Breast Cancer Wisconsin (Diagnostic)

About this 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. n the 3-dimensional space is that described in: [K. P. Bennett and O. L. Mangasarian: "Robust Linear Programming Discrimination of Two Linearly Inseparable Sets", Optimization Methods and Software 1, 1992, 23-34].

Attribute Information: 1) ID number 2) Diagnosis (M = malignant, B = benign) 3-32) Ten real-valued features are 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) 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. For instance, field 3 is Mean Radius, field 13 is Radius SE, field 23 is Worst Radius. All feature values are recoded with four significant digits.

Missing attribute values: none

Class distribution: 357 Benign, 212 Malignant

```
getwd()
```

```
## [1] "C:/Users/badal/Desktop/AEON/R use cases"
```

```
data <- read.csv('file:///C:/Users/badal/Desktop/datset_/data.csv')
head(data)
```

```
##         id diagnosis radius_mean texture_mean perimeter_mean area_mean
## 1   842302         M       17.99        10.38         122.80    1001.0
## 2   842517         M       20.57        17.77         132.90    1326.0
## 3 84300903         M       19.69        21.25         130.00    1203.0
## 4 84348301         M       11.42        20.38          77.58     386.1
## 5 84358402         M       20.29        14.34         135.10    1297.0
## 6   843786         M       12.45        15.70          82.57     477.1
##   smoothness_mean compactness_mean concavity_mean concave.points_mean
## 1         0.11840          0.27760         0.3001             0.14710
## 2         0.08474          0.07864         0.0869             0.07017
## 3         0.10960          0.15990         0.1974             0.12790
## 4         0.14250          0.28390         0.2414             0.10520
## 5         0.10030          0.13280         0.1980             0.10430
## 6         0.12780          0.17000         0.1578             0.08089
##   symmetry_mean fractal_dimension_mean radius_se texture_se perimeter_se
## 1        0.2419                0.07871    1.0950     0.9053        8.589
```

```
## 2        0.1812                0.05667    0.5435     0.7339        3.398
## 3        0.2069                0.05999    0.7456     0.7869        4.585
## 4        0.2597                0.09744    0.4956     1.1560        3.445
## 5        0.1809                0.05883    0.7572     0.7813        5.438
## 6        0.2087                0.07613    0.3345     0.8902        2.217
##   area_se smoothness_se compactness_se concavity_se concave.points_se
## 1  153.40      0.006399        0.04904      0.05373           0.01587
## 2   74.08      0.005225        0.01308      0.01860           0.01340
## 3   94.03      0.006150        0.04006      0.03832           0.02058
## 4   27.23      0.009110        0.07458      0.05661           0.01867
## 5   94.44      0.011490        0.02461      0.05688           0.01885
## 6   27.19      0.007510        0.03345      0.03672           0.01137
##   symmetry_se fractal_dimension_se radius_worst texture_worst
## 1     0.03003             0.006193        25.38         17.33
## 2     0.01389             0.003532        24.99         23.41
## 3     0.02250             0.004571        23.57         25.53
## 4     0.05963             0.009208        14.91         26.50
## 5     0.01756             0.005115        22.54         16.67
## 6     0.02165             0.005082        15.47         23.75
##   perimeter_worst area_worst smoothness_worst compactness_worst
## 1          184.60     2019.0           0.1622            0.6656
## 2          158.80     1956.0           0.1238            0.1866
## 3          152.50     1709.0           0.1444            0.4245
## 4           98.87      567.7           0.2098            0.8663
## 5          152.20     1575.0           0.1374            0.2050
## 6          103.40      741.6           0.1791            0.5249
##   concavity_worst concave.points_worst symmetry_worst
## 1          0.7119               0.2654         0.4601
## 2          0.2416               0.1860         0.2750
## 3          0.4504               0.2430         0.3613
## 4          0.6869               0.2575         0.6638
## 5          0.4000               0.1625         0.2364
## 6          0.5355               0.1741         0.3985
##   fractal_dimension_worst  X
## 1                 0.11890 NA
## 2                 0.08902 NA
## 3                 0.08758 NA
## 4                 0.17300 NA
## 5                 0.07678 NA
## 6                 0.12440 NA

data$id <- NULL
data$X <- NULL
data$diagnosis <- as.factor(data$diagnosis)
str(data)

## 'data.frame':    569 obs. of  31 variables:
##  $ diagnosis              : Factor w/ 2 levels "B","M": 2 2 2 2 2 2 2 2 2
2 ...
##  $ radius_mean            : num  18 20.6 19.7 11.4 20.3 ...
```

```
##  $ texture_mean           : num  10.4 17.8 21.2 20.4 14.3 ...
##  $ perimeter_mean         : num  122.8 132.9 130 77.6 135.1 ...
##  $ area_mean              : num  1001 1326 1203 386 1297 ...
##  $ smoothness_mean        : num  0.1184 0.0847 0.1096 0.1425 0.1003 ...
##  $ compactness_mean       : num  0.2776 0.0786 0.1599 0.2839 0.1328 ...
##  $ concavity_mean         : num  0.3001 0.0869 0.1974 0.2414 0.198 ...
##  $ concave.points_mean    : num  0.1471 0.0702 0.1279 0.1052 0.1043 ...
##  $ symmetry_mean          : num  0.242 0.181 0.207 0.26 0.181 ...
##  $ fractal_dimension_mean : num  0.0787 0.0567 0.06 0.0974 0.0588 ...
##  $ radius_se              : num  1.095 0.543 0.746 0.496 0.757 ...
##  $ texture_se             : num  0.905 0.734 0.787 1.156 0.781 ...
##  $ perimeter_se           : num  8.59 3.4 4.58 3.44 5.44 ...
##  $ area_se                : num  153.4 74.1 94 27.2 94.4 ...
##  $ smoothness_se          : num  0.0064 0.00522 0.00615 0.00911 0.01149
...
