MRR Project

Noah KWA MOUTOME - Victor TAN

2023-11-13

Introduction

Nowadays, in the realm of healthcare and data analytic, diagnosis of breast cancer has become a crucial task: In 2020, 685 000 deaths due to breast cancer were recorded according to the World Health Organization. Though, roughly half of breast cancers occur in women without any specific risk factors other than sex and age. It means that the diagnosis and prediction of malignant tumors is of the utmost importance, as it enables cancer patients to be treated as soon and effectively as possible or even prevent risk. This project is about the diagnosis and prediction 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 develop predictive models capable to predict if a tumor is benign or malignant given pictures of tumors' cells.

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

Preprocessing and Data Exploration

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

```
## [1] "Number of missing values:  0"
```

There's no missing values in this dataset, but we have to drop the last unnamed column and the "id" one which is not usefull. 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:

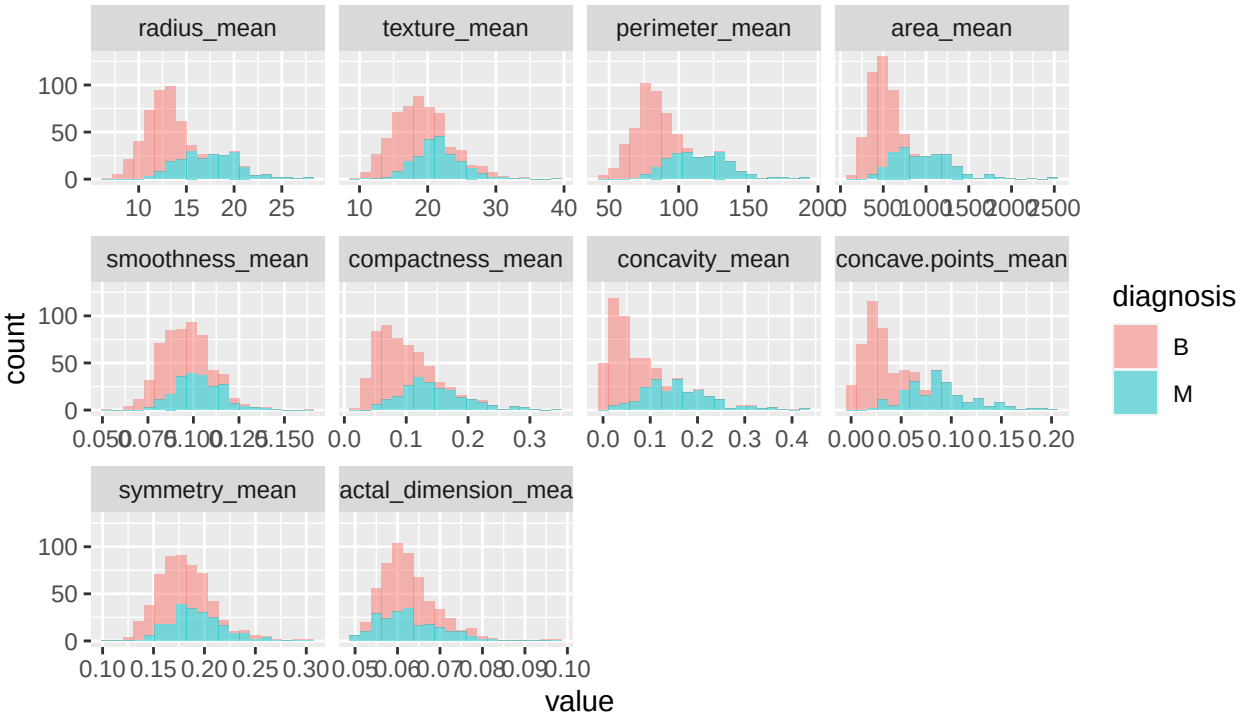
a) radius (mean of distances from center to points on the perimeter)
b) texture (standard deviation of gray-scale values)
c) perimeter
d) area
e) smoothness (local variation in radius lengths)
f) compactness (perimeter^2 / area - 1.0)
g) concavity (severity of concave portions of the contour)
h) concave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

