Breast Cancer Wisconsin Diagnostic

L. Favero

Contents

Introduction

Context

[. . .]

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."

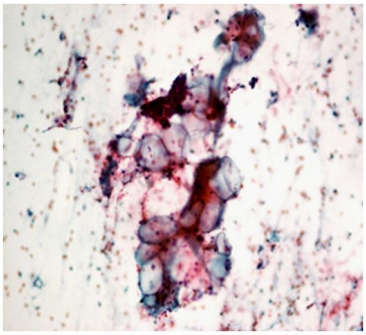


Figure 1: fine needle aspirate (FNA) of a breast mass

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

Name of the variables	type	Description
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 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/

2. Question of interest

The aim is to **predict whether the cancer is benign or malignant with only few features**

3. Choice of data

There is a lot of data, we choose only the data concerning the mean. The following analysis has been done also for the worst-type of data and lead to similar conclusion. See in annex the inspection of these 2 subframes.

4. Plan of the analysis

- Inspect the data to understand it better
- Check if the sample proportion of benign and malignant cancer is representative of the whole population
- Select variables of interest by discarding according to the correlation level
- Built GLM models and select the best model with anova, discuss the goodness of this model
- Apply PCA to see to what extent dimension can be reduced and perform a GLM model with relevant dimension.
- Conclude by addressing the accuracy of these GLM models with a subset of the data

In annex, I built a GLM model with the library glmnet and I find the same selection of variable.

I. Preprocessing

1. Required libraries

```
# Check if packages are installed and if not install 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
```

```
if(!require(pacman)) {
  install.packages("equatiomatic")
}
if(!require(devtools)) {
  install.packages("devtools")
  install_github("kassambara/factoextra")
}
```

```
## Loading required package: devtools

## Loading required package: usethis
```

```
pacman::p_load(pander,       # Rmarkdown visualization
               GGally,
               ggfortify,
               ggplot2,
               #MASS,        # for stepAIC
               here,         # to load the path
               kableExtra,   # Rmarkdown table visualization
               papaja,
               glmnet,       # GLM implementation and analysis library
               equatiomatic, # Rmarkdown model equation visualization
               patchwork,    # arrange subplot
               devtools,     # tool for PCA
               factoextra,   # tool for PCA

)
```

```
##
## Your package installed
```

2. Import data

```
path = here("LUCILE") # get relative path
setwd(path) # set working directory
df <-
  read.csv('data.csv', stringsAsFactors = 1)# load data from github repository
```

Let's delete ID number and the last variable - which is full of NA- because there are not relevant.

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

We will work only with the mean-type data, so let's create a new frame for the variables of this type.

```
df_mean <- data.frame(
  "diagnosis"          = df$diagnosis,
  "radius"             = df$radius_mean,
  "texture"            = df$texture_mean,
  "perimeter"          = df$perimeter_mean,
  "area"               = df$area_mean,
  "smoothness"         = df$smoothness_mean,
  "compactness"        = df$compactness_mean,
  "concavity"          = df$concavity_mean,
  "concave.points"     = df$concave.points_mean,
  "symmetry"           = df$symmetry_mean,
  "fractal_dimension"  = df$fractal_dimension_mean
)
```

3. Inspection

Structure

```
str(df_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 ...
```

Head

```
pander(head(df_mean))
```

Table 2: Table continues below

diagnosis	radius	texture	perimeter	area	smoothness	compactness
M	17.99	10.38	122.8	1001	0.1184	0.2776
M	20.57	17.77	132.9	1326	0.08474	0.07864
M	19.69	21.25	130	1203	0.1096	0.1599
M	11.42	20.38	77.58	386.1	0.1425	0.2839
M	20.29	14.34	135.1	1297	0.1003	0.1328
M	12.45	15.7	82.57	477.1	0.1278	0.17

concavity	concave.points	symmetry	fractal_dimension
0.3001	0.1471	0.2419	0.07871
0.0869	0.07017	0.1812	0.05667
0.1974	0.1279	0.2069	0.05999
0.2414	0.1052	0.2597	0.09744
0.198	0.1043	0.1809	0.05883
0.1578	0.08089	0.2087	0.07613

Summary

```
pander(summary(df_mean))
```

Table 4: Table continues below

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
NA	Median :13.370	Median :18.84	Median : 86.24	Median : 551.1
NA	Mean :14.127	Mean :19.29	Mean : 91.97	Mean : 654.9
NA	3rd Qu.:15.780	3rd Qu.:21.80	3rd Qu.:104.10	3rd Qu.: 782.7
NA	Max. :28.110	Max. :39.28	Max. :188.50	Max. :2501.0

Table 5: Table continues below

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

4. Proportion of benign vs malignant cancer

```
kable(prop.table(table(df_mean$diagnosis)),col.names =
        c("Severity","Frequency"))
```

Severity	Frequency
B	0.6274165
M	0.3725835

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

However, this proportion is more or less 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]

