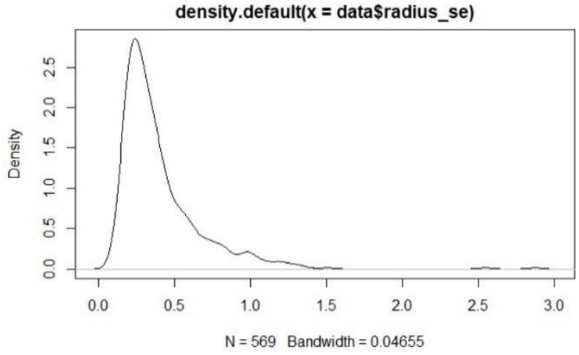


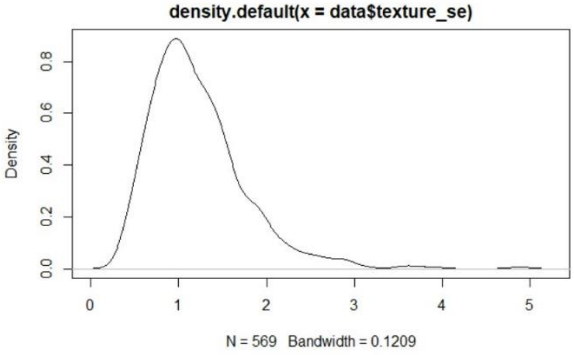

density.default(x = data$compactness_mean)
Density
N = 569 Bandwidth = 0.01237

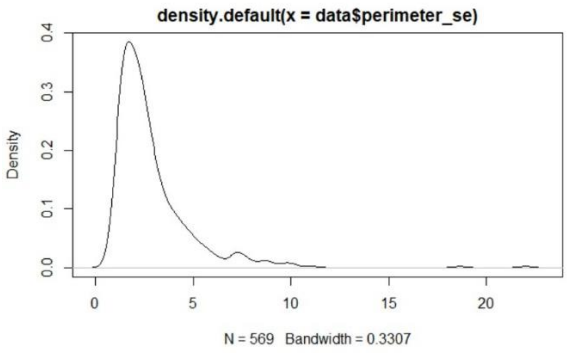

density.default(x = data$concavity_mean)
Density
N = 569 Bandwidth = 0.0191

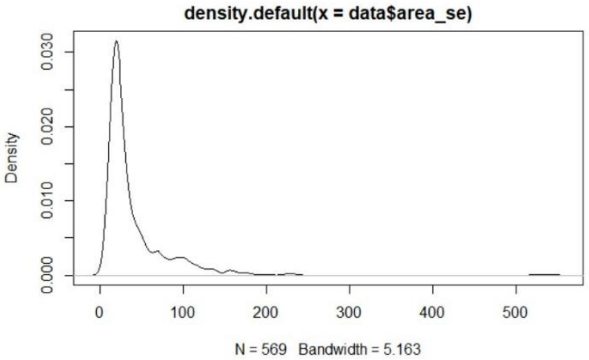

density.default(x = data$concave.points_mean)
Density
N = 569 Bandwidth = 0.009819

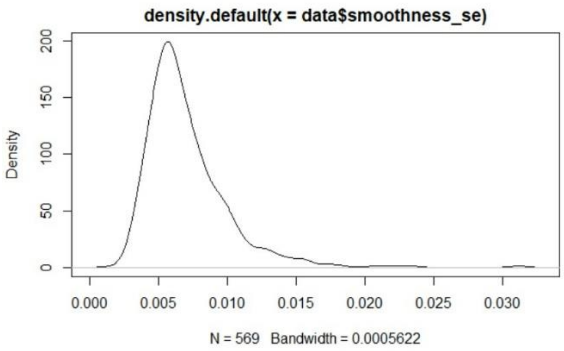

density.default(x = data$fractal_dimension_mean)
Density
N = 569 Bandwidth = 0.00159