##  $ compactness_se         : num  0.049 0.0131 0.0401 0.0746 0.0246 ...
##  $ concavity_se           : num  0.0537 0.0186 0.0383 0.0566 0.0569 ...
##  $ concave.points_se      : num  0.0159 0.0134 0.0206 0.0187 0.0188 ...
##  $ symmetry_se            : num  0.03 0.0139 0.0225 0.0596 0.0176 ...
##  $ fractal_dimension_se   : num  0.00619 0.00353 0.00457 0.00921 0.00511
...
##  $ radius_worst           : num  25.4 25 23.6 14.9 22.5 ...
##  $ texture_worst          : num  17.3 23.4 25.5 26.5 16.7 ...
##  $ perimeter_worst        : num  184.6 158.8 152.5 98.9 152.2 ...
##  $ area_worst             : num  2019 1956 1709 568 1575 ...
##  $ smoothness_worst       : num  0.162 0.124 0.144 0.21 0.137 ...
##  $ compactness_worst      : num  0.666 0.187 0.424 0.866 0.205 ...
##  $ concavity_worst        : num  0.712 0.242 0.45 0.687 0.4 ...
##  $ concave.points_worst   : num  0.265 0.186 0.243 0.258 0.163 ...
##  $ symmetry_worst         : num  0.46 0.275 0.361 0.664 0.236 ...
##  $ fractal_dimension_worst: num  0.1189 0.089 0.0876 0.173 0.0768 ...
```

```
any(is.na(data))
```

```
## [1] FALSE
```

```
summary(data)
```

```
##  diagnosis  radius_mean      texture_mean   perimeter_mean
##  B:357     Min.   : 6.981   Min.   : 9.71   Min.   : 43.79
##  M:212     1st Qu.:11.700   1st Qu.:16.17   1st Qu.: 75.17
##            Median :13.370   Median :18.84   Median : 86.24
##            Mean   :14.127   Mean   :19.29   Mean   : 91.97
##            3rd Qu.:15.780   3rd Qu.:21.80   3rd Qu.:104.10
##            Max.   :28.110   Max.   :39.28   Max.   :188.50
##    area_mean      smoothness_mean   compactness_mean  concavity_mean
##  Min.   : 143.5   Min.   :0.05263   Min.   :0.01938   Min.   :0.00000
##  1st Qu.: 420.3   1st Qu.:0.08637   1st Qu.:0.06492   1st Qu.:0.02956
##  Median : 551.1   Median :0.09587   Median :0.09263   Median :0.06154
##  Mean   : 654.9   Mean   :0.09636   Mean   :0.10434   Mean   :0.08880
##  3rd Qu.: 782.7   3rd Qu.:0.10530   3rd Qu.:0.13040   3rd Qu.:0.13070
```

```
##  Max.   :2501.0   Max.   :0.16340   Max.   :0.34540   Max.   :0.42680
##  concave.points_mean symmetry_mean    fractal_dimension_mean
##  Min.   :0.00000     Min.   :0.1060   Min.   :0.04996
##  1st Qu.:0.02031     1st Qu.:0.1619   1st Qu.:0.05770
##  Median :0.03350     Median :0.1792   Median :0.06154
##  Mean   :0.04892     Mean   :0.1812   Mean   :0.06280
##  3rd Qu.:0.07400     3rd Qu.:0.1957   3rd Qu.:0.06612
##  Max.   :0.20120     Max.   :0.3040   Max.   :0.09744
##    radius_se        texture_se      perimeter_se       area_se
##  Min.   :0.1115   Min.   :0.3602   Min.   : 0.757   Min.   :  6.802
##  1st Qu.:0.2324   1st Qu.:0.8339   1st Qu.: 1.606   1st Qu.: 17.850
##  Median :0.3242   Median :1.1080   Median : 2.287   Median : 24.530
##  Mean   :0.4052   Mean   :1.2169   Mean   : 2.866   Mean   : 40.337
##  3rd Qu.:0.4789   3rd Qu.:1.4740   3rd Qu.: 3.357   3rd Qu.: 45.190
##  Max.   :2.8730   Max.   :4.8850   Max.   :21.980   Max.   :542.200
##  smoothness_se      compactness_se      concavity_se
##  Min.   :0.001713   Min.   :0.002252   Min.   :0.00000
##  1st Qu.:0.005169   1st Qu.:0.013080   1st Qu.:0.01509
##  Median :0.006380   Median :0.020450   Median :0.02589
##  Mean   :0.007041   Mean   :0.025478   Mean   :0.03189
##  3rd Qu.:0.008146   3rd Qu.:0.032450   3rd Qu.:0.04205
##  Max.   :0.031130   Max.   :0.135400   Max.   :0.39600
##  concave.points_se   symmetry_se       fractal_dimension_se
##  Min.   :0.000000   Min.   :0.007882   Min.   :0.0008948
##  1st Qu.:0.007638   1st Qu.:0.015160   1st Qu.:0.0022480
##  Median :0.010930   Median :0.018730   Median :0.0031870
##  Mean   :0.011796   Mean   :0.020542   Mean   :0.0037949
##  3rd Qu.:0.014710   3rd Qu.:0.023480   3rd Qu.:0.0045580
##  Max.   :0.052790   Max.   :0.078950   Max.   :0.0298400
##   radius_worst   texture_worst   perimeter_worst    area_worst
##  Min.   : 7.93   Min.   :12.02   Min.   : 50.41   Min.   : 185.2
##  1st Qu.:13.01   1st Qu.:21.08   1st Qu.: 84.11   1st Qu.: 515.3
##  Median :14.97   Median :25.41   Median : 97.66   Median : 686.5
##  Mean   :16.27   Mean   :25.68   Mean   :107.26   Mean   : 880.6
##  3rd Qu.:18.79   3rd Qu.:29.72   3rd Qu.:125.40   3rd Qu.:1084.0
##  Max.   :36.04   Max.   :49.54   Max.   :251.20   Max.   :4254.0
##  smoothness_worst  compactness_worst concavity_worst  concave.points_worst
##  Min.   :0.07117   Min.   :0.02729   Min.   :0.0000   Min.   :0.00000
##  1st Qu.:0.11660   1st Qu.:0.14720   1st Qu.:0.1145   1st Qu.:0.06493
##  Median :0.13130   Median :0.21190   Median :0.2267   Median :0.09993
##  Mean   :0.13237   Mean   :0.25427   Mean   :0.2722   Mean   :0.11461
##  3rd Qu.:0.14600   3rd Qu.:0.33910   3rd Qu.:0.3829   3rd Qu.:0.16140
##  Max.   :0.22260   Max.   :1.05800   Max.   :1.2520   Max.   :0.29100
##  symmetry_worst   fractal_dimension_worst
##  Min.   :0.1565   Min.   :0.05504
##  1st Qu.:0.2504   1st Qu.:0.07146
##  Median :0.2822   Median :0.08004
##  Mean   :0.2901   Mean   :0.08395
##  3rd Qu.:0.3179   3rd Qu.:0.09208
##  Max.   :0.6638   Max.   :0.20750
```

