MRR Project

Noah KWA MOUTOME - Victor TAN

2023-11-13

Introduction

This project is about the diagnosis of breast cancer. Thanks to the dataset "Breast Cancer Wisconsin (Diagnostic) Data Set" from the computer sciences department at the University of Wisconsin, we will try to predict if a tumor is benign or malignant.

Imports

```r
library("ggplot2")
library("MASS")
library("reshape2")
library("corrplot")
library("caret")
data <- read.csv('data.csv', header = TRUE)
```

Data exploration

Let's take a look at all the missing values of the dataset, if there's any, and clean the dataset:

```
##               diagnosis             radius_mean            texture_mean
##                       0                       0                       0
##          perimeter_mean               area_mean         smoothness_mean
##                       0                       0                       0
##        compactness_mean          concavity_mean     concave.points_mean
##                       0                       0                       0
##           symmetry_mean  fractal_dimension_mean               radius_se
##                       0                       0                       0
##              texture_se            perimeter_se                 area_se
##                       0                       0                       0
##           smoothness_se          compactness_se            concavity_se
##                       0                       0                       0
##       concave.points_se             symmetry_se    fractal_dimension_se
##                       0                       0                       0
##            radius_worst           texture_worst         perimeter_worst
##                       0                       0                       0
##              area_worst        smoothness_worst       compactness_worst
##                       0                       0                       0
##         concavity_worst    concave.points_worst          symmetry_worst
##                         0                         0                         0
## fractal_dimension_worst
##                         0
```

There's no missing values in this dataset. So now, we have 569 observations and 30 covariables to predict 1 target variable, which is the diagnosis.

Now, let's explain the covariables:

There are ten real-valued features computed for each cell nucleus:

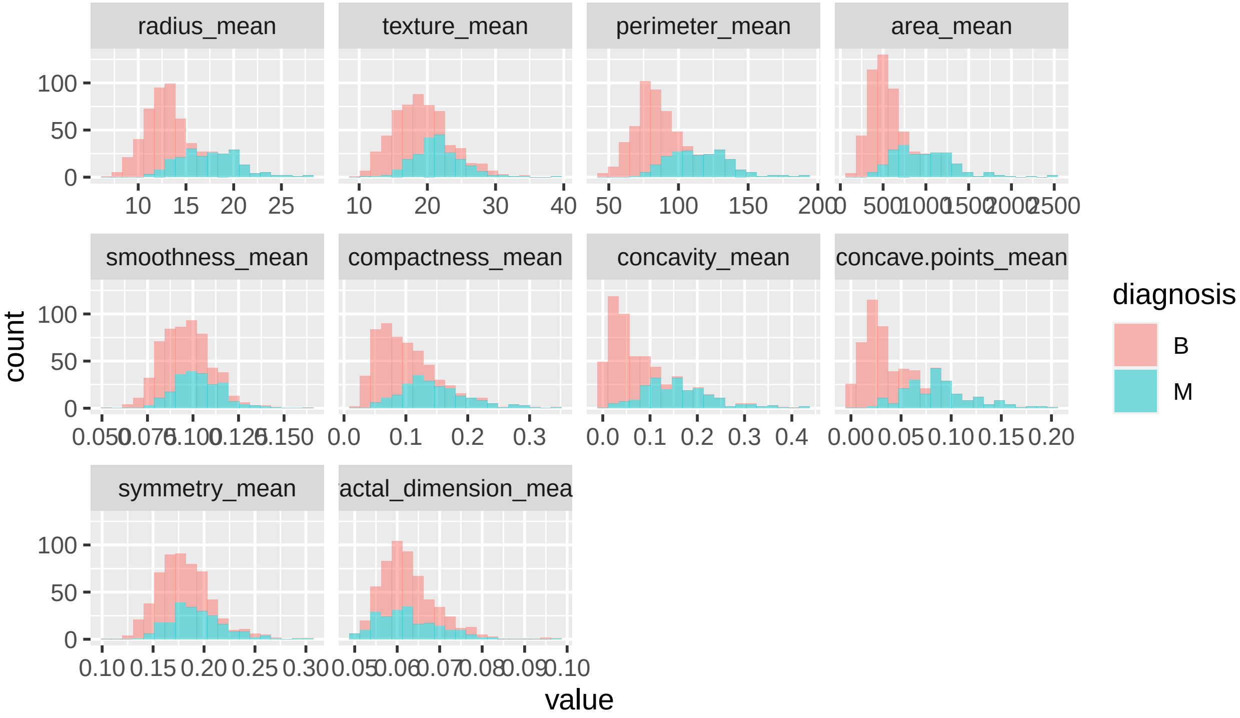
a) radius (mean of distances from center to points on the perimeter)
b) texture (standard deviation of gray-scale values)
c) perimeter
d) area
e) smoothness (local variation in radius lengths)
f) compactness (perimeterˆ2 / area - 1.0)
g) concavity (severity of concave portions of the contour)
h) concave points (number of concave portions of the contour)
i) symmetry
j) fractal dimension ("coastline approximation" - 1)

