Breast Cancer Wisconsin (Diagnostic)

L. Favero

Contents

0.1 Setup

Load libraries

```
# library(GGally)       # for ggpairs
# library(ggfortify)    # for autoplot
# library(ggplot2)      # for ggplot
# library('MASS')       # for the glm model selection

#this is to check to see if package are installed and if not to install them or just load them
if(!require(pacman)) {install.packages(c("pacman", "remotes"))}
```

```
## Loading required package: pacman
```

```
if(!require(papaja)) {remotes::install_github("crsh/papaja")}
```

```
## Loading required package: papaja
```

```
pacman::p_load(GGally, ggfortify, ggplot2, MASS, here, kableExtra, papaja, glmnet)
```

1 Information about the dataset

Link to dataset

"Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image."

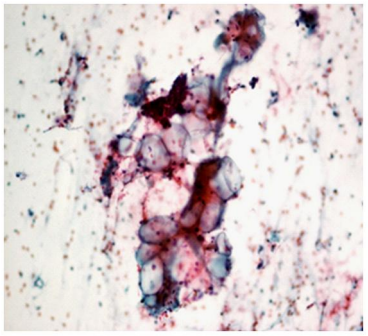


Figure 1: image of a fine needle aspirate (FNA) of a breast mass

Ten real-valued features are computed for each cell nucleus:

Name of the variables	type	Description
1) 'radius'	num	distances from center to points on the perimeter
2) 'texture'	num	standard deviation of gray-scale values
3) 'perimeter'	num	perimeter of the nucleus
4) 'area'	num	area of the nucleus
5) 'smoothness'	num	local variation in radius lengths
6) 'compactness'	num	$perimeter^2/area - 1.0$
7) 'concavity'	num	severity of concave portions of the contour
8) 'concave.points'	num	number of concave portions of the contour
9) 'symmetry'	num	symmetry of the nucleus
10)'fractal_dimension'	num	$coastlineapproximation - 1$

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. All feature values are recorded with four significant digits.

The aim is to **predict whether the cancer is benign or malignant**

diagnosis	radius	texture	perimeter	area	smoothness	compactness	concavity	concave.points	symmetry	fractal_dimension
M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	0.2419	0.07871
M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	0.1812	0.05667
M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	0.2069	0.05999
M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	0.2597	0.09744
M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	0.1809	0.05883
M	12.45	15.70	82.57	477.1	0.12780	0.17000	0.1578	0.08089	0.2087	0.07613

2 Preprocessing

Load data

```
#get relative path
path = here("LUCILE")
setwd(path) #set working directory

# df<-read.csv('/Users/lucile/Library/Mobile Documents/com~apple~CloudDocs/STUDY/NEURO/LECTURE

df<-read.csv('data.csv', stringsAsFactors = 1) # j'ai  ajouté le dataset sur github
```

2.1 First look into the data

There are a lot of variables, we should pick the most relevant ones.

Let's delete the last variable and ID number because there are not relevant.

```
df<-df[,-33]
df<-df[,-1]
```

Let's create a new frame for a the variables of type mean.

```
'data.frame':   569 obs. of  11 variables:
 $ diagnosis        : Factor w/ 2 levels "B","M": 2 2 2 2 2 2 2 2 2 2 ...
 $ radius           : num  18 20.6 19.7 11.4 20.3 ...
 $ texture          : num  10.4 17.8 21.2 20.4 14.3 ...
 $ perimeter        : num  122.8 132.9 130 77.6 135.1 ...
 $ area             : num  1001 1326 1203 386 1297 ...
 $ smoothness       : num  0.1184 0.0847 0.1096 0.1425 0.1003 ...
 $ compactness      : num  0.2776 0.0786 0.1599 0.2839 0.1328 ...
 $ concavity        : num  0.3001 0.0869 0.1974 0.2414 0.198 ...
 $ concave.points   : num  0.1471 0.0702 0.1279 0.1052 0.1043 ...
 $ symmetry         : num  0.242 0.181 0.207 0.26 0.181 ...
 $ fractal_dimension: num  0.0787 0.0567 0.06 0.0974 0.0588 ...
```

```
 diagnosis     radius          texture         perimeter          area
 B:357     Min.   : 6.981   Min.   : 9.71   Min.   : 43.79   Min.   : 143.5
```

```
 M:212     1st Qu.:11.700   1st Qu.:16.17   1st Qu.: 75.17   1st Qu.: 420.3
           Median :13.370   Median :18.84   Median : 86.24   Median : 551.1
           Mean   :14.127   Mean   :19.29   Mean   : 91.97   Mean   : 654.9
           3rd Qu.:15.780   3rd Qu.:21.80   3rd Qu.:104.10   3rd Qu.: 782.7
           Max.   :28.110   Max.   :39.28   Max.   :188.50   Max.   :2501.0
  smoothness       compactness        concavity       concave.points
 Min.   :0.05263   Min.   :0.01938   Min.   :0.00000   Min.   :0.00000
 1st Qu.:0.08637   1st Qu.:0.06492   1st Qu.:0.02956   1st Qu.:0.02031
 Median :0.09587   Median :0.09263   Median :0.06154   Median :0.03350
 Mean   :0.09636   Mean   :0.10434   Mean   :0.08880   Mean   :0.04892
 3rd Qu.:0.10530   3rd Qu.:0.13040   3rd Qu.:0.13070   3rd Qu.:0.07400
 Max.   :0.16340   Max.   :0.34540   Max.   :0.42680   Max.   :0.20120
   symmetry      fractal_dimension
 Min.   :0.1060   Min.   :0.04996
 1st Qu.:0.1619   1st Qu.:0.05770
 Median :0.1792   Median :0.06154
 Mean   :0.1812   Mean   :0.06280
 3rd Qu.:0.1957   3rd Qu.:0.06612
 Max.   :0.3040   Max.   :0.09744
```

Proportion of benign vs malignant cancer

```
prop.table(table(df_mean$diagnosis))
```

```
        B         M
0.6274165 0.3725835
```

The two types of cancer are not represented in the same proportion, this can lead to a bias. Is this proportion representative of the reality ?