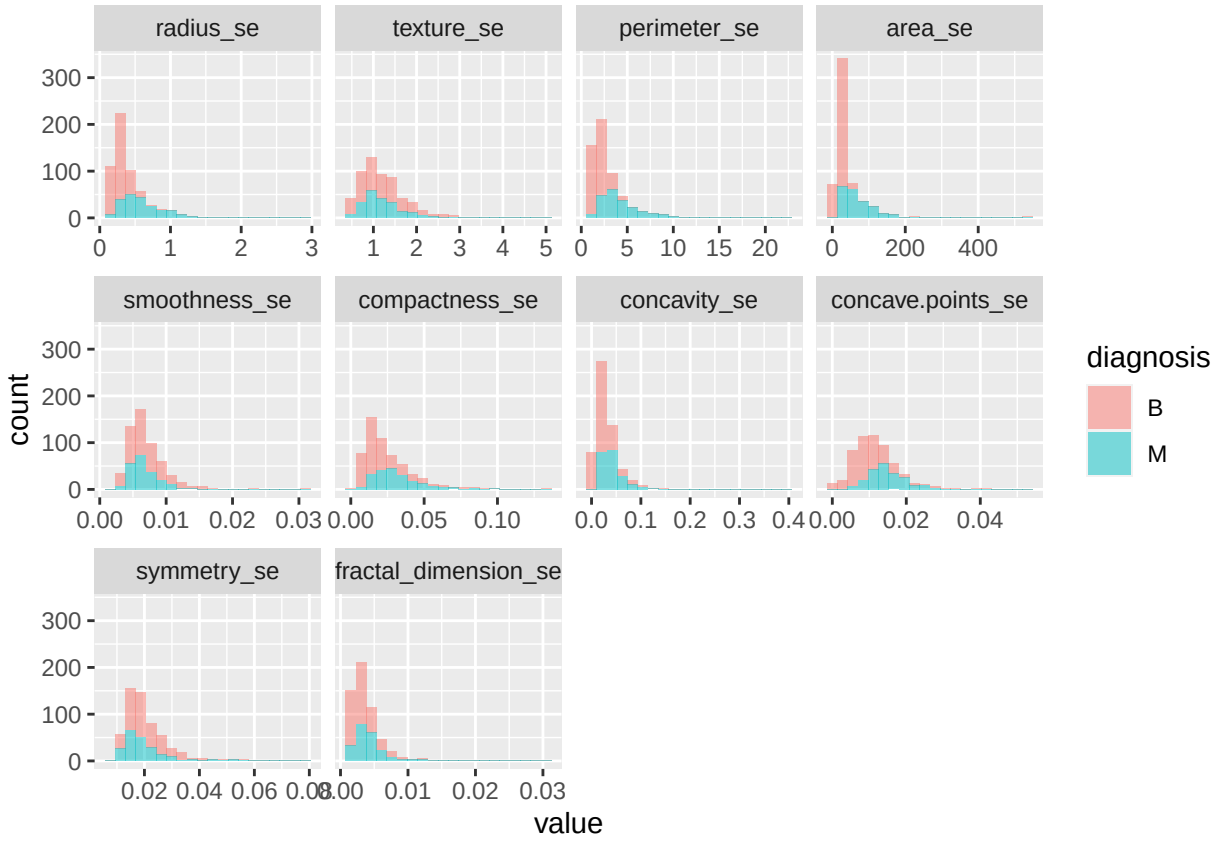
Then, the mean, the standard error and the "worst" or largest (mean of the three largest values) of these features were computed for each image, resulting in 30 features. For instance, field 3 is Mean Radius, field 13 is Radius SE, field 23 is Worst Radius. Furthermore, there are 212 Malignant diagnoses and 357 Benign diagnoses, which is a relatively balanced split: we do have enough data from each diagnoses to study the dataset correctly.