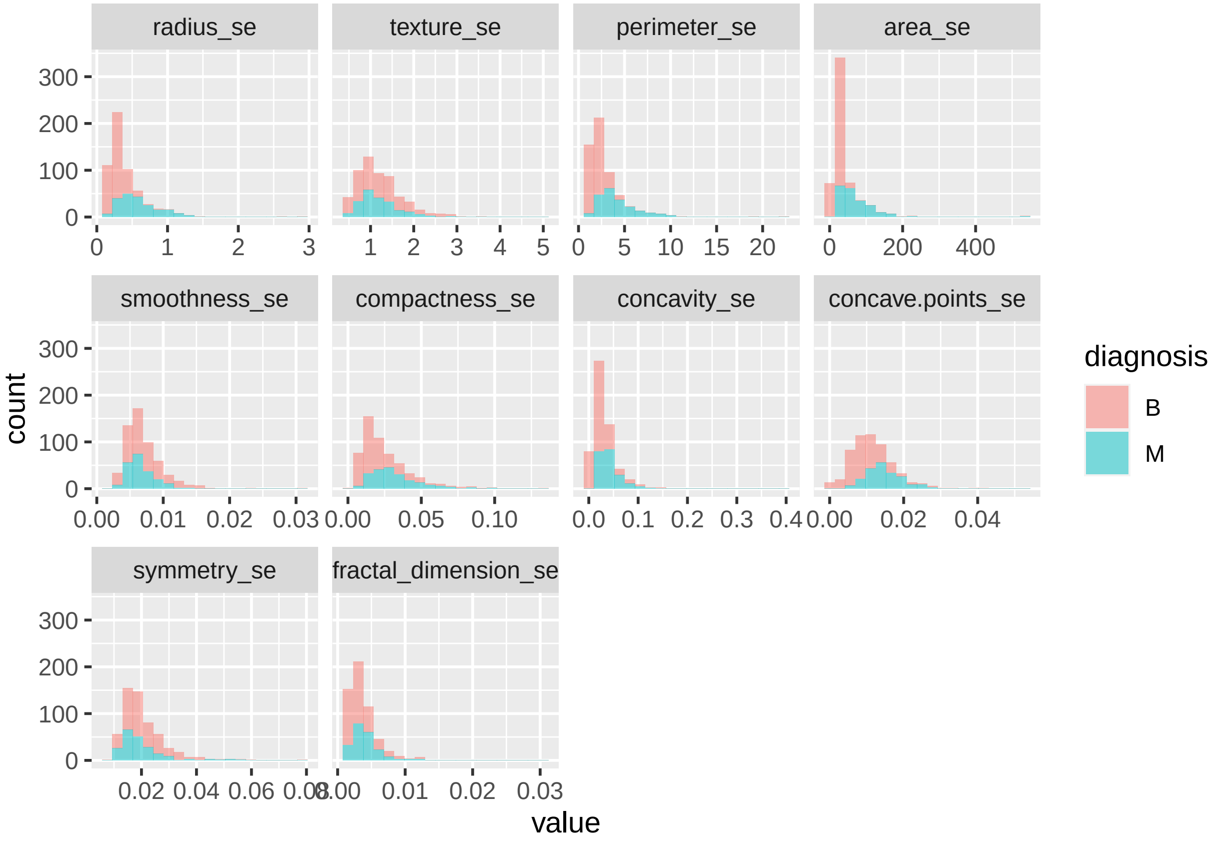
Then, the mean, the standard error and the "worst" or largest (mean of the three largest values) of these features were computed for each image, resulting in 30 features. For instance, field 3 is Mean Radius, field 13 is Radius SE, field 23 is Worst Radius.

