

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

Analyzing data/coding - Truman, Minhaz, Vincent

Putting everything together/conclusions/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

-The mean, standard error and "worst" or largest (mean of the three largest values) of these features w

-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

```
# all the library and files we'll be using
library(tidyverse)
```

```
## -- Attaching packages ----- tidyverse 1.2.1 --
## v ggplot2 3.1.0      v purrr   0.2.5
## v tibble  1.4.2      v dplyr  0.7.8
## v tidyr   0.8.2      v stringr 1.3.1
## v readr   1.1.1      v forcats 0.3.0

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##      combine
library(ICSNP)

## Loading required package: mvtnorm
## Loading required package: ICS
library(MASS)

##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##      select
library(klaR)
source("Box_M.R")

# preview of the data
cancer = read.csv("Project3-Data.csv")
head(cancer)

##      id diagnosis radius_mean texture_mean perimeter_mean area_mean
## 1  842302         M      17.99      10.38         122.80     1001.0
## 2  842517         M      20.57      17.77         132.90     1326.0
## 3 84300903         M      19.69      21.25         130.00     1203.0
## 4 84348301         M      11.42      20.38          77.58       386.1
## 5 84358402         M      20.29      14.34         135.10     1297.0
## 6  843786         M      12.45      15.70          82.57       477.1
##      smoothness_mean compactness_mean concavity_mean concave_points_mean
## 1          0.11840          0.27760          0.3001          0.14710
## 2          0.08474          0.07864          0.0869          0.07017
## 3          0.10960          0.15990          0.1974          0.12790
## 4          0.14250          0.28390          0.2414          0.10520
## 5          0.10030          0.13280          0.1980          0.10430
## 6          0.12780          0.17000          0.1578          0.08089
##      symmetry_mean fractal_dimension_mean radius_se texture_se perimeter_se
## 1          0.2419          0.07871      1.0950      0.9053          8.589
## 2          0.1812          0.05667      0.5435      0.7339          3.398
## 3          0.2069          0.05999      0.7456      0.7869          4.585
## 4          0.2597          0.09744      0.4956      1.1560          3.445
## 5          0.1809          0.05883      0.7572      0.7813          5.438
## 6          0.2087          0.07613      0.3345      0.8902          2.217
##      area_se smoothness_se compactness_se concavity_se concave_points_se
## 1    153.40      0.006399      0.04904      0.05373          0.01587
## 2     74.08      0.005225      0.01308      0.01860          0.01340
## 3     94.03      0.006150      0.04006      0.03832          0.02058
## 4     27.23      0.009110      0.07458      0.05661          0.01867
## 5     94.44      0.011490      0.02461      0.05688          0.01885
## 6     27.19      0.007510      0.03345      0.03672          0.01137
##      symmetry_se fractal_dimension_se radius_worst texture_worst
```

```
## 1      0.03003      0.006193      25.38      17.33
## 2      0.01389      0.003532      24.99      23.41
## 3      0.02250      0.004571      23.57      25.53
## 4      0.05963      0.009208      14.91      26.50
## 5      0.01756      0.005115      22.54      16.67
## 6      0.02165      0.005082      15.47      23.75
##      perimeter_worst area_worst smoothness_worst compactness_worst
## 1      184.60      2019.0      0.1622      0.6656
## 2      158.80      1956.0      0.1238      0.1866
## 3      152.50      1709.0      0.1444      0.4245
## 4       98.87      567.7      0.2098      0.8663
## 5      152.20      1575.0      0.1374      0.2050
## 6      103.40      741.6      0.1791      0.5249
##      concavity_worst concave_points_worst symmetry_worst
## 1      0.7119      0.2654      0.4601
## 2      0.2416      0.1860      0.2750
## 3      0.4504      0.2430      0.3613
## 4      0.6869      0.2575      0.6638
## 5      0.4000      0.1625      0.2364
## 6      0.5355      0.1741      0.3985
##      fractal_dimension_worst
## 1      0.11890
## 2      0.08902
## 3      0.08758
## 4      0.17300
## 5      0.07678
## 6      0.12440
```

```
# 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