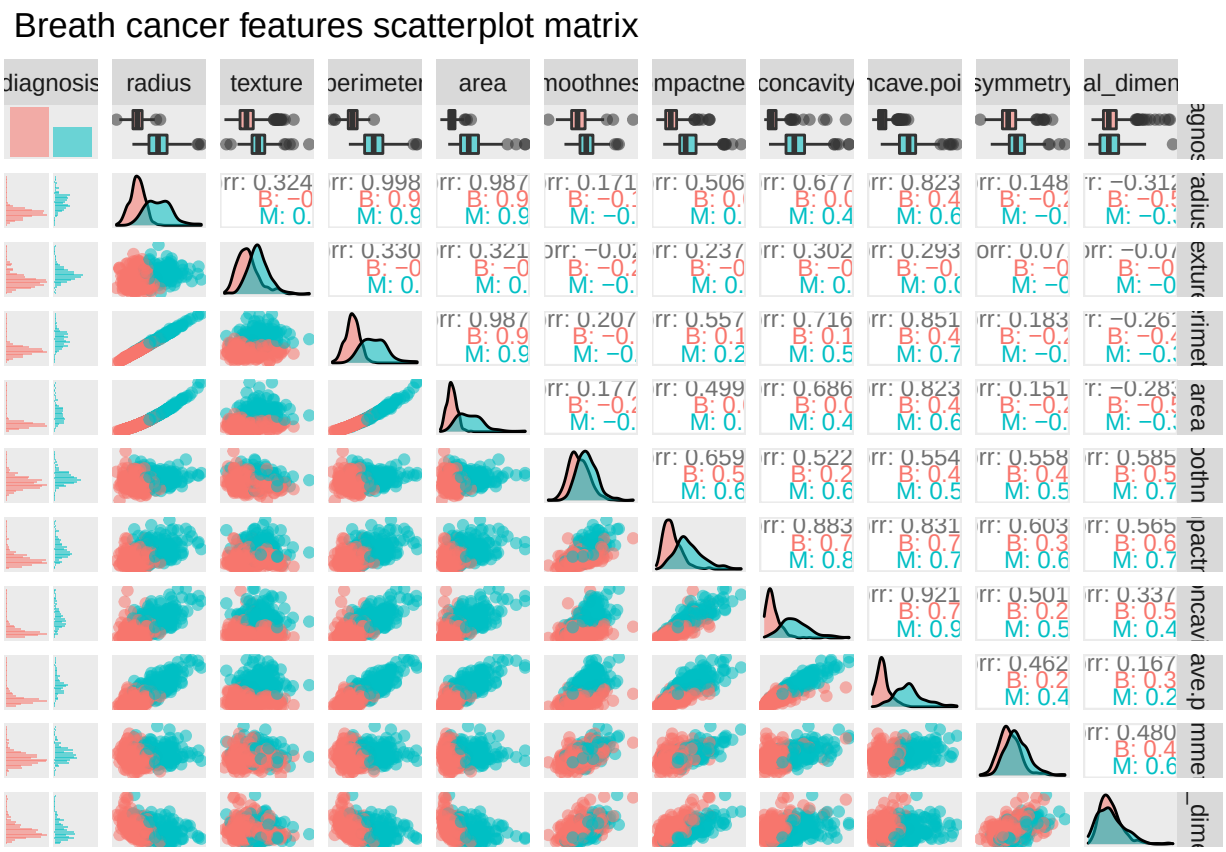
II. Selection of variables of interest

We want to remove the variables with high correlation to avoid problems during the modelisation. According to the description of the data, radius, perimeter, area and compactness should be correlated since it exists a formula between these variables. Let’s verify that.

1. Correlation

ggpairs

```
ggpairs(
  df_mean,
  aes(color = diagnosis, alpha = 0.5),
  upper = list(continuous = wrap(
    "cor", size = 3, alignPercent = 1
  )),
  axisLabels = "none",
  legends = TRUE
) +
  labs(title = " Breath cancer features scatterplot matrix") +
  theme(panel.grid = element_blank(), axis.ticks = element_blank())
```



Description :

- On the lower triangular part of the matrix, scatterplots of each pair of feature for benign (red) and malignant (blue) type are displayed.
- On the diagonal, the variable distribution is displayed.
- On the upper triangular part Pearson correlation is displayed.

Observations :

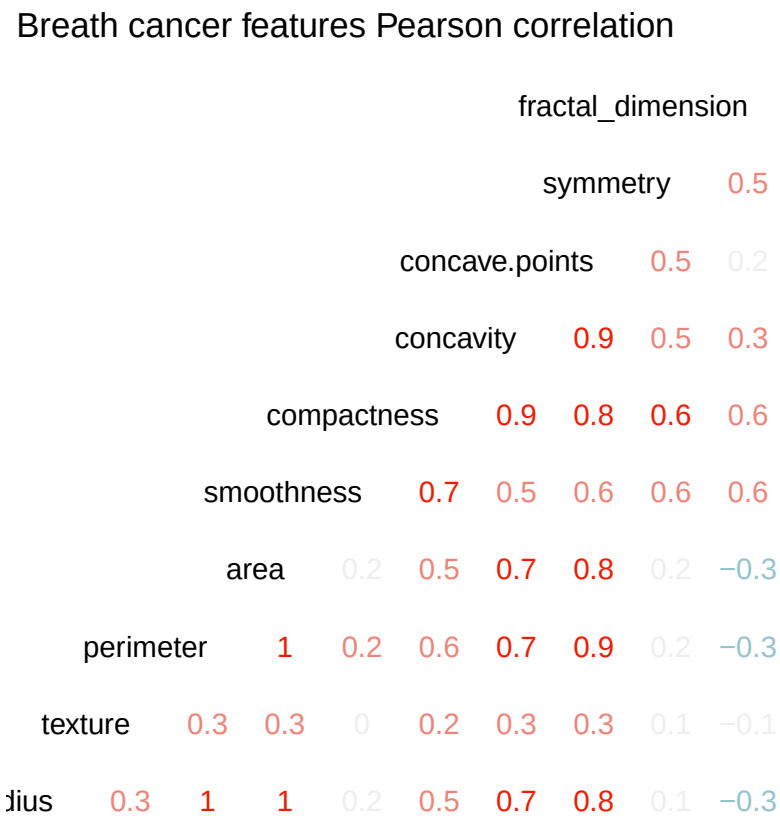
- In each subplot a distinction can be made according to the type of 'diagnosis'.
- The observations coming from malignant cancer seem to be in general bigger than the data coming from benign cancer.