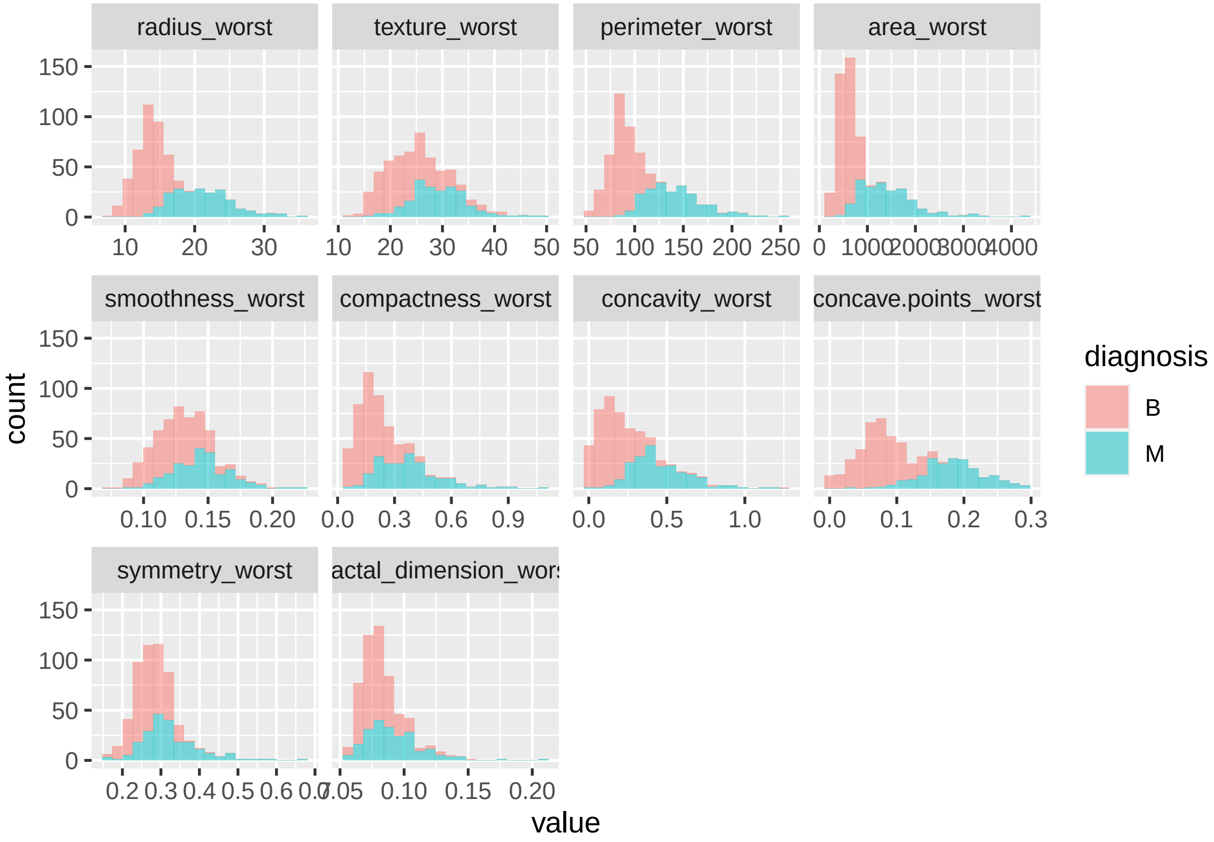
Description of the dataset

Data visualization

Let's now visualize the data to see if there's any outliers or if the data is normally distributed.

