CancerDataR

Minhaz Khan, Truman Zheng, Navin Chandradat, Vincent La, Bobak Ahmar

11/13/2018

Tasks:

Summary of data, Introduction - Truman

Organizing presentation - everyone

Presenting - Truman, Minhaz, Vincent

Anaylizing data/Coding - Truman, Minhaz, Vincent

Putting everything together/cConclusions/Report - Navin, Bobak

introduction

Breast cancer is a malignant cell growth in the breast. If it is left untreated the cancer can spread to other parts of the human body and it can be very deadly. There are generally two type of tumors non-cancerous and cancerous and the difference between the two is important, Benign tumor is non-cancerous and not dangerous on its own, but a malignant tumor, means the mass is cancerous.

Our goal for this project is to predict whether the cancer is benign or malignant and to determine what actually contribute to the classification of the two types.

We are given the following: Attribute Information: 1) ID number 2) Diagnosis (M = malignant, B = benign) 3-32)

a) radius (mean of distances from center to points on the perimeter)
b) texture (standard deviation of gray-scale values)
c) perimeter
d) area
e) smoothness (local variation in radius lengths) f)compactness (perimeter^2 / area - 1.0)
f) concavity (severity of concave portions of the contour)
g) concave points (number of concave portions of the contour)
h) symmetry
i) fractal dimension("coastline approximation" - 1)

The mean, standard error and "worst" or largest (mean of the three largest values) of these features were computed for each image, resulting in 30 features Class distribution: 357 benign, 212 malignant

summary of the data

We first take a small look at the data set and loading library/files we will need.

```r
# all the library and files we'll be using
library(tidyverse)
library(gridExtra)
library(ICSNP)
library(MASS)
library(klaR)
source("Box_M.R")
```

```
# preview of the data
cancer = read.csv("Project3-Data.csv")
head(cancer[1:5])
```

```
##         id diagnosis radius_mean texture_mean perimeter_mean
## 1   842302         M       17.99        10.38         122.80
## 2   842517         M       20.57        17.77         132.90
## 3 84300903         M       19.69        21.25         130.00
## 4 84348301         M       11.42        20.38          77.58
## 5 84358402         M       20.29        14.34         135.10
## 6   843786         M       12.45        15.70          82.57
```

```
# number of variables we have
num_var = ncol(cancer) - 1
num_var
```

```
## [1] 31
```

```
# number of observation we have
num_obs = nrow(cancer)
num_obs
```

```
## [1] 569
```

```
# the number of each type of tumor
table(cancer$diagnosis)
```

```
##
##   B   M
## 357 212
```