This first observation supports the hypothesis that the value of some features is different according to the severity of the cancer.

ggcorr

The following function permit to visualize better the correlation

```
ggcorr(df_mean,geom = "text",nbreaks = 5,hjust = 1,label = FALSE,
       label_alpha = 0.7) +
      labs(title = " Breath cancer features Pearson correlation") +
      theme(legend.position ="none")
```



2. Variables selection

- As expected, radius, perimeter and area are highly correlated ($r \approx 1$)
- Surprisingly, concavity, compactness and concave.points have a strong correlation.($r \approx 0.8$ or $r \approx 0.9$)

Note:

Even if compactness is define as $perimeter^2/area - 1.0$ the r between this variable and area or perimeter is not 1 because the correlation shows only the linear dependency and their relation is not linear.

Linear models of correlated variables

We want to discard the variables: perimeter, area and compactness. To be sure that these variables can be explained by the remaining variables, we set a linear model to express the potential discarded variable according to the other and address the goodness of the model by looking the adjusted R^2.

```
m_perimeter <-
  lm(
    data = df_mean,
    perimeter ~ radius + texture + area + smoothness + compactness + concavity +
      concave.points + symmetry + fractal_dimension
  )
pander(summary(m_perimeter))
```

Perimeter

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	2.627	0.8317	3.159	0.00167
radius	6.126	0.05223	117.3	0
texture	-0.002105	0.005885	-0.3577	0.7207
area	0.003868	0.0004676	8.272	9.65e-16
smoothness	-8.998	2.816	-3.196	0.001473
compactness	34.07	1.517	22.46	4.27e-80
concavity	3.865	0.9842	3.927	9.671e-05
concave.points	4.283	2.785	1.538	0.1246
symmetry	-1.93	1.127	-1.712	0.08751
fractal_dimension	-41.19	8.19	-5.029	6.643e-07

Table 8: Fitting linear model: perimeter ~ radius + texture + area + smoothness + compactness + concavity + concave.points + symmetry + fractal_dimension

Observations	Residual Std. Error	R^2	Adjusted R^2
569	0.5538	0.9995	0.9995

```
m_area <-
  lm(
    data = df_mean,
    area ~ radius + texture + perimeter + smoothness + compactness + concavity +
      concave.points + symmetry + fractal_dimension
  )
pander(summary(m_area))
```

Area

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-1033	56.77	-18.2	1.821e-58
radius	-81.34	22.3	-3.647	0.0002901
texture	0.4089	0.5023	0.8142	0.4159
perimeter	28.2	3.409	8.272	9.65e-16
smoothness	92.1	242.5	0.3797	0.7043
compactness	-2169	153.3	-14.15	4.538e-39
concavity	221.1	84.67	2.611	0.00927
concave.points	295.4	237.9	1.241	0.215
symmetry	92.13	96.43	0.9554	0.3398
fractal_dimension	6413	661.5	9.696	1.192e-20

Table 10: Fitting linear model: area ~ radius + texture + perimeter + smoothness + compactness + concavity + concave.points + symmetry + fractal_dimension

Observations	Residual Std. Error	R^2	Adjusted R^2
569	47.28	0.9822	0.9819

```
m_compactness <-
  lm(
    data = df_mean,
    compactness ~ radius + texture + perimeter + smoothness + area + concavity +
      concave.points + symmetry + fractal_dimension
  )
pander(summary(m_compactness))
```

Compactness

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-0.2053	0.01457	-14.09	8.313e-39
radius	-0.07702	0.004234	-18.19	2.042e-58
texture	0.0002376	0.0001185	2.004	0.04555
perimeter	0.01392	0.0006198	22.46	4.27e-80
smoothness	0.1759	0.05694	3.089	0.002109
area	-0.0001216	8.592e-06	-14.15	4.538e-39
concavity	0.0641	0.01998	3.208	0.001414
concave.points	0.1077	0.05622	1.915	0.05598
symmetry	0.09448	0.0225	4.2	3.104e-05
fractal_dimension	2.348	0.137	17.14	3.229e-53

Table 12: Fitting linear model: compactness ~ radius + texture + perimeter + smoothness + area + concavity + concave.points + symmetry + fractal_dimension

Observations	Residual Std. Error	R^2	Adjusted R^2
569	0.01119	0.9558	0.9551

Variables discarded

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

```
df_mean_reduc <-df_mean[-c(4,5,7)]
```

III. GLM

1. Set a first GLM model

We set a GLM model with the remaining features. Since we have a problem of classification, we use the binomial family.

