Modelos Regresion

Maria Jose Bustamante - Nicolas Jadan

Biomedicina

```
library(ggplot2)
library(ggpubr)
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(glmnet) ## regresiones logisitcas
```

```
Loading required package: Matrix

Loaded glmnet 4.1-7
```

```
library(caret) ### bayes y knn
```

```
Loading required package: lattice
```

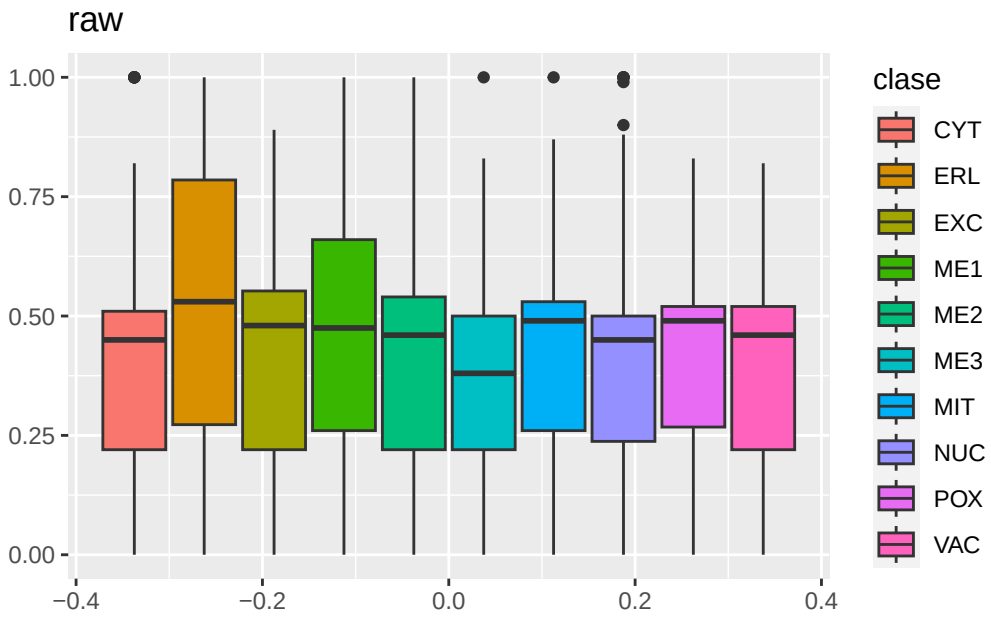
```r
library(e1071) ## bayes

#Quitamos la primera columna
datos <- read.table("./yeast.data",header = F)[,-1]

#Funciones de transformacion
min.max.mean <- function(X) apply(X,2,function(x) (x-mean(x))/(max(x)-min(x)))
min.max.median <- function(X) apply(X,2,function(x) (x-median(x))/(max(x)-min(x)))
min.max <- function(X) apply(X,2,function(x) (x-min(x))/(max(x)-min(x)))
zscore <- function(X) apply(X,2,function(x) (x-mean(x))/sd(x))
l2 <- function(X) apply(X,2,function(x) x/sqrt(sum(x^2)))


#Particion de datosdatos <- as.data.frame(datos)
datos.numericos <- datos[, which(unlist(lapply(datos, is.numeric)))]
clase <- datos$V10 <- as.factor(datos$V10)