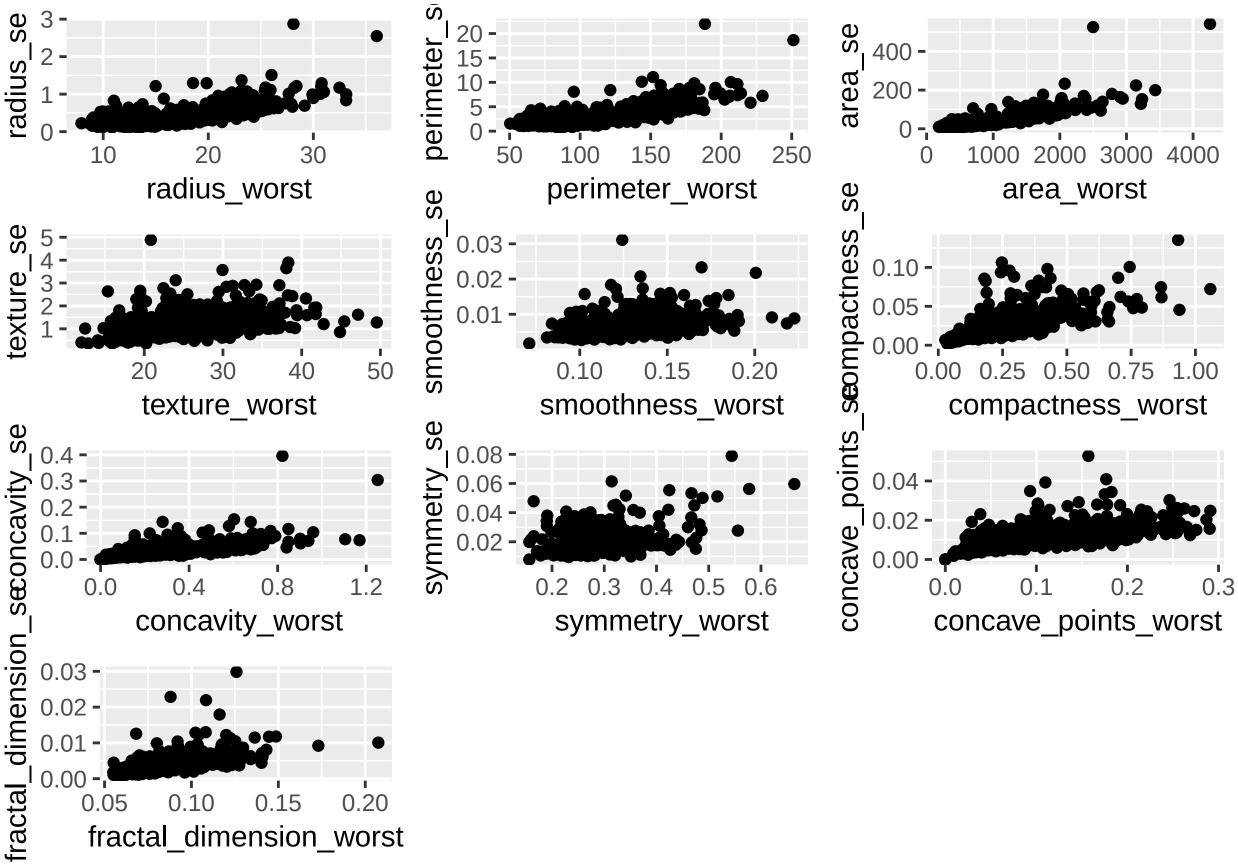
some visuals of the data

Standard errors vs worst cases

```
# filter out the first 2 columns
filcancer1=cancer[-1:-2]

x1<-ggplot(filcancer1, aes(x=radius_worst, y = radius_se))+geom_point()
x2<-ggplot(filcancer1, aes(x=perimeter_worst, y = perimeter_se))+geom_point()
x3<-ggplot(filcancer1, aes(x=area_worst, y = area_se))+geom_point()
#If we look at the first plot of texture's worst against standard errors, we can see
#non constant variance due to the cone shape of the data.
x4<-ggplot(filcancer1, aes(x=texture_worst, y = texture_se))+geom_point()
#Again we have a cone shape in the plot of smoothness worst versus standard error.
x5<-ggplot(filcancer1, aes(x=smoothness_worst, y = smoothness_se))+geom_point()
x6<-ggplot(filcancer1, aes(x=compactness_worst, y = compactness_se))+geom_point()
x7<-ggplot(filcancer1, aes(x=concavity_worst, y = concavity_se))+geom_point()
x8<-ggplot(filcancer1, aes(x=symmetry_worst, y = symmetry_se))+geom_point()
x9<-ggplot(filcancer1, aes(x=concave_points_worst, y = concave_points_se))+geom_point()
x10<-ggplot(filcancer1, aes(x=fractal_dimension_worst, y = fractal_dimension_se))+geom_point()
#Looking at the data, most points are near the the origin. However, the further stages
```

```
#in cancer seem to have higher standard errors. Also, most of the properties seem to
#follow a slightly curved distribution.
grid.arrange(x1,x2,x3,x4,x5,x6,x7,x8,x9,x10,ncol=3)
```