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

```
##
## Call:
## glm(formula = diagnosis ~ radius + texture + smoothness + concavity +
##     concave.points + symmetry + fractal_dimension, family = binomial,
##     data = df_mean_reduc)
##
## 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
```

- The algorithm converge: the number of fisher scoring iterations is reasonable.
- There are a lot of variable not significant. To solve this we can try to remove them.
- The ratio of the residual deviance by its degrees of freedom is $153/561 = 0.272$ where the dispersion parameter is 1. There is underdispersion. To solve this we can use the quasibinomial family.

2. Model selection

By performing several anova test, we will see that we can remove the features concavity, symmetry and fractal_dimension, because there is not a significant difference between the model with theses variables and the one without.

fractal_dimension

```r
m1 <-
  glm(
    data = df_mean_reduc,
    diagnosis ~  radius + texture + smoothness + concavity + concave.points +
      symmetry,
    family = binomial
  )
summary(m1)
```

```
##
## Call:
## glm(formula = diagnosis ~ radius + texture + smoothness + concavity +
##     concave.points + symmetry, family = binomial, data = df_mean_reduc)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.31157  -0.14140  -0.03545   0.02138   3.12564
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -34.26429    5.83732  -5.870 4.36e-09 ***
## radius           1.02392    0.20523   4.989 6.07e-07 ***
## texture          0.37400    0.06241   5.992 2.07e-09 ***
## smoothness      60.88592   30.33864   2.007   0.0448 *
## concavity        7.64062    7.05960   1.082   0.2791
## concave.points  50.19961   26.15735   1.919   0.0550 .
## symmetry        15.58198   10.71215   1.455   0.1458
## ---
## 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: 156.25  on 562  degrees of freedom
## AIC: 170.25
##
## Number of Fisher Scoring iterations: 8
```

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

```
## Analysis of Deviance Table
##
## Model 1: diagnosis ~ radius + texture + smoothness + concavity + concave.points +
##     symmetry + fractal_dimension
## Model 2: diagnosis ~ radius + texture + smoothness + concavity + concave.points +
##     symmetry
```

```
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       561     153.35
## 2       562     156.25 -1  -2.9022  0.08846 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- The 2 models have a similar AIC
- From the anova test there is no significant difference between the 2 models

symmetry

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

```
##
## Call:
## glm(formula = diagnosis ~ radius + texture + smoothness + concavity +
##     concave.points, family = binomial, data = df_mean_reduc)
##
## 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
```

```r
anova(m, m2, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: diagnosis ~ radius + texture + smoothness + concavity + concave.points +
##     symmetry + fractal_dimension
## Model 2: diagnosis ~ radius + texture + smoothness + concavity + concave.points
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       561     153.35
## 2       563     158.34 -2  -4.9887  0.08255 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- The 2 models have a similar AIC
- From the anova test there is no significant difference between the 2 models

concavity

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

```
##
## Call:
## glm(formula = diagnosis ~ radius + texture + smoothness + concave.points,
##     family = binomial, data = df_mean_reduc)
##
## 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(m, m3, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: diagnosis ~ radius + texture + smoothness + concavity + concave.points +
##     symmetry + fractal_dimension
## Model 2: diagnosis ~ radius + texture + smoothness + concave.points
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       561     153.35
## 2       564     160.32 -3  -6.9717   0.0728 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- The 2 models have a similar AIC
- From the anova test there is no significant difference between the 2 models

smoothness

```
m4 <-
  glm(data = df_mean_reduc,
      diagnosis ~ radius + texture + concave.points,
      family = binomial)
summary(m4)
```

```
##
## Call:
## glm(formula = diagnosis ~ radius + texture + concave.points,
##     family = binomial, data = df_mean_reduc)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.36043  -0.15742  -0.04644   0.02774   2.82699
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -21.16474    2.51027  -8.431  < 2e-16 ***
```

```
## radius           0.65637    0.12532   5.238 1.63e-07 ***
## texture          0.32593    0.05529   5.895 3.75e-09 ***
## concave.points 101.16839   13.02057   7.770 7.86e-15 ***
## ---
## 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: 164.38  on 565  degrees of freedom
## AIC: 172.38
##
## Number of Fisher Scoring iterations: 8
```

```
anova(m, m4, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: diagnosis ~ radius + texture + smoothness + concavity + concave.points +
##     symmetry + fractal_dimension
## Model 2: diagnosis ~ radius + texture + concave.points
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       561     153.35
## 2       565     164.38 -4  -11.035  0.02618 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- The model without smoothness has an AIC much bigger than the other model
- From the anova test there is a significant difference between the 2 models
- We cannot remove the feature smoothness of the model.