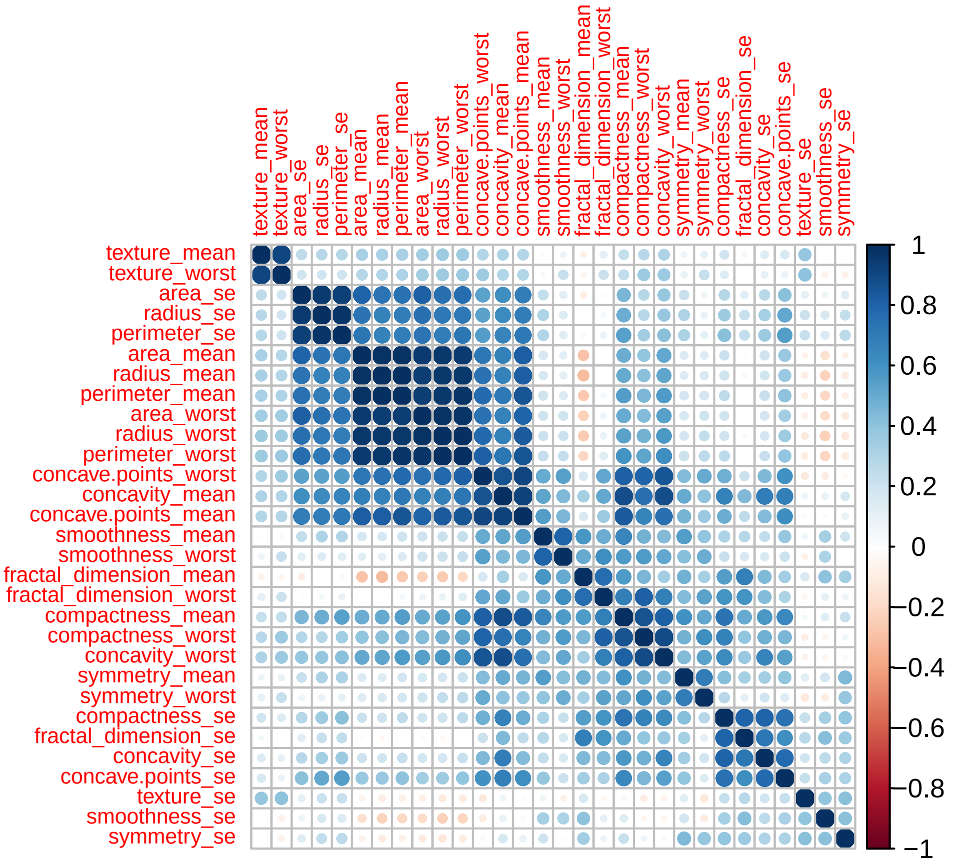

radius_mean
texture_mean
perimeter_mean
area_mean
smoothness_mean
compactness_mean
concavity_mean
concave.points_mean
symmetry_mean
actal_dimension_mea
count
100
50
0
10 15 20 25
10 20 30 40
50 100 150 200
0 500 1000 1500 2000 2500
0.050 0.075 0.100 0.125 0.150
0.0 0.1 0.2 0.3
0.0 0.1 0.2 0.3 0.4
0.00 0.05 0.10 0.15 0.20
0.10 0.15 0.20 0.25 0.30
0.05 0.06 0.07 0.08 0.09 0.10
value
diagnosis
B
M


radius_se
texture_se
perimeter_se
area_se
smoothness_se
compactness_se
concavity_se
concave.points_se
symmetry_se
fractal_dimension_se
count
value
0
100
200
300
0
1
2
3
4
5
10
15
20
400
0.00
0.01
0.02
0.03
0.05
0.10
0.0
0.1
0.2
0.3
0.4
0.04
0.06
0.08
diagnosis
B
M



It looks like most of the features are normally distributed, and there seems to be no outliers. However, there's no clear separation between benign and malignant for most of the features except "concave.points_worst", "perimeter_worst", "radius_worst" and "concave.points_mean". So a visualization of the values of these features is not enough.

Let's now look at the coorelation plot :



We can see that there's a strong correlation between some features, which can be a problem for the model.

We can now try 3 diffents approaches, and we will apply the same methods on each of them in the future:

- No covariables deletion
- Covariables deletion by highest correlation
- Covariables deletion by PCA

No covariables deletion

We still use the same dataset.

