R_Final_Project

Gaurav_Kothari, Sanika Moghe, Chaitanya Vyas

November 15, 2018

Import Librarires

```
Loadlibraries=function(){
library(rlang)
library(modelr)
library(nycflights13)
library(lubridate)
library(ISLR)
  print("The libraries have been loaded.")
  }
```

Feature Selection and Data Elimination

```
#Dividing Data into testing and Trainig set with the help of 'sample' function.

Data<-read.csv("bc_data.csv")
bc_new_data<-Data[-1]

set.seed(400)

data_sample <- sample(nrow(bc_new_data), nrow(bc_new_data)*3/4)
data_train <- bc_new_data[data_sample,]
data_test <- bc_new_data[-data_sample,]

#feature selection

library(corrplot)
```

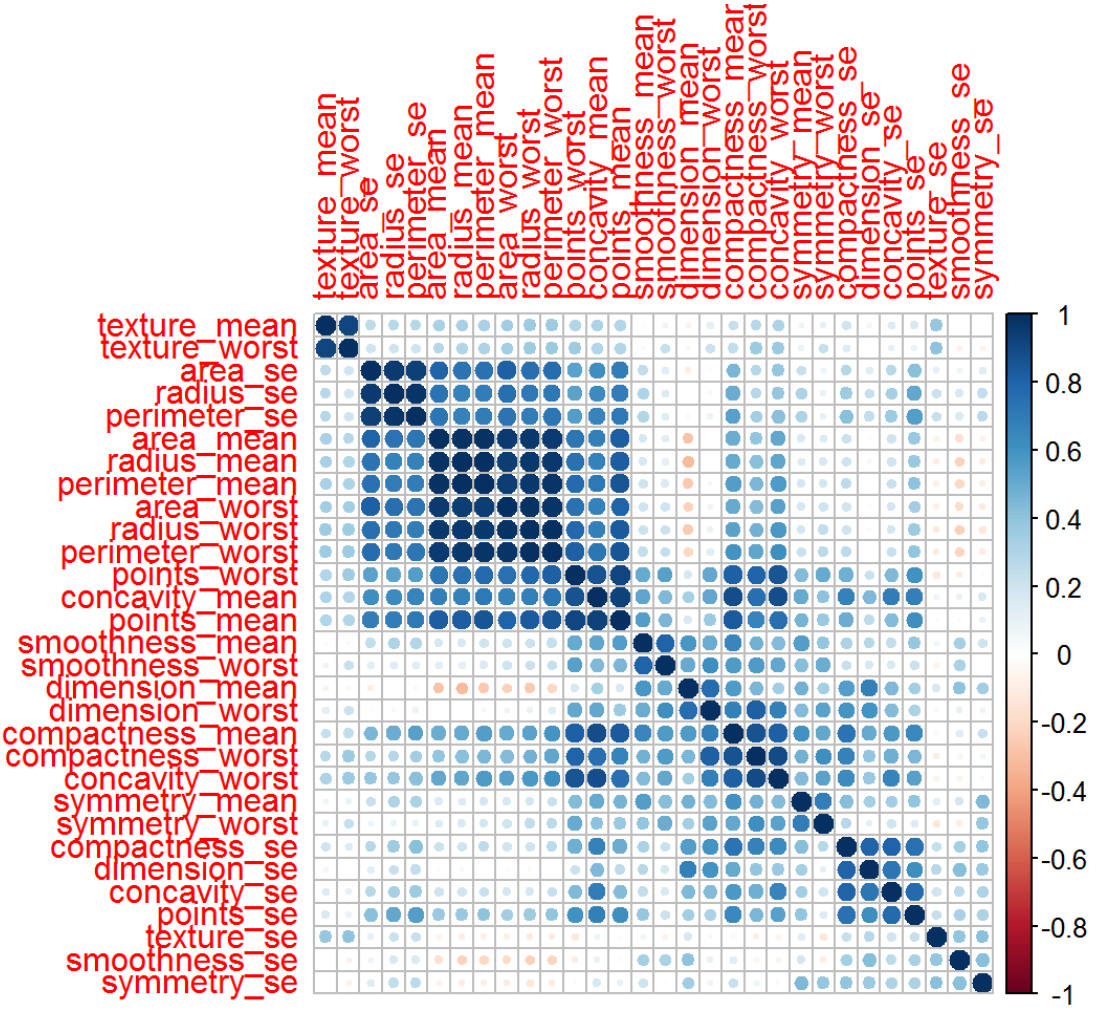
```
## corrplot 0.84 loaded
```

```
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
#Here first let's find the corelation between different features.
corMatMy <- cor(bc_new_data[2:31])
corrplot(corMatMy, order= "hclust")
```



```
#As we can see from the plot that there are number of features which are highly
 correlated. It is advisable to remove the highly correlated features (for ex.
 area and radius) to reduce the complexity of models.

#Here, we are removing all the feaures which has correlation of 0.7 or more.

highlyCor <- colnames(data_train[,-1])[findCorrelation(corMatMy, cutoff = 0.7,
 verbose = TRUE)]
```

```
## Compare row 7  and column  8 with corr  0.921
##   Means:  0.571 vs 0.389 so flagging column 7
## Compare row 8  and column  6 with corr  0.831
##   Means:  0.542 vs 0.377 so flagging column 8
## Compare row 6  and column  28 with corr  0.816
##   Means:  0.524 vs 0.365 so flagging column 6
## Compare row 28  and column  27 with corr  0.855
##   Means:  0.507 vs 0.354 so flagging column 28
## Compare row 27  and column  26 with corr  0.892
##   Means:  0.457 vs 0.343 so flagging column 27
## Compare row 23  and column  21 with corr  0.994
##   Means:  0.456 vs 0.333 so flagging column 23
## Compare row 21  and column  3 with corr  0.969
##   Means:  0.422 vs 0.324 so flagging column 21
## Compare row 3  and column  24 with corr  0.942
##   Means:  0.384 vs 0.316 so flagging column 3
## Compare row 26  and column  30 with corr  0.81
##   Means:  0.4 vs 0.313 so flagging column 26
## Compare row 24  and column  1 with corr  0.941
##   Means:  0.356 vs 0.302 so flagging column 24
## Compare row 1  and column  4 with corr  0.987
##   Means:  0.308 vs 0.298 so flagging column 1
## Compare row 4  and column  13 with corr  0.727
##   Means:  0.27 vs 0.294 so flagging column 13
## Compare row 4  and column  11 with corr  0.733
##   Means:  0.244 vs 0.29 so flagging column 11
## Compare row 4  and column  14 with corr  0.8
##   Means:  0.213 vs 0.294 so flagging column 14
## Compare row 18  and column  16 with corr  0.744
##   Means:  0.36 vs 0.292 so flagging column 18
## Compare row 16  and column  17 with corr  0.801
##   Means:  0.394 vs 0.288 so flagging column 16
## Compare row 17  and column  20 with corr  0.727
##   Means:  0.292 vs 0.272 so flagging column 17
## Compare row 5  and column  25 with corr  0.805
##   Means:  0.33 vs 0.268 so flagging column 5
## Compare row 10  and column  30 with corr  0.767
##   Means:  0.372 vs 0.256 so flagging column 10
## Compare row 22  and column  2 with corr  0.912
##   Means:  0.253 vs 0.243 so flagging column 22
## All correlations <= 0.7
```

```
highlyCor
```

```
##  [1] "concavity_mean"    "points_mean"       "compactness_mean"
##  [4] "points_worst"      "concavity_worst"   "perimeter_worst"
##  [7] "radius_worst"      "perimeter_mean"    "compactness_worst"
## [10] "area_worst"        "radius_mean"       "perimeter_se"
## [13] "radius_se"         "points_se"         "compactness_se"
## [16] "area_se"           "concavity_se"      "smoothness_mean"
## [19] "dimension_mean"    "texture_worst"
```

```
#feature elimination
train_data_cor <- data_train[, which(!colnames(data_train) %in% highlyCor)]

test_data_cor<-data_test[, which(!colnames(data_test)%in% highlyCor)]

#Feature eliminated whole dataset

data_cor<-bc_new_data[, which(!colnames(bc_new_data)%in% highlyCor)]

#After the elimination, there are total 10 most important features
```