3. Deal with underdispersion

Let's set a GLM model with a quasibinomial family to solve the issue of underdispersion.

```
m_selected_quas <-
  glm(
    data = df_mean_reduc,
    diagnosis ~ radius + texture + smoothness + concave.points,
    family = quasibinomial
  )
(summary(m_selected_quas))
```

```
##
## Call:
## glm(formula = diagnosis ~ radius + texture + smoothness + concave.points,
```

```
##     family = quasibinomial, data = df_mean_reduc)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.42132  -0.15010  -0.04247   0.02603   2.86598
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -28.57552    3.25917  -8.768  < 2e-16 ***
## radius           0.85081    0.11585   7.344 7.30e-13 ***
## texture          0.35845    0.04052   8.847  < 2e-16 ***
## smoothness      52.26403   17.65979   2.959  0.00321 **
## concave.points  78.73692   11.23385   7.009 6.88e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasibinomial family taken to be 0.458343)
##
##     Null deviance: 751.44  on 568  degrees of freedom
## Residual deviance: 160.32  on 564  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 8
```

The formula of the retained model is:

```
extract_eq(m_selected_quas,use_coefs = TRUE,wrap = TRUE,terms_per_line = 2)
```

$$\log\left[\frac{E(\widehat{\text{diagnosis}})}{1-E(\widehat{\text{diagnosis}})}\right] = -28.58 + 0.85(\text{radius}) + 0.36(\text{texture}) + 52.26(\text{smoothness}) + 78.74(\text{concave. points}) \quad (1)$$

The ratio of the residual deviance by its degrees of freedom is $160.32/564 = 0.284$ where the dispersion parameter is 0.458. These two values are close, we can validated this model.

See in annex the diagnostic plot of the retained model and an other GLM modelisation less conservative with glmnet.

IV. PCA

1. Variability explained by each PC

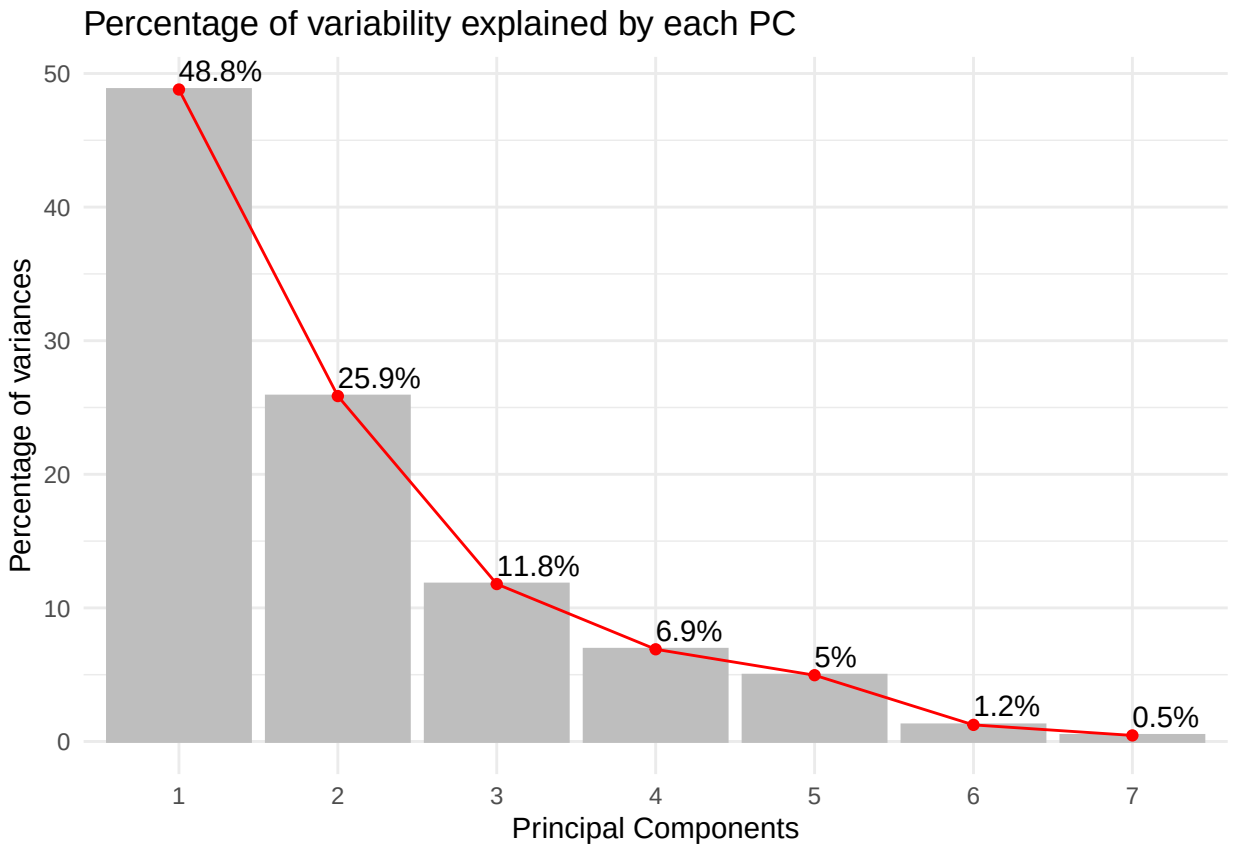
Still in the perspective of predicting the severity of the breast cancer, we will reduce the dimensionality of the data with the principal components analysis and model it in the space of most relevant principal components.

```
p <- prcomp(df_mean_reduc[,-1], scale=TRUE)
```

Let's see the percentage of variability explained by each principal components.

