Assignment 1

Rodríguez González, Álvaro

10/6/2021

Preprocesamiento: Carga datos y librerias

Cargamos las librerias a usar:

```
library(tidyverse)
library(GGally)
library(MLTools)
library(caret)
library(ROCR)
```

Cargamos los datos:

```
datos <- read.table("./data/Diabetes.csv", sep = ";", header = TRUE)
```

Análisis Exploratorio

Resumen general del datasets

```
str(datos)
```

```
## 'data.frame':    768 obs. of  9 variables:
##  $ PREGNANT     : int  6 1 8 1 0 5 3 10 2 8 ...
##  $ GLUCOSE      : int  148 85 183 89 137 116 78 115 197 125 ...
##  $ BLOODPRESS   : int  72 66 64 66 40 74 50 0 70 96 ...
##  $ SKINTHICKNESS: int  35 29 0 23 35 0 32 0 45 0 ...
##  $ INSULIN      : int  0 0 0 94 168 0 88 0 543 0 ...
##  $ BODYMASSINDEX: num  33.6 26.6 23.3 28.1 43.1 25.6 31 35.3 30.5 0 ...
##  $ PEDIGREEFUNC : num  0.627 0.351 0.672 0.167 2.288 ...
##  $ AGE          : int  50 31 32 21 33 30 26 29 53 54 ...
##  $ DIABETES     : int  1 0 1 0 1 0 1 0 1 1 ...
```

```
summary(datos)
```

```
##     PREGNANT         GLUCOSE        BLOODPRESS     SKINTHICKNESS
##  Min.   : 0.000   Min.   :  0.0   Min.   :  0.00   Min.   : 0.00
##  1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 62.00   1st Qu.: 0.00
##  Median : 3.000   Median :117.0   Median : 72.00   Median :23.00
##  Mean   : 3.845   Mean   :120.9   Mean   : 69.11   Mean   :20.54
##  3rd Qu.: 6.000   3rd Qu.:140.2   3rd Qu.: 80.00   3rd Qu.:32.00
##  Max.   :17.000   Max.   :199.0   Max.   :122.00   Max.   :99.00
##     INSULIN      BODYMASSINDEX    PEDIGREEFUNC         AGE
##  Min.   :  0.0   Min.   : 0.00   Min.   :0.0780   Min.   :21.00
##  1st Qu.:  0.0   1st Qu.:27.30   1st Qu.:0.2437   1st Qu.:24.00
```

```
##  Median : 30.5   Median :32.00   Median :0.3725   Median :29.00
##  Mean   : 79.8   Mean   :31.99   Mean   :0.4719   Mean   :33.24
##  3rd Qu.:127.2   3rd Qu.:36.60   3rd Qu.:0.6262   3rd Qu.:41.00
##  Max.   :846.0   Max.   :67.10   Max.   :2.4200   Max.   :81.00
##     DIABETES
##  Min.   :0.000
##  1st Qu.:0.000
##  Median :0.000
##  Mean   :0.349
##  3rd Qu.:1.000
##  Max.   :1.000
```

Realizamos algunos cambios en el dataset:

```
datos <- datos %>%
  mutate(DIABETES = ifelse(DIABETES == 1, "Si", "No"))
datos <- datos %>%
  mutate(DIABETES = as.factor(DIABETES))
str(datos)
```

```
## 'data.frame':    768 obs. of  9 variables:
##  $ PREGNANT     : int  6 1 8 1 0 5 3 10 2 8 ...
##  $ GLUCOSE      : int  148 85 183 89 137 116 78 115 197 125 ...
##  $ BLOODPRESS   : int  72 66 64 66 40 74 50 0 70 96 ...
##  $ SKINTHICKNESS: int  35 29 0 23 35 0 32 0 45 0 ...
##  $ INSULIN      : int  0 0 0 94 168 0 88 0 543 0 ...
##  $ BODYMASSINDEX: num  33.6 26.6 23.3 28.1 43.1 25.6 31 35.3 30.5 0 ...
##  $ PEDIGREEFUNC : num  0.627 0.351 0.672 0.167 2.288 ...
##  $ AGE          : int  50 31 32 21 33 30 26 29 53 54 ...
##  $ DIABETES     : Factor w/ 2 levels "No","Si": 2 1 2 1 2 1 2 1 2 2 ...
```

Vemos que no hay datos nulos y por lo tanto trabajaremos con todas las filas que hemos cargado

