SVM_Tarea1

María Isabel Chuya

Tarea SVM

- Realizar Support Vector Machines con una "Evaluación de Datos en CRUDO" y datos con "Transformación de datos por logaritmo"

Cargar Librerias

Se cargan las librerias al inicio, puesto que ayuda a que todo el codigo se desarrolle de manera continua.

```
library(ggplot2)
library(e1071)
library(dplyr)
```

```
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
```

```
library(reshape2)
library(corrplot)
```

```
corrplot 0.92 loaded
```

```
library(caret)
```

```
Loading required package: lattice
```

```
library(kernlab)
```

```
Attaching package: 'kernlab'
```

```
The following object is masked from 'package:ggplot2':

    alpha
```

```
library(pROC)
```

```
Type 'citation("pROC")' for a citation.
```

```
Attaching package: 'pROC'
```

```
The following objects are masked from 'package:stats':

    cov, smooth, var
```

```
library(gridExtra)
```

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

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

    combine
```

```
library(grid)
library(ggfortify)
library(purrr)
```

```
Attaching package: 'purrr'
```

```
The following object is masked from 'package:kernlab':

    cross
```

```
The following object is masked from 'package:caret':

    lift
```

```
library(nnet)
library(ggstatsplot)
```

```
You can cite this package as:
     Patil, I. (2021). Visualizations with statistical details: The 'ggstatsplot' approach.
     Journal of Open Source Software, 6(61), 3167, doi:10.21105/joss.03167
```

```
library(knitr)
library(lavaan)
```

```
This is lavaan 0.6-15
lavaan is FREE software! Please report any bugs.
```

```
library(doParallel) # parallel processing
```

```
Loading required package: foreach
```

```
Attaching package: 'foreach'
```

```
The following objects are masked from 'package:purrr':

    accumulate, when

Loading required package: iterators

Loading required package: parallel
```

```r
registerDoParallel()
require(foreach)
require(iterators)
require(parallel)
```

Cargar Datos

- Para cargar los datos se establece una variable y se lee los datos cargados en la misma carpeta “read.csv(”./cancer.csv”), header=T)”
- Se usa el comando “head()” para visualizar las primeras filas de un conjunto de datos o un objeto en forma de tabla.
- Se usa el comando “summary(datos)” para obtener las estadisticas descriptivas de las variables
- Se usa el comando “str(datos)” para realizar el analisis estadistico de los datos

```r
# Cargar los datos
datos <- read.csv("./cancer.csv",header = T)
datos.numericos <- datos[, which(unlist(lapply(datos, is.numeric)))]
colnames(datos.numericos) <- paste0("Var", rep(1:11))

# Explorar los datos
head(datos)  # Ver las primeras filas del conjunto de datos
```

```
  diagnostico mean_radius mean_texture mean_perimeter mean_area mean_smoothnes
1           M       17.99        10.38         122.80    1001.0        0.11840
2           M       20.57        17.77         132.90    1326.0        0.08474
3           M       19.69        21.25         130.00    1203.0        0.10960
4           M       11.42        20.38          77.58     386.1        0.14250
5           M       20.29        14.34         135.10    1297.0        0.10030
6           M       12.45        15.70          82.57     477.1        0.12780
```