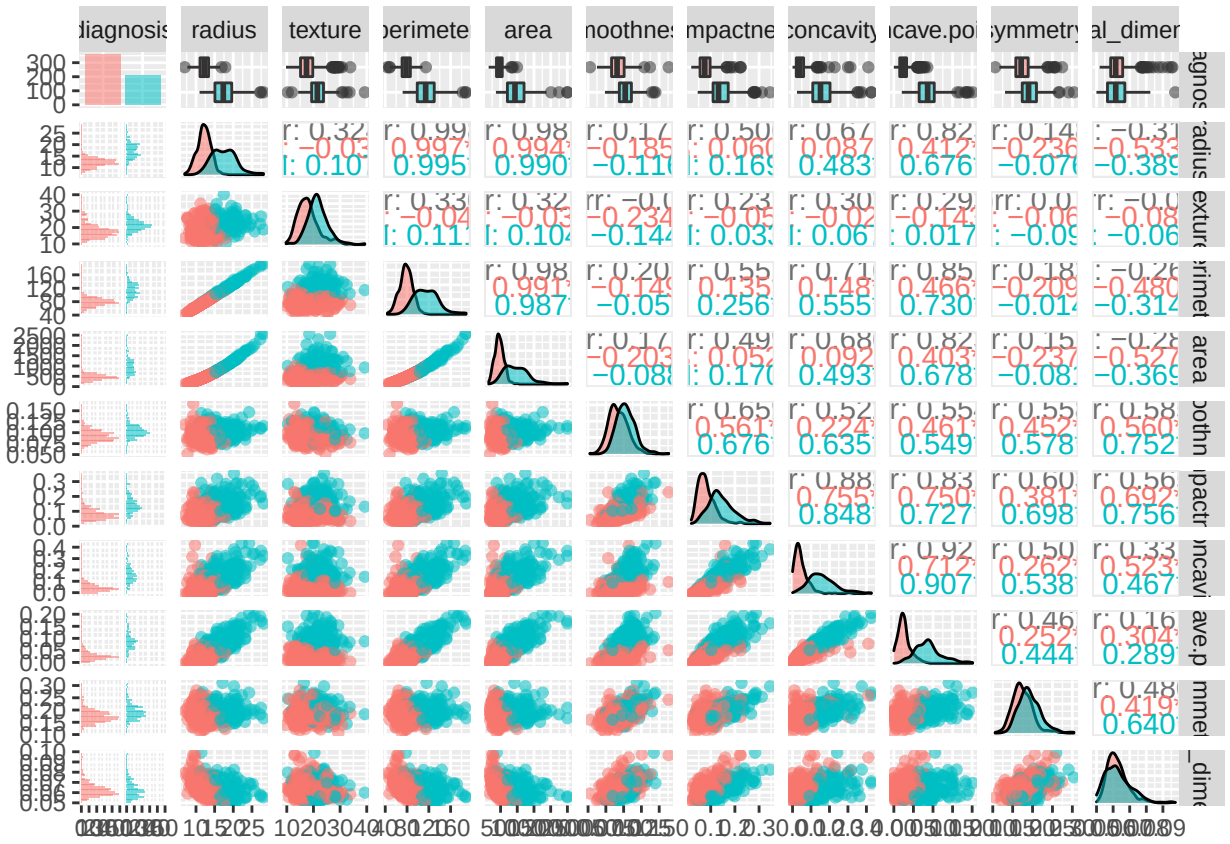
"The benign to malignant ratio (B:M ratio) among breast biopsies (number of benign breast lesions divided by number of breast cancers) is widely believed to be around 4:1 or 5:1" [2]

Blabla citation in parentesis (James 1890) blbabla, or citation in text James (1890)

2.2 Selection of variables of interest

According to the description of the data, some variables are likely to be correlated. We will address this correlation with the mean group of variable.

Hypothesis about the correlation between variables : *radius and smoothness should be correlated* radius, perimeter, area and compactness should be perfectly correlated since it exists a formula between these variables * concavity and symmetry should be correlated * texture and fractal_dimension should not have any correlation




```
## [1] "B" "M"
```

By eye, the variables seem to be different according to the type of 'diagnosis' (first row of the plot), the variables coming from malignant cancer seem to be in general bigger than the data coming from bening cancer.

As expected, radius, perimeter and area are highly correlated;
and texture and fractal_dimension don't have strong correlation.

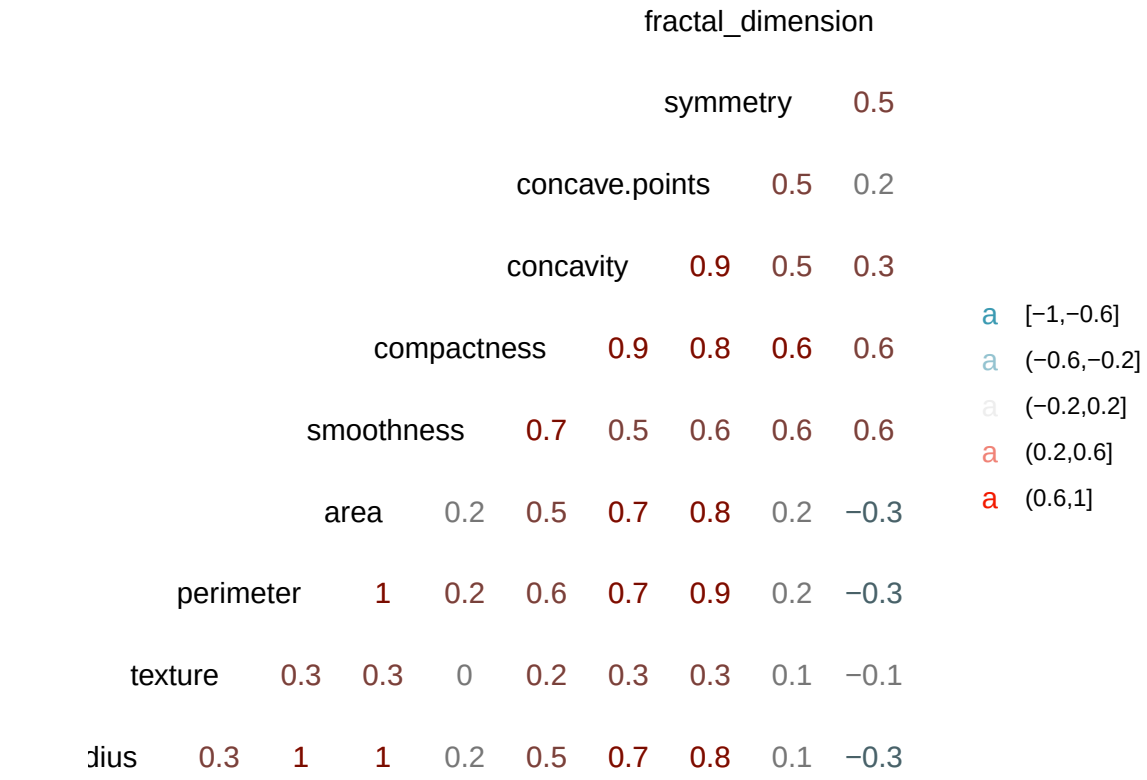
Surprisingly, radius and smoothness are not very correlated, and the compactness doesn't show any strong correlation.

Concavity and compactness have a strong correlation. In annexe we show that we have the same correlations with the standard deviation group and extreme group of variable.

The following function permit to see better the correlation :

```
ggcorr(df_mean, geom = "text", nbreaks = 5, hjust = 1, label = TRUE, label_alpha = 0.5)
```

```
## Warning in ggcorr(df_mean, geom = "text", nbreaks = 5, hjust = 1, label =
## TRUE, : data in column(s) 'diagnosis' are not numeric and were ignored
```



Even if compactness is define as $perimeter^2/area - 1.0$ the correlation between this variable and area or perimeter is not 1 because the correlation show only the linear dependency. The correlation between radius perimeter and area is 1. We will only keep radius.