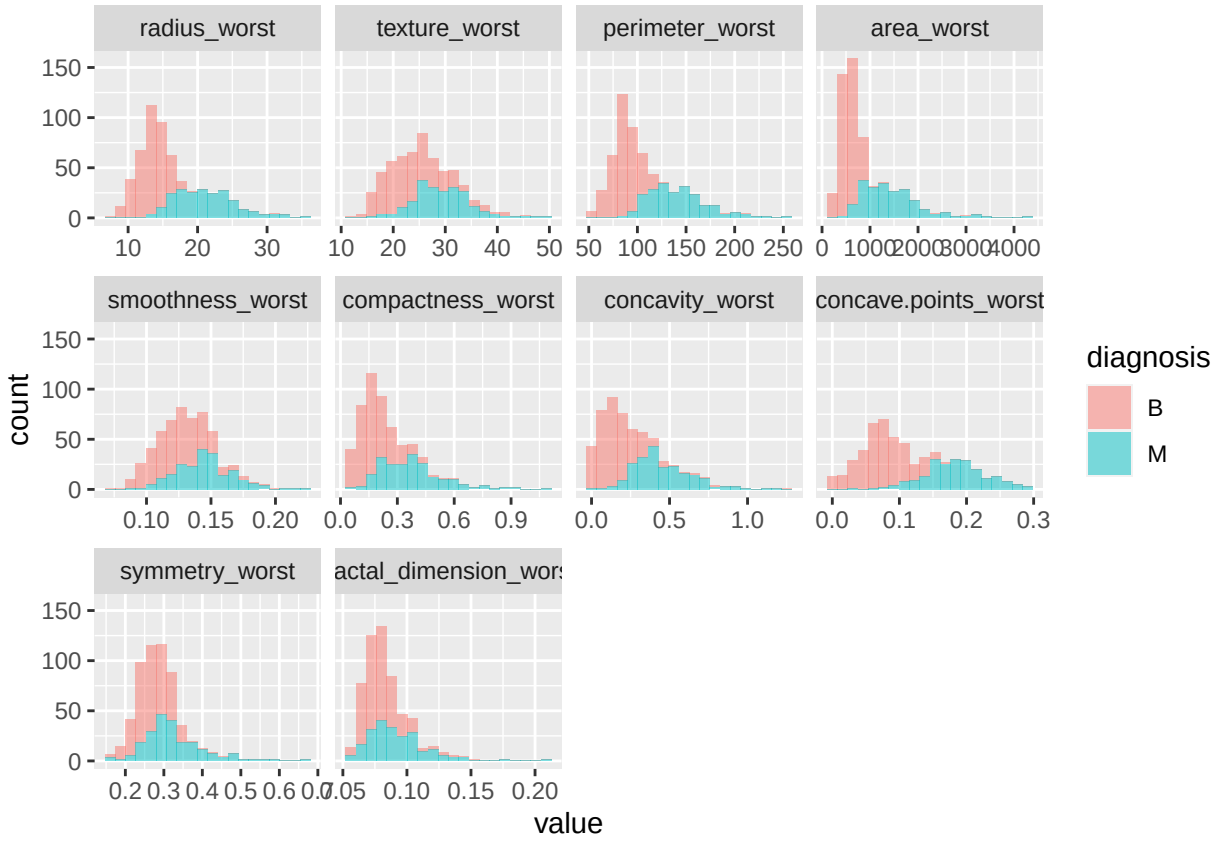
Description of the dataset

Data visualization

Let's now visualize the data to see if there's any outliers or if the data is normally distributed. These histograms are repetitive, so we will only show the means of the 10 features.