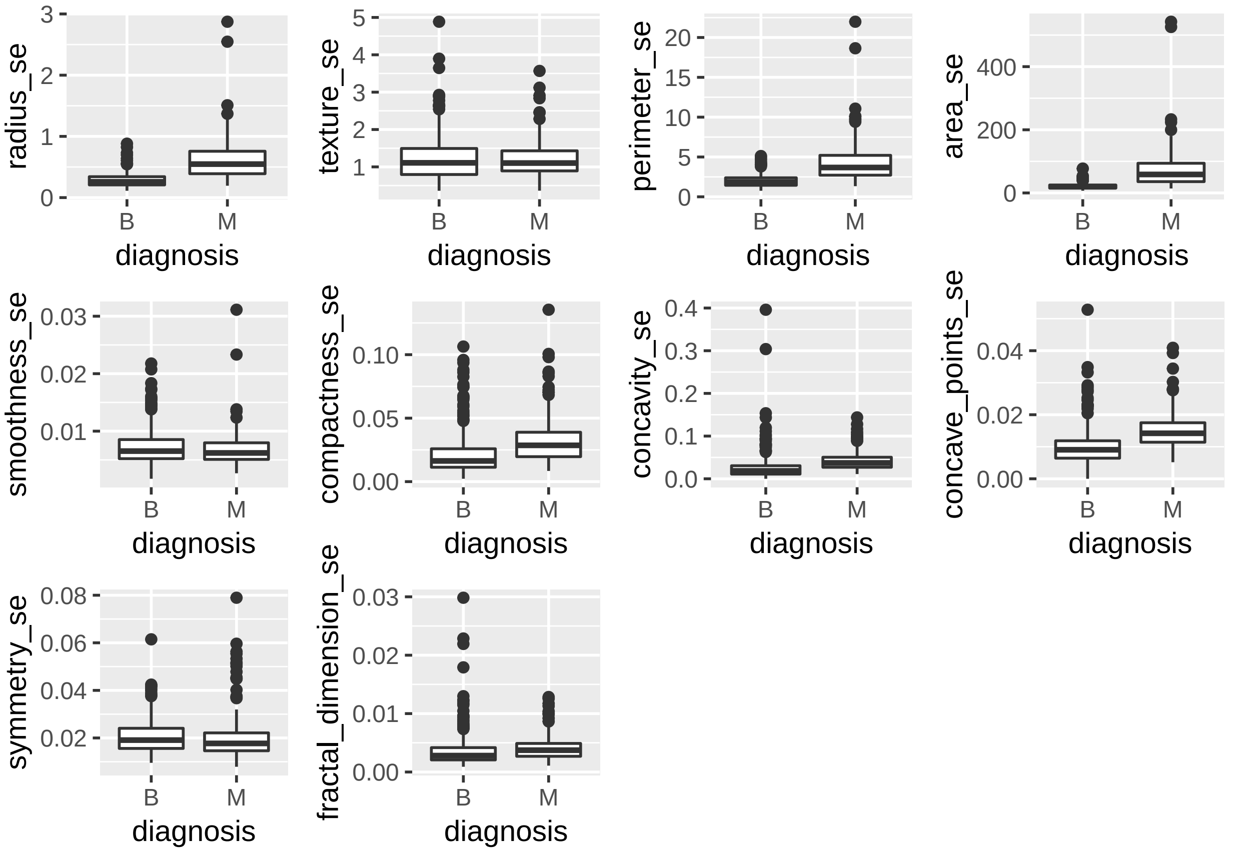


If we look at all the graphs for each category for measuring breast cancer, we can see that most data points cluster quite near to the origin, but not exactly on it. There are no data points at the origin since this would mean that the person has no reason to suspect that they have breast cancer. If we now look at the data points that are close to the center but not exactly, these points might represent either benign tumours or early stage malignant tumours, since these breasts don't really deviate that much from the average. If we look at the points further out, these points are likely middle stage malignant tumours, since they deviate a sizeable amount. Now, looking at the outliers, we have reason to suspect that these are late stage malignant tumours too, given the deformity of the breasts.

Boxplot of the SEs

```
new = cancer[-1]
new2 = new[,-c(2:11)]
new3 = new2[,-c(12:21)]
x1 = ggplot(new3, aes(x=diagnosis, y=radius_se))+geom_boxplot()
x2 = ggplot(new3, aes(x=diagnosis, y=texture_se))+geom_boxplot()
x3 = ggplot(new3, aes(x=diagnosis, y=perimeter_se))+geom_boxplot()
x4 = ggplot(new3, aes(x=diagnosis, y=area_se))+geom_boxplot()
x5 = ggplot(new3, aes(x=diagnosis, y=smoothness_se))+geom_boxplot()
x6 = ggplot(new3, aes(x=diagnosis, y=compactness_se))+geom_boxplot()
x7 = ggplot(new3, aes(x=diagnosis, y=concavity_se))+geom_boxplot()
```

```
x8 = ggplot(new3, aes(x=diagnosis, y=concave_points_se))+geom_boxplot()
x9 = ggplot(new3, aes(x=diagnosis, y=symmetry_se))+geom_boxplot()
x10 = ggplot(new3, aes(x=diagnosis, y=fractal_dimension_se))+geom_boxplot()
grid.arrange(x1,x2,x3,x4,x5,x6,x7,x8,x9,x10,ncol=4)
```



Each boxplot shows symmetry regarding the SE's, could potentially say its normally distributed since we have large sample size to back up the outliers.