How many observations have Benign or Malignant diagnosis ?

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.6.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
data %>%
  count(diagnosis) %>%
  group_by(diagnosis) %>%
  summarize(perc_dx = round((n / 569)* 100, 2))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 2 x 2
##   diagnosis perc_dx
##   <fct>       <dbl>
## 1 B            62.7
## 2 M            37.3
```

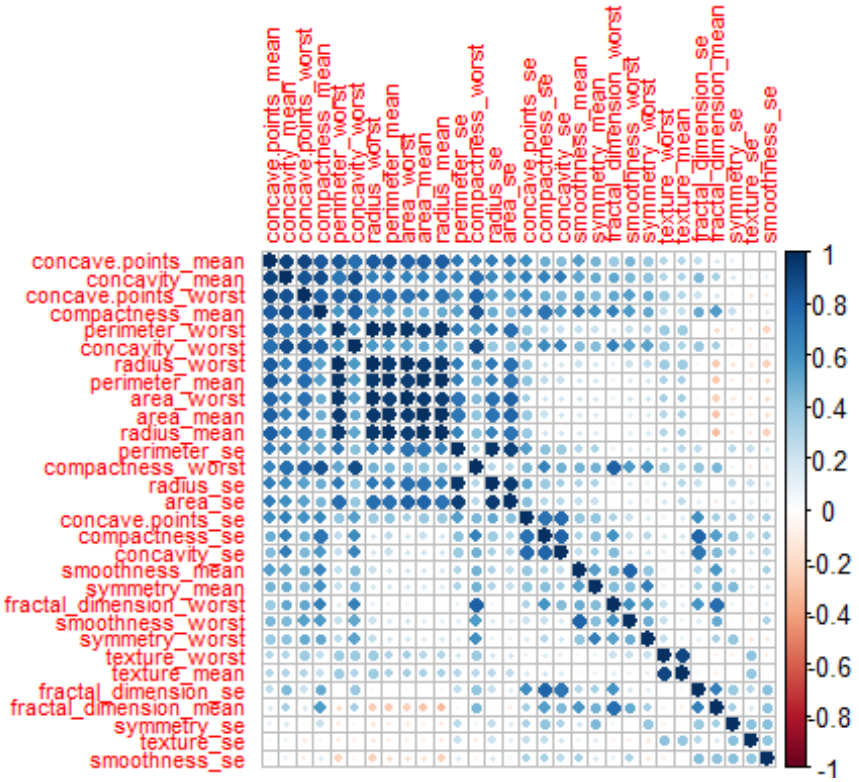
```
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 3.6.1
```