From here let's remove area, perimeter and compactness

```
m_perimeter <-lm(data = df_mean,perimeter~ radius+texture+area+smoothness+compactness+concavity
summary(m_perimeter)
```

```
##
## Call:
## lm(formula = perimeter ~ radius + texture + area + smoothness +
##     compactness + concavity + concave.points + symmetry + fractal_dimension,
##     data = df_mean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.7781 -0.1859  0.0416  0.2280  3.8482
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        2.627e+00  8.317e-01   3.159  0.00167 **
## radius             6.126e+00  5.223e-02 117.301  < 2e-16 ***
## texture           -2.105e-03  5.886e-03  -0.358  0.72073
## area               3.868e-03  4.676e-04   8.272 9.65e-16 ***
## smoothness        -8.998e+00  2.816e+00  -3.196  0.00147 **
## compactness        3.407e+01  1.517e+00  22.459  < 2e-16 ***
```

```
## concavity          3.865e+00  9.842e-01   3.927 9.67e-05 ***
## concave.points     4.283e+00  2.785e+00   1.538  0.12456
## symmetry          -1.930e+00  1.127e+00  -1.712  0.08751 .
## fractal_dimension -4.119e+01  8.190e+00  -5.029 6.64e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5538 on 559 degrees of freedom
## Multiple R-squared:  0.9995, Adjusted R-squared:  0.9995
## F-statistic: 1.214e+05 on 9 and 559 DF,  p-value: < 2.2e-16
```

```
m_area <-lm(data = df_mean,area~ radius+texture+ perimeter+smoothness+compactness+concavity+co
summary(m_area)
```

```
##
## Call:
## lm(formula = area ~ radius + texture + perimeter + smoothness +
##     compactness + concavity + concave.points + symmetry + fractal_dimension,
##     data = df_mean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -119.45  -25.53   -8.03   18.13  395.17
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -1033.1844    56.7713 -18.199  < 2e-16 ***
## radius              -81.3418    22.3034  -3.647  0.00029 ***
## texture               0.4089     0.5023   0.814  0.41588
## perimeter            28.1972     3.4086   8.272 9.65e-16 ***
## smoothness           92.0982   242.5499   0.380  0.70431
## compactness       -2169.2200   153.3091 -14.149  < 2e-16 ***
## concavity           221.0651    84.6672   2.611  0.00927 **
## concave.points      295.3508   237.9177   1.241  0.21498
## symmetry             92.1348    96.4335   0.955  0.33978
## fractal_dimension  6413.4180   661.4625   9.696  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 47.28 on 559 degrees of freedom
## Multiple R-squared:  0.9822, Adjusted R-squared:  0.9819
## F-statistic:  3434 on 9 and 559 DF,  p-value: < 2.2e-16
```

```
m_compactness <-lm(data = df_mean, compactness~ radius+texture+ perimeter+smoothness+area+conca
summary(m_compactness)
```

```
##
```

```
## Call:
## lm(formula = compactness ~ radius + texture + perimeter + smoothness +
##     area + concavity + concave.points + symmetry + fractal_dimension,
##     data = df_mean)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.044976 -0.006177 -0.000675  0.005421  0.058426
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -2.053e-01  1.457e-02 -14.091  < 2e-16 ***
## radius            -7.702e-02  4.234e-03 -18.189  < 2e-16 ***
## texture            2.376e-04  1.185e-04   2.004  0.04555 *
## perimeter          1.392e-02  6.198e-04  22.459  < 2e-16 ***
## smoothness         1.759e-01  5.694e-02   3.089  0.00211 **
## area              -1.216e-04  8.592e-06 -14.149  < 2e-16 ***
## concavity          6.410e-02  1.998e-02   3.208  0.00141 **
## concave.points     1.077e-01  5.622e-02   1.915  0.05598 .
## symmetry           9.448e-02  2.250e-02   4.200  3.1e-05 ***
## fractal_dimension  2.348e+00  1.370e-01  17.136  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01119 on 559 degrees of freedom
## Multiple R-squared:  0.9558, Adjusted R-squared:  0.9551
## F-statistic:  1343 on 9 and 559 DF,  p-value: < 2.2e-16
```

The variables area, perimeter and compactness are well explained by the other variables (Adjusted R-squared very close to 1). So we discard them.

3 GLM

```
m <-glm(data = df_mean, diagnosis~ radius+texture+smoothness+concavity+concave.points+symmetry+
summary(m)
```

```
##
## Call:
## glm(formula = diagnosis ~ radius + texture + smoothness + concavity +
##     concave.points + symmetry + fractal_dimension, family = binomial,
##     data = df_mean)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.35180  -0.13938  -0.03229   0.02046   3.15368
```

```
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -28.38387    6.66946  -4.256 2.08e-05 ***
## radius               0.88701    0.21852   4.059 4.93e-05 ***
## texture              0.37262    0.06212   5.998 2.00e-09 ***
## smoothness          78.50170   32.64920   2.404   0.0162 *
## concavity           15.52082    8.35462   1.858   0.0632 .
## concave.points      46.67203   26.16265   1.784   0.0744 .
## symmetry            16.85783   10.75613   1.567   0.1170
## fractal_dimension -101.54448   61.26233  -1.658   0.0974 .
## ---
## 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: 153.35  on 561  degrees of freedom
## AIC: 169.35
##
## Number of Fisher Scoring iterations: 8
```

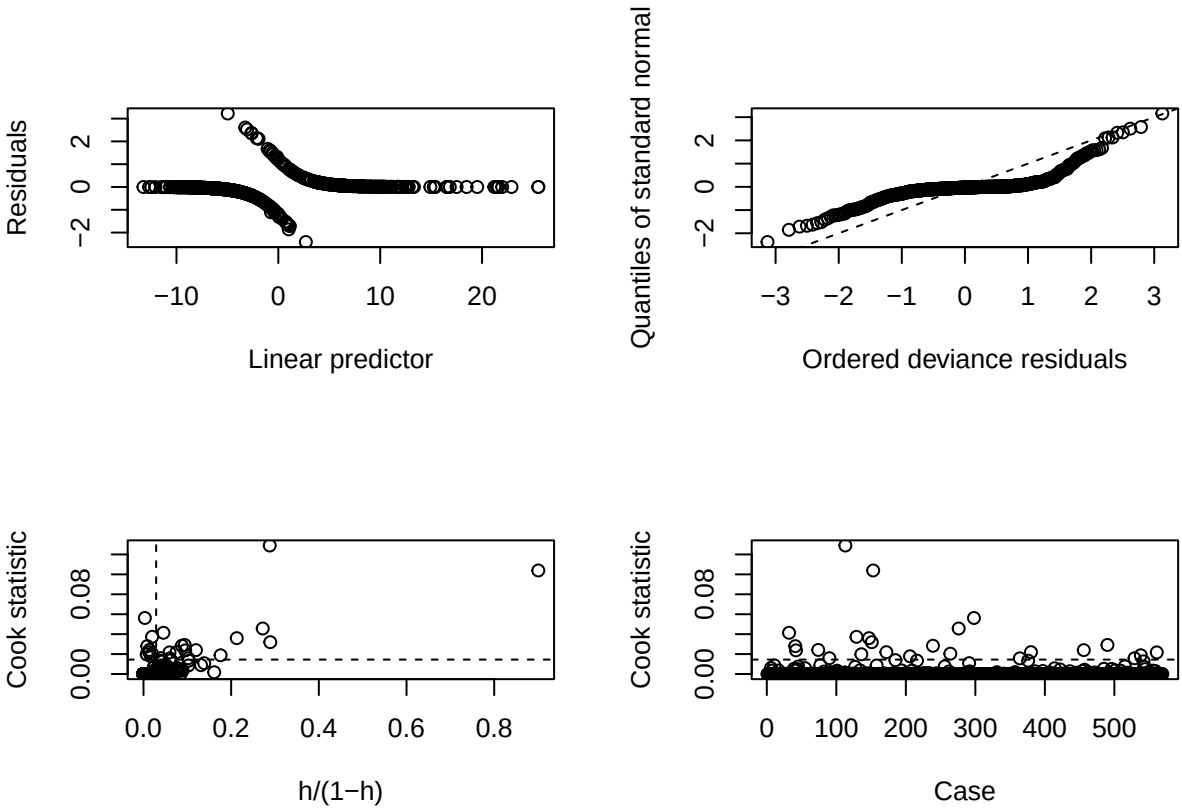
3.1 Validation

```
751.44/153.35
```

```
## [1] 4.900163
```