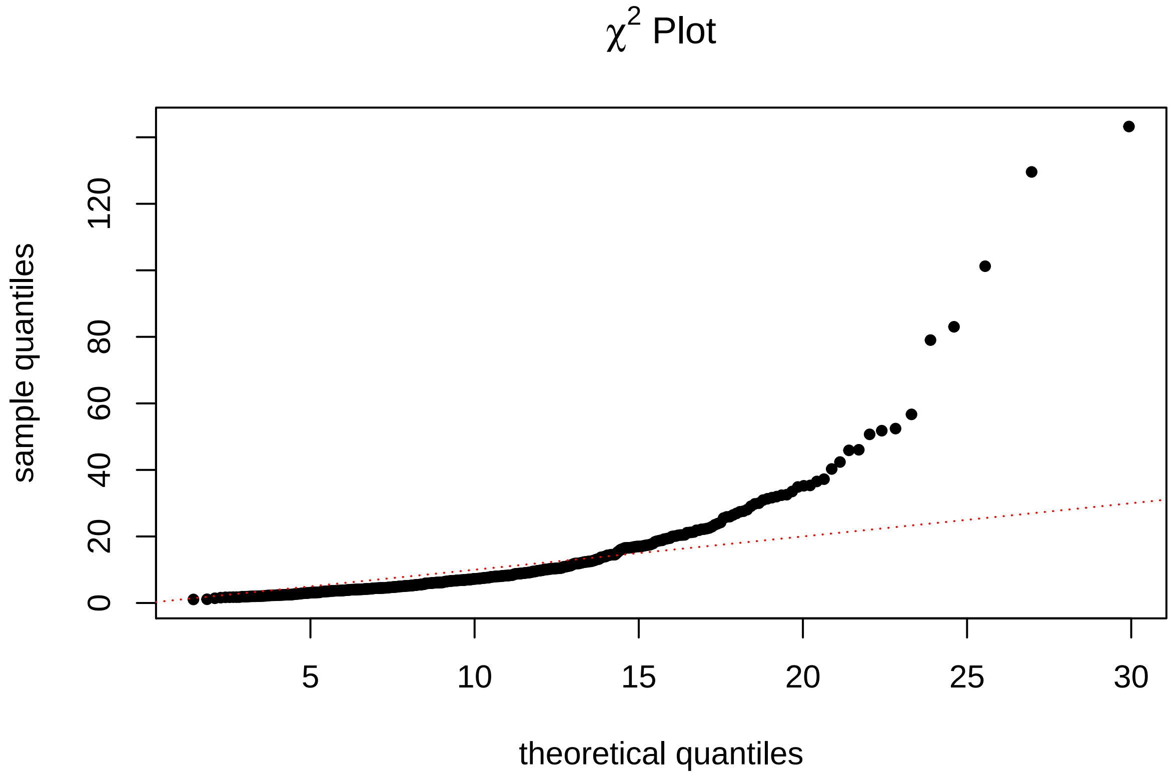
methods

Chi Square plot

before we start any analysis we want to verify the normallity of the data.

```
source("ChisqPlot.R")
can = cancer[,2:12]

# setting the independent variables into a matrix
can.matrix = as.matrix(can[,2:11])
chisqplot(can.matrix)
```



Because our data have a very large sample, I would say the normality assumption here is fine.

fitting a generalized linear model

we first starts of by fitting a generalized linear model to assess the significance of each of the variables. here we are only dealing with the "mean" variables as we believe that the other two category "standard error" and "worst" will not give us much information regarding the type of cancer.

```
# taking the mean values
can = cancer[2:12]

# changing diagnosis from chr to factor so is easy for model fitting
can$diagnosis = as.factor(can$diagnosis)

# fitting the generalized linear model
glm.fit = glm(diagnosis ~ ., data=can, family=binomial)
```