```
  mean_compactness mean_concavity mean_concave_points mean_simmetry
1          0.27760         0.3001             0.14710        0.2419
2          0.07864         0.0869             0.07017        0.1812
3          0.15990         0.1974             0.12790        0.2069
4          0.28390         0.2414             0.10520        0.2597
5          0.13280         0.1980             0.10430        0.1809
6          0.17000         0.1578             0.08089        0.2087
  mean_fractal_dimension se_radius se_texture se_perimeter se_area se_smoothnes
1                0.07871    1.0950     0.9053        8.589  153.40     0.006399
2                0.05667    0.5435     0.7339        3.398   74.08     0.005225
3                0.05999    0.7456     0.7869        4.585   94.03     0.006150
4                0.09744    0.4956     1.1560        3.445   27.23     0.009110
5                0.05883    0.7572     0.7813        5.438   94.44     0.011490
6                0.07613    0.3345     0.8902        2.217   27.19     0.007510
  se_compactness se_concavity se_concave_points se_simmetry
1        0.04904      0.05373           0.01587     0.03003
2        0.01308      0.01860           0.01340     0.01389
3        0.04006      0.03832           0.02058     0.02250
4        0.07458      0.05661           0.01867     0.05963
5        0.02461      0.05688           0.01885     0.01756
6        0.03345      0.03672           0.01137     0.02165
  se_fractal_dimension worst_radius worst_texture worst_perimeter worst_area
1             0.006193        25.38         17.33          184.60     2019.0
2             0.003532        24.99         23.41          158.80     1956.0
3             0.004571        23.57         25.53          152.50     1709.0
4             0.009208        14.91         26.50           98.87      567.7
5             0.005115        22.54         16.67          152.20     1575.0
6             0.005082        15.47         23.75          103.40      741.6
  worst_smoothnes worst_compactness worst_concavity worst_concave_points
1          0.1622            0.6656          0.7119               0.2654
2          0.1238            0.1866          0.2416               0.1860
3          0.1444            0.4245          0.4504               0.2430
4          0.2098            0.8663          0.6869               0.2575
5          0.1374            0.2050          0.4000               0.1625
6          0.1791            0.5249          0.5355               0.1741
  worst_simmetry worst_fractal_dimension
1         0.4601                0.11890
2         0.2750                0.08902
3         0.3613                0.08758
4         0.6638                0.17300
5         0.2364                0.07678
6         0.3985                0.12440
```

```r
summary(datos)  # Obtener estadísticas descriptivas de las variables
```

```
diagnostico         mean_radius      mean_texture   mean_perimeter
Length:569         Min.   : 6.981   Min.   : 9.71   Min.   : 43.79
Class :character   1st Qu.:11.700   1st Qu.:16.17   1st Qu.: 75.17
Mode  :character   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
  mean_area      mean_smoothnes    mean_compactness  mean_concavity
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
mean_concave_points mean_simmetry    mean_fractal_dimension   se_radius
Min.   :0.00000     Min.   :0.1060   Min.   :0.04996        Min.   :0.1115
1st Qu.:0.02031     1st Qu.:0.1619   1st Qu.:0.05770        1st Qu.:0.2324
Median :0.03350     Median :0.1792   Median :0.06154        Median :0.3242
Mean   :0.04892     Mean   :0.1812   Mean   :0.06280        Mean   :0.4052
3rd Qu.:0.07400     3rd Qu.:0.1957   3rd Qu.:0.06612        3rd Qu.:0.4789
Max.   :0.20120     Max.   :0.3040   Max.   :0.09744        Max.   :2.8730
  se_texture      se_perimeter       se_area         se_smoothnes
Min.   :0.3602   Min.   : 0.757   Min.   :  6.802   Min.   :0.001713
1st Qu.:0.8339   1st Qu.: 1.606   1st Qu.: 17.850   1st Qu.:0.005169
Median :1.1080   Median : 2.287   Median : 24.530   Median :0.006380
Mean   :1.2169   Mean   : 2.866   Mean   : 40.337   Mean   :0.007041
3rd Qu.:1.4740   3rd Qu.: 3.357   3rd Qu.: 45.190   3rd Qu.:0.008146
Max.   :4.8850   Max.   :21.980   Max.   :542.200   Max.   :0.031130
se_compactness      se_concavity     se_concave_points   se_simmetry
Min.   :0.002252   Min.   :0.00000   Min.   :0.000000   Min.   :0.007882
1st Qu.:0.013080   1st Qu.:0.01509   1st Qu.:0.007638   1st Qu.:0.015160
Median :0.020450   Median :0.02589   Median :0.010930   Median :0.018730
Mean   :0.025478   Mean   :0.03189   Mean   :0.011796   Mean   :0.020542
3rd Qu.:0.032450   3rd Qu.:0.04205   3rd Qu.:0.014710   3rd Qu.:0.023480
Max.   :0.135400   Max.   :0.39600   Max.   :0.052790   Max.   :0.078950
se_fractal_dimension  worst_radius   worst_texture   worst_perimeter
Min.   :0.0008948    Min.   : 7.93   Min.   :12.02   Min.   : 50.41
1st Qu.:0.0022480    1st Qu.:13.01   1st Qu.:21.08   1st Qu.: 84.11
Median :0.0031870    Median :14.97   Median :25.41   Median : 97.66
Mean   :0.0037949    Mean   :16.27   Mean   :25.68   Mean   :107.26
```