Exploratory data analysis

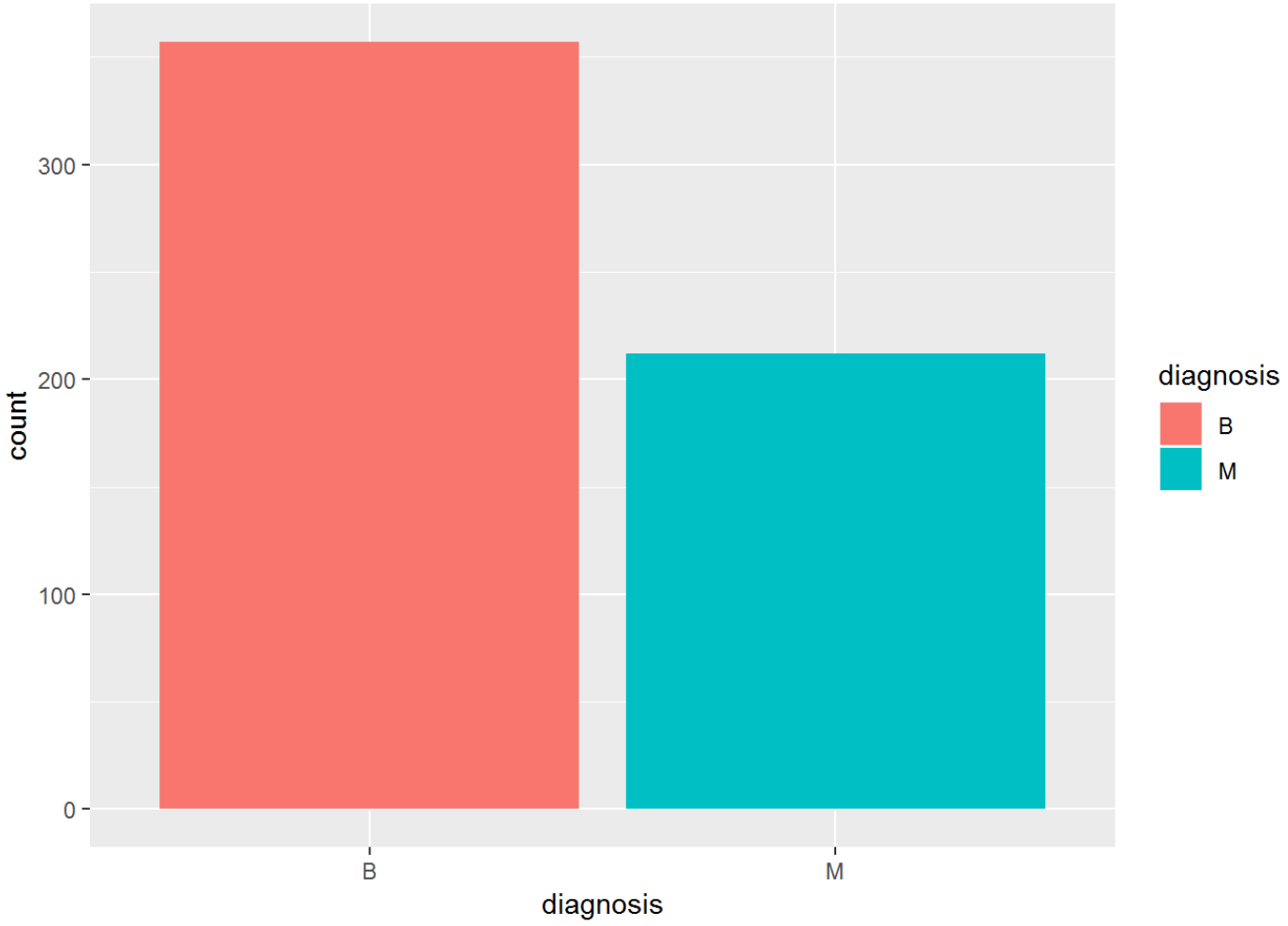
```
library(tidyverse)
```

```
## -- Attaching packages --------------------- tidyverse 1.2.1 --
```