Overdistribution ?

```
library(boot)
diag <- glm.diag(m)
glm.diag.plots(m, diag)
```



Let's see with an anova test if we can remove symmetry,concavity,concave.points and fractal_dimension

```
m1 <-glm(data = df_mean, diagnosis~ radius+texture+smoothness+concavity+concave.points+fractal_
summary(m1)
```

```
##
## Call:
## glm(formula = diagnosis ~ radius + texture + smoothness + concavity +
##     concave.points + fractal_dimension, family = binomial, data = df_mean)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.33122  -0.15084  -0.03480   0.02274   3.04740
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -26.65706    6.59839  -4.040 5.35e-05 ***
## radius              0.86784    0.22209   3.908 9.32e-05 ***
## texture             0.36277    0.06098   5.949 2.70e-09 ***
## smoothness         90.53604   33.17393   2.729  0.00635 **
## concavity          17.34487    8.23293   2.107  0.03514 *
## concave.points     45.65526   26.56395   1.719  0.08567 .
## fractal_dimension -93.18039   59.52661  -1.565  0.11750
## ---
```

```
## 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: 155.78  on 562  degrees of freedom
## AIC: 169.78
##
## Number of Fisher Scoring iterations: 8
```

```
anova(m,m1,test="Chisq")
```

Resid. Df	Resid. Dev	Df	Deviance	Pr(>Chi)
561	153.3485	NA	NA	NA
562	155.7755	-1	-2.426971	0.1192631

```
m2 <-glm(data = df_mean, diagnosis~ radius+texture+smoothness+concavity+concave.points,family=
summary(m2)
```

```
##
## Call:
## glm(formula = diagnosis ~ radius + texture + smoothness + concavity +
##     concave.points, family = binomial, data = df_mean)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.28927  -0.15267  -0.03761   0.02390   3.03440
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -32.18048    5.69324  -5.652 1.58e-08 ***
## radius           0.99766    0.20762   4.805 1.55e-06 ***
## texture          0.36496    0.06131   5.953 2.64e-09 ***
## smoothness      72.72278   30.46830   2.387   0.0170 *
## concavity       10.21913    6.99120   1.462   0.1438
## concave.points  48.66262   26.54605   1.833   0.0668 .
## ---
## 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: 158.34  on 563  degrees of freedom
## AIC: 170.34
##
## Number of Fisher Scoring iterations: 8
```

```
anova(m1,m2,test="Chisq")
```

Resid. Df	Resid. Dev	Df	Deviance	Pr(>Chi)
562	155.7755	NA	NA	NA
563	158.3372	-1	-2.56172	0.1094794

```
m3 <-glm(data = df_mean, diagnosis~ radius+texture+smoothness+concave.points,family=binomial)
summary(m3)
```

```
##
## Call:
## glm(formula = diagnosis ~ radius + texture + smoothness + concave.points,
##     family = binomial, data = df_mean)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.42132  -0.15010  -0.04247   0.02603   2.86598
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -28.57552    4.81406  -5.936 2.92e-09 ***
## radius           0.85081    0.17112   4.972 6.63e-07 ***
## texture          0.35845    0.05985   5.990 2.10e-09 ***
## smoothness      52.26403   26.08496   2.004   0.0451 *
## concave.points  78.73692   16.59332   4.745 2.08e-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: 160.32  on 564  degrees of freedom
## AIC: 170.32
##
## Number of Fisher Scoring iterations: 8
```

```
anova(m2,m3,test="Chisq")
```

Resid. Df	Resid. Dev	Df	Deviance	Pr(>Chi)
563	158.3372	NA	NA	NA
564	160.3203	-1	-1.983059	0.1590686

3.2 Visualization

(to see) ## glm with glmnet Let's separate the dataset into train and test set

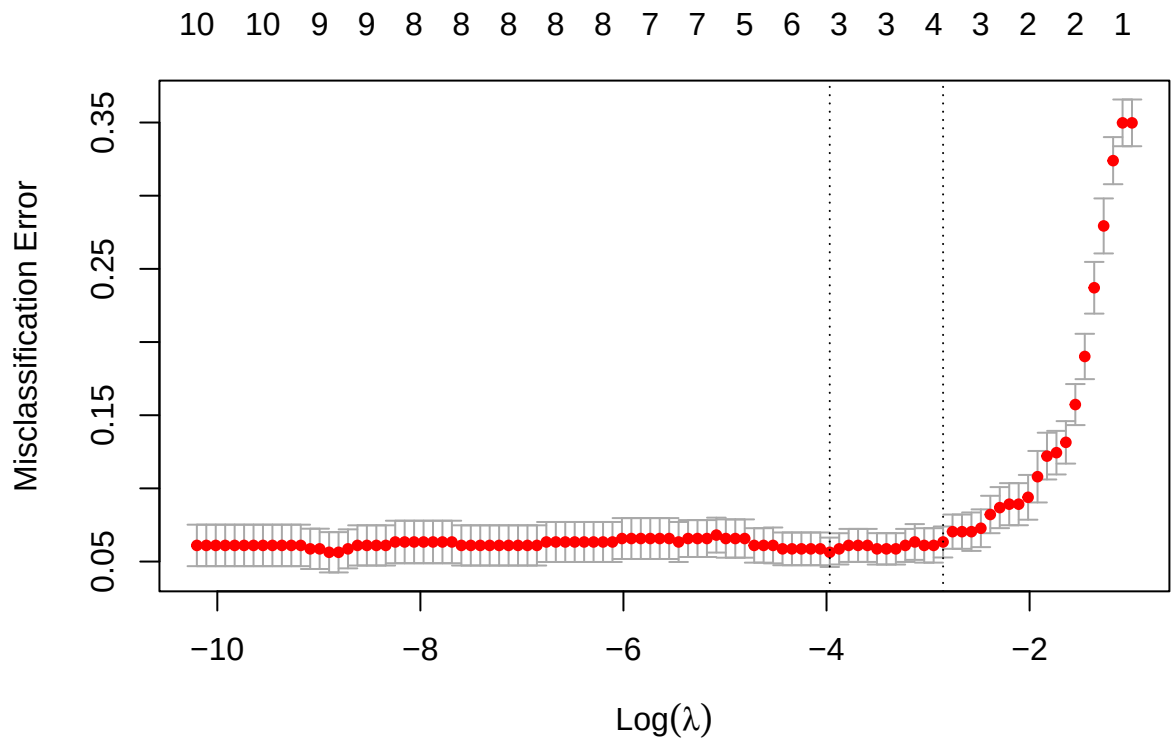
```r
## 75% of the sample size
prop_train_test <- floor(0.75 * nrow(df_mean))

## set the seed to make your partition reproducible
set.seed(123)
train_ind <- sample(seq_len(nrow(df_mean)), size = prop_train_test)
#sample = sample.split(data$num, SplitRatio = .75)
#train = subset(data, sample == TRUE)
#test  = subset(data, sample == FALSE)
train <- df_mean[train_ind, ]
test <- df_mean[-train_ind, ]

x_train <- train[,-1]
y_train <- train$diagnosis
x_test <-  test[,-1]
y_test <- test$diagnosis
```

```r
tol_length=length(levels(y_train))

# run glm : train and test with cross validation
cvfit<-cv.glmnet(as.matrix(x_train), y_train,family = "binomial", type.measure="class")
plot(cvfit)
```




```r
cvfit$lambda.min
```

```
## [1] 0.01890567
```

```
assess<-assess.glmnet(cvfit,newx=as.matrix(x_test), newy=y_test, s='lambda.min')
confusion.glmnet(cvfit, newx =as.matrix(x_test), newy = y_test, s = 'lambda.min')
```

```
##          True
## Predicted  B  M Total
##     B     79 11    90
##     M      1 52    53
##     Total 80 63   143
##
##  Percent Correct:  0.9161
```

```
as.numeric(1-assess$class)
```

```
## [1] 0.9160839
```

```
coef(cvfit, s="lambda.min")
```

```
## 11 x 1 sparse Matrix of class "dgCMatrix"
##                              s1
## (Intercept)       -11.9549638
## radius              0.3363175
## texture             0.1750409
## perimeter           .
## area                .
## smoothness          .
## compactness         .
## concavity           .
## concave.points     63.6306102
## symmetry            .
## fractal_dimension   .
```