colnames(datos.numericos) <- paste0("Var", rep(1:8))

#Procedemos a crear una lista con todas las transformaciones
datos.lista <- list(
  raw = bind_cols(datos.numericos,clase=clase),
  zscore = bind_cols(zscore(datos.numericos),
                     clase = clase),
  l2 = bind_cols(l2(datos.numericos), clase = clase),
  media = bind_cols(min.max.mean(datos.numericos), clase =
                      clase),
  mediana = bind_cols(min.max.median(datos.numericos), clase =
                        clase),
  min_max = bind_cols(min.max(datos.numericos),
  clase = clase))

#Al ser demasiadas variables, podemos realizar un melt
lista_graficos <- vector("list",length=length(datos.lista))
datos.melt <- lapply(datos.lista,reshape2::melt)
```

```
Using clase as id variables
```

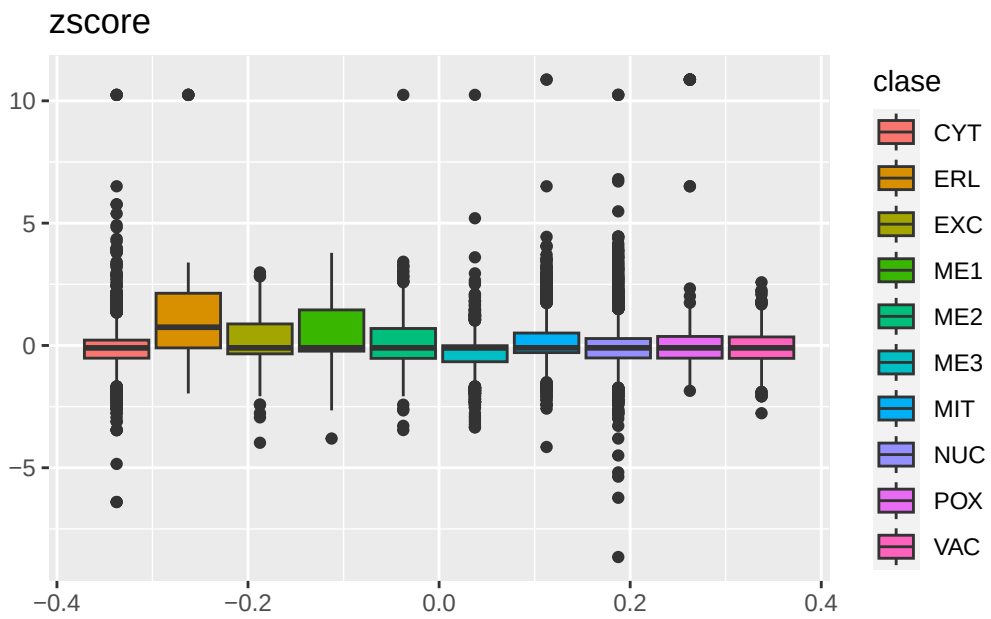
```
Using clase as id variables
Using clase as id variables
Using clase as id variables
Using clase as id variables
```

```
Using clase as id variables
```

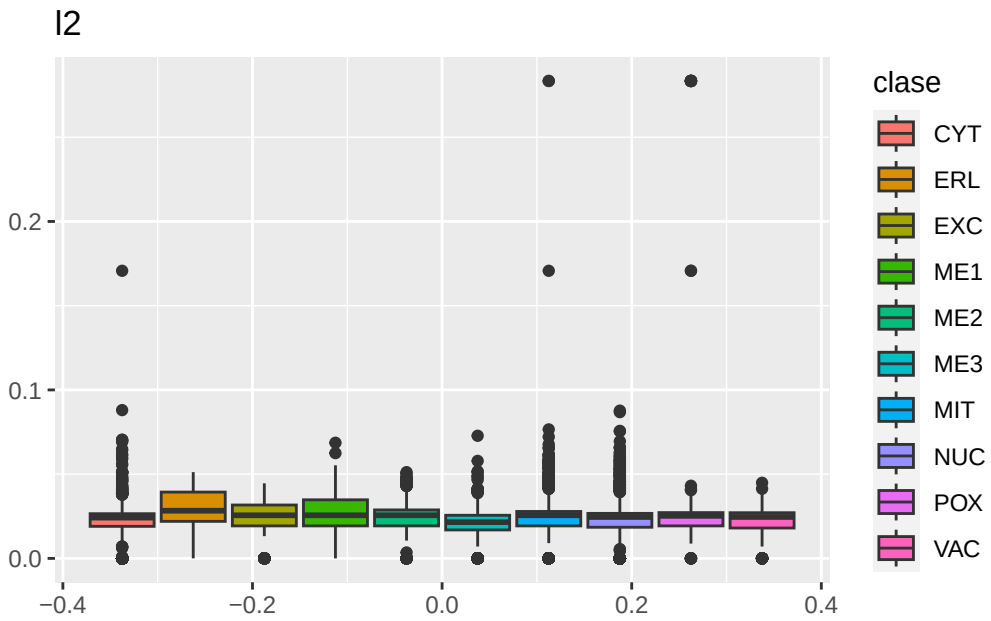
```
#Graficos
for(l in 1:length(datos.melt)){

  X <- datos.melt[[l]]