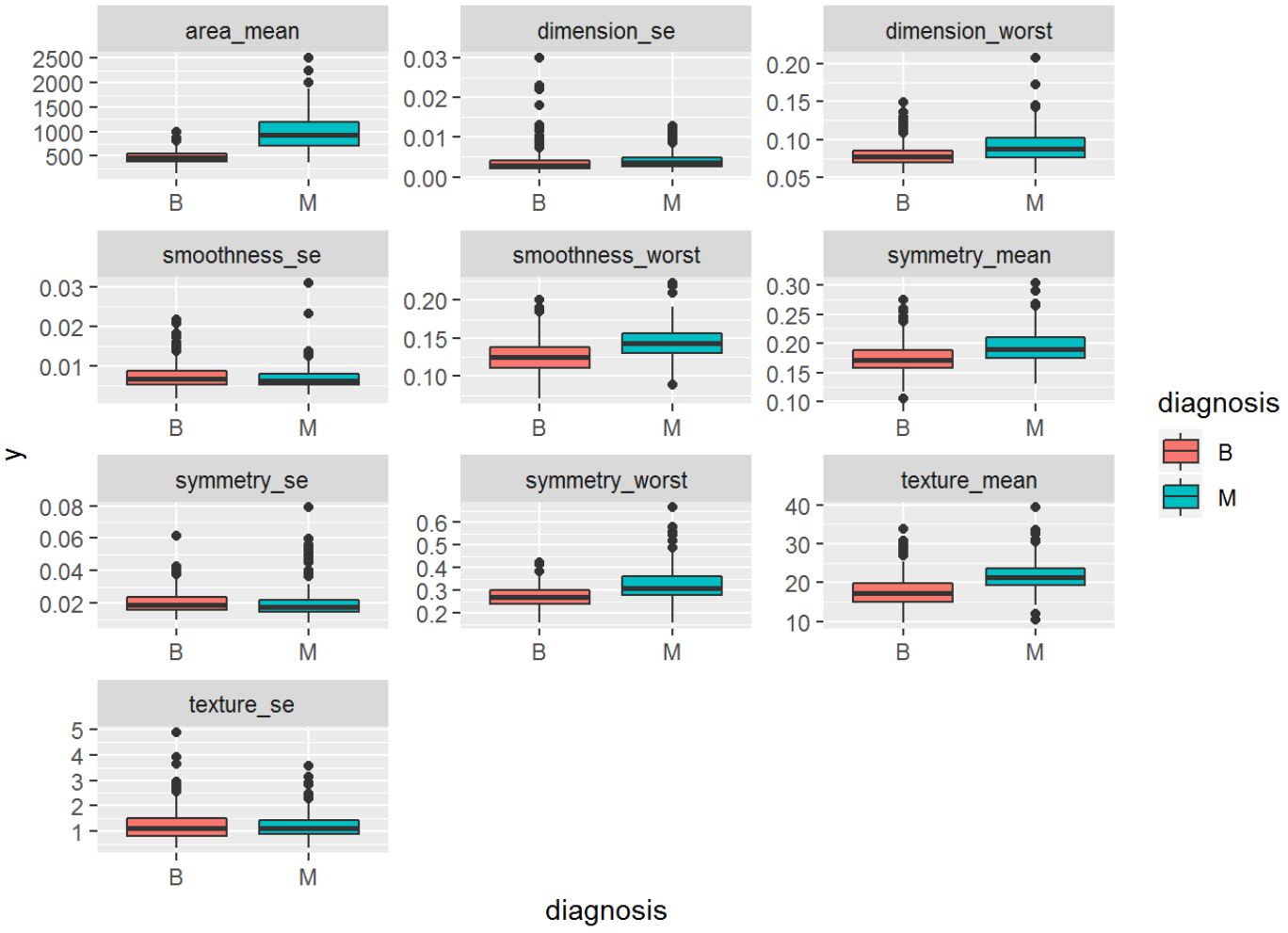
```
## v tibble  1.4.2     v purrr   0.2.5
## v tidyr   0.8.1     v dplyr   0.7.6
## v readr   1.1.1     v stringr 1.3.1
## v tibble  1.4.2     v forcats 0.3.0
```

```
## -- Conflicts ------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x purrr::lift()   masks caret::lift()
```