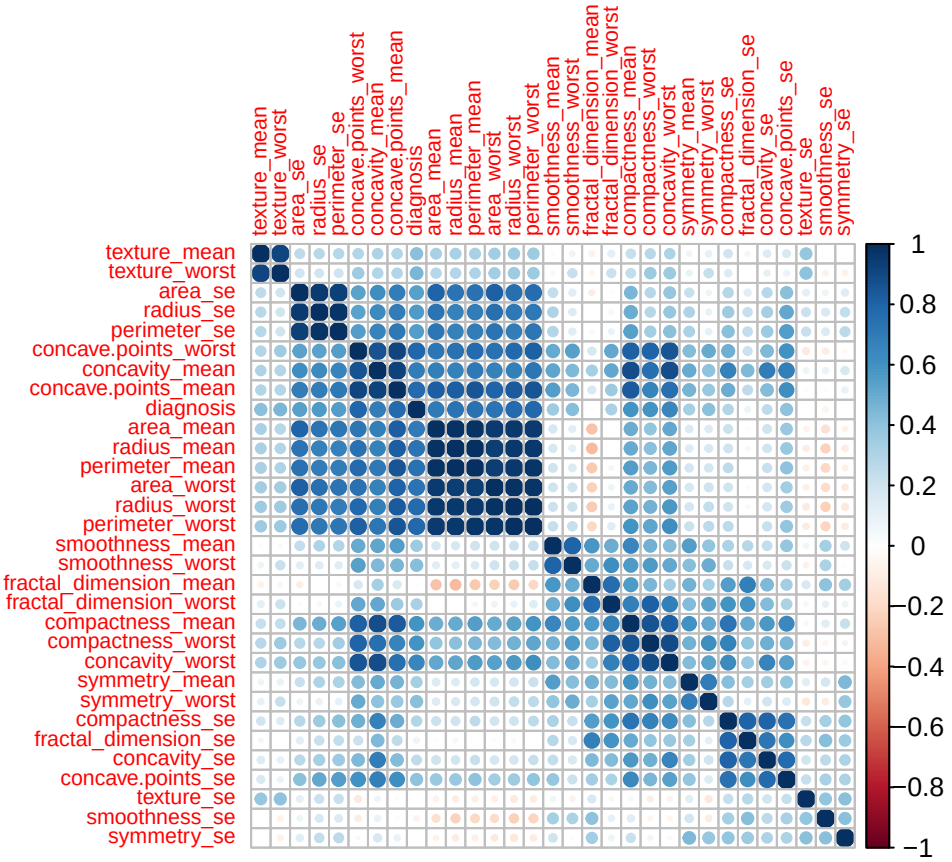



radius_se
texture_se
perimeter_se
area_se
smoothness_se
compactness_se
concavity_se
concave.points_se
symmetry_se
fractal_dimension_se
count
value
diagnosis
B
M



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.

However, there is a significant correlation between diagnosis and the following variables: diagnosis, radius mean, perimeter mean, area mean, concave.points mean, radius worst, perimeter worst, area worst, concave.points worst. That could mean that these co-variables may be important for the prediction of the target variable.

We will try 3 approaches to deal with the strong correlations between the co-variables: - No co-variables deletion (using Ridge regression to simplify the model) - Backward Selection - Co-variables deletion by highest correlation - Co-variables deletion by PCA

No covariables deletion

We still use the same dataset, and we will use the Ridge method. We will use the best lambda which minimize the MRSE.

Backward selection

We will now use the backward selection method to delete the covariables that are not useful for the model with the function "step". With this method, the co-variables left are:

```
##
## Call:  glm(formula = diagnosis ~ radius_mean + texture_mean + area_mean +
##     smoothness_mean + compactness_mean + concavity_mean + concave.points_mean +
##     symmetry_mean + fractal_dimension_mean + perimeter_se + area_se +
##     smoothness_se + compactness_se + concavity_se + concave.points_se +
##     symmetry_se + fractal_dimension_se + radius_worst + texture_worst +
##     perimeter_worst + area_worst + concavity_worst + symmetry_worst +
##     fractal_dimension_worst, family = "binomial", data = data_back)
```

```
##
## Coefficients:
##             (Intercept)             radius_mean            texture_mean
##              -5.914e+03              -6.630e+03               1.913e+02
##               area_mean         smoothness_mean        compactness_mean
##               6.077e+01               3.914e+04              -8.621e+04
##          concavity_mean     concave.points_mean           symmetry_mean
##               2.852e+04               5.886e+04              -1.964e+04
##  fractal_dimension_mean            perimeter_se                 area_se
##               1.626e+05              -1.253e+03               1.562e+02
##           smoothness_se          compactness_se            concavity_se
##              -9.793e+04               9.217e+04              -8.131e+04
##       concave.points_se             symmetry_se    fractal_dimension_se
##               4.398e+05              -1.038e+05              -1.092e+06
##            radius_worst           texture_worst         perimeter_worst
##               2.226e+03               7.269e+01               1.267e+02
##              area_worst         concavity_worst          symmetry_worst
##              -1.626e+01               6.737e+03               2.201e+04
## fractal_dimension_worst
##               5.899e+04
##
## Degrees of Freedom: 568 Total (i.e. Null);  544 Residual
## Null Deviance:       751.4
## Residual Deviance: 0.0001671     AIC: 50
```

So we get a total of 24 co-variables, and the step method only deleted 6 co-variables, which isn't really effective.

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.