```
## corrplot 0.84 loaded
```

Create corrplot : Because there are so much correlation some machine learning models can fail. We are going to create a PCA and LDA version of the data

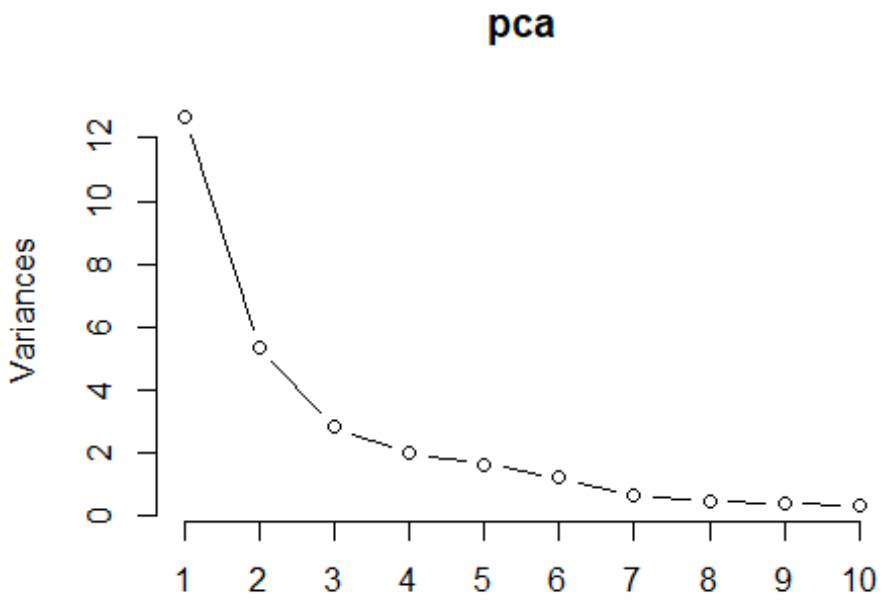
```
corrplot(cor(data[,-1]), order = "FPC", tl.cex = .7, addrect = 10)
```



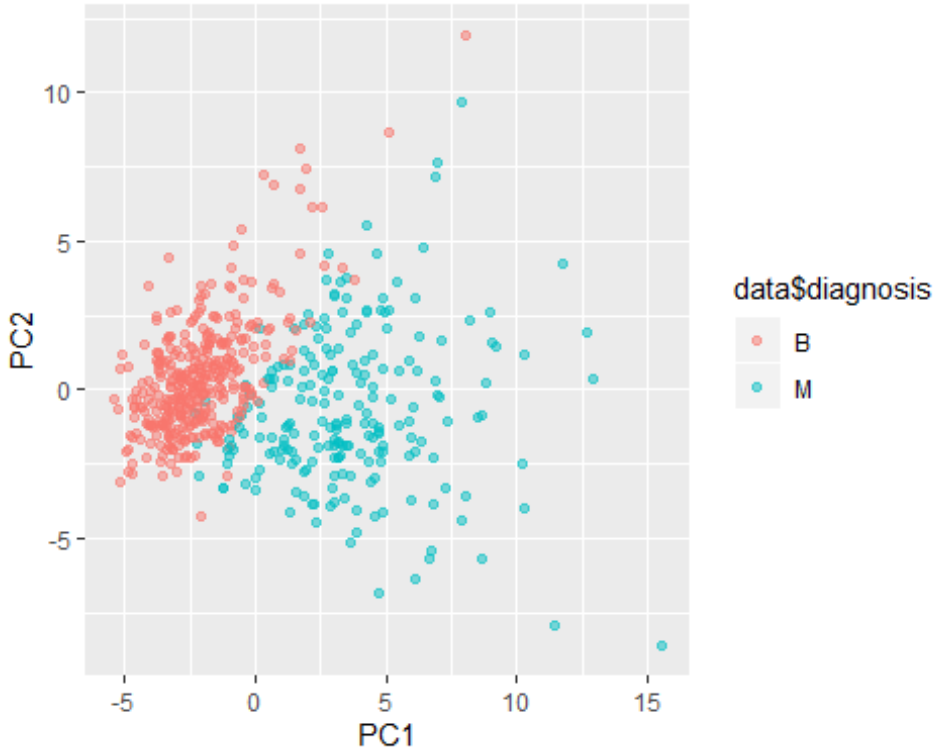
Principal Components Analysis

```
set.seed(1234)
data_index <-  sample(2,nrow(data),replace = TRUE, prob = c(0.75,0.25))
train <- data[data_index==1,]
test <- data[data_index==2,]

pca<- prcomp(data[,3:ncol(data)], center = TRUE, scale. = T)
summary(pca)

## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5     PC6
## Standard deviation     3.5602 2.3145 1.67860 1.40601 1.28301 1.09859
## Proportion of Variance 0.4371 0.1847 0.09716 0.06817 0.05676 0.04162
## Cumulative Proportion  0.4371 0.6218 0.71895 0.78712 0.84388 0.88550
##                           PC7     PC8     PC9    PC10    PC11    PC12
## Standard deviation     0.81534 0.69036 0.62876 0.58783 0.54148 0.51013
## Proportion of Variance 0.02292 0.01643 0.01363 0.01192 0.01011 0.00897
## Cumulative Proportion  0.90842 0.92485 0.93849 0.95040 0.96051 0.96948
##                           PC13    PC14    PC15   PC16    PC17    PC18
## Standard deviation     0.49123 0.39543 0.30645 0.2796 0.23982 0.22774
## Proportion of Variance 0.00832 0.00539 0.00324 0.0027 0.00198 0.00179
## Cumulative Proportion  0.97781 0.98320 0.98644 0.9891 0.99111 0.99290
##                           PC19    PC20    PC21    PC22    PC23    PC24
## Standard deviation     0.21104 0.17623 0.17248 0.16495 0.15477 0.13050
## Proportion of Variance 0.00154 0.00107 0.00103 0.00094 0.00083 0.00059
## Cumulative Proportion  0.99444 0.99551 0.99654 0.99747 0.99830 0.99889
##                           PC25    PC26    PC27    PC28    PC29
```

```
## Standard deviation     0.12436 0.08933 0.08164 0.03850 0.02635
## Proportion of Variance 0.00053 0.00028 0.00023 0.00005 0.00002
## Cumulative Proportion  0.99942 0.99970 0.99992 0.99998 1.00000

pca_df <- as.data.frame(pca$x)
plot(pca, type="l")
```




```
library(ggplot2)

## Warning: package 'ggplot2' was built under R version 3.6.1

ggplot(pca_df, aes(x=PC1, y=PC2, col=data$diagnosis)) + geom_point(alpha=0.5)
```



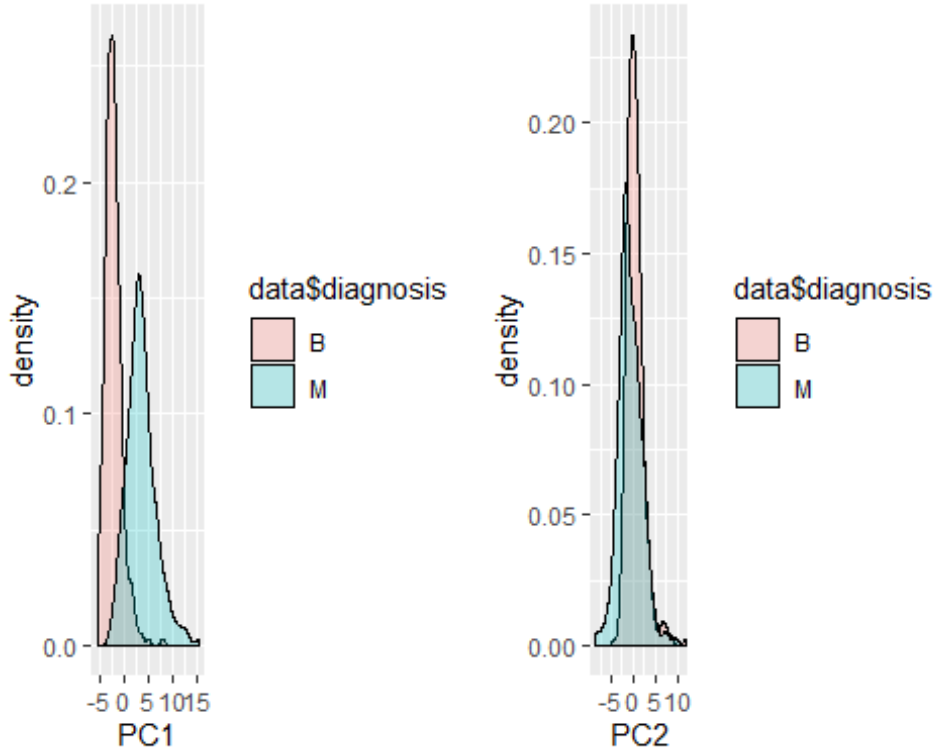