Covariables deletion by highest correlation

If two covariables are highly correlated, (for instance, if the correlation is higher than 0.8), we will delete the one that has the highest mean of correlations with the others. We can use the function "findCorrelation" from the package "caret" to provide us the said covariables that we will delete.

```
## Compare row 7  and column  8 with corr  0.921
##   Means:  0.571 vs 0.389 so flagging column 7
## Compare row 8  and column  6 with corr  0.831
##   Means:  0.542 vs 0.377 so flagging column 8
## Compare row 6  and column  28 with corr  0.816
##   Means:  0.524 vs 0.365 so flagging column 6
## Compare row 28  and column  27 with corr  0.855
##   Means:  0.507 vs 0.354 so flagging column 28
## Compare row 27  and column  26 with corr  0.892
##   Means:  0.457 vs 0.343 so flagging column 27
## Compare row 23  and column  21 with corr  0.994
##   Means:  0.456 vs 0.333 so flagging column 23
## Compare row 21  and column  3 with corr  0.969
##   Means:  0.422 vs 0.324 so flagging column 21
## Compare row 3  and column  24 with corr  0.942
##   Means:  0.384 vs 0.316 so flagging column 3
## Compare row 26  and column  30 with corr  0.81
##   Means:  0.4 vs 0.313 so flagging column 26
## Compare row 24  and column  1 with corr  0.941
##   Means:  0.356 vs 0.302 so flagging column 24
## Compare row 1  and column  4 with corr  0.987
##   Means:  0.308 vs 0.298 so flagging column 1
## Compare row 4  and column  14 with corr  0.8
##   Means:  0.27 vs 0.297 so flagging column 14
## Compare row 13  and column  11 with corr  0.973
##   Means:  0.316 vs 0.297 so flagging column 13
## Compare row 16  and column  17 with corr  0.801
##   Means:  0.413 vs 0.291 so flagging column 16
## Compare row 5  and column  25 with corr  0.805
##   Means:  0.326 vs 0.278 so flagging column 5
## Compare row 22  and column  2 with corr  0.912
##   Means:  0.212 vs 0.278 so flagging column 2
## All correlations <= 0.8
```

```
##  [1] "compactness_mean"     "concavity_mean"       "smoothness_mean"
##  [4] "concavity_worst"      "compactness_worst"    "texture_worst"
##  [7] "fractal_dimension_se" "texture_mean"         "smoothness_worst"
## [10] "perimeter_worst"      "diagnosis"            "texture_se"
## [13] "smoothness_se"        "perimeter_se"         "area_mean"
## [16] "radius_mean"
```

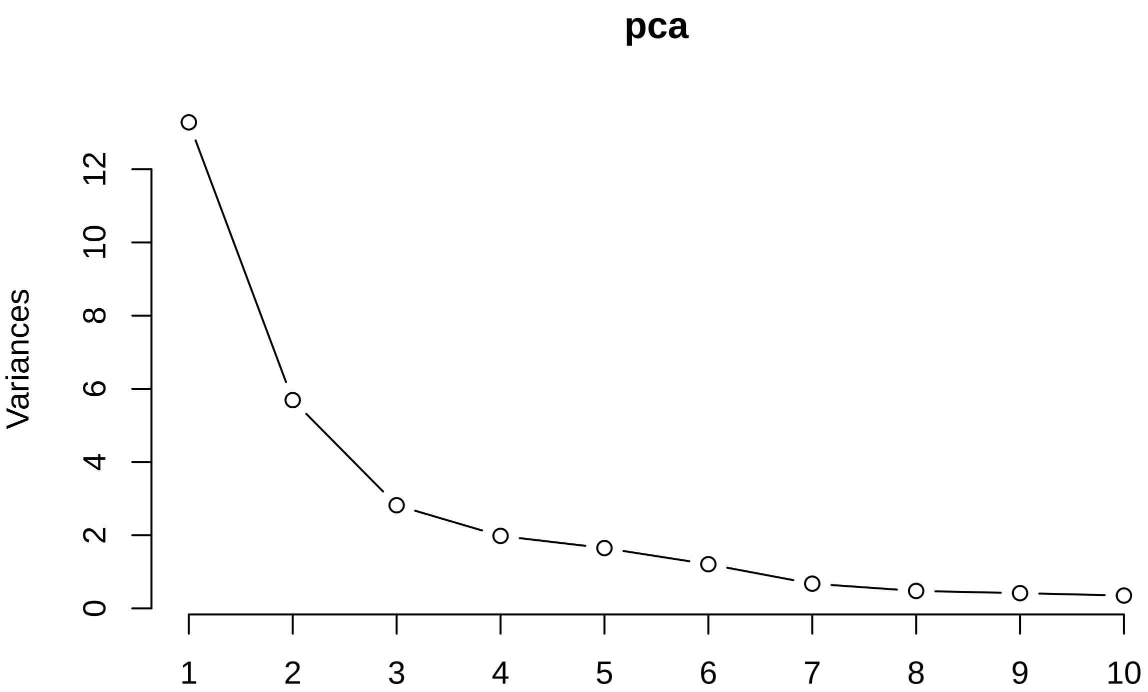
Then, we only keep the covariables that are not in the list “corr_var”.

```
##  [1] "diagnosis"               "perimeter_mean"
##  [3] "concave.points_mean"     "symmetry_mean"
##  [5] "fractal_dimension_mean"  "radius_se"
##  [7] "area_se"                 "compactness_se"
##  [9] "concavity_se"            "concave.points_se"
## [11] "symmetry_se"             "radius_worst"
## [13] "area_worst"              "concave.points_worst"
## [15] "symmetry_worst"          "fractal_dimension_worst"
```

We now have a new dataset with less covariables, only 16. However, this approch is not perfect, because we only scan the correlation between two covariables once: there could still be high correlations between variables if 3 or more covariables were correlated together, and deleting more covariables could reduce to much information.