Graph

```
fviz_eig(p,addlabels = TRUE,geom = c("bar", "line"),barfill = "grey",
         barcolor = "grey",linecolor = "red",ncp = 10) +
    labs(title = "Percentage of variability explained by each PC",
          x = "Principal Components", y = "Percentage of variances") +
    theme_minimal()
```



The first four components explain more than 90% of the data.

Table

```
pander(summary(p))
```

Table 13: Principal Components Analysis (continued below)

	PC1	PC2	PC3	PC4	PC5
radius	-0.3589	0.5034	-0.2736	0.02538	0.02717
texture	-0.1731	0.3435	0.9012	-0.04728	0.1932
smoothness	-0.3858	-0.361	-0.1145	-0.2599	0.7786
concavity	-0.505	0.09141	-0.0368	-0.1644	-0.4256
concave.points	-0.5086	0.1908	-0.1671	-0.1031	-0.06186
symmetry	-0.3616	-0.3141	0.09945	0.8707	-0.01891
fractal_dimension	-0.219	-0.5957	0.2463	-0.3657	-0.4127

	PC6	PC7
radius	0.6285	-0.3828
texture	0.01509	0.005236
smoothness	-0.09876	-0.1553
concavity	-0.6008	-0.4076
concave.points	-0.01339	0.8139
symmetry	0.04041	-0.02235
fractal_dimension	0.4819	-0.01644

Table 15: Table continues below

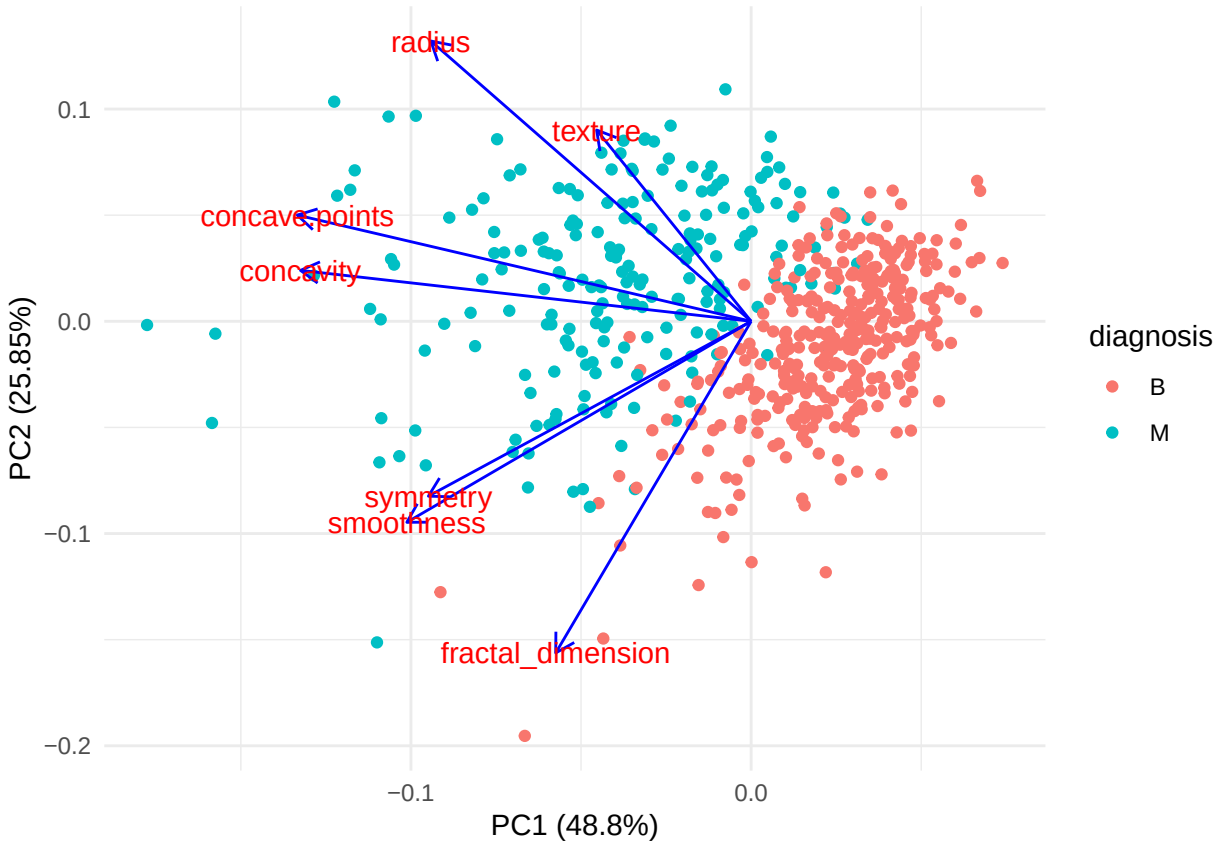
	PC1	PC2	PC3	PC4	PC5
Standard deviation	1.848	1.345	0.9083	0.695	0.5893
Proportion of Variance	0.488	0.2585	0.1178	0.06901	0.04962
Cumulative Proportion	0.488	0.7465	0.8644	0.9334	0.983

	PC6	PC7
Standard deviation	0.295	0.1784
Proportion of Variance	0.01243	0.00455
Cumulative Proportion	0.9954	1

The 4 first components explain more than 90% of the data.