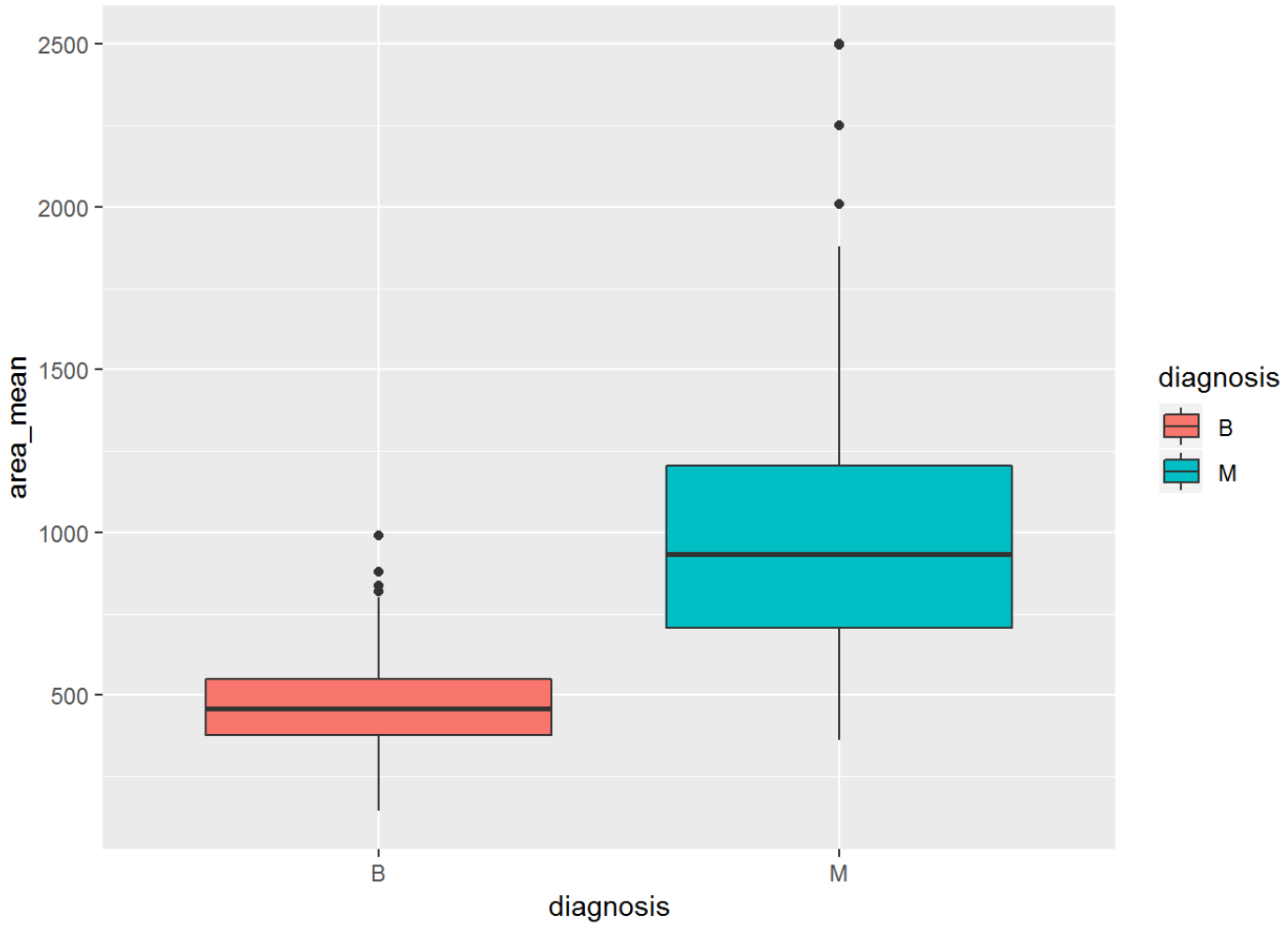
```
library(ggplot2)

#Let's check how many of the cases have the Positive cancer result


ggplot(data=Data)+
 geom_bar(mapping=aes(x=diagnosis, fill=diagnosis))
```




```
#Relationship between radius_mean and diagnosos

gather(data_cor, x, y, texture_mean:dimension_worst)%>%
ggplot(aes(x=diagnosis, y=y, fill=diagnosis))+
  geom_boxplot()+
  facet_wrap( ~ x, scales = "free", ncol = 3)
```




```
#As from all the plots, we can see that area _mean is one of the most important
 factors in the cancer diognisis.

ggplot(data=data_cor)+
  geom_boxplot(mapping=aes(x=diagnosis,y=area_mean, fill=diagnosis))
```