```
 3rd Qu.:0.0045580    3rd Qu.:18.79   3rd Qu.:29.72   3rd Qu.:125.40
 Max.   :0.0298400    Max.   :36.04   Max.   :49.54   Max.   :251.20
   worst_area     worst_smoothnes   worst_compactness worst_concavity
 Min.   : 185.2   Min.   :0.07117   Min.   :0.02729   Min.   :0.0000
 1st Qu.: 515.3   1st Qu.:0.11660   1st Qu.:0.14720   1st Qu.:0.1145
 Median : 686.5   Median :0.13130   Median :0.21190   Median :0.2267
 Mean   : 880.6   Mean   :0.13237   Mean   :0.25427   Mean   :0.2722
 3rd Qu.:1084.0   3rd Qu.:0.14600   3rd Qu.:0.33910   3rd Qu.:0.3829
 Max.   :4254.0   Max.   :0.22260   Max.   :1.05800   Max.   :1.2520
 worst_concave_points worst_simmetry   worst_fractal_dimension
 Min.   :0.00000      Min.   :0.1565   Min.   :0.05504
 1st Qu.:0.06493      1st Qu.:0.2504   1st Qu.:0.07146
 Median :0.09993      Median :0.2822   Median :0.08004
 Mean   :0.11461      Mean   :0.2901   Mean   :0.08395
 3rd Qu.:0.16140      3rd Qu.:0.3179   3rd Qu.:0.09208
 Max.   :0.29100      Max.   :0.6638   Max.   :0.20750
```

```r
# Calcular medidas de tendencia central
mean(datos$variable)  # Calcular la media de una variable
```

```
Warning in mean.default(datos$variable): argument is not numeric or logical:
returning NA
```

```
[1] NA
```

```r
median(datos$variable)  # Calcular la mediana de una variable
```

```
NULL
```

```r
table(datos$variable)  # Obtener la tabla de frecuencias de una variable categórica
```

```
< table of extent 0 >
```

```r
# Calcular medidas de dispersión
sd(datos$variable)  # Calcular la desviación estándar de una variable
```

```
[1] NA
```

```r
range(datos$variable)  # Obtener el rango de una variable
```

```
Warning in min(x, na.rm = na.rm): no non-missing arguments to min; returning
Inf
```

```
Warning in max(x, na.rm = na.rm): no non-missing arguments to max; returning
-Inf
```

```
[1]  Inf -Inf
```

```r
str(datos)
```

```
'data.frame':   569 obs. of  31 variables:
 $ diagnostico            : chr  "M" "M" "M" "M" ...
 $ mean_radius            : num  18 20.6 19.7 11.4 20.3 ...
 $ mean_texture           : num  10.4 17.8 21.2 20.4 14.3 ...
 $ mean_perimeter         : num  122.8 132.9 130 77.6 135.1 ...
 $ mean_area              : num  1001 1326 1203 386 1297 ...
 $ mean_smoothnes         : num  0.1184 0.0847 0.1096 0.1425 0.1003 ...
 $ mean_compactness       : num  0.2776 0.0786 0.1599 0.2839 0.1328 ...
 $ mean_concavity         : num  0.3001 0.0869 0.1974 0.2414 0.198 ...
 $ mean_concave_points    : num  0.1471 0.0702 0.1279 0.1052 0.1043 ...
 $ mean_simmetry          : num  0.242 0.181 0.207 0.26 0.181 ...
 $ mean_fractal_dimension : num  0.0787 0.0567 0.06 0.0974 0.0588 ...
 $ se_radius              : num  1.095 0.543 0.746 0.496 0.757 ...
 $ se_texture             : num  0.905 0.734 0.787 1.156 0.781 ...
 $ se_perimeter           : num  8.59 3.4 4.58 3.44 5.44 ...
 $ se_area                : num  153.4 74.1 94 27.2 94.4 ...
 $ se_smoothnes           : num  0.0064 0.00522 0.00615 0.00911 0.01149 ...
 $ se_compactness         : num  0.049 0.0131 0.0401 0.0746 0.0246 ...
 $ se_concavity           : num  0.0537 0.0186 0.0383 0.0566 0.0569 ...
 $ se_concave_points      : num  0.0159 0.0134 0.0206 0.0187 0.0188 ...
 $ se_simmetry            : num  0.03 0.0139 0.0225 0.0596 0.0176 ...
 $ se_fractal_dimension   : num  0.00619 0.00353 0.00457 0.00921 0.00511 ...
 $ worst_radius           : num  25.4 25 23.6 14.9 22.5 ...
 $ worst_texture          : num  17.3 23.4 25.5 26.5 16.7 ...
 $ worst_perimeter        : num  184.6 158.8 152.5 98.9 152.2 ...
 $ worst_area             : num  2019 1956 1709 568 1575 ...
 $ worst_smoothnes        : num  0.162 0.124 0.144 0.21 0.137 ...
```