Resumen de pacientes con diabetes

```
datos %>%
  count(DIABETES) %>%
  mutate(frecuenciaAbosulta = n, frecuenciaRelativa = frecuenciaAbosulta/sum(n), n = NULL)
```

```
##   DIABETES frecuenciaAbosulta frecuenciaRelativa
## 1       No                500          0.6510417
## 2       Si                268          0.3489583
```

```
ggplot(datos) +
  geom_boxplot(aes(AGE))
```

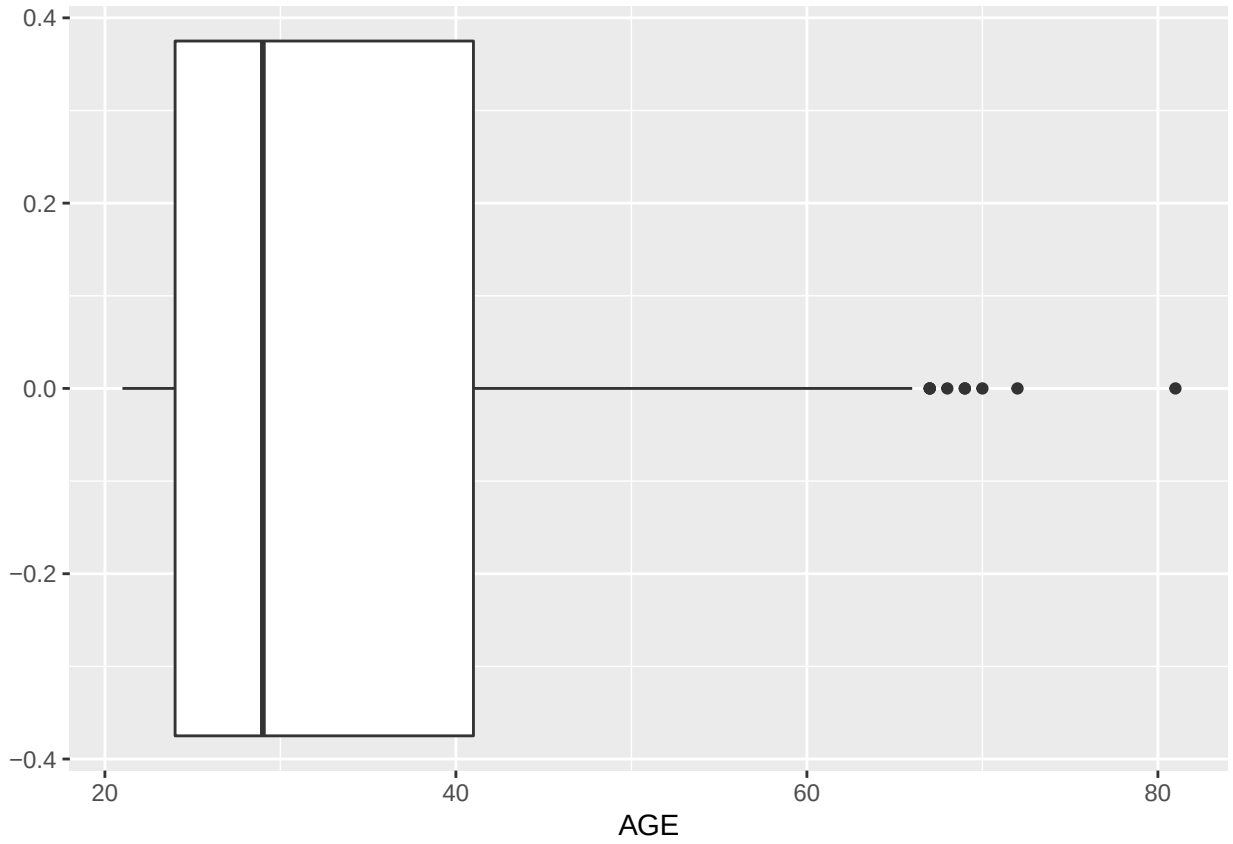


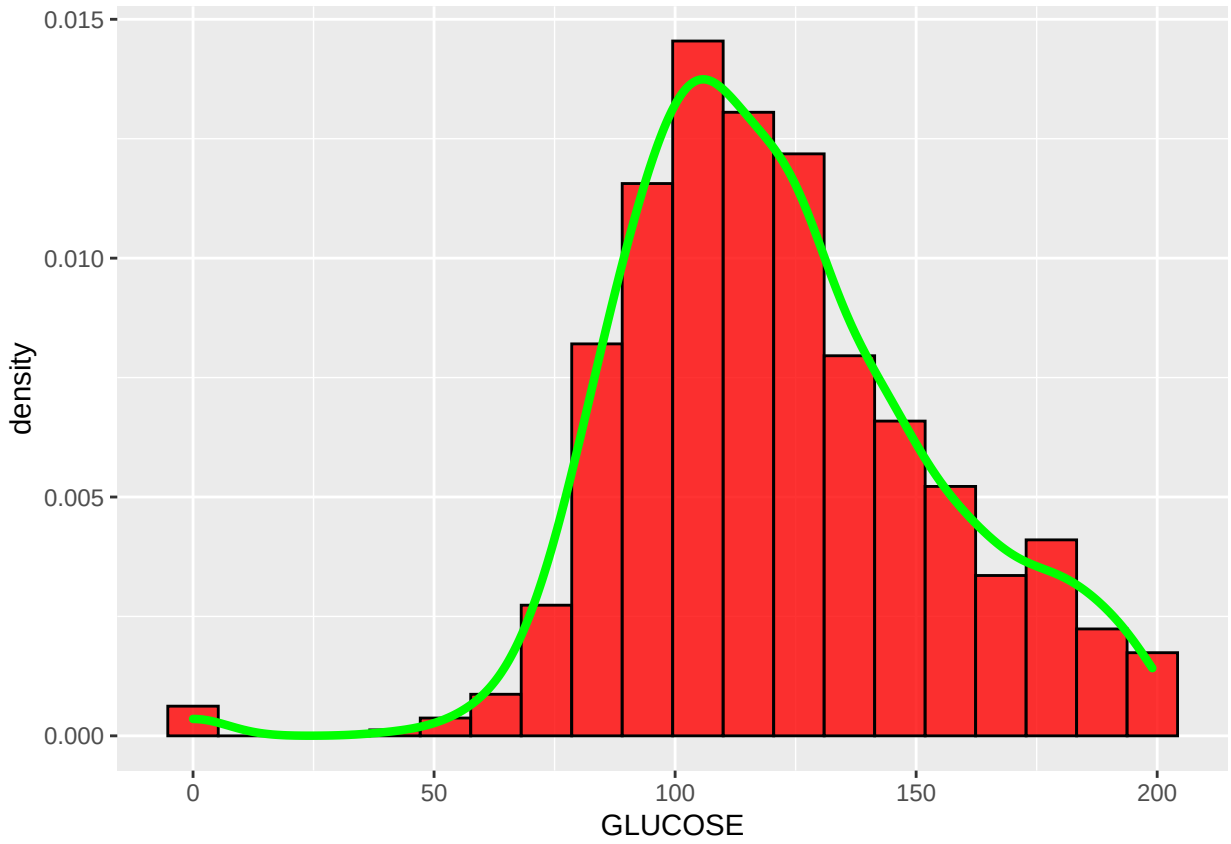
Tabla de frecuencias absolutas de pregnant

```
table(datos$PREGNANT)
```

```
##
##   0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  17
## 111 135 103  75  68  57  50  45  38  28  24  11   9  10   2   1   1
```