```
library(gridExtra)
```

```
## Warning: package 'gridExtra' was built under R version 3.6.1
```

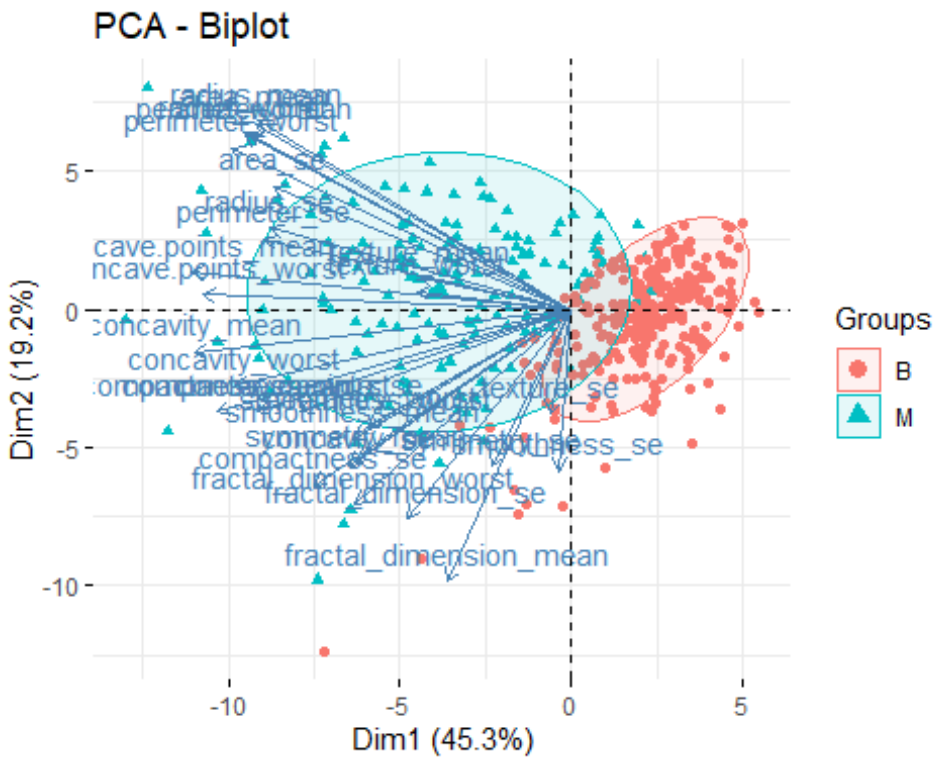
```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
g_pc1 <- ggplot(pca_df, aes(x=PC1, fill=data$diagnosis)) +
geom_density(alpha=0.25)
g_pc2 <- ggplot(pca_df, aes(x=PC2, fill=data$diagnosis)) +
geom_density(alpha=0.25)
grid.arrange(g_pc1, g_pc2, ncol=2)
```




```
library(factoextra)

## Warning: package 'factoextra' was built under R version 3.6.1

## Welcome! Related Books: `Practical Guide To Cluster Analysis in R` at
https://goo.gl/13EFCZ

res.pca <- prcomp(train[,-1],  scale = TRUE)

fviz_pca_biplot(res.pca, label="var", habillage= train$diagnosis,
                addEllipses=TRUE, ellipse.level=0.8)
```




```
traning <- predict(res.pca,train)
traning <- data.frame(traning,train[1])

testing <- predict(res.pca,test)
testing <- data.frame(testing,test[1])

library(nnet)
mlr <- relevel(traning$diagnosis, ref = "B")
model1<- multinom(diagnosis~.,
                  data = traning)

## # weights:  32 (31 variable)
## initial  value 291.814963
## iter  10 value 44.241986
## iter  20 value 11.941272
## iter  30 value 0.669574
## iter  40 value 0.000977
## final  value 0.000073
## converged

summary(model1)

## Call:
## multinom(formula = diagnosis ~ ., data = traning)
##
## Coefficients:
##                   Values    Std. Err.
```

```
## (Intercept)   -73.14116 290.6375063
## PC1          -263.07805 102.0300639
## PC2           148.61557 128.8672462
## PC3           -85.15951 178.9360474
## PC4           -86.42985 154.5636023
## PC5            67.63058  80.6419206
## PC6            12.79680 137.0406646
## PC7          -135.50721 172.1728976
## PC8           -25.94348  80.0353729
## PC9           259.90488  84.9918136
## PC10          -86.14869 102.1976171
## PC11         -112.85663  44.2866691
## PC12         -106.67274  75.8392741
## PC13         -202.14795  44.1645127
## PC14          -83.50829  72.4791395
## PC15         -217.66296  25.6317243
## PC16          154.15706  18.6992112
## PC17         -300.23255  15.8764646
## PC18          143.15476  32.0177548
## PC19         -266.36788  32.9420472
## PC20          689.31154  22.9996278
## PC21         -496.92812  20.4585274
## PC22          128.82238  15.7122286
## PC23         -270.67992  12.7435197
## PC24        -1053.77979  21.9412765
## PC25         -540.31723   2.6594127
## PC26         -935.27056   9.1299110
## PC27         -806.36370   4.5181373
## PC28          -86.18085   1.3857107
## PC29          -43.80111   0.9509789
## PC30          -95.51290   1.3743798
##
## Residual Deviance: 0.0001450812
## AIC: 62.00015
```

```
prd <- predict(model1, traning)
table <- table(prd, traning$diagnosis)
table
```

```
##
## prd   B   M
##   B 270   0
##   M   0 151
```

```
sum(diag(table))/sum(table)
```

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