```
$ worst_compactness      : num  0.666 0.187 0.424 0.866 0.205 ...
$ worst_concavity        : num  0.712 0.242 0.45 0.687 0.4 ...
$ worst_concave_points   : num  0.265 0.186 0.243 0.258 0.163 ...
$ worst_simmetry         : num  0.46 0.275 0.361 0.664 0.236 ...
$ worst_fractal_dimension: num  0.1189 0.089 0.0876 0.173 0.0768 ...
```

```
#datos<-datos[,1:11]
```

```
#head(datos)
```

Evaluacion de datos en CRUDO

Diagrama de cajas

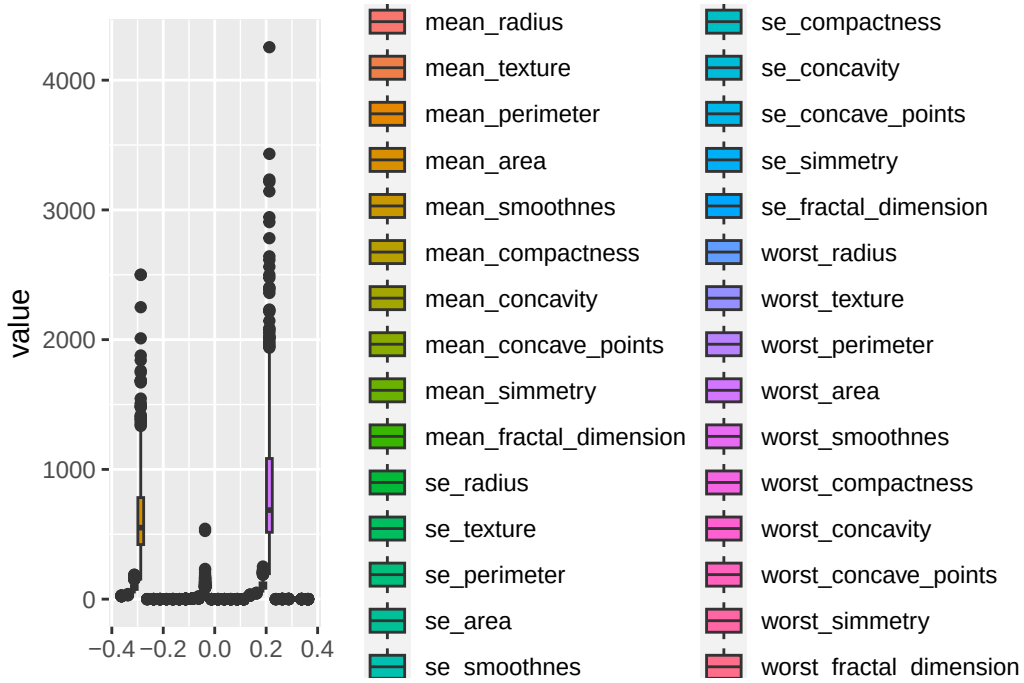
Los diagramas de caja, también conocidos como diagramas de caja y bigotes, son gráficos que muestran la distribución de una variable usando cuartiles para que podamos inferir visualmente algunas características sobre su dispersión, ubicación y simetría.

- Se utiliza el comando "geom_boxplot()" para graficar el diagrama de cajas de todas las variables presentes en los datos de "cancer.csv"