Let's look with the first lambda:

```
cvfit$lambda.1se
```

```
## [1] 0.05773519
```

```
assess<-assess.glmnet(cvfit,newx=as.matrix(x_test), newy=y_test, s='lambda.1se')
confusion.glmnet(cvfit, newx =as.matrix(x_test), newy = y_test, s = 'lambda.1se')
```

```
##          True
## Predicted  B  M Total
##     B     80 13    93
##     M      0 50    50
##     Total 80 63   143
##
##  Percent Correct:  0.9091
```

```
as.numeric(1-assess$class)
```

```
## [1] 0.9090909
```

```
coef(cvfit, s="lambda.1se")
```

```
## 11 x 1 sparse Matrix of class "dgCMatrix"
##                              s1
## (Intercept)       -6.87636978
## radius             0.01235359
## texture            0.06882308
## perimeter          0.02960914
## area               .
## smoothness         .
## compactness        .
## concavity          .
## concave.points    40.52201326
## symmetry           .
## fractal_dimension  .
```

4 PCA

```
p1 <-prcomp(df_mean[-1])
summary(p1)
```

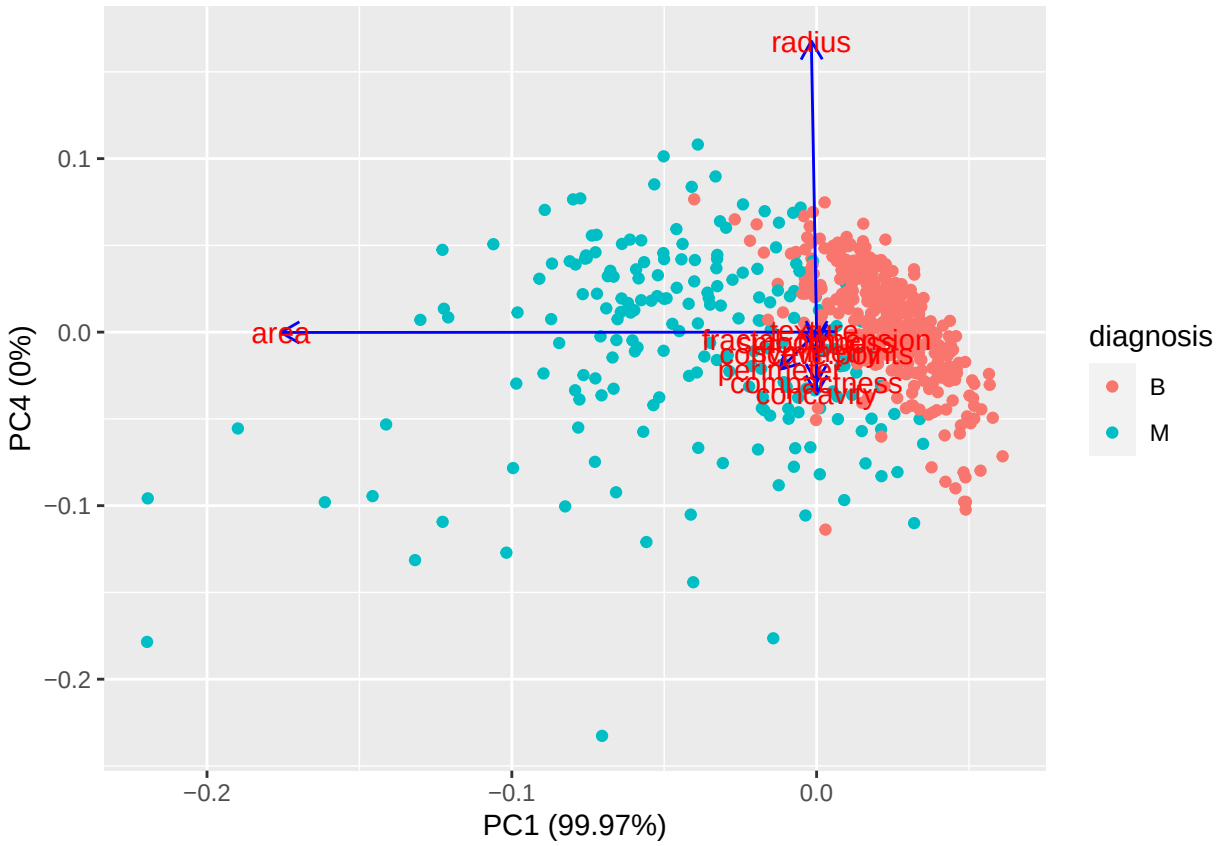
```
## Importance of components:
##                             PC1     PC2     PC3    PC4     PC5     PC6     PC7
## Standard deviation     352.7496 4.20290 3.86547 0.2281 0.03302 0.02246 0.01498
## Proportion of Variance   0.9997 0.00014 0.00012 0.0000 0.00000 0.00000 0.00000
## Cumulative Proportion    0.9997 0.99988 1.00000 1.0000 1.00000 1.00000 1.00000
##                            PC8      PC9     PC10
## Standard deviation     0.0107 0.006723 0.002738
## Proportion of Variance 0.0000 0.000000 0.000000
## Cumulative Proportion  1.0000 1.000000 1.000000
```

```
#str(p1)
p1
```

```
## Standard deviations (1, .., p=10):
##  [1] 3.527496e+02 4.202899e+00 3.865469e+00 2.280785e-01 3.302431e-02
##  [6] 2.246078e-02 1.498314e-02 1.070032e-02 6.723036e-03 2.737508e-03
##
```

```
## Rotation (n x k) = (10 x 10):
##                                PC1           PC2           PC3          PC4
## radius             -9.865063e-03  0.0771733355  0.1024301301  0.952456868
## texture            -3.916220e-03  0.7783081363 -0.6278541228  0.004445528
## perimeter          -6.796386e-02  0.6213843010  0.7698582420 -0.121414104
## area               -9.976313e-01 -0.0461513749 -0.0509958616 -0.001113681
## smoothness         -7.064044e-06  0.0001775170  0.0007770653 -0.034043607
## compactness        -7.468080e-05  0.0037175357  0.0035919739 -0.164145250
## concavity          -1.550715e-04  0.0039201536  0.0027419692 -0.201758030
## concave.points     -9.058071e-05  0.0015036949  0.0016864113 -0.069072812
## symmetry           -1.177025e-05  0.0009092201  0.0010449234 -0.062495480
## fractal_dimension   5.664885e-06  0.0001237651  0.0001339522 -0.024038634
##                                PC5           PC6           PC7           PC8
## radius             -2.595576e-01  1.773711e-02 -8.169307e-02 -4.256383e-02
## texture             1.634922e-04  5.052041e-04 -3.071585e-04  3.820028e-04
## perimeter           3.993975e-02 -2.997531e-03  1.355597e-02  6.738197e-03
## area                2.176553e-05  5.861401e-05 -7.944077e-05 -5.039876e-05
## smoothness         -1.179914e-01  1.872529e-01 -3.601308e-01  5.742689e-01
## compactness        -2.583014e-01  2.297613e-01 -7.553151e-01 -4.991668e-01
## concavity          -8.333273e-01 -4.220235e-01  2.369160e-01 -8.406286e-02
## concave.points     -2.625501e-01  5.965391e-02 -2.037723e-01  6.355082e-01
## symmetry           -2.905607e-01  8.537795e-01  4.208549e-01 -7.454169e-02
## fractal_dimension  -5.099665e-02  3.493464e-02 -1.348186e-01 -5.193119e-02
##                                PC9          PC10
## radius             -9.842129e-03 -3.726245e-03
## texture             1.432045e-04  5.680747e-05
## perimeter           2.186348e-03  1.117907e-03
## area               -9.791094e-06 -2.263191e-05
## smoothness          6.966649e-01 -7.094988e-02
## compactness        -1.077020e-01 -1.491696e-01
## concavity           1.528201e-01 -1.352082e-02
## concave.points     -6.903978e-01  2.626377e-02
## symmetry           -2.043042e-03  6.930804e-03
## fractal_dimension   5.430797e-02  9.857884e-01
```

```
autoplot(p1,x=1, y=4,data=df_mean,colour = "diagnosis", loadings = TRUE, loadings.colour = "blu
         loadings.label = TRUE)
```