```
testing <- data.frame(testing,test[1])

prd1 <- predict(model1, testing)
```

```
table1 <- table(prd1, testing$diagnosis)
table1
```

```
##
## prd1  B  M
##    B 82  7
##    M  5 54
```

```
sum(diag(table1))/sum(table1)
```

```
## [1] 0.9189189
```

LDA

```
library(MASS)
```

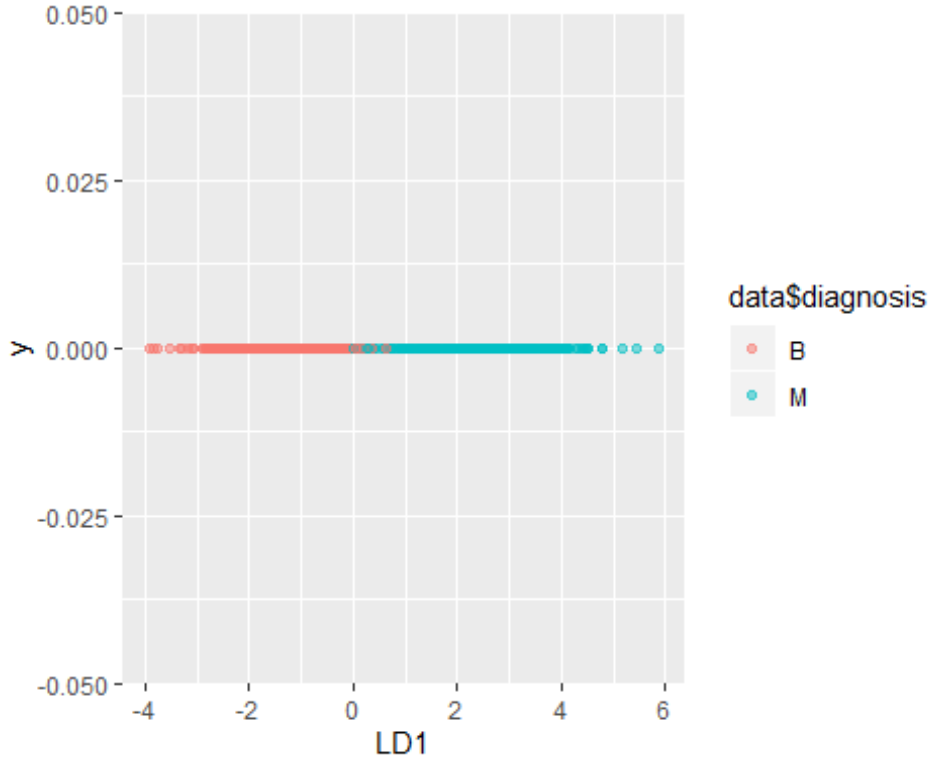
```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

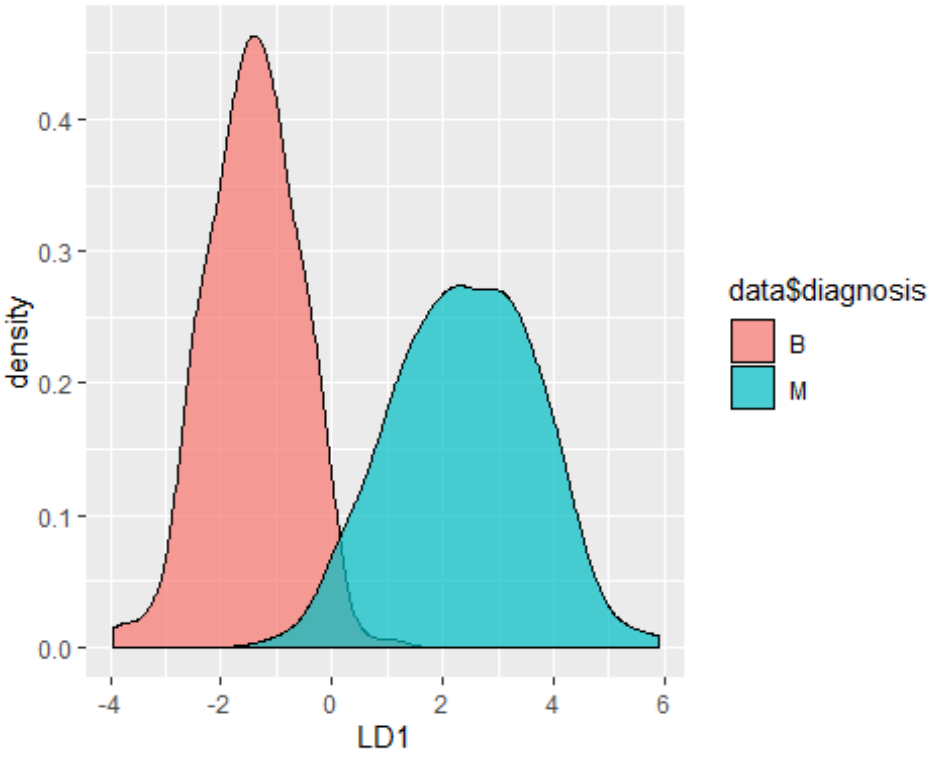
```
lda_res <- lda(diagnosis~., data)

lda_res
```

```
## Call:
## lda(diagnosis ~ ., data = data)
##
## Prior probabilities of groups:
##         B         M
## 0.6274165 0.3725835
##
## Group means:
##   radius_mean texture_mean perimeter_mean area_mean smoothness_mean
## B    12.14652     17.91476       78.07541  462.7902      0.09247765
## M    17.46283     21.60491      115.36538  978.3764      0.10289849
##   compactness_mean concavity_mean concave.points_mean symmetry_mean
## B       0.08008462     0.04605762          0.02571741      0.174186
## M       0.14518778     0.16077472          0.08799000      0.192909
##   fractal_dimension_mean radius_se texture_se perimeter_se  area_se
## B             0.06286739 0.2840824   1.220380     2.000321 21.13515
## M             0.06268009 0.6090825   1.210915     4.323929 72.67241
##   smoothness_se compactness_se concavity_se concave.points_se symmetry_se
## B   0.007195902     0.02143825   0.02599674       0.009857653  0.02058381
## M   0.006780094     0.03228117   0.04182401       0.015060472  0.02047240
##   fractal_dimension_se radius_worst texture_worst perimeter_worst
## B          0.003636051     13.37980      23.51507        87.00594
## M          0.004062406     21.13481      29.31821       141.37033
##   area_worst smoothness_worst compactness_worst concavity_worst
## B   558.8994        0.1249595         0.1826725       0.1662377
```