```
#diagrama de cajas unido
datos.melt<-reshape2::melt((datos))
```

```
Using diagnostico as id variables
```

```
ggplot(datos.melt,aes(y=value,fill=variable))+geom_boxplot()
```



Grafica de Densidad

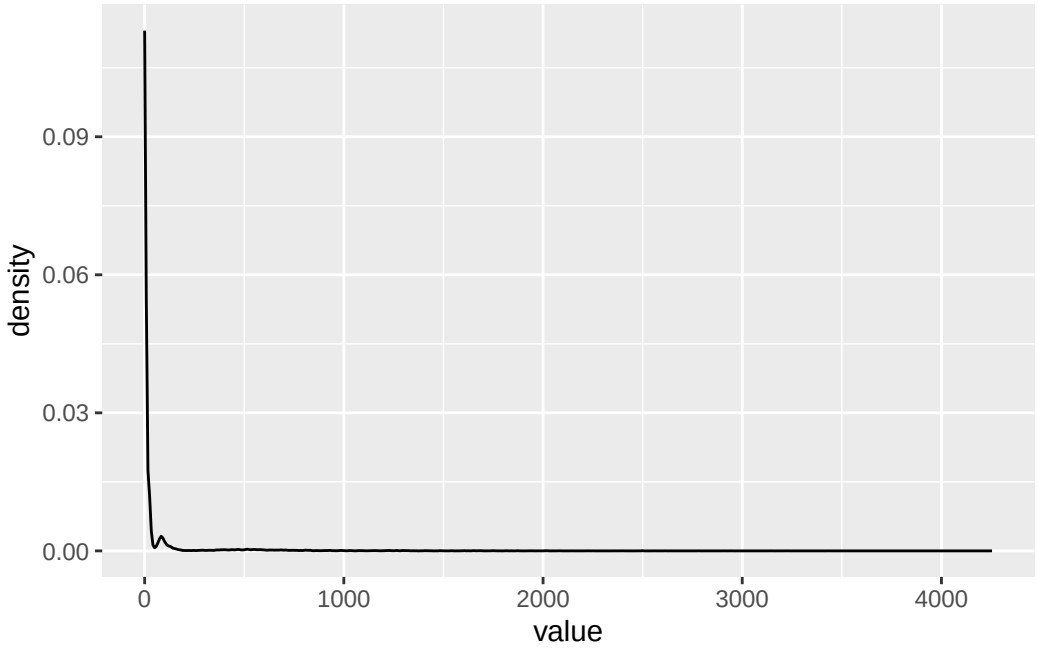
Un gráfico de densidad muestra cómo se distribuyen los datos cuantitativos en un rango continuo o período de tiempo. Este gráfico es una adaptación de un histograma que utiliza el suavizado kernel para trazar valores, lo que permite distribuciones más suaves al eliminar el ruido.

- Se utiliza el comando "geom_density()" para graficar el diagrama de cajas de todas las variables presentes en los datos de "cancer.csv"