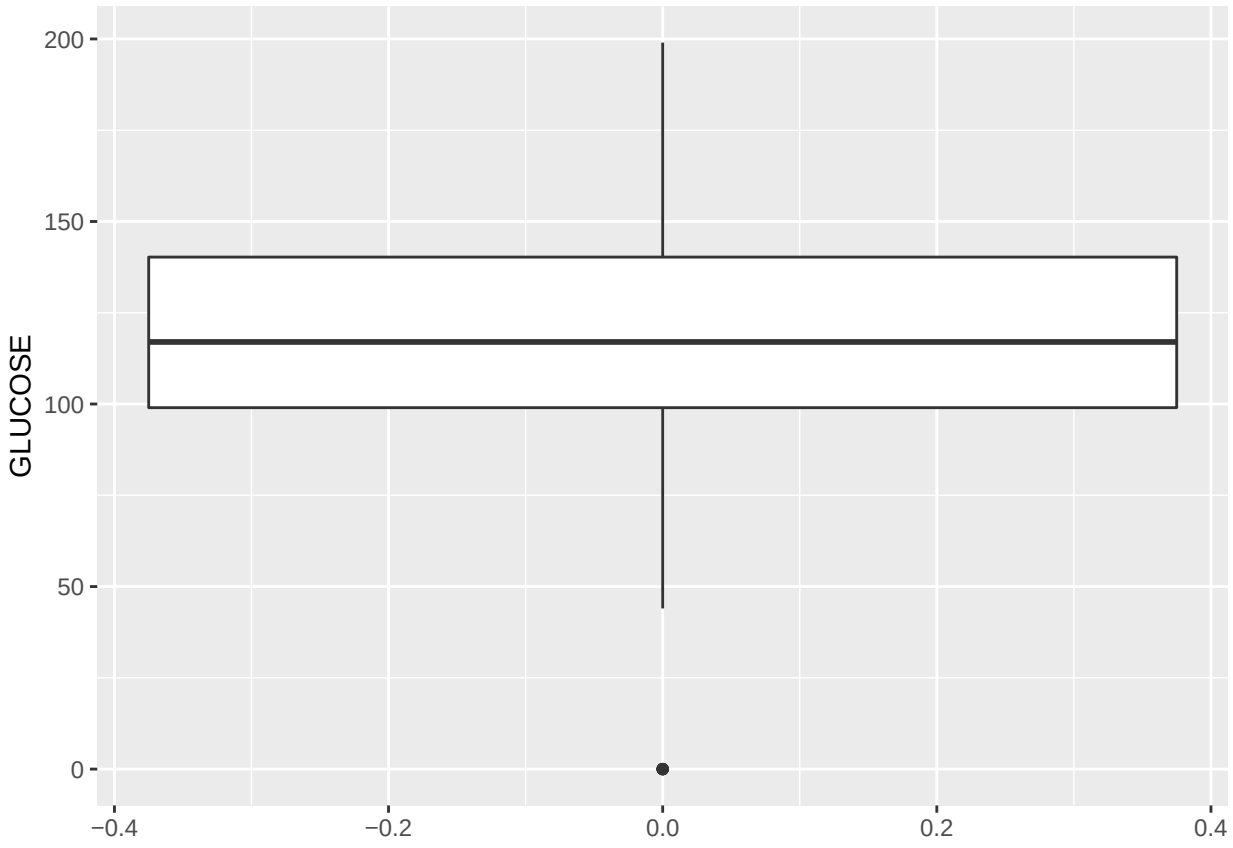
Histograma de Glucose

```
ggplot(datos, aes(x = GLUCOSE)) +
  geom_histogram(aes(y = stat(density)), fill = "red", color = "black", bins = 20, alpha = 0.8) +
  geom_density(color = "green", size = 1.5)
```