  nombre <- names(datos.melt)[l]
  lista_graficos[[l]] <- ggplot(X,aes(y=value,fill=clase))+geom_boxplot()+ggtitle(nombre)+


}

names(lista_graficos) <- paste0("plot",1:length(datos.lista))

lista_graficos$plot1
```



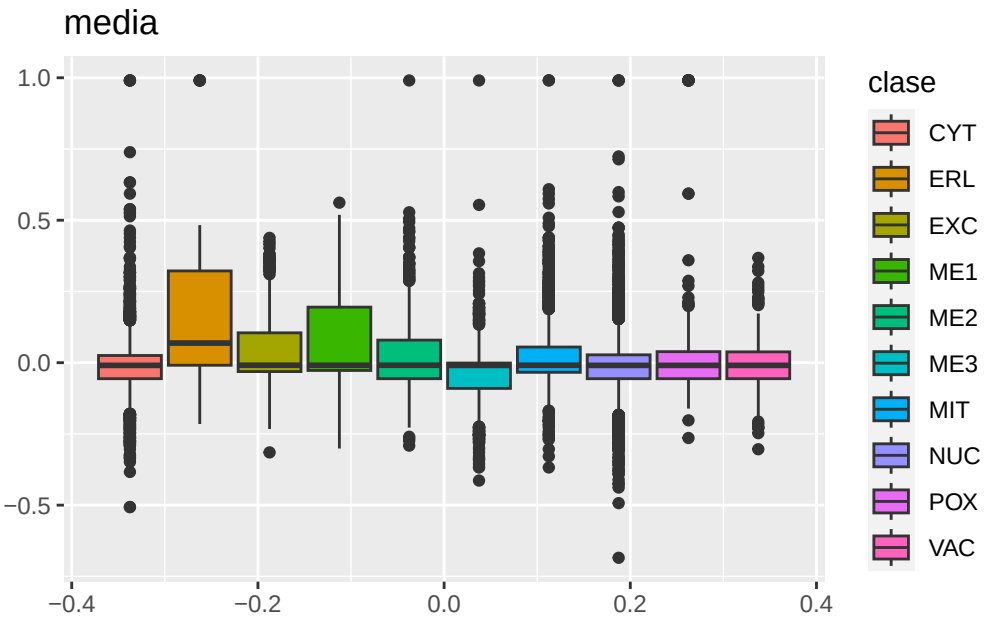

```
lista_graficos$plot2
```



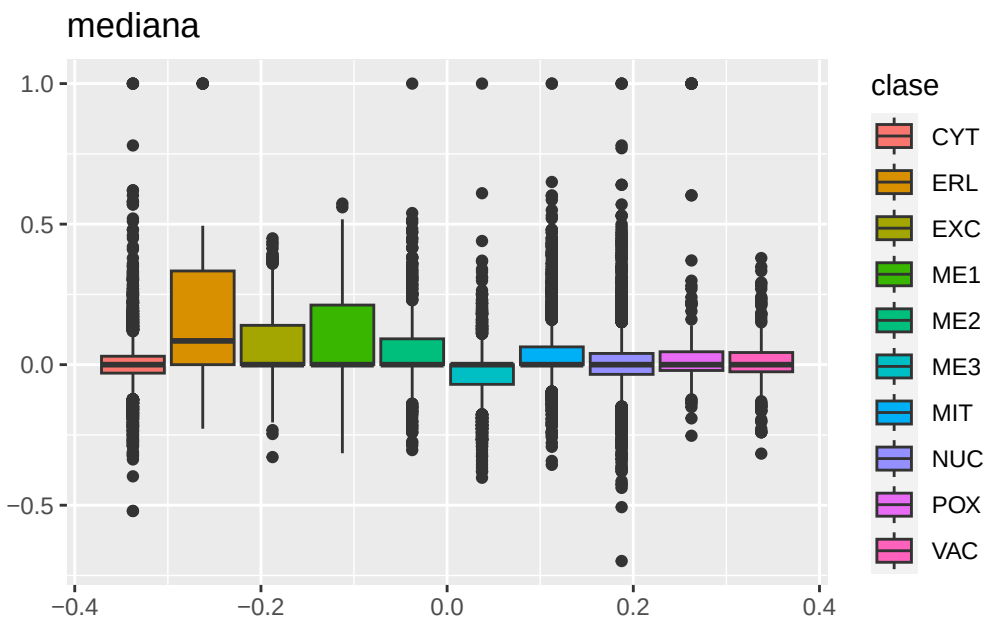

```
lista_graficos$plot3
```



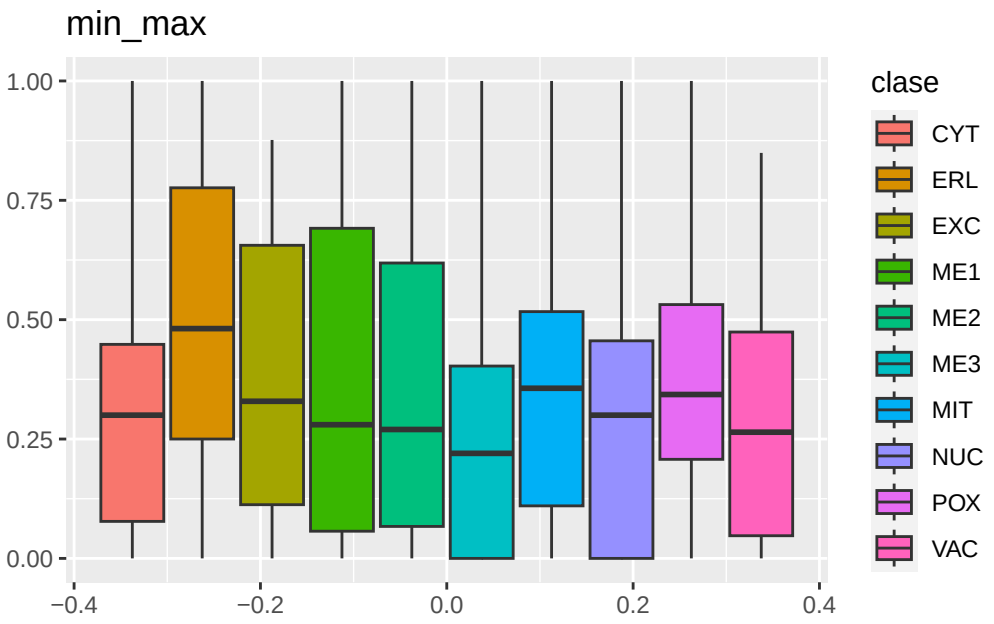
```
lista_graficos$plot4
```



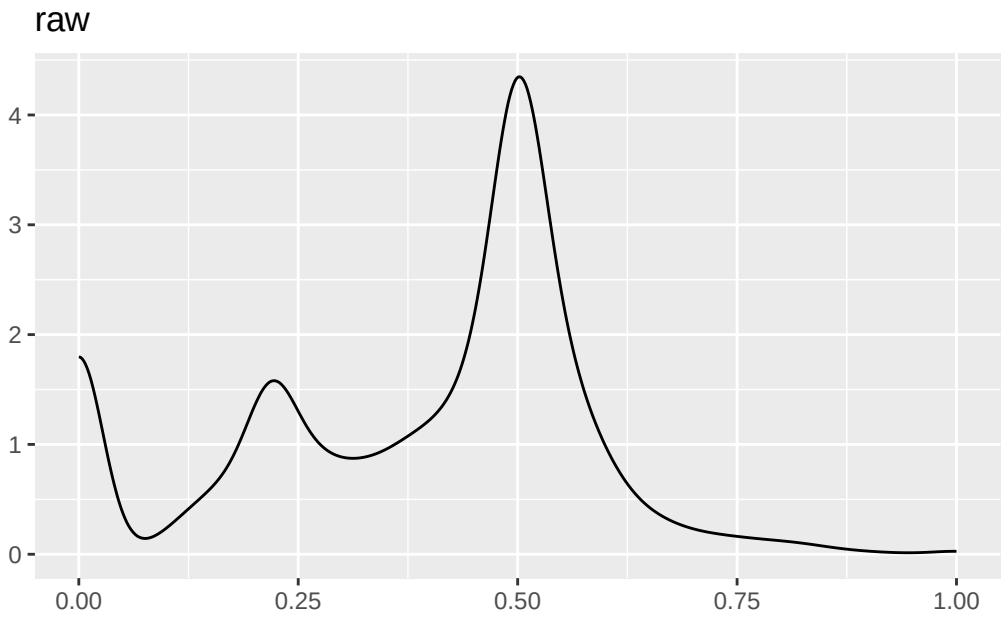

```
lista_graficos$plot5
```



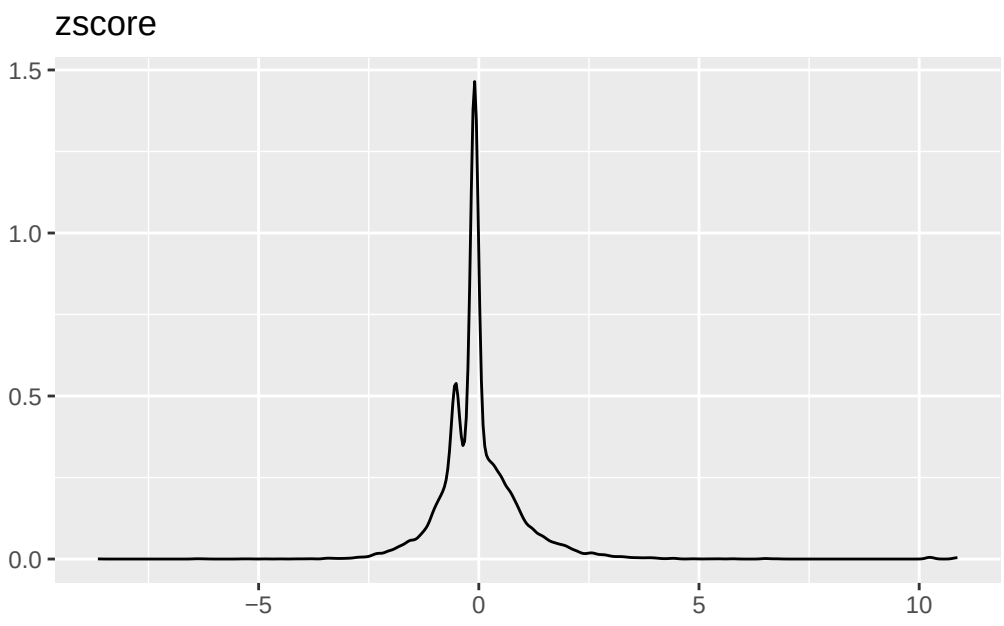

```
lista_graficos$plot6
```



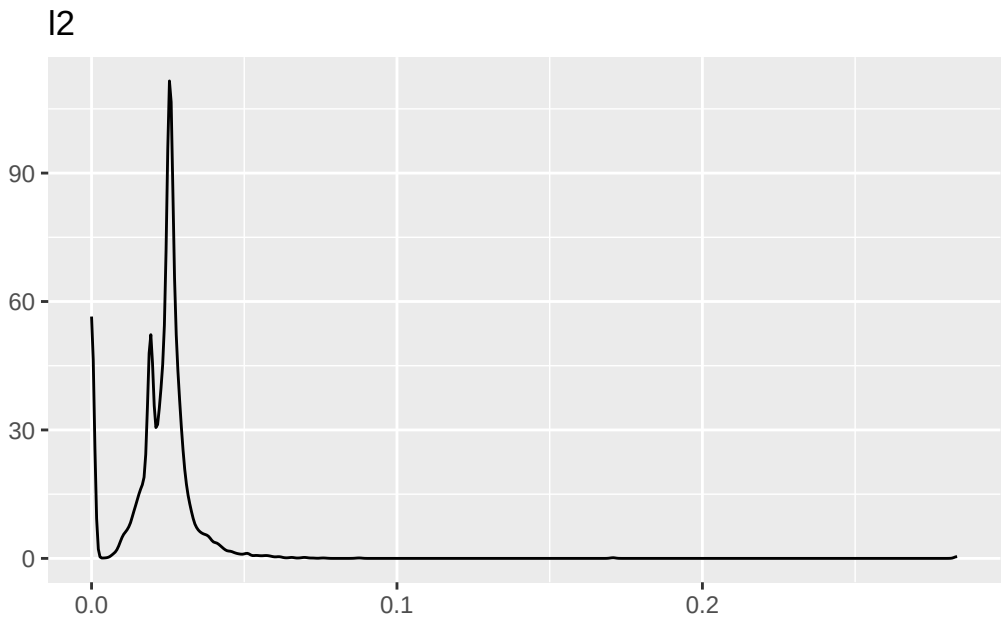
```r
#Grafico de densidad
for(l in 1:length(datos.melt)){

  X <- datos.melt[[l]]
  nombre <- names(datos.melt)[l]
  lista_graficos[[l]] <- ggplot(X,aes(x=value))+geom_density()+ggtitle(nombre)+xlab("")+yl


}

names(lista_graficos) <- paste0("plot",1:length(datos.lista))

lista_graficos$plot1
```



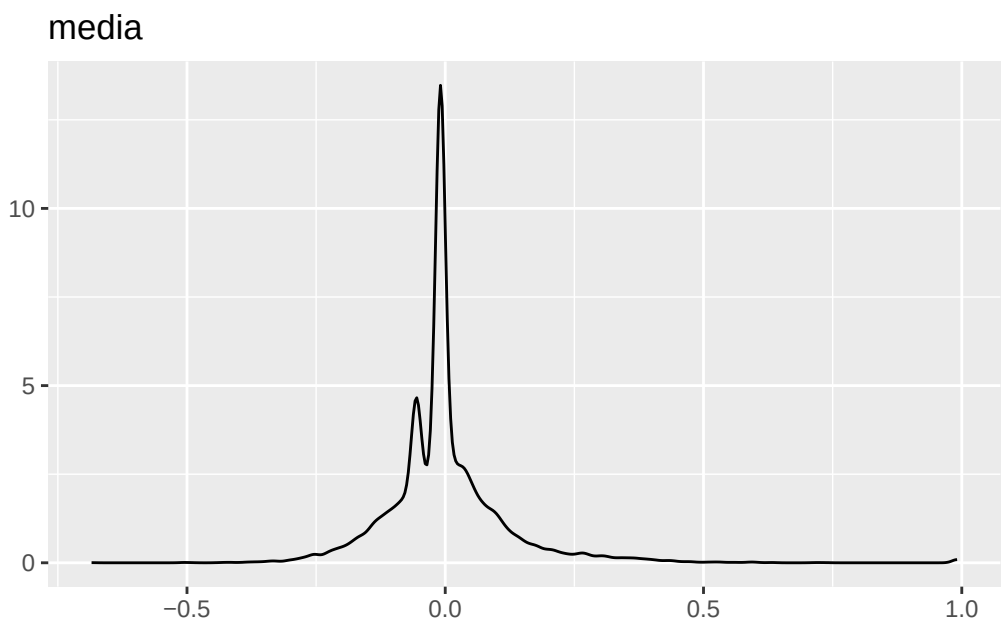

```r
lista_graficos$plot2
```



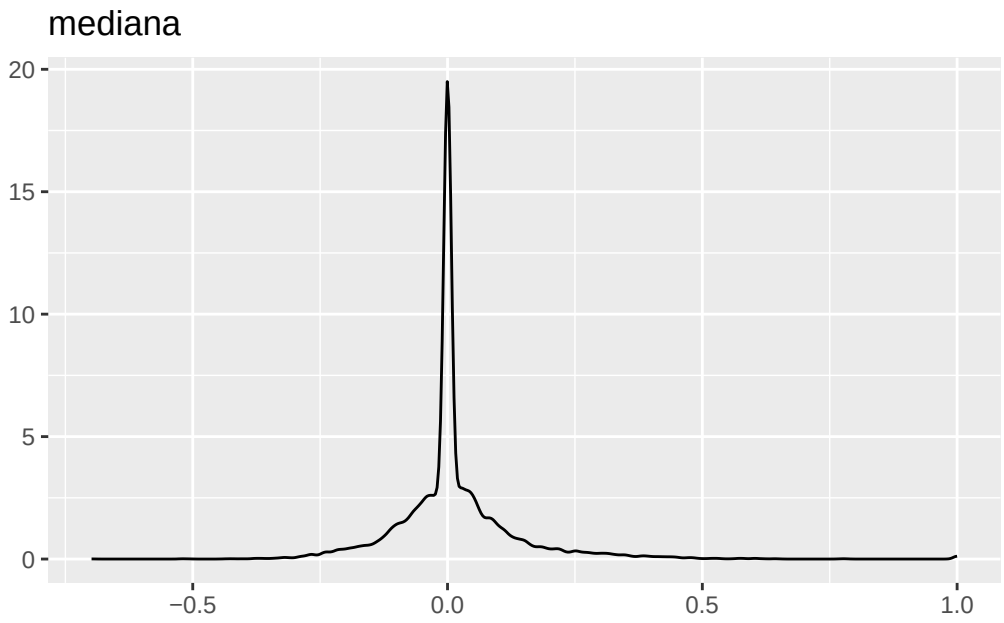

```
lista_graficos$plot3
```