```
datos.melt1<-reshape2::melt((datos))
```

```
Using diagnostico as id variables
```

```
ggplot(datos.melt1,aes(x=value))+geom_density()
```



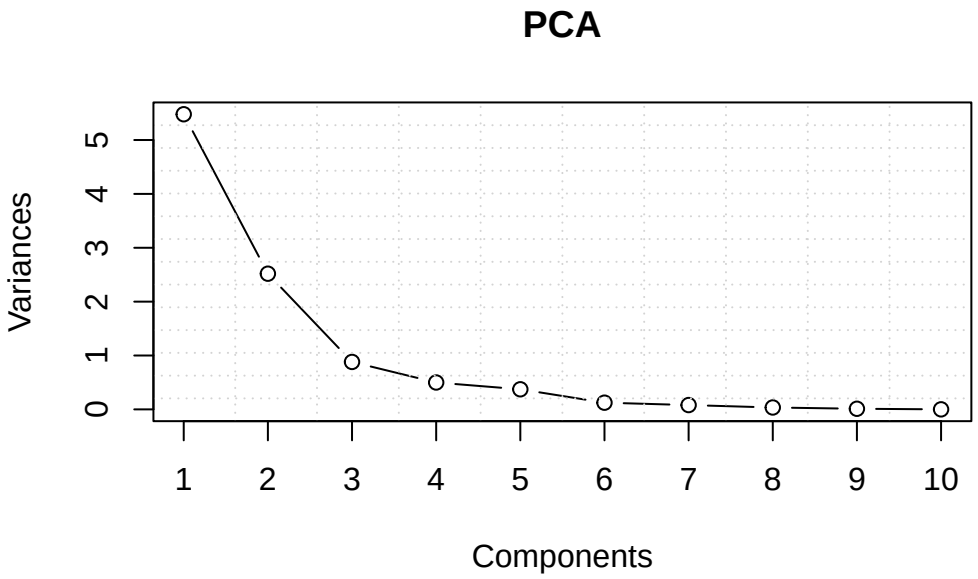
- Transformar en función logarítmica porque los algoritmos están diseñados la mayoría por distribuciones normales, aunque el que se está observando funcionaria con todo tipo de datos, sin embargo para corroborar se necesita realizar un análisis de componentes principales.
- Se transforma a función logarítmica, puesto que la gráfica de densidad no es parecida a una gráfica de distribución normal.

Analisis de Componentes Principales (PCA) en CRUDO

Una de las técnicas de aprendizaje no supervisado, el análisis de componentes principales (PCA), se usa con frecuencia junto con el análisis exploratorio de datos.

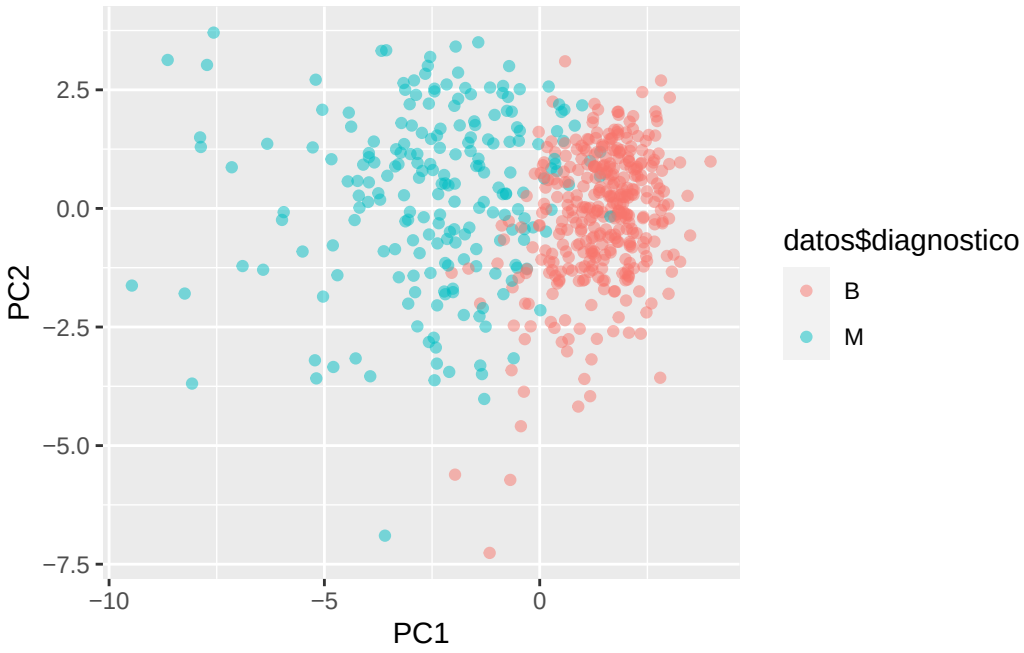
- Se realiza primero un Scree Plot y seguido se realiza un Biplot para el analisis del comportamiento de los componentes principales.
  - SCREE PLOT