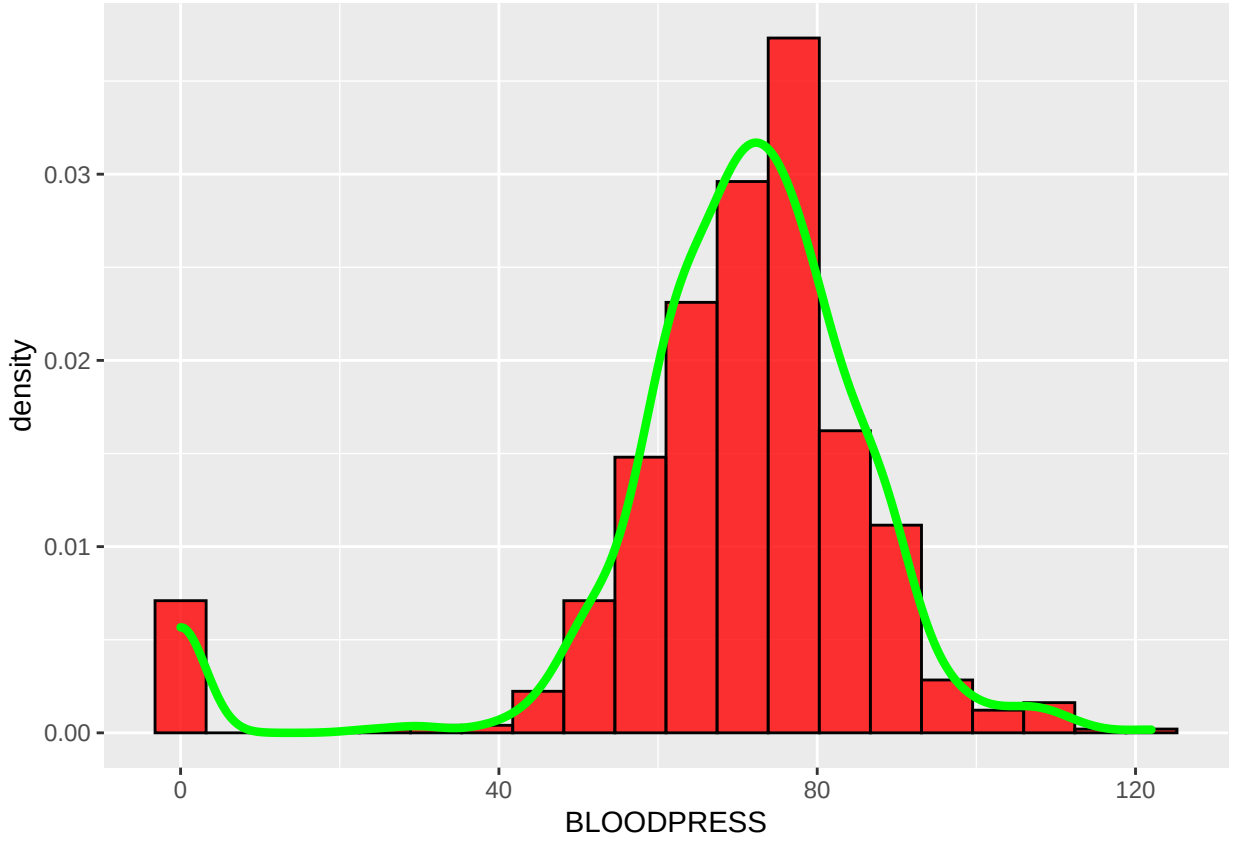


Analizamos la varibale Glucosa

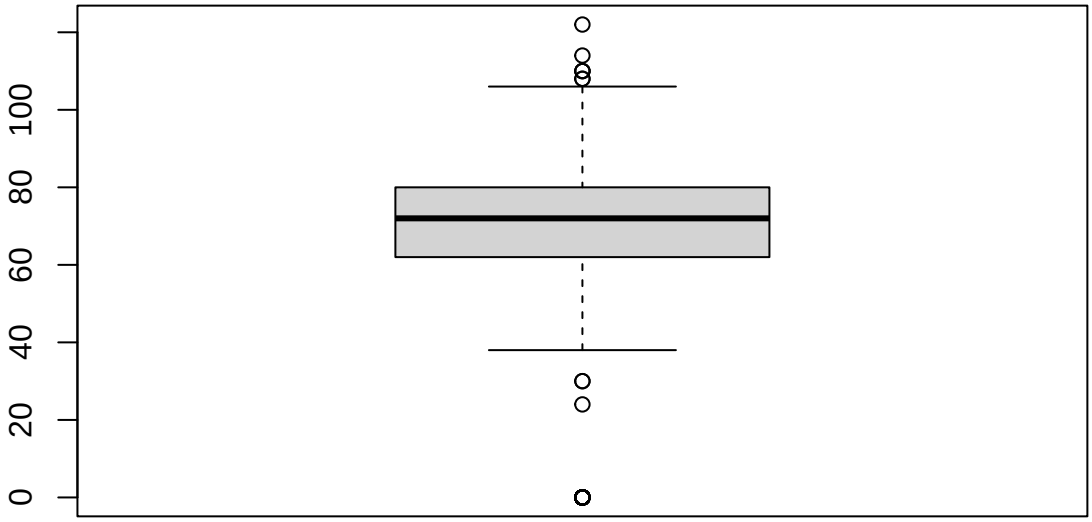
```
ggplot(datos) +
  geom_boxplot(aes(y = GLUCOSE))
```




```
ggplot(datos, aes(x = BLOODPRESS)) +
  geom_histogram(aes(y = stat(density)), color = "black", fill = "red", bins = 20, alpha = 0.8) +
  geom_density(aes(BLOODPRESS), color = "green", size = 1.5)
```