The first PCA component explain the 99% of the data. Interpretation distribution ? k nearst ?

```
df_mean_pca <- cbind(df_mean, p1$x)
summary(df_mean_pca)
```

```
##  diagnosis     radius          texture        perimeter          area
##  B:357     Min.   : 6.981   Min.   : 9.71   Min.   : 43.79   Min.   : 143.5
##  M:212     1st Qu.:11.700   1st Qu.:16.17   1st Qu.: 75.17   1st Qu.: 420.3
##            Median :13.370   Median :18.84   Median : 86.24   Median : 551.1
##            Mean   :14.127   Mean   :19.29   Mean   : 91.97   Mean   : 654.9
##            3rd Qu.:15.780   3rd Qu.:21.80   3rd Qu.:104.10   3rd Qu.: 782.7
##            Max.   :28.110   Max.   :39.28   Max.   :188.50   Max.   :2501.0
##    smoothness       compactness        concavity       concave.points
##  Min.   :0.05263   Min.   :0.01938   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.08637   1st Qu.:0.06492   1st Qu.:0.02956   1st Qu.:0.02031
##  Median :0.09587   Median :0.09263   Median :0.06154   Median :0.03350
##  Mean   :0.09636   Mean   :0.10434   Mean   :0.08880   Mean   :0.04892
##  3rd Qu.:0.10530   3rd Qu.:0.13040   3rd Qu.:0.13070   3rd Qu.:0.07400
##  Max.   :0.16340   Max.   :0.34540   Max.   :0.42680   Max.   :0.20120
##     symmetry      fractal_dimension       PC1                PC2
##  Min.   :0.1060   Min.   :0.04996   Min.   :-1848.3   Min.   :-24.6830
##  1st Qu.:0.1619   1st Qu.:0.05770   1st Qu.: -128.4   1st Qu.: -2.4398
##  Median :0.1792   Median :0.06154   Median :  103.9   Median : -0.4213
##  Mean   :0.1812   Mean   :0.06280   Mean   :    0.0   Mean   :  0.0000
```

```
##  3rd Qu.:0.1957   3rd Qu.:0.06612   3rd Qu.:  235.2   3rd Qu.:  2.5400
##  Max.   :0.3040   Max.   :0.09744   Max.   :  513.5   Max.   : 16.8702
##       PC3                PC4                PC5                 PC6
##  Min.   :-24.0804   Min.   :-1.26579   Min.   :-0.307819   Min.   :-0.0887264
##  1st Qu.: -1.8838   1st Qu.:-0.11037   1st Qu.:-0.010592   1st Qu.:-0.0142114
##  Median :  0.4681   Median : 0.02708   Median : 0.003205   Median : 0.0002414
##  Mean   :  0.0000   Mean   : 0.00000   Mean   : 0.000000   Mean   : 0.0000000
##  3rd Qu.:  2.3750   3rd Qu.: 0.15251   3rd Qu.: 0.015512   3rd Qu.: 0.0131804
##  Max.   : 12.0763   Max.   : 0.58826   Max.   : 0.148701   Max.   : 0.0979141
##       PC7                  PC8                  PC9
##  Min.   :-0.0756378   Min.   :-0.0491463   Min.   :-3.214e-02
##  1st Qu.:-0.0091623   1st Qu.:-0.0060271   1st Qu.:-3.923e-03
##  Median : 0.0004192   Median : 0.0004688   Median : 3.682e-05
##  Mean   : 0.0000000   Mean   : 0.0000000   Mean   : 0.000e+00
##  3rd Qu.: 0.0087381   3rd Qu.: 0.0066076   3rd Qu.: 3.991e-03
##  Max.   : 0.0577082   Max.   : 0.0485694   Max.   : 2.320e-02
##       PC10
##  Min.   :-9.149e-03
##  1st Qu.:-1.727e-03
##  Median :-6.735e-05
##  Mean   : 0.000e+00
##  3rd Qu.: 1.448e-03
##  Max.   : 1.195e-02
```

```
glm_pca <-glm(data= df_mean_pca, df_mean_pca$diagnosis ~PC1+PC2+PC3+PC4+PC5+PC6, family = binor
```