```
cancer.pca <- prcomp(datos[, 2:11], center=TRUE, scale=TRUE)
plot(cancer.pca, type="l", main='')
grid(nx = 10, ny = 14)
title(main = "PCA", sub = NULL, xlab = "Components")
box()
```



- – BIPLOT

```
pca_df <- as.data.frame(cancer.pca$x)
ggplot(pca_df, aes(x=PC1, y=PC2, col=datos$diagnostico)) + geom_point(alpha=0.5)
```



SVM con datos en crudo

1. Dividir los datos en entrenamiento y prueba

```
n<-nrow(datos)
set.seed(123456)

datos$diagnostico<-as.factor(datos$diagnostico)

train <- sample(n,floor(n*0.7))
datos.train <- datos[train,]
datos.test  <- datos[-train,]


#datos.test <- dato.log[train,]
#datos.test  <- datos.log[-train,]


#set.seed(123456)
#sample <- sample(c(TRUE, FALSE), nrow(datos), replace=TRUE, prob=c(0.7,0.3))
#datos.train  <- datos[sample, ]
```

```
#datos.test   <- datos[!sample, ]
```

2. Formacion de modelos

   Se usa el comando “Kernel” para la formación de modelos.

   - Se crea para clasifier.lineal y para clasifier.gauss

```
clasifier.lineal<-ksvm(diagnostico~ .,data=datos.train,kernel="vanilladot")
```

```
 Setting default kernel parameters
```

```
clasifier.gauss<-ksvm(diagnostico~.,data=datos.train,kernel = "rbfdot")
clasifier.lineal
```

```
Support Vector Machine object of class "ksvm"

SV type: C-svc  (classification)
 parameter : cost C = 1

Linear (vanilla) kernel function.

Number of Support Vectors : 33

Objective Function Value : -21.2999
Training error : 0.012563
```

```
clasifier.gauss
```

```
Support Vector Machine object of class "ksvm"

SV type: C-svc  (classification)
 parameter : cost C = 1

Gaussian Radial Basis kernel function.
 Hyperparameter : sigma =  0.0405393061667693

Number of Support Vectors : 107

Objective Function Value : -47.3373
Training error : 0.015075
```

3. Evaluación de rendimiento del modelo

   Cuando se utiliza "ConfusionMatrix" para resolver un problema de clasificación, donde el resultado puede ser dos o más clases, la matriz de confusión se utiliza como indicador de rendimiento.

   - Se usa para cada una de los modelos realizados, lineal y de gauss.

```
prediction.linear<-predict(clasifier.lineal,datos.test);res.linear<-table(prediction.linea
prediction.gauss<-predict(clasifier.gauss,datos.test);res.gauss<-table(prediction.gauss,da
```

```
cmatrix1 <- confusionMatrix(res.linear)
print(cmatrix1)
```

```
Confusion Matrix and Statistics


prediction.linear   B   M
                B 107   3
                M   2  59

               Accuracy : 0.9708
                 95% CI : (0.9331, 0.9904)
    No Information Rate : 0.6374
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9365

 Mcnemar's Test P-Value : 1

            Sensitivity : 0.9817
            Specificity : 0.9516
         Pos Pred Value : 0.9727
         Neg Pred Value : 0.9672
             Prevalence : 0.6374
         Detection Rate : 0.6257
   Detection Prevalence : 0.6433
      Balanced Accuracy : 0.9666

       'Positive' Class : B
```

```r
cmatrix2<-confusionMatrix(res.gauss)
print(cmatrix2)
```

```
Confusion Matrix and Statistics


prediction.gauss   B   M
               B 107   3
               M   2  59

               Accuracy : 0.9708
                 95% CI : (0.9331, 0.9904)
    No Information Rate : 0.6374
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9365

 Mcnemar's Test P-Value : 1

            Sensitivity : 0.9817
            Specificity : 0.9516
         Pos Pred Value : 0.9727
         Neg Pred Value : 0.9672
             Prevalence : 0.6374
         Detection Rate : 0.6257
   Detection Prevalence : 0.6433
      Balanced Accuracy : 0.9666

       'Positive' Class : B
```

4. Validación cruzada quintuple OPCIONAL

La *Validación Cruzada* o *k-fold Cross Validation* consiste en tomar los datos originales y crear a partir de ellos dos conjuntos separados: un primer conjunto de entrenamiento y prueba, y un segundo conjunto de validación.