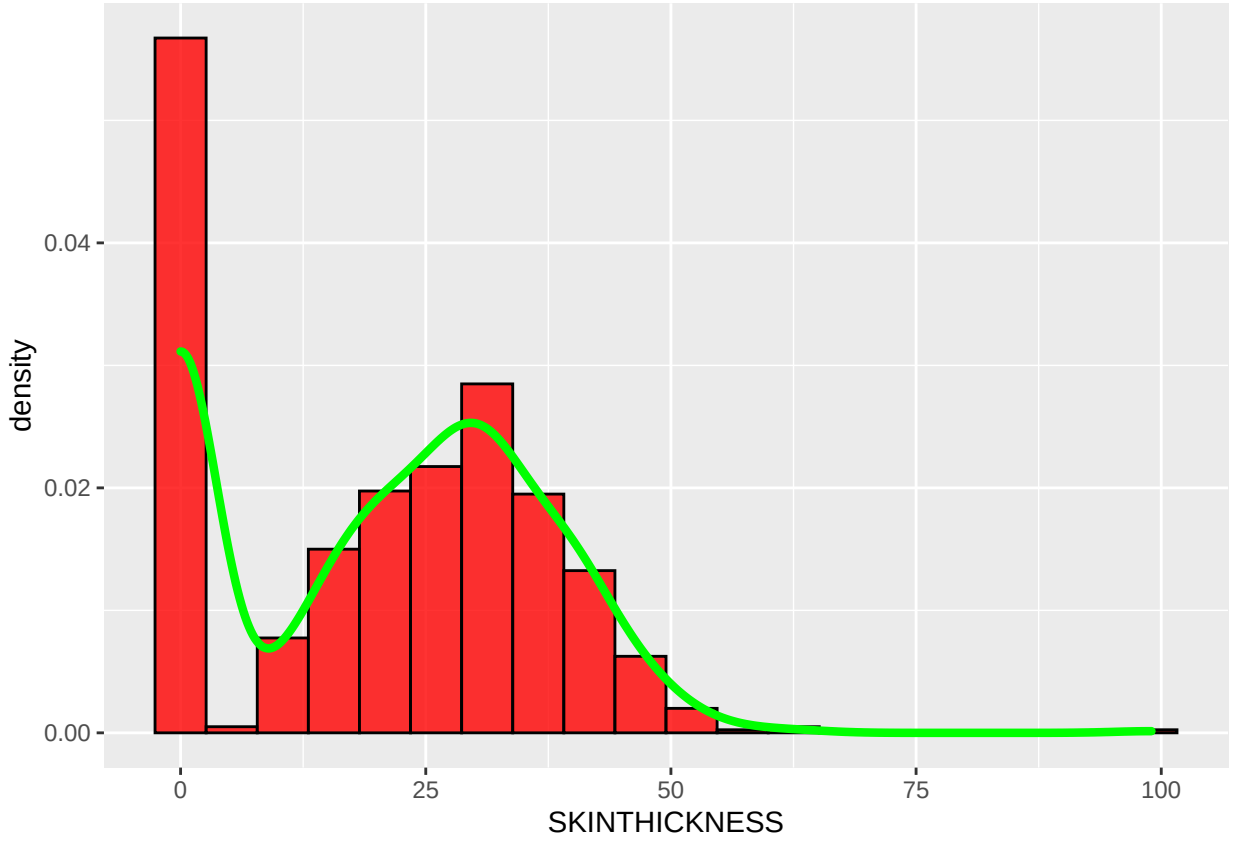



```
boxplot(datos$BLOODPRESS)
```

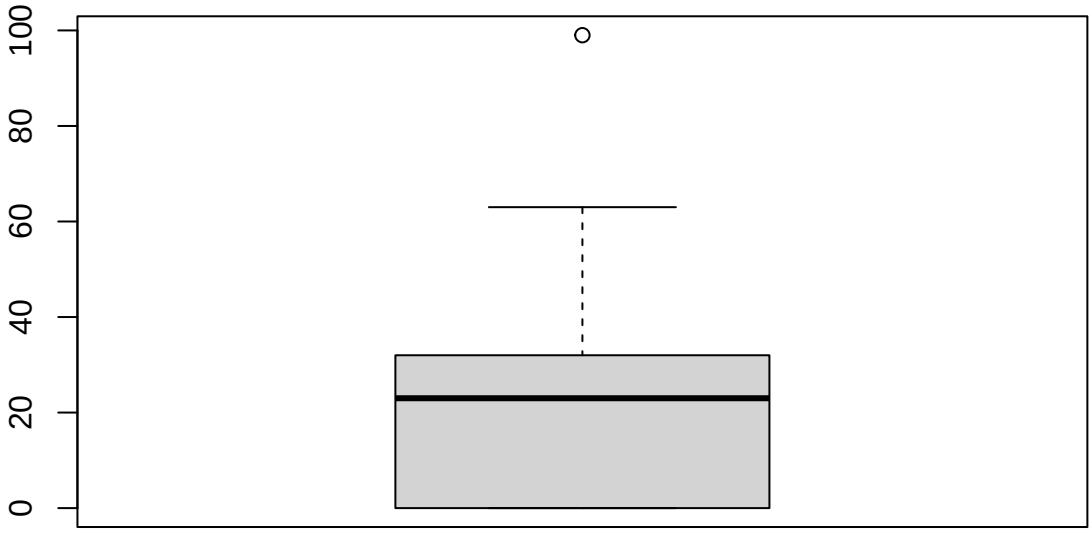


Analizamos la variable SKINTHICKNESS

```
ggplot(datos, aes(x = SKINTHICKNESS)) +
  geom_histogram(aes(y = stat(density)), color = "black", fill = "red", bins = 20, alpha = 0.8) +
  geom_density(aes(SKINTHICKNESS), color = "green", size = 1.5)
```