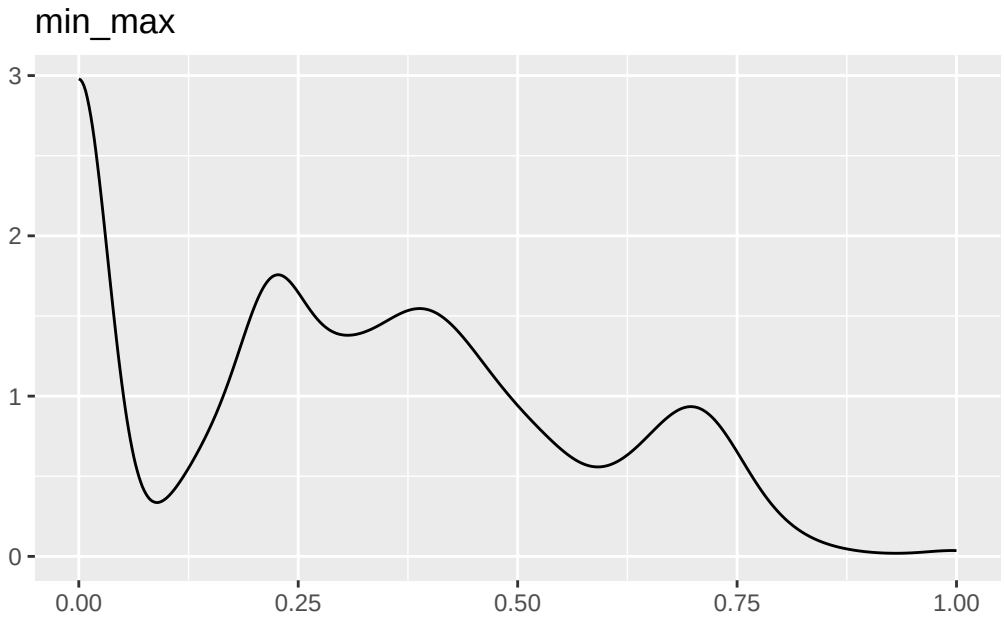
```
lista_graficos$plot4
```




```
lista_graficos$plot5
```




```
lista_graficos$plot6
```



```r
#Fijamos la semilla y la muestra
set.seed(123456789)
trControl <- trainControl(method = 'cv', number = 10)
n  <- nrow(datos)
idx <- sample(1:n,size=n*0.7,replace=F)
lambda_seq <- seq(0.01, 1, by = 0.01)

#Para conjunto de datos podemos realizar el split
entrenamiento <- lapply(datos.lista, function(x) x[idx,])
test <- lapply(datos.lista, function(x) x[-idx,])

#Regresion logistica Lineal
set.seed(123456789)
myfnlog <- function(x) train(clase ~ ., data = x, method = "multinom", trControl = trContr
logistica.lista <- lapply(entrenamiento,myfnlog)
logisita.pred <- vector("list",length = length(datos.lista))
for(l in 1:length(datos.lista)){
  logisita.pred[[l]] <- predict(logistica.lista[[l]],test[[l]])
}
names(logisita.pred) <- names(datos.lista)
accuracy <- vector("numeric",length = length(datos.lista))
for(l in 1:length(datos.lista)){
  accuracy[l] <- confusionMatrix(test$raw$clase,logisita.pred[[l]])$overall[1]
}