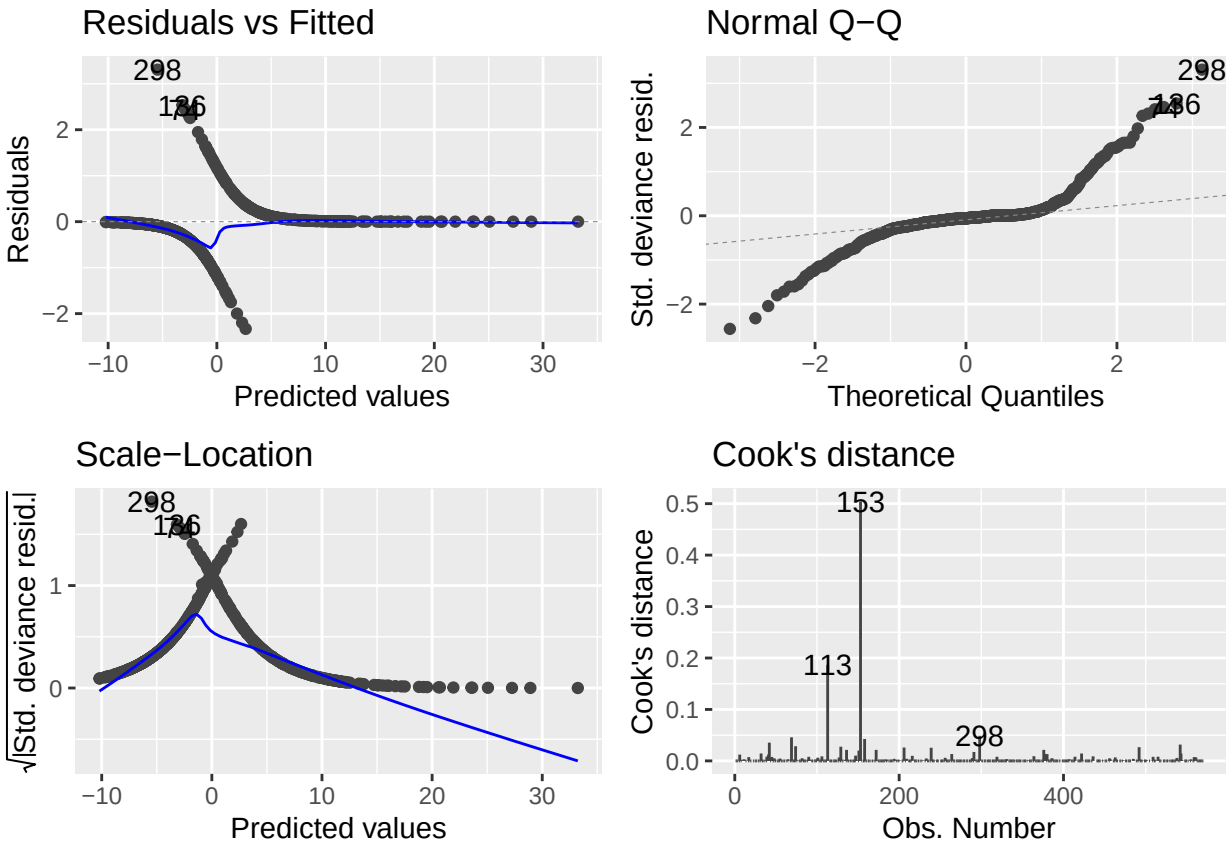
```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(glm_pca)
```

```
##
## Call:
## glm(formula = df_mean_pca$diagnosis ~ PC1 + PC2 + PC3 + PC4 +
##     PC5 + PC6, family = binomial, data = df_mean_pca)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3325  -0.1995  -0.0565   0.0168   3.3065
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.264462   0.480395  -0.551  0.58197
## PC1          -0.018446   0.002775  -6.647 2.99e-11 ***
## PC2           0.343023   0.105645   3.247  0.00117 **
## PC3          -0.112416   0.119567  -0.940  0.34712
## PC4          -6.524728   1.671987  -3.902 9.53e-05 ***
```

```
## PC5          -31.829078   7.097893  -4.484 7.32e-06 ***
## PC6           27.805384   9.323062   2.982  0.00286 **
## ---
## 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: 174.12  on 562  degrees of freedom
## AIC: 188.12
##
## Number of Fisher Scoring iterations: 8
```

```
autoplot(glm_pca,1:4)
```



---

5 Annexe

5.1 More informations about the dataset

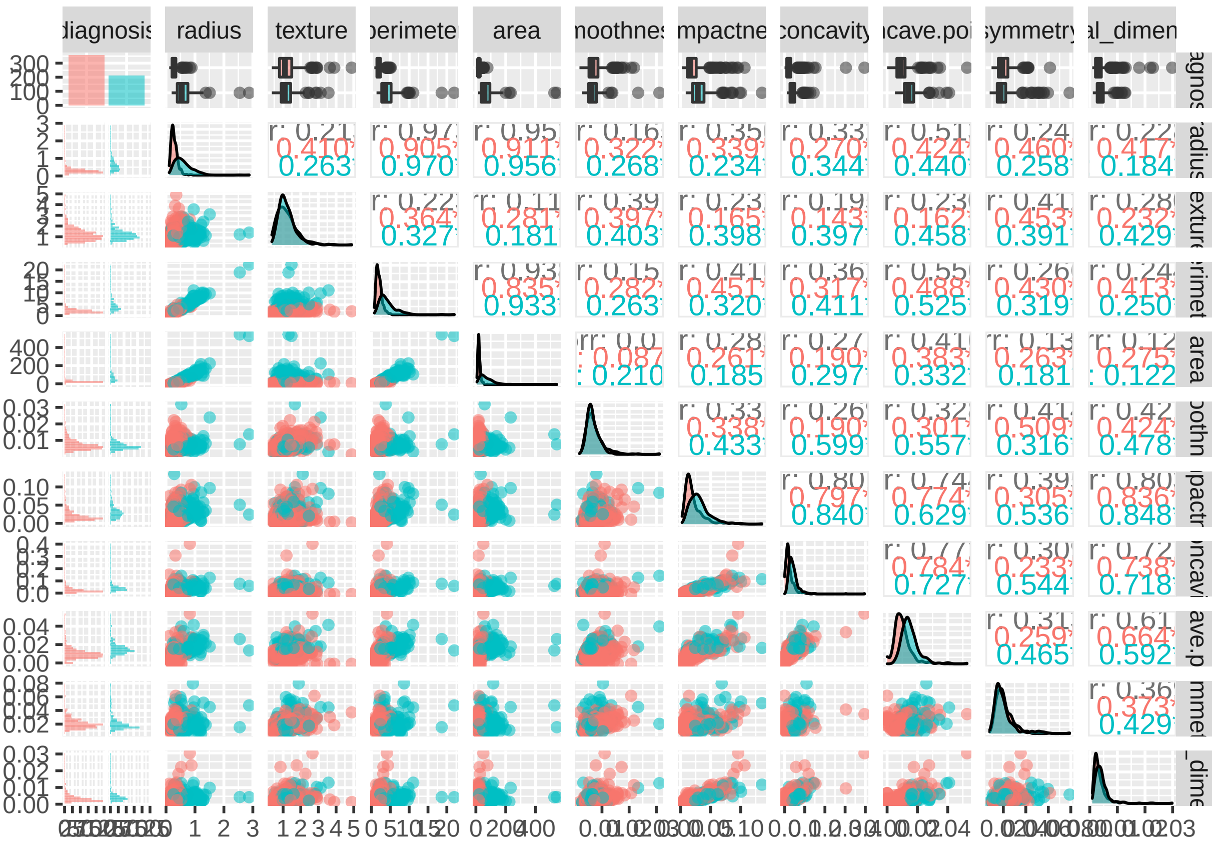
The 3-dimensional space is that described in: [3].