Logistic Regression

```
#Logistic Regression

lg=glm(diagnosis~texture_mean+area_mean+symmetry_mean+texture_se+symmetry_se+sm
oothness_se+dimension_se+smoothness_worst+symmetry_worst+dimension_worst, data=
train_data_cor, family=binomial)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(lg)
```

```
##
## Call:
## glm(formula = diagnosis ~ texture_mean + area_mean + symmetry_mean +
##     texture_se + symmetry_se + smoothness_se + dimension_se +
##     smoothness_worst + symmetry_worst + dimension_worst, family = binomial,
##     data = train_data_cor)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.67938  -0.06438  -0.00568   0.00087   2.87724
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -5.173e+01  8.473e+00  -6.105 1.03e-09 ***
## texture_mean      4.052e-01  1.116e-01   3.631 0.000283 ***
## area_mean         2.726e-02  4.485e-03   6.078 1.22e-09 ***
## symmetry_mean     5.952e+00  2.196e+01   0.271 0.786352
## texture_se        1.349e+00  9.018e-01   1.496 0.134767
## symmetry_se      -3.705e+01  6.280e+01  -0.590 0.555151
## smoothness_se     2.904e+02  2.544e+02   1.142 0.253648
## dimension_se     -1.234e+03  6.192e+02  -1.994 0.046198 *
## smoothness_worst  9.153e+01  3.527e+01   2.595 0.009458 **
## symmetry_worst    1.467e+01  1.272e+01   1.154 0.248699
## dimension_worst   1.131e+02  6.852e+01   1.650 0.098952 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 557.44  on 425  degrees of freedom
## Residual deviance:  71.05  on 415  degrees of freedom
## AIC: 93.05
##
## Number of Fisher Scoring iterations: 9
```

```
#Removing features which has larger p values

lg2=glm(diagnosis~texture_mean+area_mean+dimension_se+smoothness_worst, data=tr
ain_data_cor, family=binomial)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

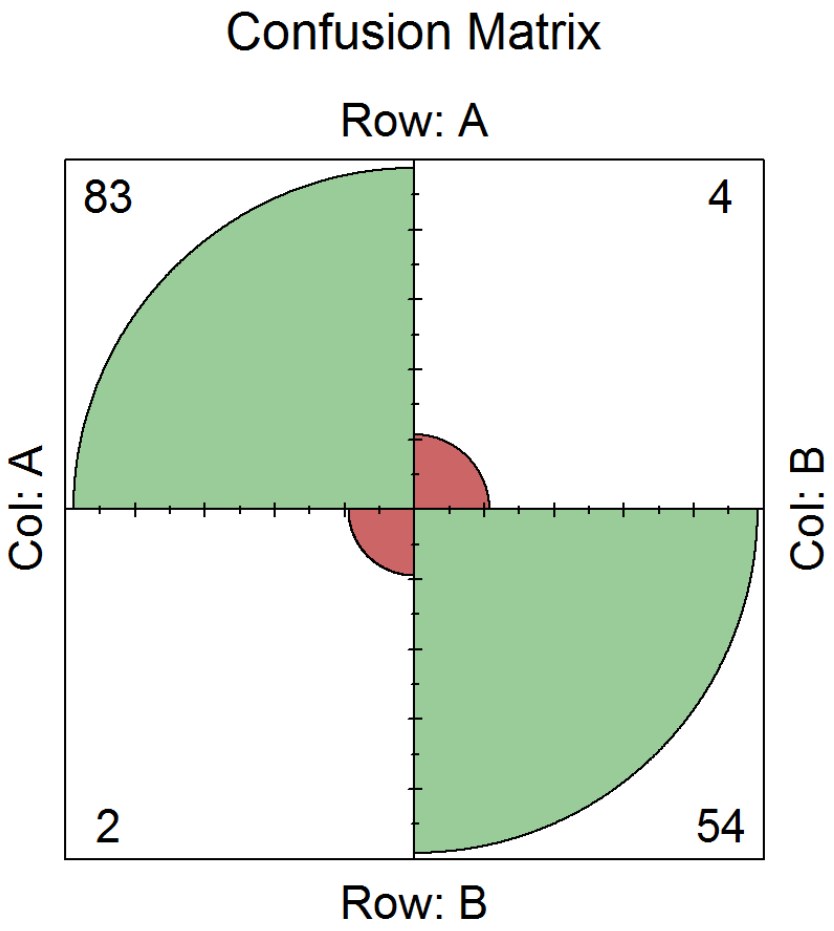
```
summary(lg2)
```

```
##
## Call:
## glm(formula = diagnosis ~ texture_mean + area_mean + dimension_se +
##     smoothness_worst, family = binomial, data = train_data_cor)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.27195  -0.09347  -0.01074   0.00194   3.04315
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -42.94778    6.28789  -6.830 8.48e-12 ***
## texture_mean        0.42583    0.08612   4.945 7.62e-07 ***
## area_mean           0.02408    0.00361   6.670 2.56e-11 ***
## dimension_se     -265.92955  243.24485  -1.093    0.274
## smoothness_worst  145.55148   24.17866   6.020 1.75e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 557.445  on 425  degrees of freedom
## Residual deviance:  86.839  on 421  degrees of freedom
## AIC: 96.839
##
## Number of Fisher Scoring iterations: 9
```

```
probability = predict(lg,test_data_cor,type = "response")
pred.glm = rep("B", length(probability))
pred.glm[probability > 0.5] = "M"
with(test_data_cor, table(pred.glm,diagnosis))
```

```
##         diagnosis
## pred.glm  B  M
##        B 83  4
##        M  2 54
```

```
ctable_lg <- as.table(matrix(c(83, 4, 2, 54), nrow = 2, byrow = TRUE))
fourfoldplot(ctable_lg, color = c("#CC6666", "#99CC99"),
             conf.level = 0, margin = 1, main = "Confusion Matrix")
```