names(accuracy) <- names(datos.lista)
accuracy_logis<-accuracy

#Ridge
set.seed(123456789)
myfnridge <- function(x) train(clase ~ ., data = x, method = "glmnet", trControl = trContr

logistica.lista <- lapply(entrenamiento,myfnridge)
logisita.pred <- vector("list",length = length(datos.lista))
for(l in 1:length( datos.lista)){
  logisita.pred[[l]] <- predict(logistica.lista[[l]],test[[l]])
}
names(logisita.pred) <- names(datos.lista)
accuracy <- vector("numeric",length = length(datos.lista))
for(l in 1:length(datos.lista)){
  accuracy[l] <- confusionMatrix(test$raw$clase,logisita.pred[[l]])$overall[1]
}
```

```r
names(accuracy) <- names(datos.lista)
accuracy_ridge <- accuracy

#Lasso
set.seed(123456789)
myfnlasso <- function(x) train(clase ~ ., data = x, method = "glmnet", trControl = trContr

logistica.lista <- lapply(entrenamiento,myfnlasso)
logisita.pred <- vector("list",length = length(datos.lista))
for(l in 1:length( datos.lista)){
  logisita.pred[[l]] <- predict(logistica.lista[[l]],test[[l]])
}
names(logisita.pred) <- names(datos.lista)