Covariables deletion by PCA

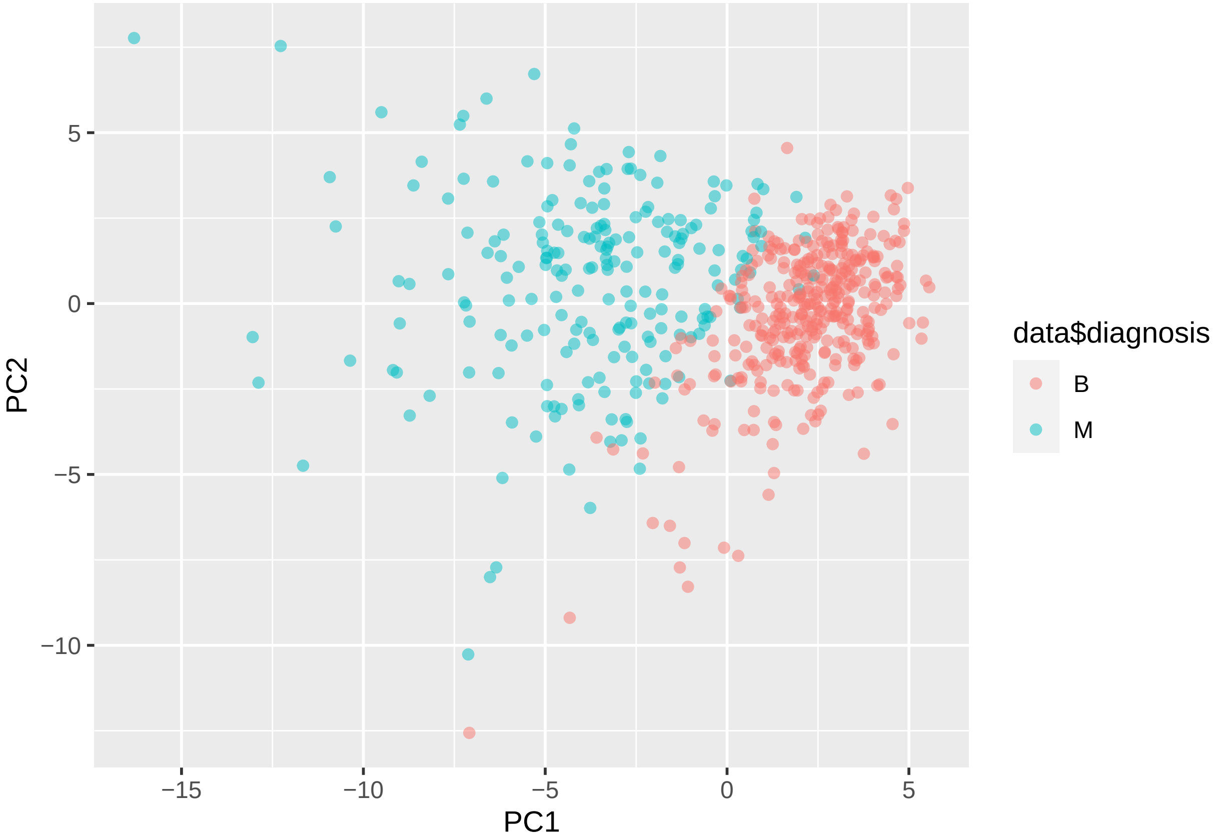
We will now use the PCA method to reduce the number of covariables. We will use the function “prcomp” from the package “stats” to do so. Because each covariable has a different scale, we will use the parameters “center = TRUE” and “scale. = TRUE” to standardize the covariables.