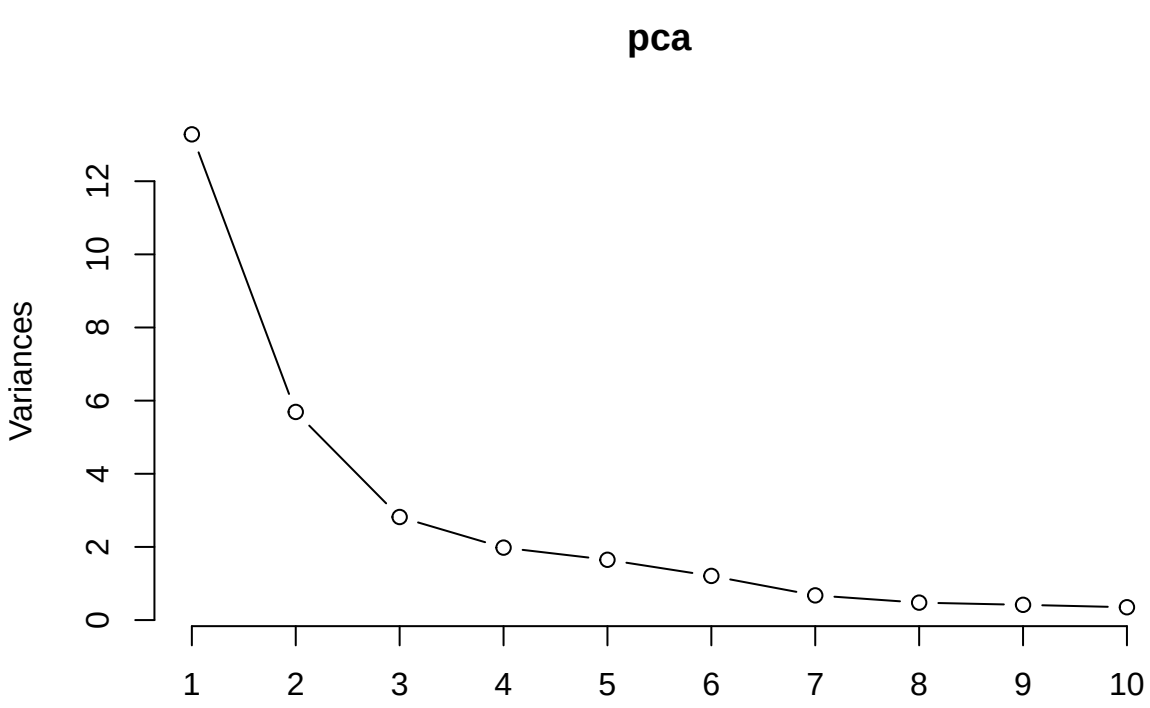
Then, we only keep the covariables that are not in the list "corr_var", which gives us:

```
##  [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 15, which is half of the initial variables. 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 too much information.

Covariables deletion by PCA