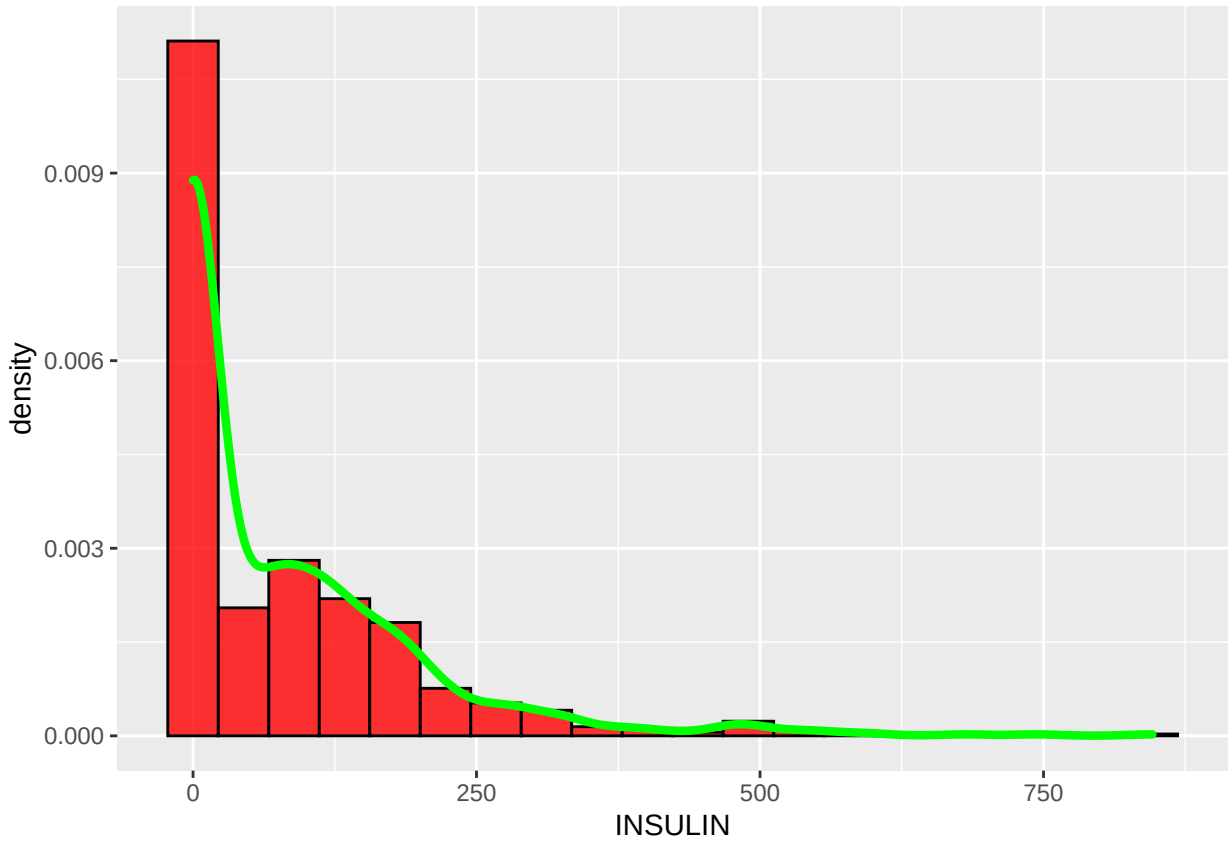



```
boxplot(datos$SKINTHICKNESS)
```

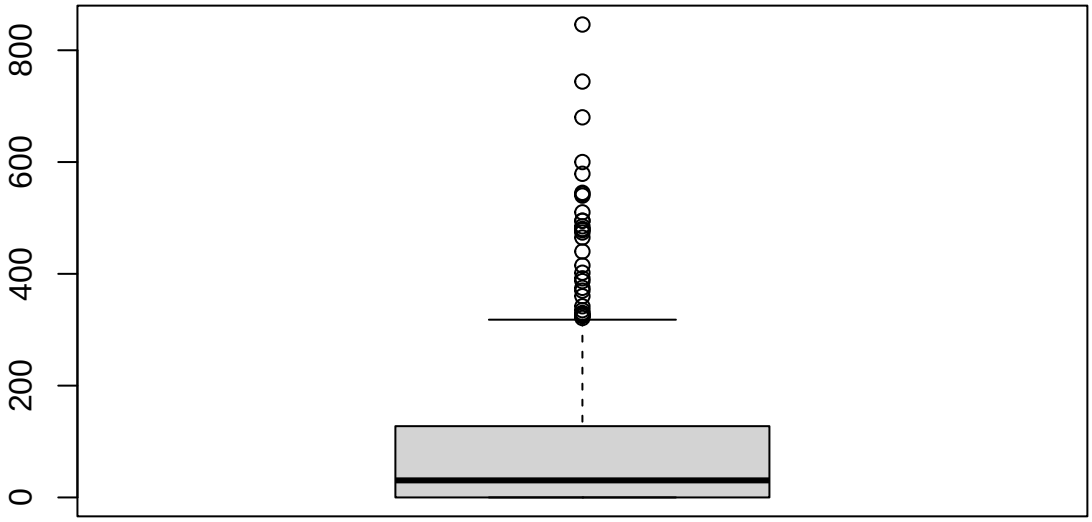


Vemos la variable INSULIN

```
ggplot(datos, aes(x = INSULIN)) +