```
#Accuracy = ((83+54) *100)/143 = 95%
```

LDA

```
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

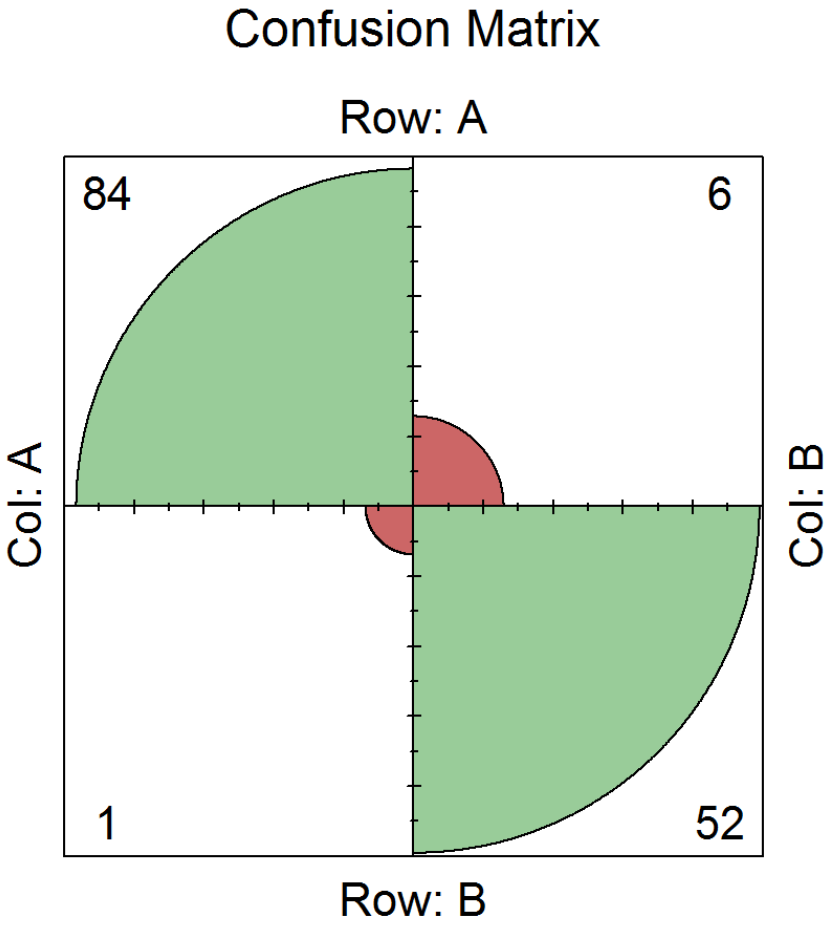
```
lda=lda(diagnosis~texture_mean+area_mean+symmetry_mean+texture_se+symmetry_se+s
moothness_se+dimension_se+smoothness_worst+symmetry_worst+dimension_worst, data
=train_data_cor)

lda
```

```
## Call:
## lda(diagnosis ~ texture_mean + area_mean + symmetry_mean + texture_se +
##     symmetry_se + smoothness_se + dimension_se + smoothness_worst +
##     symmetry_worst + dimension_worst, data = train_data_cor)
##
## Prior probabilities of groups:
##         B         M
## 0.6384977 0.3615023
##
## Group means:
##   texture_mean area_mean symmetry_mean texture_se symmetry_se
## B      17.7643  465.8368     0.1737430   1.215479  0.02038661
## M      21.3676  990.9383     0.1936701   1.178033  0.02016447
##   smoothness_se dimension_se smoothness_worst symmetry_worst
## B   0.007185886  0.003661715        0.1256989      0.2698110
## M   0.006909532  0.003956604        0.1455915      0.3223422
##   dimension_worst
## B      0.07971963
## M      0.09106253
##
## Coefficients of linear discriminants:
##                             LD1
## texture_mean       0.087912110
## area_mean          0.003553614
## symmetry_mean     -0.943622562
## texture_se         0.078271225
## symmetry_se      -17.794526729
## smoothness_se     19.749761526
## dimension_se     -28.128207151
## smoothness_worst  15.087628365
## symmetry_worst     7.333486692
## dimension_worst    9.952983943
```

```
pred.lda = predict(lda, test_data_cor)
with(test_data_cor, table(pred.lda$class,diagnosis))
```

```
##    diagnosis
##       B  M
##   B 84  6
##   M  1 52
```

```
ctable_lda<- as.table(matrix(c(84, 6, 1, 52), nrow = 2, byrow = TRUE))
fourfoldplot(ctable_lda, color = c("#CC6666", "#99CC99"),
             conf.level = 0, margin = 1, main = "Confusion Matrix")
```




```
#Accuracy = ((84+52)*100/143)= 95%
```

QDA

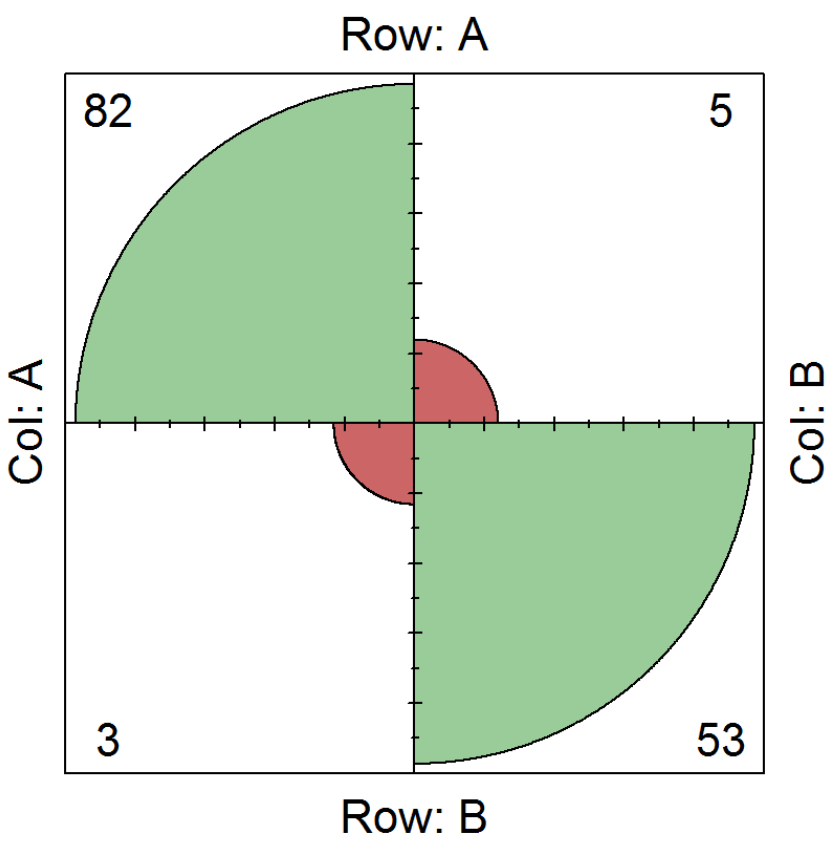
```
library(MASS)

qda=qda(diagnosis~., data=train_data_cor)

qda
```

```
## Call:
## qda(diagnosis ~ ., data = train_data_cor)
##
## Prior probabilities of groups:
##         B         M
## 0.6384977 0.3615023
##
## Group means:
##   texture_mean area_mean symmetry_mean texture_se smoothness_se
## B      17.7643  465.8368     0.1737430   1.215479   0.007185886
## M      21.3676  990.9383     0.1936701   1.178033   0.006909532
##   symmetry_se dimension_se smoothness_worst symmetry_worst dimension_worst
## B  0.02038661  0.003661715        0.1256989      0.2698110      0.07971963
## M  0.02016447  0.003956604        0.1455915      0.3223422      0.09106253
```

```
pred.qda = predict(qda, test_data_cor)
with(test_data_cor,table(pred.qda$class, test_data_cor$diagnosis))
```

```
##
##      B  M
##   B 82  5
##   M  3 53
```

```
ctable_qda<- as.table(matrix(c(82, 5, 3, 53), nrow = 2, byrow = TRUE))
fourfoldplot(ctable_qda, color = c("#CC6666", "#99CC99"),
             conf.level = 0, margin = 1, main = "Confusion Matrix")
```