This database is also available through the UW CS ftp server: ftp ftp.cs.wisc.edu cd math-prog/cpo-dataset/machine-learn/WDBC/

5.2 Correlation in the 'standard deviation' and 'worst' group

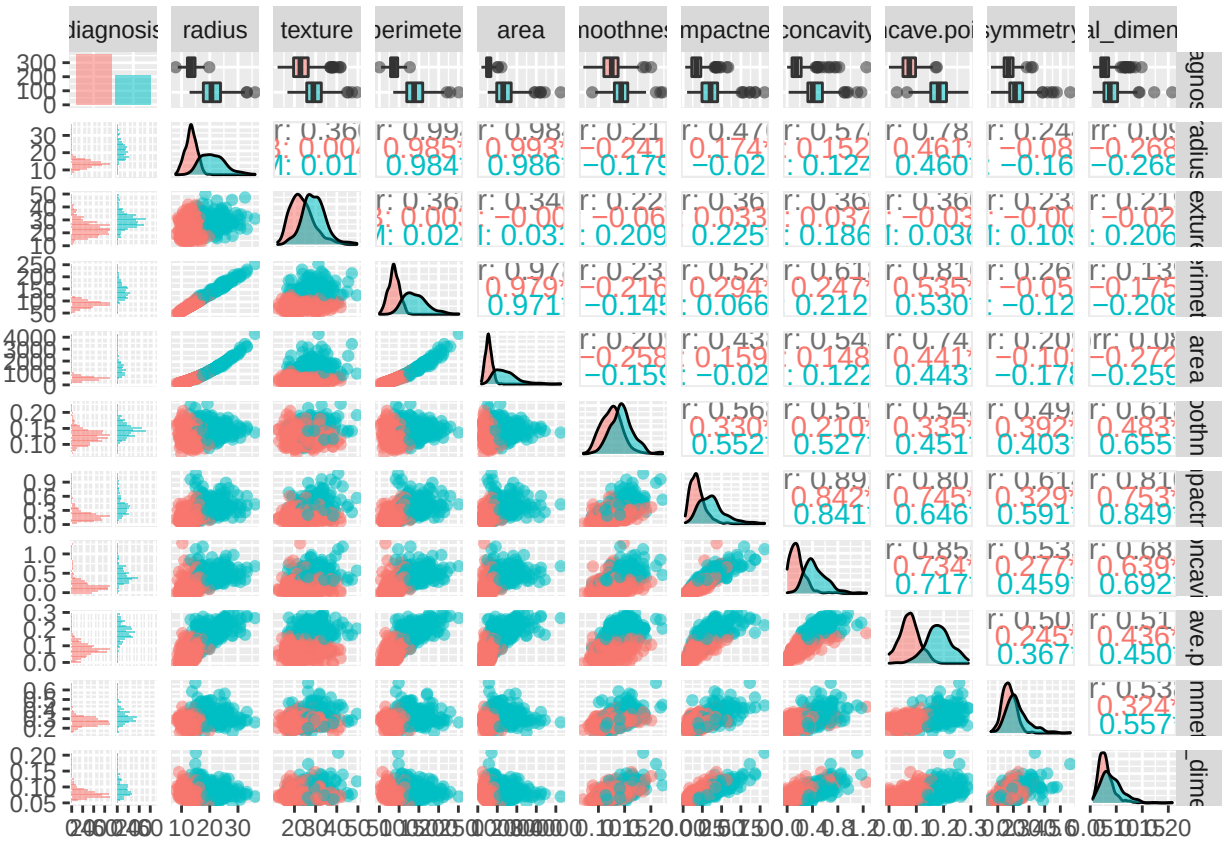
```
## 'data.frame':    569 obs. of  11 variables:
##  $ diagnosis         : Factor w/ 2 levels "B","M": 2 2 2 2 2 2 2 2 2 2 ...
##  $ radius            : num  1.095 0.543 0.746 0.496 0.757 ...
##  $ texture           : num  0.905 0.734 0.787 1.156 0.781 ...
##  $ perimeter         : num  8.59 3.4 4.58 3.44 5.44 ...
##  $ area              : num  153.4 74.1 94 27.2 94.4 ...
##  $ smoothness        : num  0.0064 0.00522 0.00615 0.00911 0.01149 ...
##  $ compactness       : num  0.049 0.0131 0.0401 0.0746 0.0246 ...
##  $ concavity         : num  0.0537 0.0186 0.0383 0.0566 0.0569 ...
##  $ concave.points    : num  0.0159 0.0134 0.0206 0.0187 0.0188 ...
##  $ symmetry          : num  0.03 0.0139 0.0225 0.0596 0.0176 ...
##  $ fractal_dimension.: num  0.00619 0.00353 0.00457 0.00921 0.00511 ...
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
## 'data.frame':    569 obs. of  11 variables:
##  $ diagnosis         : Factor w/ 2 levels "B","M": 2 2 2 2 2 2 2 2 2 2 ...
##  $ radius            : num  25.4 25 23.6 14.9 22.5 ...
##  $ texture           : num  17.3 23.4 25.5 26.5 16.7 ...
##  $ perimeter         : num  184.6 158.8 152.5 98.9 152.2 ...
##  $ area              : num  2019 1956 1709 568 1575 ...
##  $ smoothness        : num  0.162 0.124 0.144 0.21 0.137 ...
##  $ compactness       : num  0.666 0.187 0.424 0.866 0.205 ...
##  $ concavity         : num  0.712 0.242 0.45 0.687 0.4 ...
##  $ concave.points    : num  0.265 0.186 0.243 0.258 0.163 ...
##  $ symmetry          : num  0.46 0.275 0.361 0.664 0.236 ...
##  $ fractal_dimension.: num  0.1189 0.089 0.0876 0.173 0.0768 ...
```

```
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
```



We get the same results as for the mean group.

---

6 References

- [1] https://www.researchgate.net/figure/a-b-Fine-needle-aspiration-cytology-of-the-breast-lesion-showed-singly-lying_fig1_41548857
- [2] https://pubmed.ncbi.nlm.nih.gov/7091922/
- [3] K. P. Bennett and O. L. Mangasarian: "Robust Linear Programming Discrimination of Two Linearly Inseparable Sets," Optimization Methods and Software 1, 1992, 23-34

7 Version of R used

```
## R version 4.1.1 (2021-08-10)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS Big Sur 10.16
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.1/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.1/Resources/lib/libRlapack.dylib
```

```
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] boot_1.3-28       glmnet_4.1-3      Matrix_1.3-4      kableExtra_1.3.4
##  [5] here_1.0.1        MASS_7.3-54       ggfortify_0.4.12  GGally_2.1.2
##  [9] ggplot2_3.3.5     papaja_0.1.0.9997 pacman_0.5.1
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.7         svglite_2.0.0      lattice_0.20-44    tidyr_1.1.4
##  [5] rprojroot_2.0.2    digest_0.6.28      foreach_1.5.1      utf8_1.2.2
##  [9] R6_2.5.1           plyr_1.8.6         evaluate_0.14      httr_1.4.2
## [13] highr_0.9          pillar_1.6.3       rlang_0.4.11       rstudioapi_0.13
## [17] rmarkdown_2.11     labeling_0.4.2     splines_4.1.1      webshot_0.5.2
## [21] stringr_1.4.0      munsell_0.5.0      compiler_4.1.1     xfun_0.26
## [25] pkgconfig_2.0.3    systemfonts_1.0.3  shape_1.4.6        htmltools_0.5.2
## [29] tidyselect_1.1.1   tibble_3.1.5       gridExtra_2.3      codetools_0.2-18
## [33] reshape_0.8.8      fansi_0.5.0        viridisLite_0.4.0  crayon_1.4.1
## [37] dplyr_1.0.7        withr_2.4.2        grid_4.1.1         gtable_0.3.0
## [41] lifecycle_1.0.1    magrittr_2.0.1     scales_1.1.1       stringi_1.7.5
## [45] farver_2.1.0       xml2_1.3.3         ellipsis_0.3.2     generics_0.1.0
## [49] vctrs_0.3.8        RColorBrewer_1.1-2 iterators_1.0.13   tools_4.1.1
## [53] glue_1.4.2         purrr_0.3.4        fastmap_1.1.0      survival_3.2-11
## [57] yaml_2.2.1         colorspace_2.0-2   rvest_1.0.2        knitr_1.36
```

8 References

James, William. 1890. “The Perception of Reality.” *Principles of Psychology* 2: 283–324.