```r
# modelo 5-crossvalidation
model.5v.linear <- train(diagnostico ~ ., datos.train, method='svmLinear',
                trControl= trainControl(method='cv', number=5),
                tuneGrid= NULL, tuneLength=10 ,trace = FALSE)
```

```r
# plot(model.5v, alpha=0.6)
summary(model.5v.linear)
```

```
Length  Class   Mode
     1   ksvm     S4
```

```r
prediction <- predict(model.5v.linear, datos.test)                              # predict
res.linear.2<-table(prediction, datos.test$diagnostico)                                  #

# predict can also return the probability for each class:
cm_nb <- confusionMatrix(res.linear.2)
print(cm_nb)
```

```
Confusion Matrix and Statistics


prediction   B   M
         B 107   3
         M   2  59

               Accuracy : 0.9708
                 95% CI : (0.9331, 0.9904)
    No Information Rate : 0.6374
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9365

 Mcnemar's Test P-Value : 1

            Sensitivity : 0.9817
            Specificity : 0.9516
         Pos Pred Value : 0.9727
         Neg Pred Value : 0.9672
             Prevalence : 0.6374
         Detection Rate : 0.6257
   Detection Prevalence : 0.6433
      Balanced Accuracy : 0.9666

       'Positive' Class : B
```

```r
# modelo 5-crossvalidation
model.5v.radial <- train(diagnostico ~ ., datos.train, method='svmRadial',
                trControl= trainControl(method='cv', number=5),
                tuneGrid= NULL, tuneLength=10 ,trace = FALSE)

# plot(model.5v, alpha=0.6)
summary(model.5v.radial)
```

```
Length  Class   Mode
     1   ksvm     S4
```

```r
prediction <- predict(model.5v.radial, datos.test)                              # predict
res.radial.2<-table(prediction, datos.test$diagnostico)                                  #

# predict can also return the probability for each class:
cm_nb <- confusionMatrix(res.radial.2)
print(cm_nb)
```

```
Confusion Matrix and Statistics


prediction   B   M
         B 107   1
         M   2  61

               Accuracy : 0.9825
                 95% CI : (0.9496, 0.9964)
    No Information Rate : 0.6374
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9622

 Mcnemar's Test P-Value : 1

            Sensitivity : 0.9817
            Specificity : 0.9839
         Pos Pred Value : 0.9907
         Neg Pred Value : 0.9683
             Prevalence : 0.6374
         Detection Rate : 0.6257
```

```
   Detection Prevalence : 0.6316
      Balanced Accuracy : 0.9828

       'Positive' Class : B
```

Bootstrap

```
# Por defecto es Bootstrap, con 25 repeticiones para 3 posibles decay
# y 3 posibles sizes
model.bootstrap.linear <- train(diagnostico ~ ., datos.train, method='svmLinear', trace =
# we also add parameter 'preProc = c("center", "scale"))' at train() for centering and sca

summary(model.bootstrap.linear)
```

```
Length  Class   Mode
     1   ksvm     S4
```

```
prediction <- predict(model.bootstrap.linear, datos.test)                          # pred
res.gauss.2<-table(prediction, datos.test$diagnostico)                                 #

# predict can also return the probability for each class:
# prediction <- predict(model.bootstrap.linear, datos.test, type="prob")
# head(prediction)
confusionMatrix(res.gauss.2)
```

```
Confusion Matrix and Statistics


prediction   B   M
         B 107   3
         M   2  59

               Accuracy : 0.9708
                 95% CI : (0.9331, 0.9904)
    No Information Rate : 0.6374
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9365
```

```
Mcnemar's Test P-Value : 1

           Sensitivity : 0.9817
           Specificity : 0.9516
        Pos Pred Value : 0.9727
        Neg Pred Value : 0.9672
            Prevalence : 0.6374
        Detection Rate : 0.6257
  Detection Prevalence : 0.6433
     Balanced Accuracy : 0.9666

      'Positive' Class : B
```