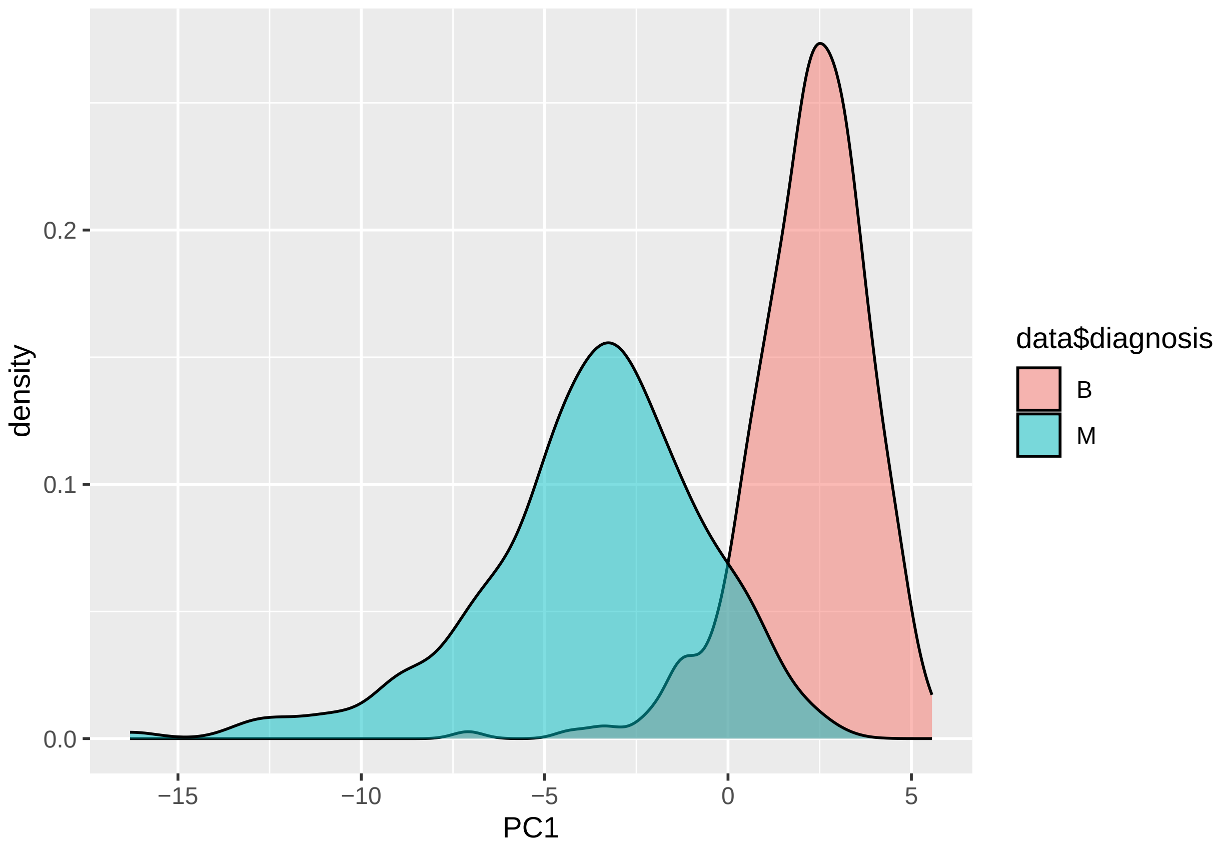
```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     3.6444 2.3857 1.67867 1.40735 1.28403 1.09880 0.82172
## Proportion of Variance 0.4427 0.1897 0.09393 0.06602 0.05496 0.04025 0.02251
## Cumulative Proportion  0.4427 0.6324 0.72636 0.79239 0.84734 0.88759 0.91010
##                            PC8    PC9    PC10   PC11    PC12    PC13    PC14
## Standard deviation     0.69037 0.6457 0.59219 0.5421 0.51104 0.49128 0.39624
## Proportion of Variance 0.01589 0.0139 0.01169 0.0098 0.00871 0.00805 0.00523
## Cumulative Proportion  0.92598 0.9399 0.95157 0.9614 0.97007 0.97812 0.98335
##                           PC15    PC16    PC17    PC18    PC19    PC20   PC21
## Standard deviation     0.30681 0.28260 0.24372 0.22939 0.22244 0.17652 0.1731
## Proportion of Variance 0.00314 0.00266 0.00198 0.00175 0.00165 0.00104 0.0010
## Cumulative Proportion  0.98649 0.98915 0.99113 0.99288 0.99453 0.99557 0.9966
##                           PC22    PC23   PC24    PC25    PC26    PC27    PC28
## Standard deviation     0.16565 0.15602 0.1344 0.12442 0.09043 0.08307 0.03987
## Proportion of Variance 0.00091 0.00081 0.0006 0.00052 0.00027 0.00023 0.00005
## Cumulative Proportion  0.99749 0.99830 0.9989 0.99942 0.99969 0.99992 0.99997
##                           PC29    PC30
## Standard deviation     0.02736 0.01153
## Proportion of Variance 0.00002 0.00000
## Cumulative Proportion  1.00000 1.00000
```