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

```
summary(glm.fit)
```

```
##
## Call:
## glm(formula = diagnosis ~ ., family = binomial, data = can)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.95590  -0.14839  -0.03943   0.00429   2.91690
##
## Coefficients:
```

```
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)              -7.35952   12.85259  -0.573   0.5669
## radius_mean              -2.04930    3.71588  -0.551   0.5813
## texture_mean              0.38473    0.06454   5.961  2.5e-09 ***
## perimeter_mean           -0.07151    0.50516  -0.142   0.8874
## area_mean                 0.03980    0.01674   2.377   0.0174 *
## smoothness_mean          76.43227   31.95492   2.392   0.0168 *
## compactness_mean         -1.46242   20.34249  -0.072   0.9427
## concavity_mean            8.46870    8.12003   1.043   0.2970
## concave_points_mean      66.82176   28.52910   2.342   0.0192 *
## symmetry_mean            16.27824   10.63059   1.531   0.1257
## fractal_dimension_mean -68.33703   85.55666  -0.799   0.4244
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 751.44  on 568  degrees of freedom
## Residual deviance: 146.13  on 558  degrees of freedom
## AIC: 168.13
##
## Number of Fisher Scoring iterations: 9
```

Theres various information in the summary but look at the coefficients, we have estimate, SE, z-score, and p-value, the p-value that is less than 0.05 indicates significance, that is those variable has an impact on either cancer being M or B. Example: for a unit increase in texture mean the odd of cancer being M (vs B) increases by exp(0.38473).

Now that we know which of the variables are actually significant, we will fit the model again with only those significant variables.

```
# new model after removing insignificant variables
glm.fit2 = glm(diagnosis ~ ., data=can[,c(1,3,5,6,9)], family=binomial)
```

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

```
summary(glm.fit2)
```

```
##
## Call:
## glm(formula = diagnosis ~ ., family = binomial, data = can[,
##     c(1, 3, 5, 6, 9)])
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.31798  -0.15623  -0.04212   0.01662   2.84201
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -23.677816   3.882774  -6.098 1.07e-09 ***
## texture_mean           0.362687   0.060544   5.990 2.09e-09 ***
## area_mean              0.010342   0.002002   5.165 2.40e-07 ***
## smoothness_mean       59.471304  25.965153   2.290    0.022 *
## concave_points_mean   76.571210  16.427864   4.661 3.15e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 751.44  on 568  degrees of freedom
## Residual deviance: 156.44  on 564  degrees of freedom
## AIC: 166.44
##
## Number of Fisher Scoring iterations: 8
```

Taking a look at the different AIC values of the two model, we see that the AIC values for the reduced model are smaller than the full model, this tells us that not only are some of the variable are insignificant but it also will effect the accuracy of our result.

With the following we can get a rough probability of type of cancer with given values.

```
# a function to get the probability of cancer being type M
prob = function(x1,x2,x3,x4){
  x = exp(-23.677816 + 0.362687*x1 + 0.010342*x2 + 59.471304*x3 + 76.571210*x4)
  pix = x/(1+x)
  return(pix)
}
```

Discriminant Analysis

Now we shall take a look at another method, here we use discriminant analysis, Discriminant analysis is a technique that is used to analyze the research data when the criterion or the dependent variable is categorical and the predictor or the independent variable is interval in nature (which is what we have here).