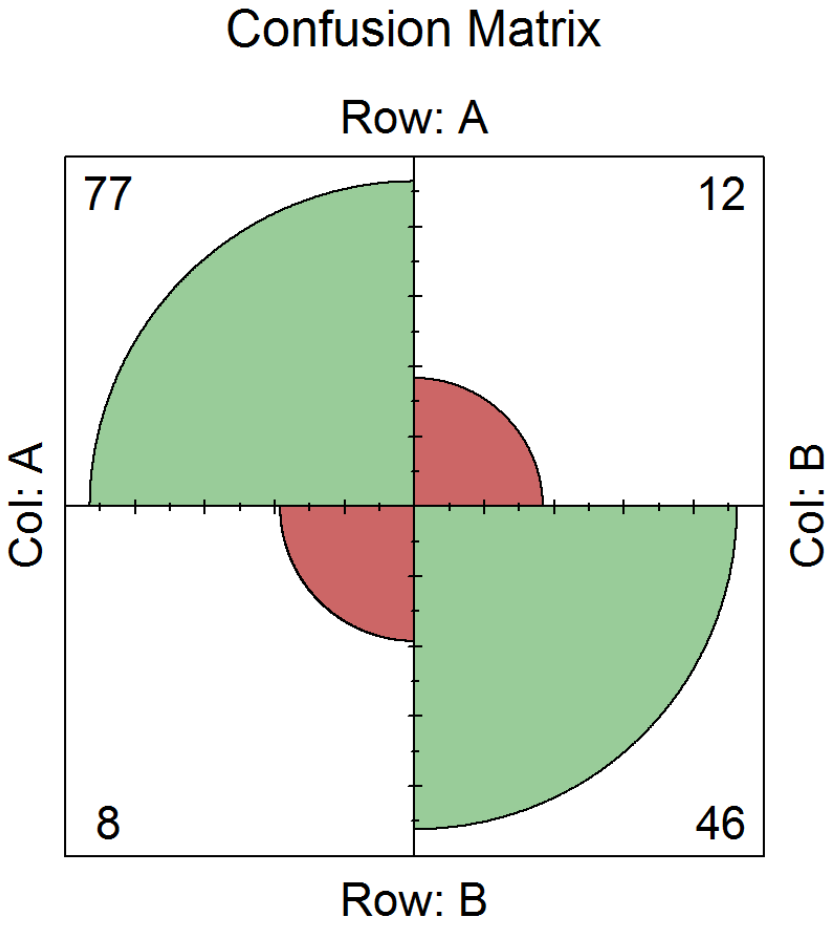

```
#Accuracy = ((82+53)*100)/143 = 94%
```

KNN

```
library(class)

knntrain=train_data_cor[,2:11]
knntest=test_data_cor[,2:11]
knnlabel=train_data_cor[,1]

#KNN with K=1
knn.pred1=knn(knntrain,knntest,knnlabel,k=1)

table(knn.pred1,test_data_cor$diagnosis)
```

```
##
## knn.pred1  B  M
##         B 77 12
##         M  8 46
```

```
ctable_knn1<- as.table(matrix(c(77, 12, 8, 46), nrow = 2, byrow = TRUE))
fourfoldplot(ctable_knn1, color = c("#CC6666", "#99CC99"),
             conf.level = 0, margin = 1, main = "Confusion Matrix")
```



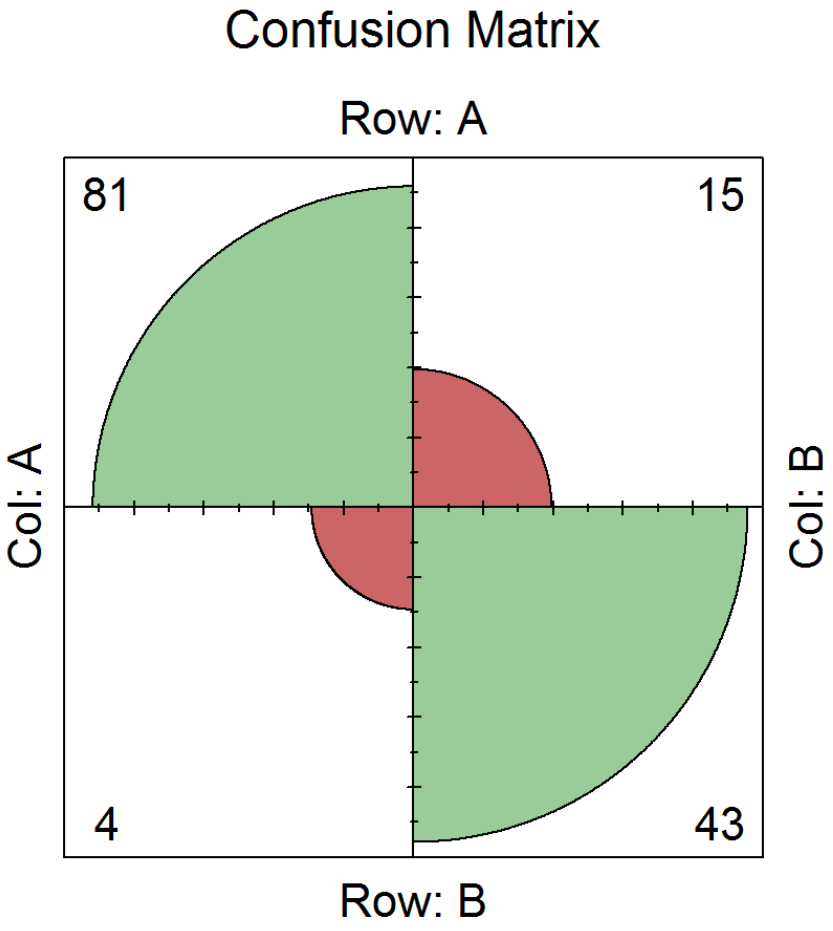

```
mean(knn.pred1==test_data_cor$diagnosis)
```

```
## [1] 0.8601399
```

```
#Accuracy = ((77+46)*100/143) = 86%


#KNN with K=100

knn.pred2=knn(knntrain,knntest,knnlabel,k=100)

table(knn.pred2,test_data_cor$diagnosis)
```

```
##
## knn.pred2  B  M
##         B 81 15
##         M  4 43
```

```
ctable_knn10<- as.table(matrix(c(81, 15, 4, 43), nrow = 2, byrow = TRUE))
fourfoldplot(ctable_knn10, color = c("#CC6666", "#99CC99"),
             conf.level = 0, margin = 1, main = "Confusion Matrix")
```