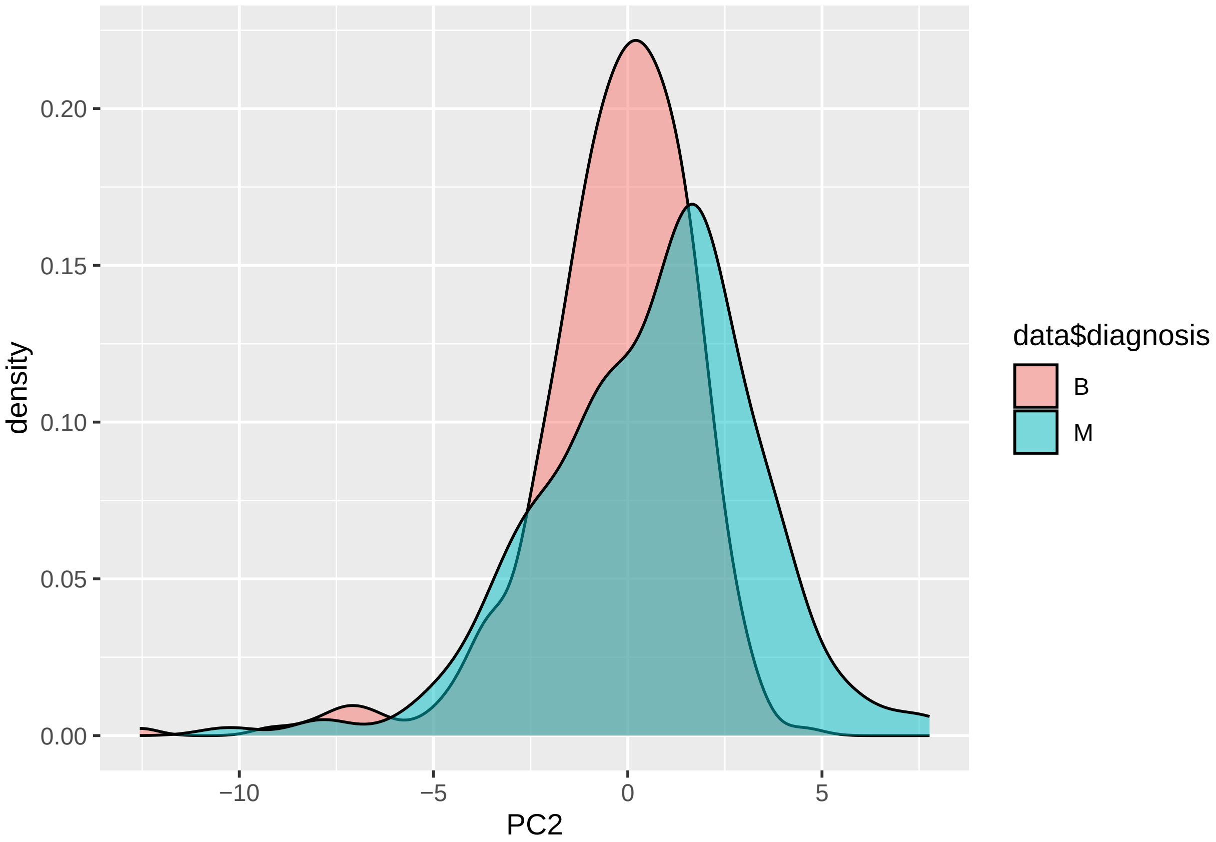
Let’s arbitrarily choose to keep enough covariables to explain at least 99% of the variance. According to the summary, we need to keep 17 covariables to do so (99.11% of the variance explained).

Though, 17 covariables means 17 dimensions, which is too much to visualize. So we can also choose to keep only 2 covariables, which will allow us to visualize the data. Together, they explain 63.24% of the variance.



Visually, we can see that there's a clear separation between benign and malignant tumors, but only thanks to the first principal component. Let's draw their respective distributions:


density
0.2
0.1
0.0
−15
−10
−5
0
5
PC1
data$diagnosis
B
M



With the first PC1, we can see that the means of the distributions are far enough from each other to be able to separate them. However, with the second PC2, the means are too close to each other, so we can't separate them with this principal component. And given that the following principal components explain even less variance, we can't use them to separate the distributions either.