accuracy <- vector("numeric",length = length(datos.lista))
for(l in 1:length(datos.lista)){
  accuracy[l] <- confusionMatrix(test$raw$clase,logisita.pred[[l]])$overall[1]
}
names(accuracy) <- names(datos.lista)
accuracy_lasso <- accuracy

#Knn
set.seed(123456789)
myfnknn <- function(x) train(clase ~ ., data = x, method = "knn", trControl = trControl, t

logistica.lista <- lapply(entrenamiento,myfnknn)
logisita.pred <- vector("list",length = length(datos.lista))
for(l in 1:length( datos.lista)){
  logisita.pred[[l]] <- predict(logistica.lista[[l]],test[[l]])
}
names(logisita.pred) <- names(datos.lista)
accuracy <- vector("numeric",length = length(datos.lista))
for(l in 1:length(datos.lista)){
  accuracy[l] <- confusionMatrix(test$raw$clase,logisita.pred[[l]])$overall[1]
}
names(accuracy) <- names(datos.lista)
accuracy_knn <- accuracy


#Naive Bayes
set.seed(123456789)
myfnknn <- function(x) train(clase ~ ., data = x, method = "naive_bayes", trControl = trCo
```

```r
logistica.lista <- lapply(entrenamiento,myfnknn)
logisita.pred <- vector("list",length = length(datos.lista))