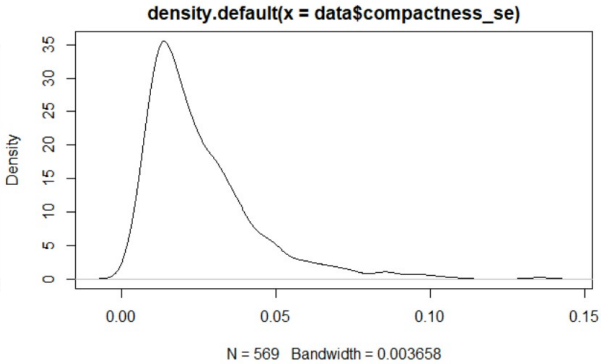

density.default(x = data$radius_se)
Density
N = 569 Bandwidth = 0.04655

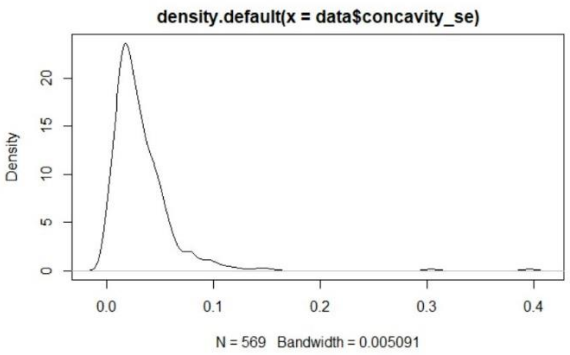

density.default(x = data$texture_se)
Density
N = 569 Bandwidth = 0.1209

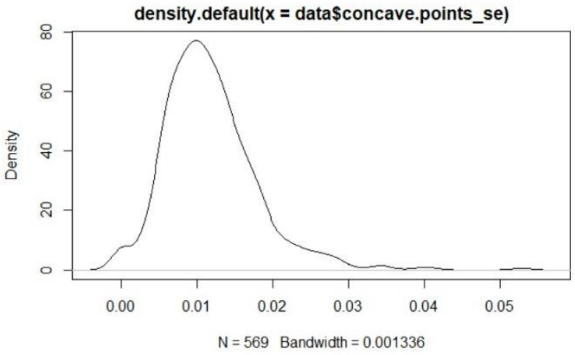

density.default(x = data$perimeter_se)
Density
N = 569 Bandwidth = 0.3307

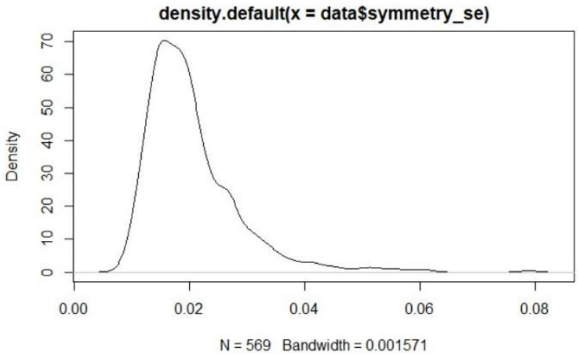

density.default(x = data$area_se)
Density
N = 569 Bandwidth = 5.163

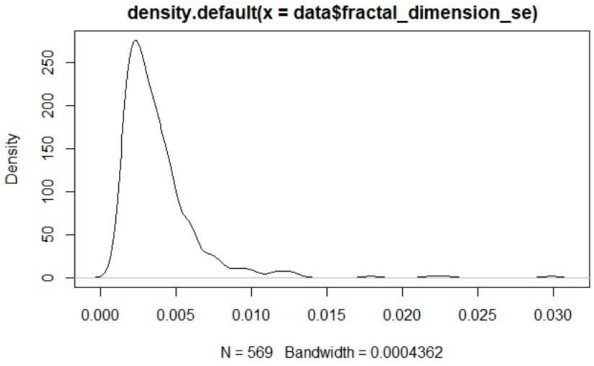

density.default(x = data$smoothness_se)
Density
N = 569 Bandwidth = 0.0005622