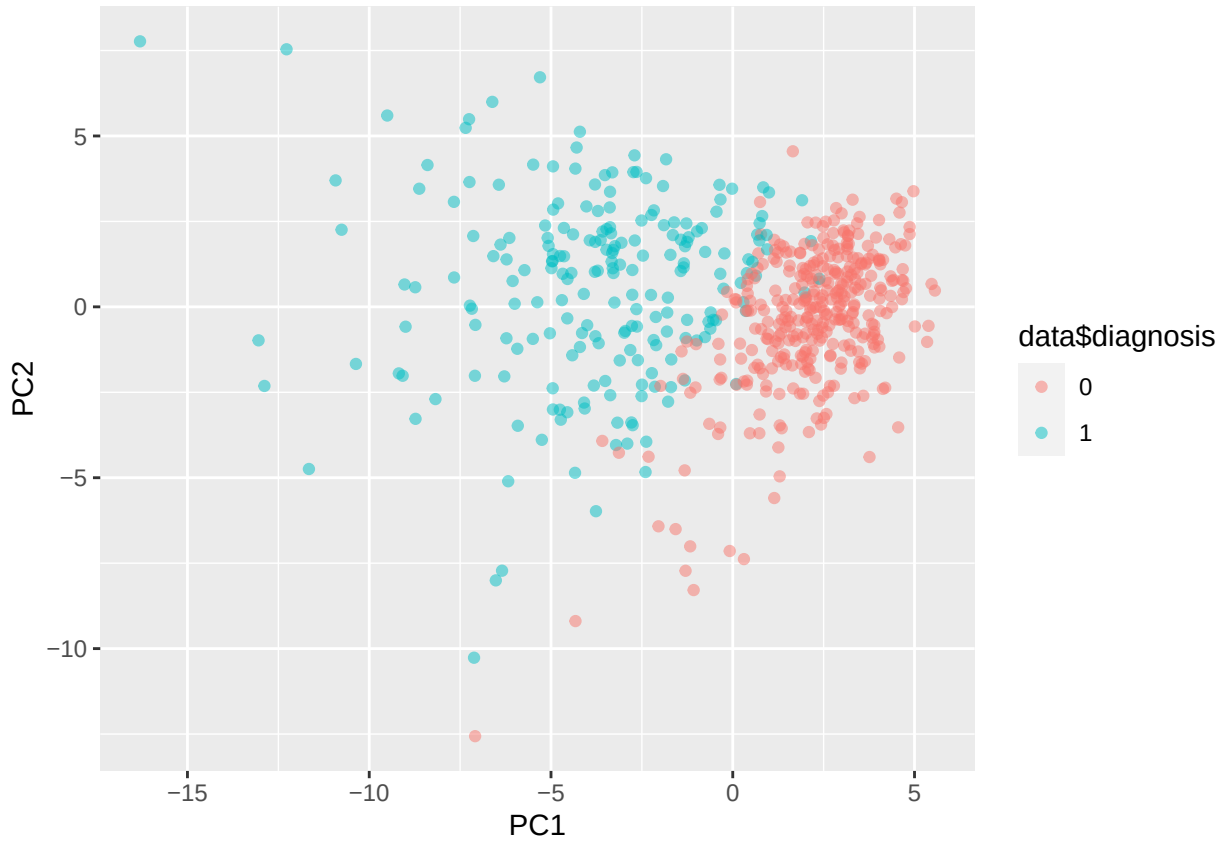
We will now use the PCA method to reduce the number of covariables. We will use the function "prcomp" from the package "stats" to do so. Because each covariable has a different scale, we will use the parameters "center = TRUE" and "scale. = TRUE" to standardize the covariables.



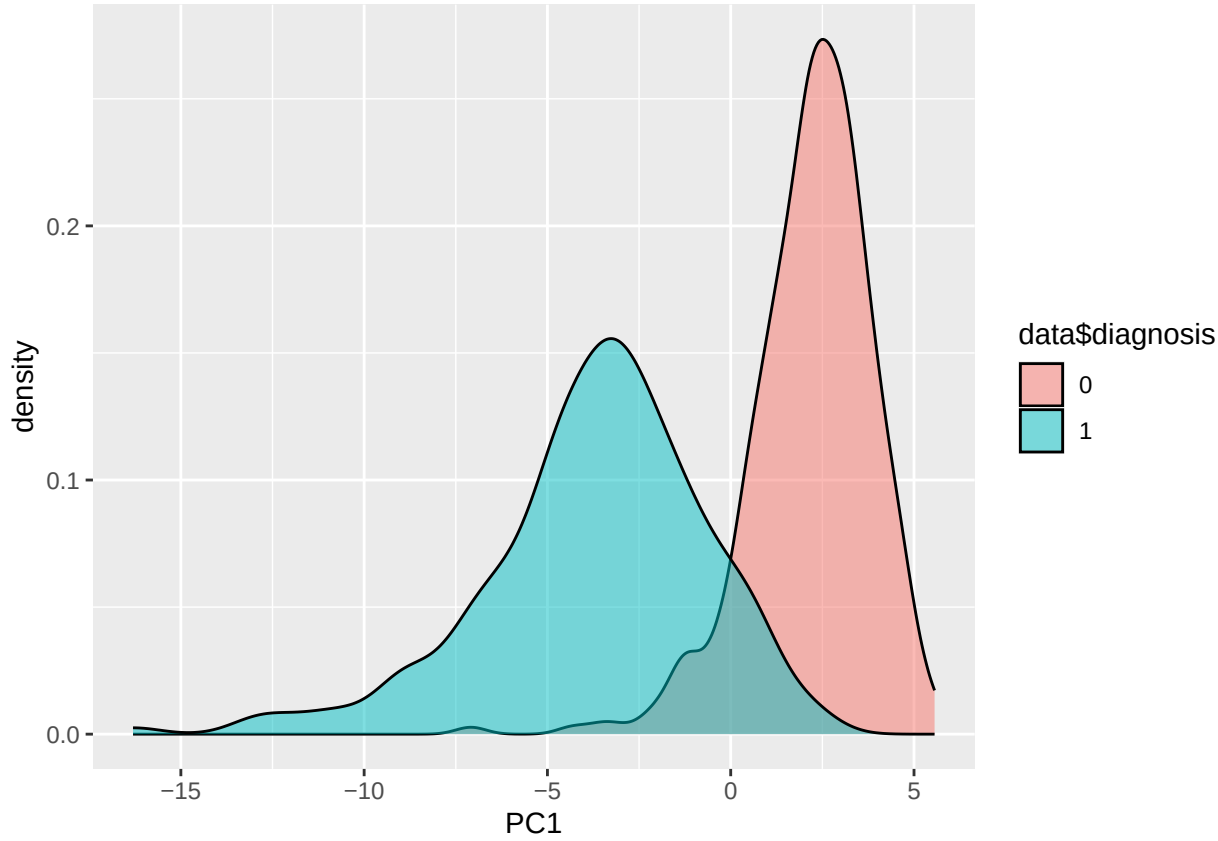