```
#Obtaining the standard errors and worst cases.
```

```
filcancer1=cancer[-1:-2]
```

```
standard_errors = filcancer1[,11:20]
```

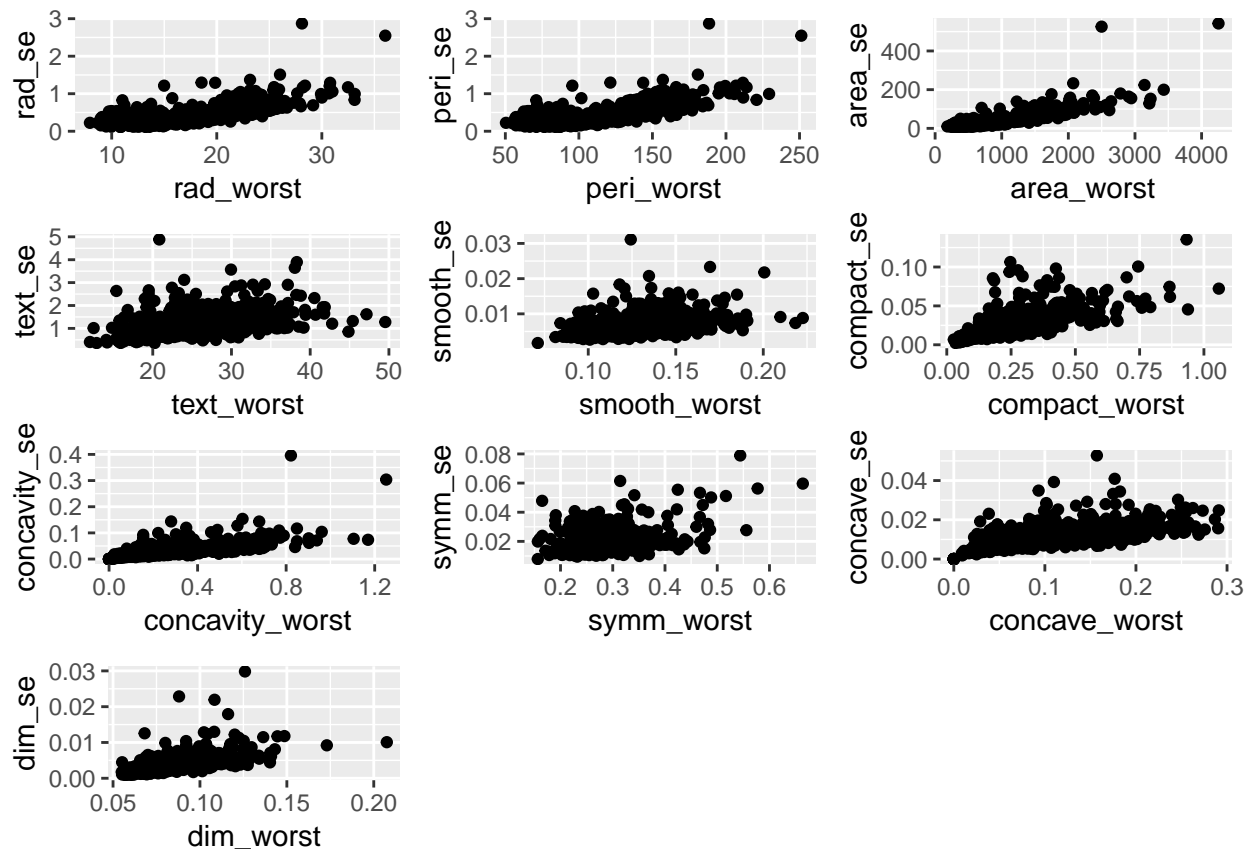
```
worst_cases = filcancer1[,21:30]
```

```

# Setting up the data frame for each of the possible properties of the boob.
worst_data = data.frame(worst_cases)
se_data = data.frame(standard_errors)
df_radius = data.frame(cbind(rad_worst=worst_data$radius_worst,rad_se=se_data$radius_se))
df_perimeter = data.frame(cbind(peri_worst=worst_data$perimeter_worst,peri_se=se_data$radius_se))
df_area = data.frame(cbind(area_worst=worst_data$area_worst,area_se=se_data$area_se))
df_texture = data.frame(cbind(text_worst=worst_data$texture_worst,text_se=se_data$texture_se))
df_smoothness = data.frame(cbind(smooth_worst=worst_data$smoothness_worst,smooth_se=se_data$smoothness_se))
df_compact = data.frame(cbind(compact_worst=worst_data$compactness_worst,compact_se=se_data$compactness_se))
df_concavity = data.frame(cbind(concavity_worst=worst_data$concavity_worst,concavity_se=se_data$concavity_se))
df_concave_points = data.frame(cbind(concave_worst=worst_data$concave_points_worst,concave_se=se_data$concave_points_se))
df_symmetry = data.frame(cbind(symm_worst=worst_data$symmetry_worst,symm_se=se_data$symmetry_se))
df_dimension = data.frame(cbind(dim_worst=worst_data$fractal_dimension_worst,dim_se = se_data$fractal_dimension_se))