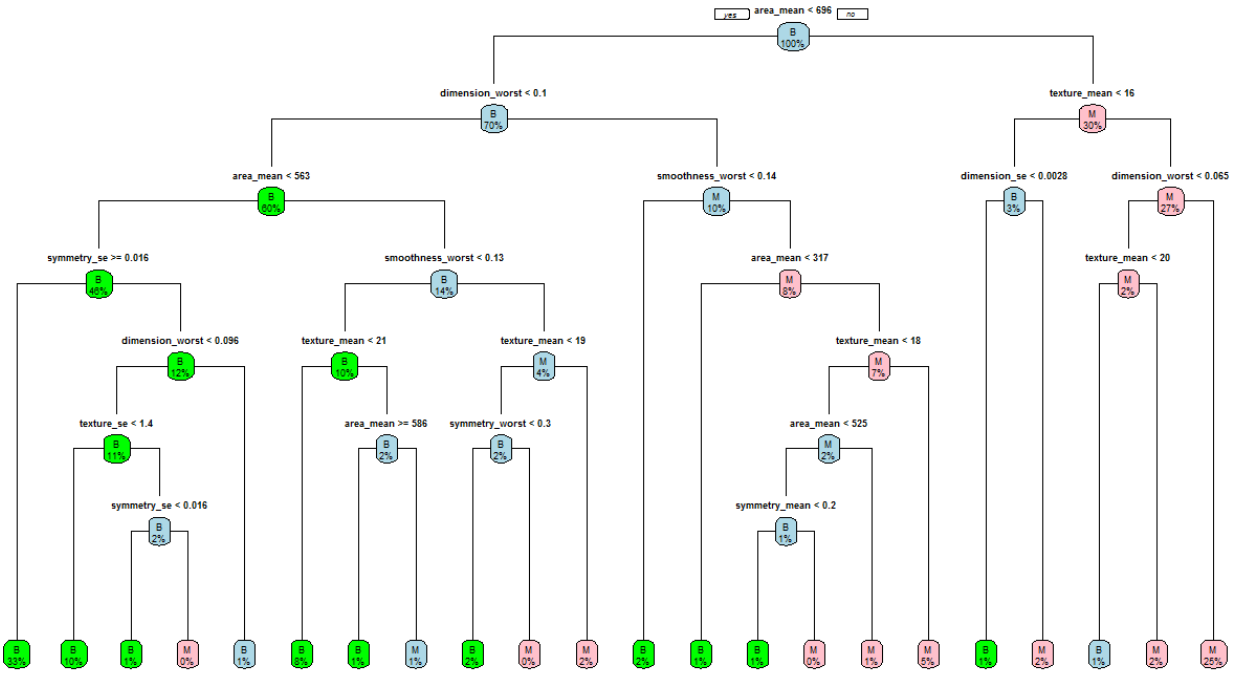
```
mean(knn.pred2==test_data_cor$diagnosis)
```

```
## [1] 0.8671329
```

```
#Accuracy = ((59+31)*100)/100 = 90%

#As we can see here, logistic regression and LDA has the highest accuracy, foll
owed by QDA. KNN seems to be the least accurate model.
```

Decision Tree

```
library(rpart)
#?rpart.plot
#View(rpart)
library(rpart.plot)

set.seed(42)
fit <- rpart(diagnosis ~ .,
             data = data_cor,
             method = "class",
             control = rpart.control(xval = 10,
                                     minbucket = 2,
                                     cp = 0),
              parms = list(split = "information"))

rpart.plot(fit, type = 1 , extra = 100, box.palette = c("green","light blue","p
ink"))
```


yes
area_mean < 696
no
B
100%
dimension_worst < 0.1
B
70%
texture_mean < 16
M
30%
area_mean < 563
B
90%
smoothness_worst < 0.14
M
10%
dimension_se < 0.0028
B
3%
dimension_worst < 0.065
M
27%
symmetry_se >= 0.016
B
48%
smoothness_worst < 0.13
B
14%
area_mean < 317
M
8%
texture_mean < 20
M
2%
dimension_worst < 0.096
B
12%
texture_mean < 21
B
10%
texture_mean < 19
M
4%
texture_mean < 18
M
7%
texture_se < 1.4
B
11%
area_mean >= 586
B
2%
symmetry_worst < 0.3
B
2%
area_mean < 525
M
2%
symmetry_se < 0.016
B
2%
symmetry_mean < 0.2
B
1%
B
43%
B
10%
B
1%
M
0%
B
1%
B
5%
B
1%
M
1%
B
2%
M
0%
M
2%
B
2%
B
1%
B
1%
M
0%
M
1%
M
5%
B
1%
M
2%
B
1%
M
2%
M
26%