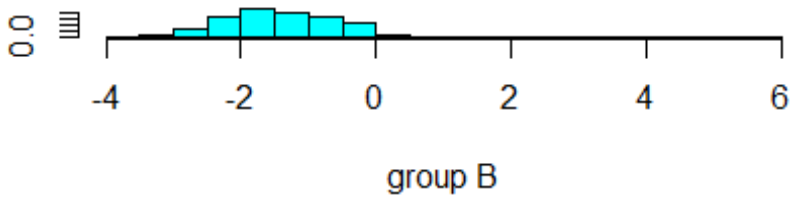
```
## M  1422.2863        0.1448452         0.3748241       0.4506056
##   concave.points_worst symmetry_worst fractal_dimension_worst
## B           0.07444434      0.2702459              0.07944207
## M           0.18223731      0.3234679              0.09152995
##
## Coefficients of linear discriminants:
##                                  LD1
## radius_mean              -1.075583600
## texture_mean              0.022450225
## perimeter_mean            0.117251982
## area_mean                 0.001569797
## smoothness_mean           0.418282533
## compactness_mean        -20.852775912
## concavity_mean            6.904756198
## concave.points_mean      10.578586272
## symmetry_mean             0.507284238
## fractal_dimension_mean    0.164280222
## radius_se                 2.148262164
## texture_se               -0.033380325
## perimeter_se             -0.111228320
## area_se                  -0.004559805
## smoothness_se            78.305030179
## compactness_se            0.320560148
## concavity_se            -17.609967822
## concave.points_se        52.195471457
## symmetry_se               8.383223501
## fractal_dimension_se    -35.296511336
## radius_worst              0.964016085
## texture_worst             0.035360398
## perimeter_worst          -0.012026798
## area_worst               -0.004994466
## smoothness_worst          2.681188528
## compactness_worst         0.331697102
## concavity_worst           1.882716394
## concave.points_worst      2.293242388
## symmetry_worst            2.749992654
## fractal_dimension_worst  21.255049570

library(dplyr)
lda_df <- predict(lda_res, data)$x %>% as.data.frame()

ggplot(lda_df, aes(x=LD1, y=0, col=data$diagnosis)) + geom_point(alpha=0.5)
```



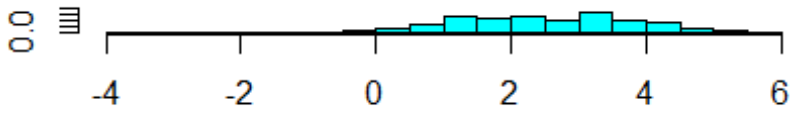

```
ggplot(lda_df, aes(x=LD1, fill=data$diagnosis)) + geom_density(alpha=0.7)
```



```
prd <- predict(lda_res, train)
ldahist(data = prd$x[,1], g = train$diagnosis)
```



group B



group M

```
p.train <- predict(lda_res, train)$class
tab <-  table(predicted = p.train, Actual = train$diagnosis)
tab
```

```
##           Actual
## predicted   B   M
##         B 268  11
##         M   2 140
```

```
sum(diag(tab)/sum(tab))
```

```
## [1] 0.9691211
```

```
p.test <- predict(lda_res, test)$class
tab2 <- table(predicted = p.test, Actual = test$diagnosis)
tab2
```

```
##           Actual
## predicted  B  M
##         B 87  7
##         M  0 54
```

```
sum(diag(tab2)/sum(tab2))
```

```
## [1] 0.9527027
```

We have found a model based on LDA preprocessed data with good results over the test set.