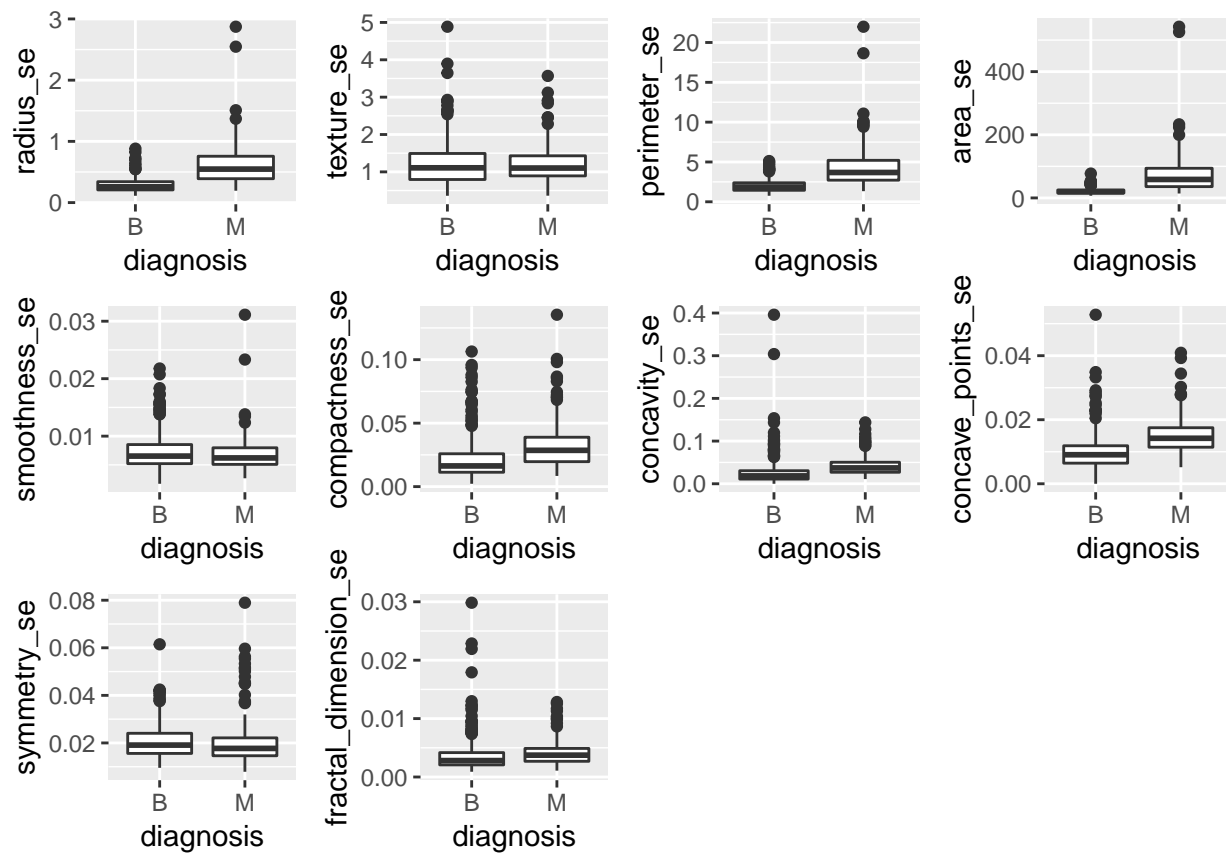
x1<-ggplot(df_radius, aes(x=rad_worst, y = rad_se))+geom_point()
x2<-ggplot(df_perimeter, aes(x=peri_worst, y = peri_se))+geom_point()
x3<-ggplot(df_area, aes(x=area_worst, y = area_se))+geom_point()
#If we look at the first plot of texture's worst against standard errors, we can see non constant variance
x4<-ggplot(df_texture, aes(x=text_worst, y = text_se))+geom_point()
#Again we have a cone shape in the plot of smoothness worst versus standard error.
x5<-ggplot(df_smoothness, aes(x=smooth_worst, y = smooth_se))+geom_point()
x6<-ggplot(df_compact, aes(x=compact_worst, y = compact_se))+geom_point()
x7<-ggplot(df_concavity, aes(x=concavity_worst, y = concavity_se))+geom_point()
x8<-ggplot(df_symmetry, aes(x=symm_worst, y = symm_se))+geom_point()