for(l in 1:length( datos.lista)){
  logisita.pred[[l]] <- predict(logistica.lista[[l]],test[[l]])
}
names(logisita.pred) <- names(datos.lista)
accuracy <- vector("numeric",length = length(datos.lista))
for(l in 1:length(datos.lista)){
  accuracy[l] <- confusionMatrix(test$raw$clase,logisita.pred[[l]])$overall[1]
}
names(accuracy) <- names(datos.lista)
accuracy_bayes <- accuracy
```

```r
#Matriz Accuracy
#Crear la matriz de 5x6
matriz <- matrix(nrow = 5, ncol = 6)

# Asignar los vectores a las filas de la matriz
matriz[1, ] <- accuracy_logis
matriz[2, ] <- accuracy_ridge
matriz[3, ] <- accuracy_lasso
matriz[4, ] <- accuracy_knn
matriz[5, ] <- accuracy_bayes

# Asignar nombres a las filas y columnas
filas <- c("Regresion Logis", "Ridge", "Lassso", "Knn", "Naive Bayes")
columnas <- c("raw", "zscore", "l2", "media", "mediana", "min_max")

# Asignar nombres a las filas y columnas de la matriz
rownames(matriz) <- filas
colnames(matriz) <- columnas

# Imprimir la matriz
print(matriz)
```

```
                    raw    zscore        l2     media   mediana   min_max