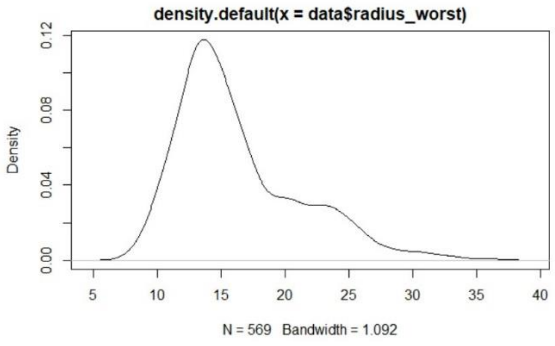

density.default(x = data$compactness_se)
Density
N = 569 Bandwidth = 0.003658

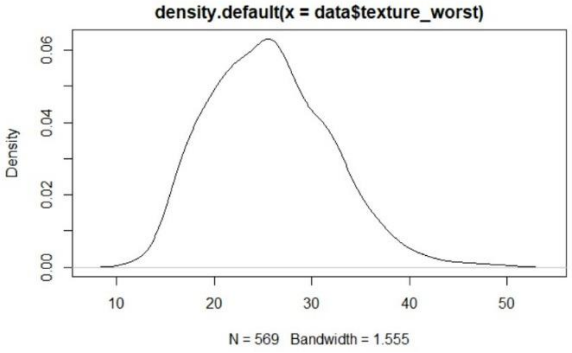

density.default(x = data$concavity_se)
Density
N = 569 Bandwidth = 0.005091

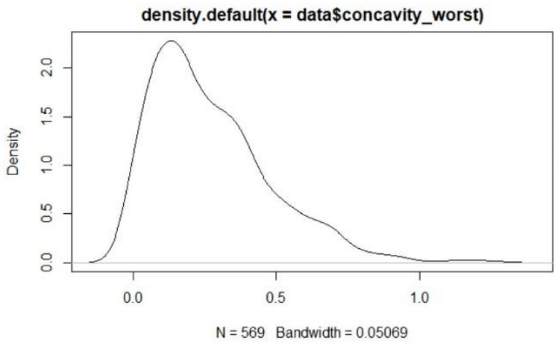

density.default(x = data$concave.points_se)
Density
N = 569 Bandwidth = 0.001336

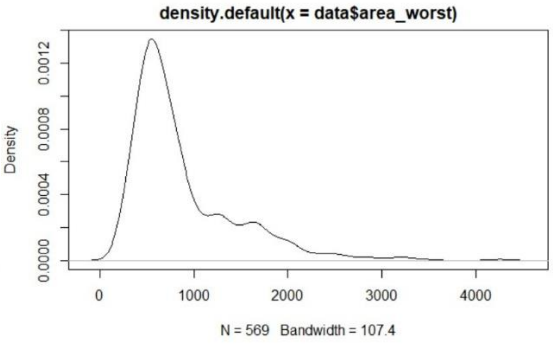

density.default(x = data$symmetry_se)
Density
N = 569 Bandwidth = 0.001571

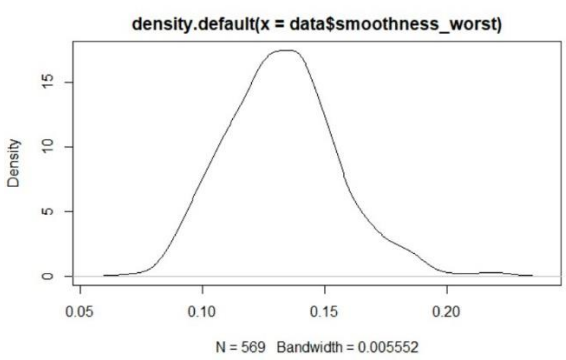

density.default(x = data$fractal_dimension_se)
Density
N = 569 Bandwidth = 0.0004362