x9<-ggplot(df_concave_points, aes(x=concave_worst, y = concave_se))+geom_point()
x10<-ggplot(df_dimension, aes(x=dim_worst, y = dim_se))+geom_point()
#Looking at the data, most points are near the the origin. However, the further stages in cancer seem to
grid.arrange(x1,x2,x3,x4,x5,x6,x7,x8,x9,x10,ncol=3)

```



boxplot of the SEs

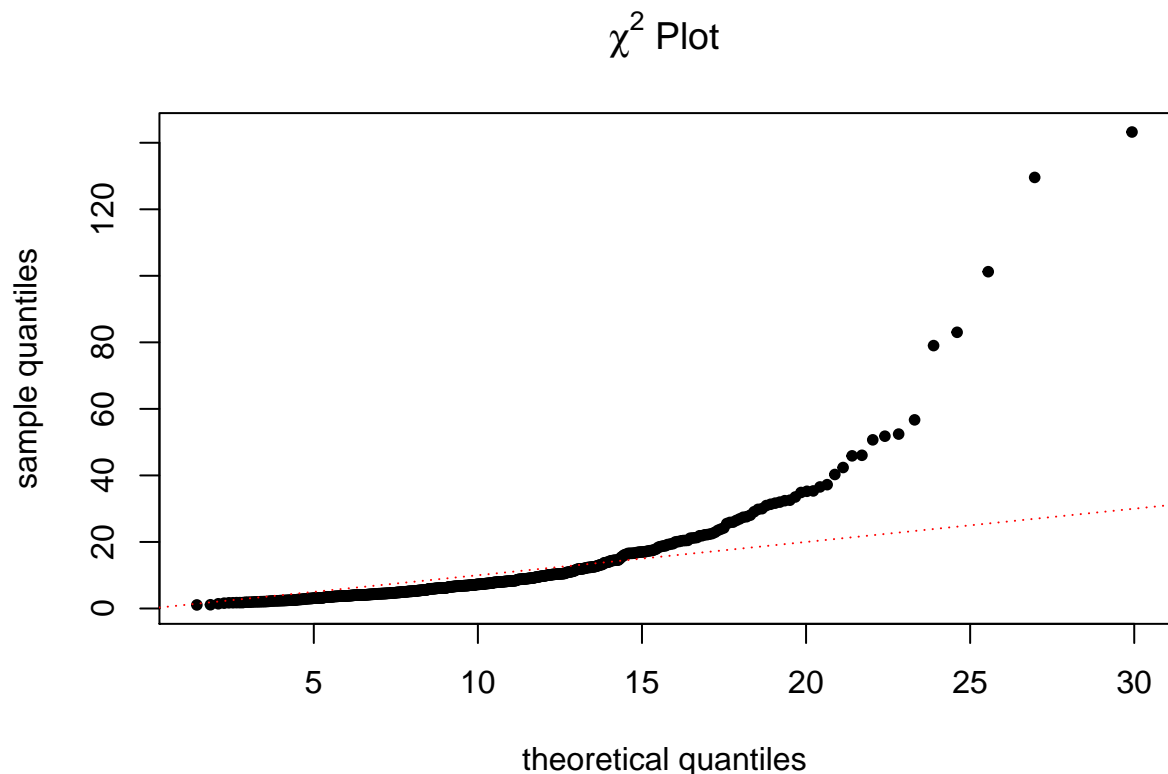
```
new = cancer[-1]
new2 = new[, -c(2:11)]
new3 = new2[, -c(12:21)]
x1 = ggplot(new3, aes(x=diagnosis, y=radius_se))+geom_boxplot()
x2 = ggplot(new3, aes(x=diagnosis, y=texture_se))+geom_boxplot()
x3 = ggplot(new3, aes(x=diagnosis, y=perimeter_se))+geom_boxplot()
x4 = ggplot(new3, aes(x=diagnosis, y=area_se))+geom_boxplot()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