Regresion Logis 0.5919283 0.5919283 0.5964126 0.5919283 0.5941704 0.5919283
Ridge           0.5919283 0.5941704 0.5941704 0.5941704 0.5941704 0.5941704
Lassso          0.5919283 0.5919283 0.5919283 0.5919283 0.5919283 0.5919283
Knn             0.5852018 0.5717489 0.5986547 0.5941704 0.5874439 0.6031390
```

```
Naive Bayes     0.4080717 0.5336323 0.4596413 0.5336323 0.4125561 0.4125561
```

```r
# Encontrar el número máximo y su posición en la matriz
indice_maximo <- which.max(matriz)
fila_maximo <- row(matriz)[indice_maximo]
columna_maximo <- col(matriz)[indice_maximo]

# Obtener el nombre de la fila y columna correspondientes
nombre_fila <- rownames(matriz)[fila_maximo]
nombre_columna <- colnames(matriz)[columna_maximo]

# Imprimir el número máximo y su posición
cat("El mayor porcentaje de los diferentes metodos es", matriz[fila_maximo, columna_maximo
```

```
El mayor porcentaje de los diferentes metodos es 0.603139 %
```

```r
cat("Con el metodo", nombre_fila, "y la transformacion", nombre_columna, "\n")
```

```
Con el metodo Knn y la transformacion min_max
```