2. Observations in PC plans

```
autoplot(p,x = 1,y = 2, data = df_mean_reduc,colour = "diagnosis",
  loadings = TRUE,loadings.colour = "blue",loadings.label = TRUE)+
  theme_minimal()
```



By plotting the data in the plan of the first 2 principal components we see a clear separation between benign and malignant type of cancer. That mean that knowing the location in this plan of a new observation should allow to predict the severity of the cancer. Let's try this approach by applying a GLM model to the first two PC.

Note:

We tried to plot the data in all the possible pairs formed by the first four components. The clearest separation between the type of cancer occurs in the plan PC1-PC2.

3. GLM model with PCA

Implementation

We want to predict 'diagnosis' only with PC1 and PC2 with the help of a GLM model. We try with the binomial family but as before there is an underdispersion issue. We use the quasibinomial family instead.

```
df_mean_pca <- cbind(df_mean_reduc, p$x)
glm_pca <-glm(data= df_mean_pca, df_mean_pca$diagnosis ~PC1+PC2,
              family = quasibinomial)
```

Validation

```
summary(glm_pca)
```

```
##
## Call:
## glm(formula = df_mean_pca$diagnosis ~ PC1 + PC2, family = quasibinomial,
##     data = df_mean_pca)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.74789  -0.16378  -0.04399   0.02463   3.13937
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.6441     0.1600  -4.026 6.44e-05 ***
## PC1          -3.1546     0.2727 -11.569  < 2e-16 ***
## PC2           2.4026     0.2384  10.077  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasibinomial family taken to be 0.6184117)
##
##     Null deviance: 751.44  on 568  degrees of freedom
## Residual deviance: 162.50  on 566  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 8
```

- The algorithm converge: the number of fisher scoring iterations is reasonable.
- The p-value of the two PC is significant.
- The ratio of the residual deviance by its degrees of freedom is $162.50/565 = 0.288$ where the dispersion parameter is 0.618. These two values are close, we can validated this model.

Conclusion

1.Features allowing to predict severity of cancer ?

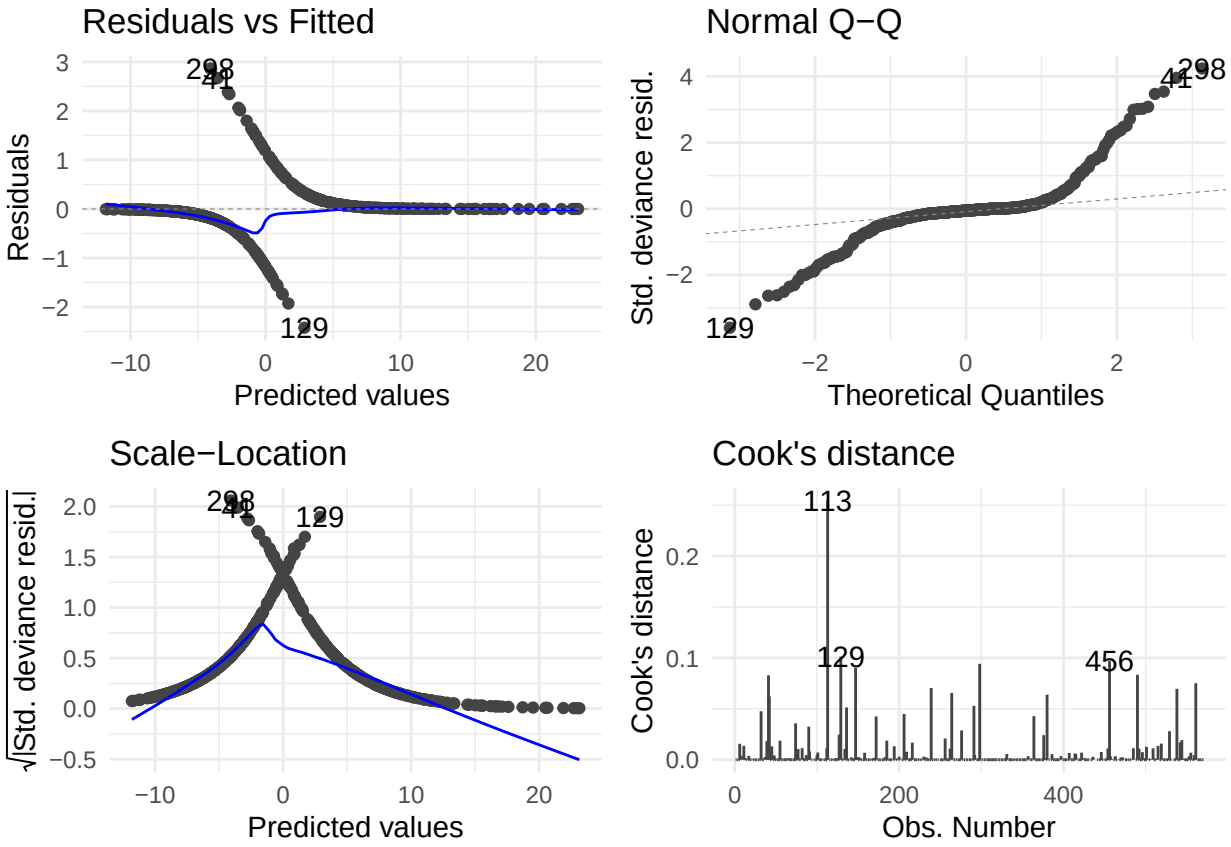
[...]