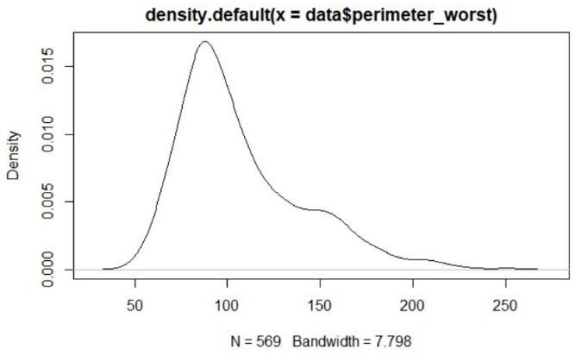

density.default(x = data$radius_worst)
Density
N = 569 Bandwidth = 1.092

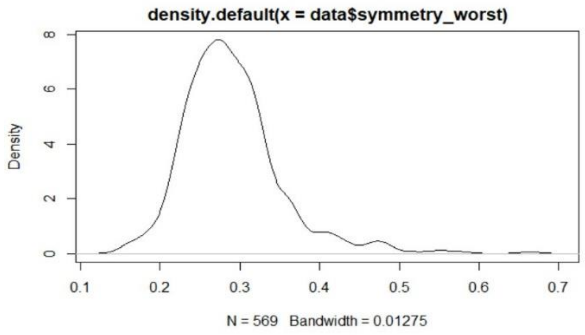

density.default(x = data$texture_worst)
Density
N = 569 Bandwidth = 1.555

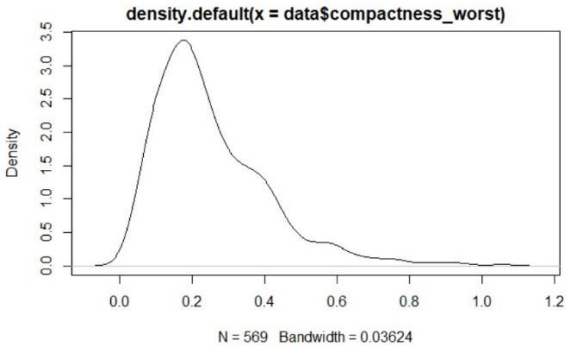

density.default(x = data$concavity_worst)
Density
N = 569 Bandwidth = 0.05069

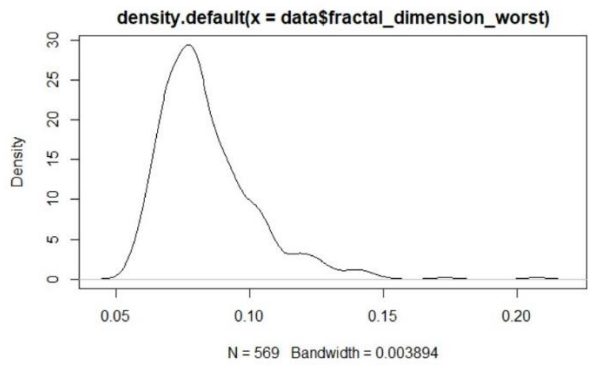

density.default(x = data$area_worst)
Density
N = 569 Bandwidth = 107.4

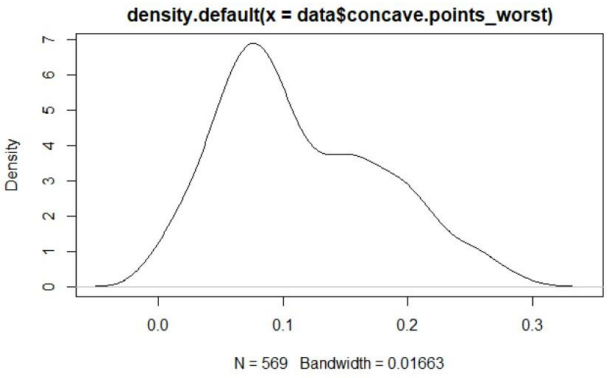












5)Independent and Dependent feature?