```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     3.6444 2.3857 1.67867 1.40735 1.28403 1.09880 0.82172
## Proportion of Variance 0.4427 0.1897 0.09393 0.06602 0.05496 0.04025 0.02251
## Cumulative Proportion  0.4427 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
```

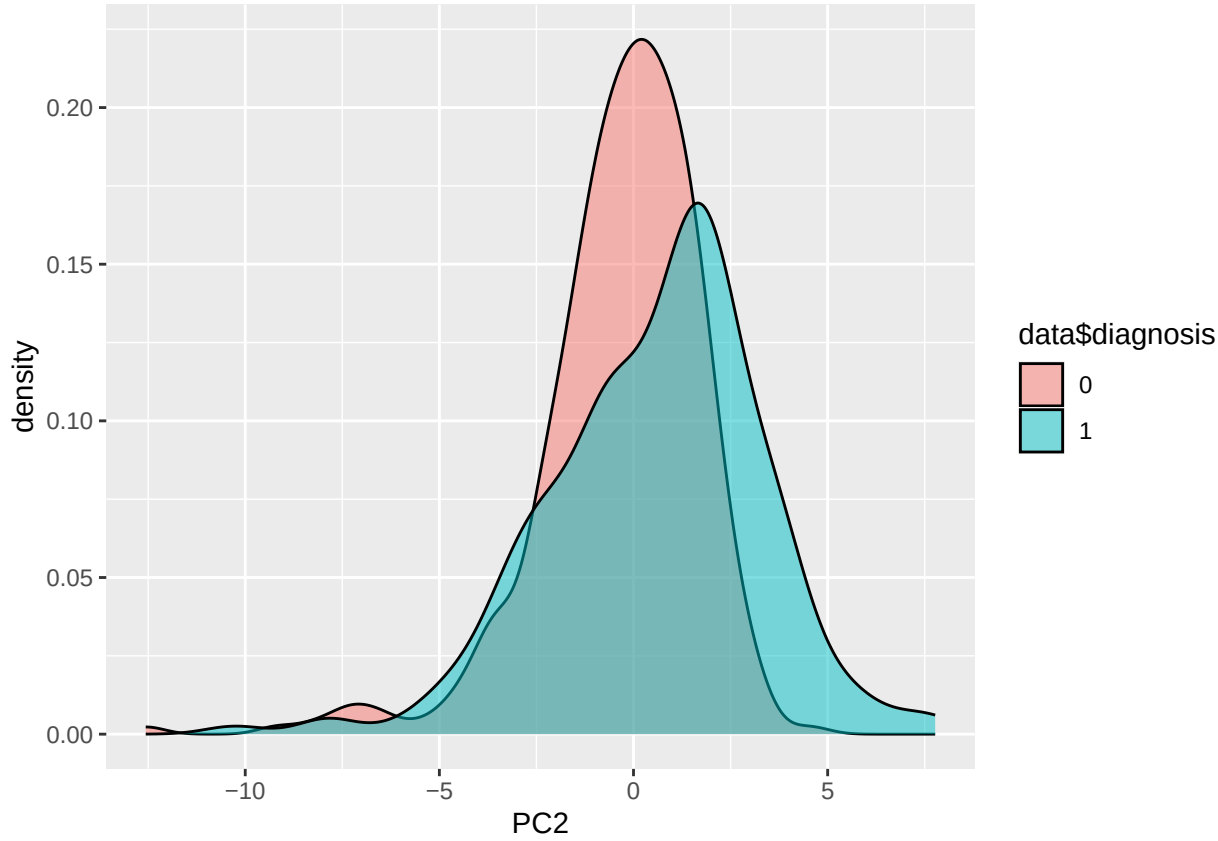
Let's arbitrarily choose to keep enough covariables to explain at least 99% of the variance. According to the summary, we need to keep 17 covariables to do so (99.11% of the variance explained).

Though, 17 covariables means 17 dimensions, which is too much to visualize. So we can temporarily choose to keep 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: If we guess the center of the two clusters of diagnosis, we can see that their PC1 values are far enough from each other to be able to separate them (around -3 for the malignant tumors and 2.5 for the benign ones), but their PC2 values are too close to each other (around 0), so we can't separate them with this principal component. Let's draw only the first distribution on PC1 as the PC2 one is not useful (the reader can still see the PC2 distribution in the code):


density
0.2
0.1
0.0
−15
−10
−5
0
5
PC1
data$diagnosis
0
1

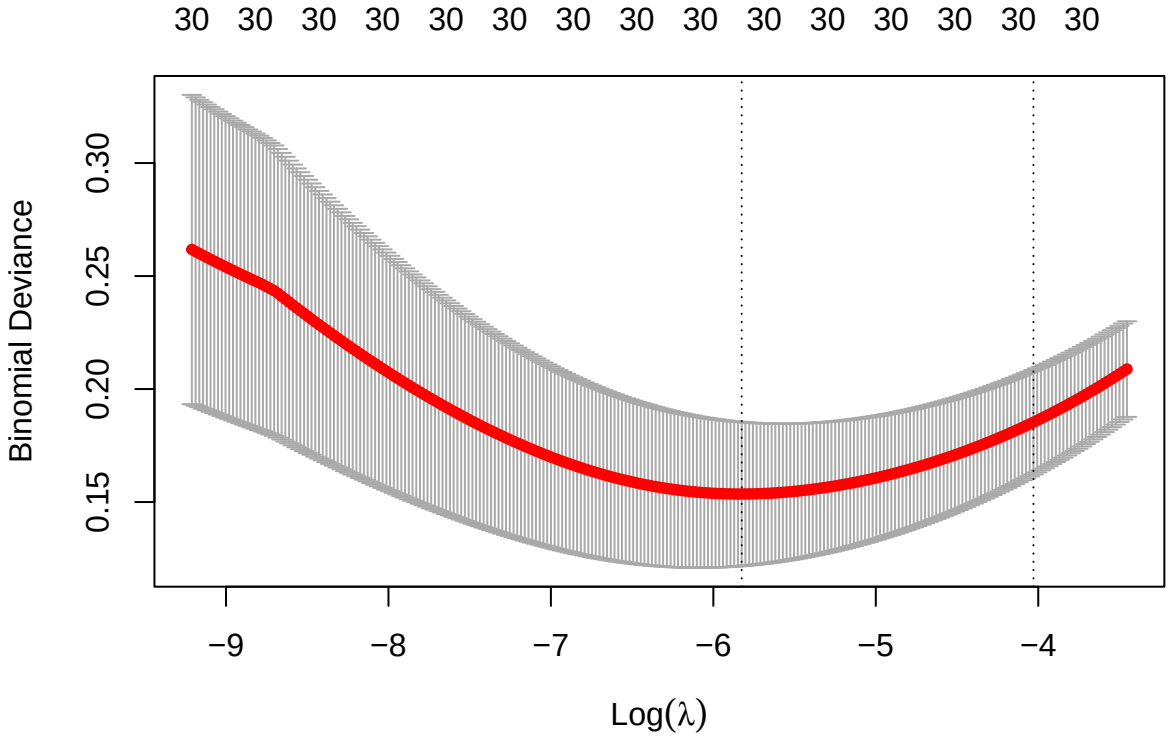


With the first PC1, we can see that the means of the distributions are far enough from each other to be able to separate them. But the PC1 only explains 44.27% of the variance, so it doesn't really give us a lot of information. Let's keep the 17 covariables for the next steps.

Model training and prediction

We will use a k-fold method to build a model training with the 4 approaches, and predict the diagnosis. We will use K = 5, meaning that 20% of the 569 observations will compose the test set and 80% will compose the training set.

Ridge on total dataset