methods # Chi Square plot before we start any analysis we want to verify the normality 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)
```



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

fitting a generalized linear model

we first start 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
```

there's 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 variables have 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 ~ texture_mean + area_mean + smoothness_mean + concave_points_mean, data=can, family=binomial)
```

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

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

```
##
## Call:
## glm(formula = diagnosis ~ texture_mean + area_mean + smoothness_mean +
##      concave_points_mean, family = binomial, data = can)
##
## 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 actually better 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 LA 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

```
# splitting 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 ~ radius_mean+texture_mean+perimeter_mean+area_mean+smoothness_mean+compactness_mean,
cancer.qda
```

```
## Call:
## qda(diagnosis ~ radius_mean + texture_mean + perimeter_mean +
##      area_mean + smoothness_mean + compactness_mean + concavity_mean +
##      concave_points_mean + symmetry_mean + fractal_dimension_mean,
##      data = cantrain)
##
## Prior probabilities of groups:
##           B           M
## 0.6588235 0.3411765
##
## Group means:
##      radius_mean texture_mean perimeter_mean area_mean smoothness_mean
## B      12.10727      18.22902       77.83478  460.1232      0.09285304
## M      17.45448      21.43250      115.14819  977.8397      0.10225155
##      compactness_mean concavity_mean concave_points_mean symmetry_mean
## B           0.08111161      0.04497214      0.02577725      0.1760844
## M           0.13997836      0.15358543      0.08574810      0.1905328
##      fractal_dimension_mean
## B              0.06298929
## M              0.06246810
```

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

```
##
```

```
##           B      M
##      B 131      2
##      M   14    82

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

## [1] 0.069869
```

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
## 0.6129944 0.3870056
##
## Group means:
## texture_mean area_mean smoothness_mean concave_points_mean
## B      17.96760 465.1654      0.0918877      0.02540562
## M      21.38255 942.9737      0.1024308      0.08477416

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

```
##
##           B      M
##      B 135      6
##      M   5     69

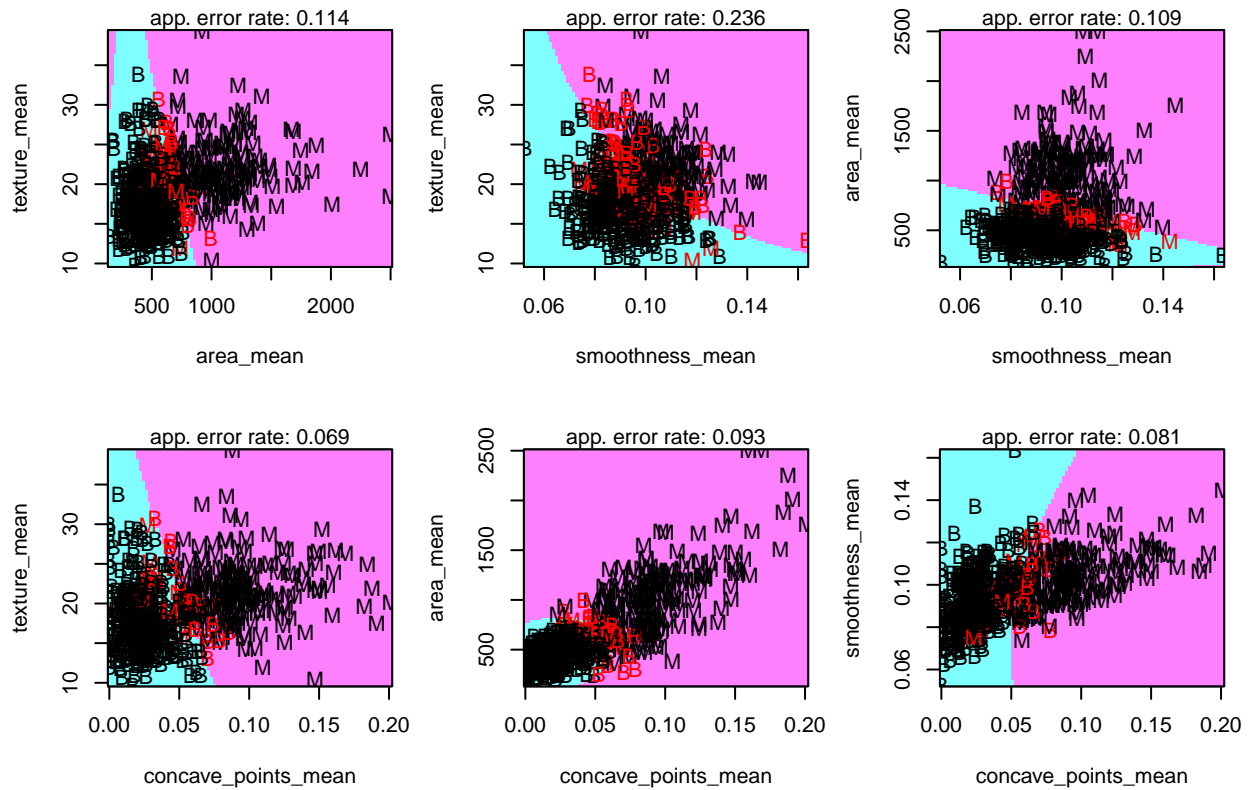
n <- sum(confusionTest)
aer <- (n - sum(diag(confusionTest))) / n
aer

## [1] 0.05116279
```

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")
```

Partition Plot



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 necessary.