: - Independent Feature: diagnosis

Dependent Feature:
radius_mean
texture_mean
perimeter_mean
area_mean
smoothness_mean
compactness_mean
concavity_mean
concave points_mean
symmetry_mean
fractal_dimension_mean
radius_se
texture_se
perimeter_se

area_se
smoothness_se
compactness_se
concavity_se
concave points_se
symmetry_se
fractal_dimension_se
radius_worst
texture_worst
perimeter_worst
area_worst
smoothness_worst
compactness_worst
concavity_worst
concave points_worst
symmetry_worst
fractal_dimension_worst

6)Why and how selection/engineering/scaling were performed?

:- Feature scaling had to be done as the featured ranged from magnitude of 10^1 to 10^3 so using

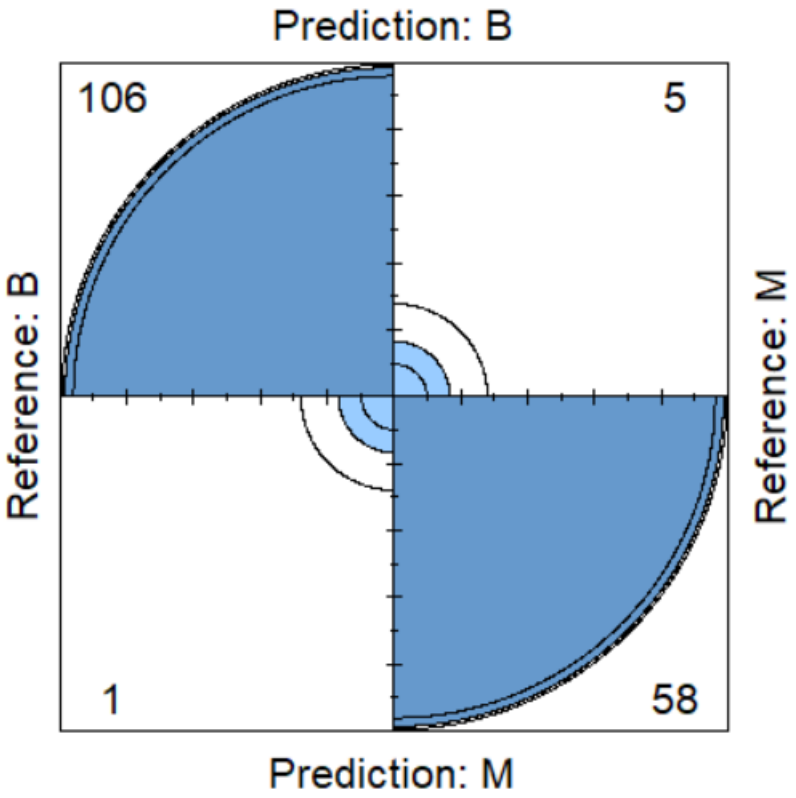
```
set.seed(1234)
data_index <- createDataPartition(data$diagnosis, p=0.7, list = FALSE)
train_data <- data[data_index, -1]
test_data <- data[-data_index, -1]

fitControl <- trainControl(method="cv",
                           number = 5,
                           preProcOptions = list(thresh = 0.9),
                           classProbs = TRUE,
                           summaryFunction = twoClassSummary)
```

Feature scaling was done.

7)Which Classifier was chosen and why?

Random Forest Classifier was chosen on this dataset as its accuracy was found to be better than Naïve Bayes (94.71% which greater than NB 91.3%). However KNN and boosted tree Accuracy (97%) was more than Random Forest , after hyper parameter tuning we obtained the accuracy close to boosted tree method (96.47%).


Prediction: B
106
5
Reference: B
Reference: M
1
58
Prediction: M