```
##                   predictions_ridge
## true_values_ridge   0   1
##                 0 352   5
##                 1   8 204
```

```
## [1] "Accuracy:  0.98"
```

Covariables deletion by Backward selection

```
##                  predictions_back
## true_values_back   0   1
##                0 344  13
##                1  11 201
```

```
## [1] "Accuracy:  0.96"
```

Covariables deletion by highest correlation

```
##                   predictions_clear
## true_values_clear   0   1
##                 0 345  12
##                 1  15 197
```

```
## [1] "Accuracy:  0.95"
```

Covariables deletion by PCA

```
##                predictions_pca
## true_values_pca   0   1
##               0 343  14
##               1  10 202
```

```
## [1] "Accuracy:  0.96"
```

We can see that the Ridge method is the most accurate, followed by PCA by only a slight percentage. Moreover, the Ridge method gives the lowest False Negative, which is the only value that we want to minimize: False positives (when the prediction gives "Malignant" when the true value is "Benign") are not really a issue. But False negatives (when the prediction gives "Benign" when the true value is "Malignant") are the worst case scenario, as the patient might be in danger without our knowledge.

Conclusion

The results of our models obtained from the breast cancer classification project seem to be really accurate overall. The highest accuracy, Sensitivity and Specificity of the prediction on 569 observations is achieved by the Ridge Model. It may hold promise for supporting clinical decision-making of the diagnosis of tumors given images. On a broader scale, we can then decide to make another type of model, where we can make a greater penalization on False negatives, which are the cases we want to avoid. It will probably reduce the accuracy of the current model, but would be more helpful in our scenarios.