First we have to compute a two-sample Hotelling T-Squared test and compute Bartlett's test for homogeneous covariance matrices. with this we can determine whether or not to use Linear DA or quadratic DA as one requires equal covariance and the other one does not (LDA require equal covariance).

```
# again we are only working with the means
can = cancer[,2:12]

# setting the independent variables into a matrix
can.matrix = as.matrix(can[,2:11])

fit=manova(can.matrix ~ can$diagnosis)
summary(fit, test="Hotelling-Lawley")
```

```
##                 Df Hotelling-Lawley approx F num Df den Df    Pr(>F)
## can$diagnosis   1           2.1522   120.09     10    558 < 2.2e-16 ***
## Residuals     567
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# create separate data sets for Benign and Malignant tumors.
cancer1 <- can[can[,1]=="M",2:11]
cancer2 <- can[can[,1]=="B",2:11]

HotellingsT2(cancer1,cancer2)
```

```
##
##  Hotelling's two sample T2-test
##
## data:  cancer1 and cancer2
## T.2 = 120.09, df1 = 10, df2 = 558, p-value < 2.2e-16
## alternative hypothesis: true location difference is not equal to c(0,0,0,0,0,0,0,0,0,0)
```

```
n1 = dim(cancer1)[1]
n2 = dim(cancer2)[1]

Box_M(can.matrix, n=c(n1, n2))
```

```
## Test result:
##               [,1]
## Box.M-C 221.7243
## p.value   0.0000
```

Here we see that we do not have equal covariance and so we'll be using QDA instead of LDA for better performance/accuracy.

discriminant analysis with all 10 variables

We first start with all 10 variables just so we can have a comparison later with the reduced model.

```
# spliting the data into 2 set, training and testing
training_sample <- sample(c(TRUE, FALSE), nrow(can), replace = T, prob = c(0.6,0.4))
cantrain <- can[training_sample, ]
cantest <- can[!training_sample, ]

# the model
cancer.qda <- qda(diagnosis ~ ., data=cantrain)
cancer.qda
```

```
## Call:
## qda(diagnosis ~ ., data = cantrain)
##
## Prior probabilities of groups:
##         B         M
## 0.6140845 0.3859155
##
## Group means:
##   radius_mean texture_mean perimeter_mean area_mean smoothness_mean
## B    12.07255     17.91294       77.58917  457.2165      0.09231959
## M    17.50504     21.57511      115.53511  982.8642      0.10258453
##   compactness_mean concavity_mean concave_points_mean symmetry_mean
## B       0.07955138     0.04398185           0.0257272     0.1740610
## M       0.14276124     0.15907270           0.0876708     0.1944701
##   fractal_dimension_mean
## B             0.06279739
## M             0.06221197
```

```
#Confusion test
set.seed(1)
confusionTest <- table(cantest$diagnosis, predict(cancer.qda, newdata=cantest)$class)
confusionTest
```

```
##
##       B   M
##   B 134   5
##   M   8  67
```

```
n <- sum(confusionTest)
aer <- (n - sum(diag(confusionTest))) / n
aer
```

```
## [1] 0.06074766
```

Discriminant analysis with the significant variables

We now do the same thing but with the reduced model.

```
# splitting data into 2sets, training and testing
can2 = can[,c(1,3,5,6,9)]
training_sample2 <- sample(c(TRUE, FALSE), nrow(can2), replace = T, prob = c(0.6,0.4))
cantrain2 <- can2[training_sample2, ]
cantest2 <- can2[!training_sample2, ]

# the model
cancer.qda2 <- qda(diagnosis ~ ., data=cantrain2, CV=FALSE)
cancer.qda2
```

```
## Call:
## qda(diagnosis ~ ., data = cantrain2, CV = FALSE)
##
## Prior probabilities of groups:
##         B         M
## 0.6340058 0.3659942
##
## Group means:
##   texture_mean area_mean smoothness_mean concave_points_mean
## B     17.98995  457.4986      0.09226236          0.02516845
## M     21.73795  983.6764      0.10292252          0.08891394
```

```
# testing the accuracy of our model
set.seed(1)
qda.test <- predict(cancer.qda2,cantest2)
cantest2$qda <- qda.test$class
confusionTest <-table(cantest2$qda,cantest2$diagnosis)
confusionTest
```