2. test accuracy of the prediction with these feature

Annex

1.diagnostic plot of retained glm model

```
autoplot(m_selected_quas,1:4)+ theme_minimal()
```



2. glmnet

First, let's separate the data into train and test set. (This will permit to be more accurate when we test the goodness of the model)

```
# set the seed to make partition reproducible
set.seed(123)

prop_train_test <- floor(0.75 * nrow(df_mean_reduc))

# 0.75 of the data fall randomly into the training set and the remaining is for the test set
train_ind <- sample(seq_len(nrow(df_mean_reduc)), size = prop_train_test)
train <- df_mean_reduc[train_ind, ]
test <- df_mean_reduc[-train_ind, ]
```

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

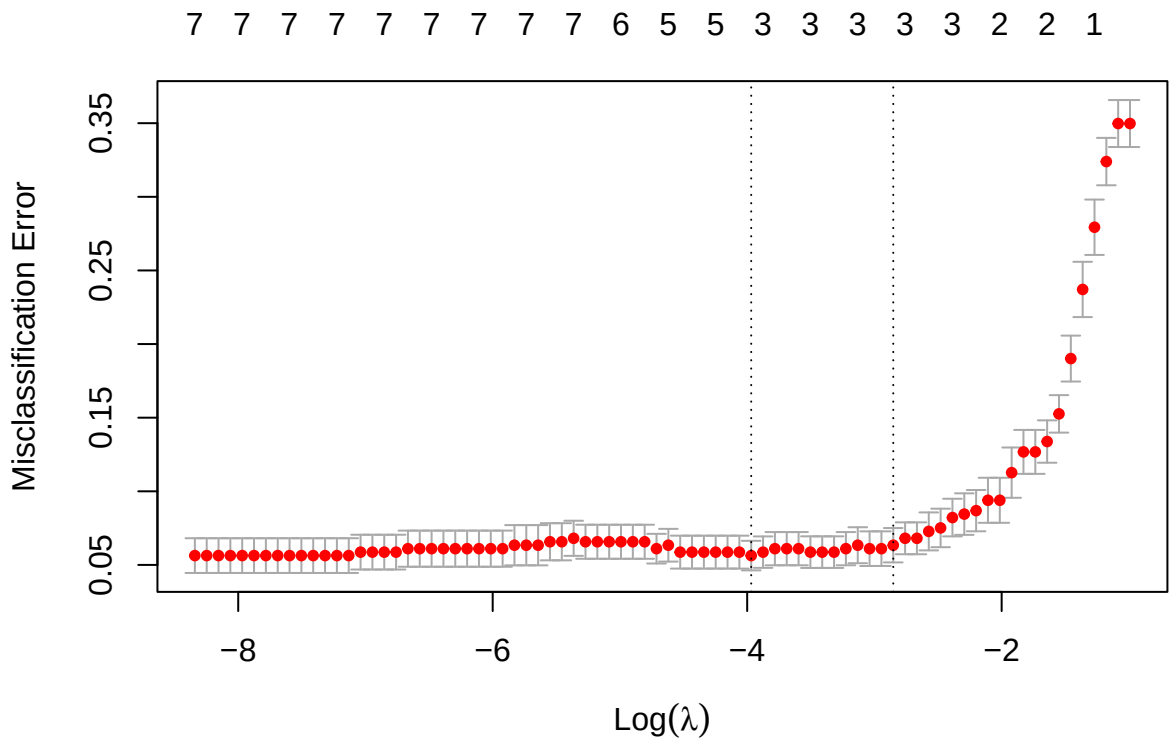
Apply GLM with cross validation

```
library("glmnet")
tol_length=length(levels(y_train))

cvfit<-cv.glmnet(as.matrix(x_train), y_train,family = "binomial",
                 type.measure="class")
```

Show the misclassification error according to the regularization hyperparameter λ.

```
plot(cvfit)
```



We will now see the for two particular values of this regularization hyperparameter the accurary of the model with the confusion matrix and which features are retained.

lambda.min

Let's choose this particular value of λ

```
pander(cvfit$lambda.min)
```

0.01891

The confusion matrix is:

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

	B	M
B	79	11
M	1	52

The accuracy is very good:

```
assess<-assess.glmnet(cvfit,newx=as.matrix(x_test),
                      newy=y_test, s='lambda.min')
pander(as.numeric(1-assess$class))
```

0.9161

Let's see the features retained by this model :

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

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

This model is less conservative than our previous GLM model. Indeed, the variable "smoothness" is not taken into account.

lambda.1se

Let's choose this particular value of λ:

```
pander(cvfit$lambda.1se)
```

0.05774

The confusion matrix is:

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

	B	M
B	80	13
M	0	50

The accuracy is very good:

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

0.9091

Let's see the features retained by this model :

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

```
## 8 x 1 sparse Matrix of class "dgCMatrix"
##                              s1
## (Intercept)       -6.91104517
## radius             0.20042878
## texture            0.07001007
## smoothness         .
## concavity          .
## concave.points    42.10066651
## symmetry           .
## fractal_dimension  .
```

Again, this model is less conservative than our previous GLM model.

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