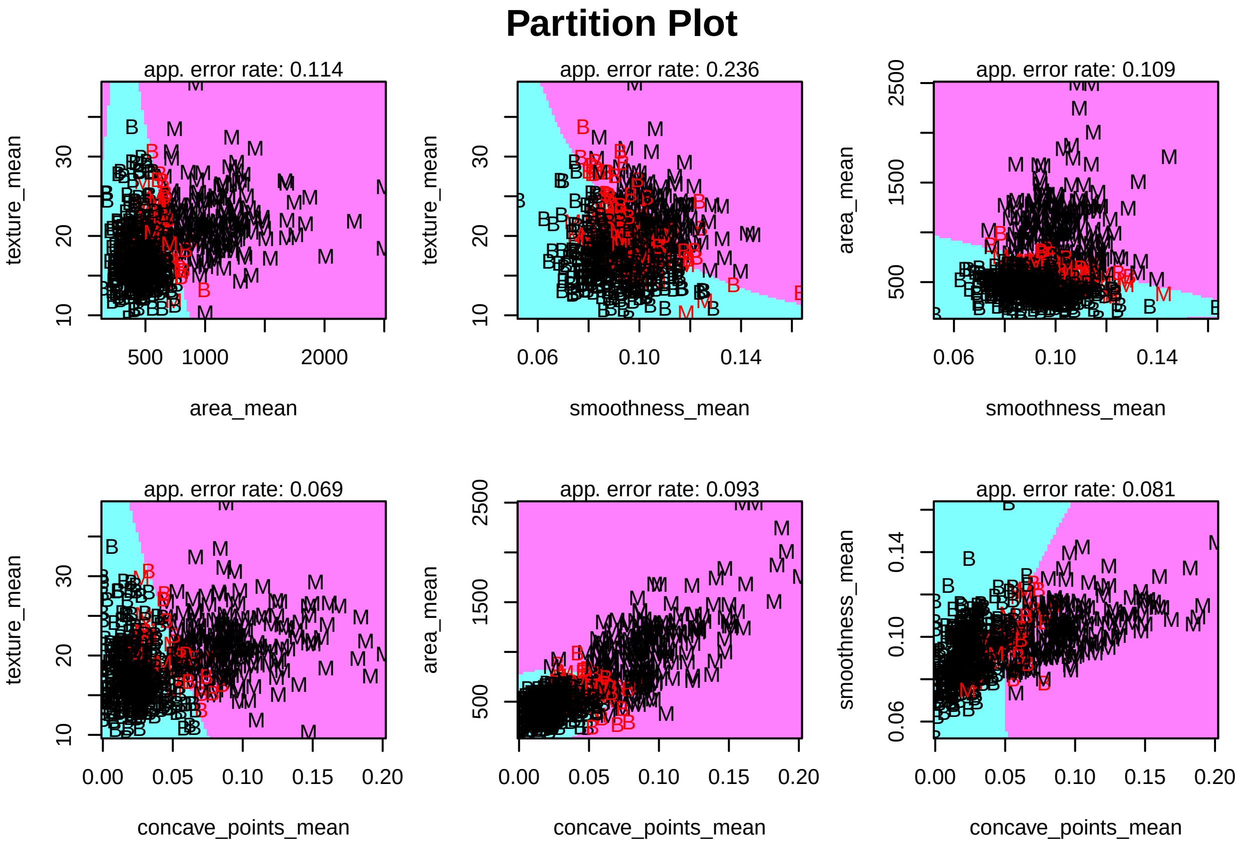
```
##
##       B   M
##   B 130  10
##   M   7  75
```

```
n <- sum(confusionTest)
aer <- (n - sum(diag(confusionTest))) / n
aer
```

```
## [1] 0.07657658
```

Taking a look at the errors of the two model, full vs reduced we see that the difference between the 2 is negligible (the AER for the two are very close). This also however tells us that with only 4 variables, texture mean, area mean, smoothness mean and concave points mean we can accurately predict 90% of the class of observation which is very good.

```
# here are just some more visuals of the data
partimat(diagnosis ~ ., data=can2, method="qda")
```



From this we can also confirm that the accuracy of our model is indeed good.

conclusion

From the two analysis we done, from discriminant analysis to simply fitting a generalized linear model we can clearly conclude that the dependent variable cancer type or diagnosis hugely depend on simply four variables, that is, it's mainly depend on texture mean, area mean, smoothness mean and concave points mean and from these four variables we can determine the odds patient's cancer type and so from that can determine whether treatment are neccessary.

reference

Fayed, L., & Paul, D. (n.d.). Differences Between a Malignant and Benign Tumor. Retrieved from https://www.verywellhealth.com/what-does-malignant-and-benign-mean-514240

Sign In. (n.d.). Retrieved from https://rpubs.com/Nolan/298913