  geom_histogram(aes(y = stat(density)), color = "black", fill = "red", bins = 20, alpha = 0.8) +
  geom_density(aes(INSULIN), size = 1.5, color = "green")
```

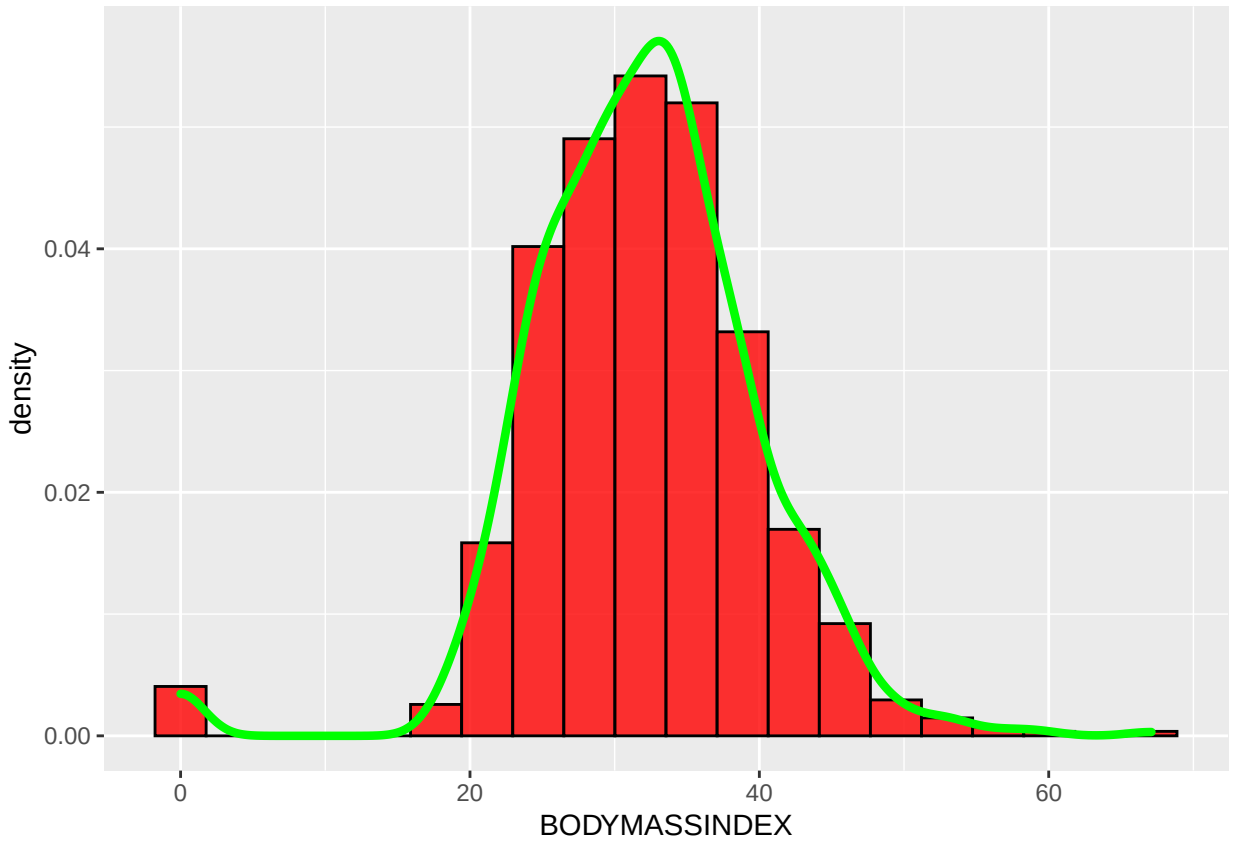



```
boxplot(datos$INSULIN)
```



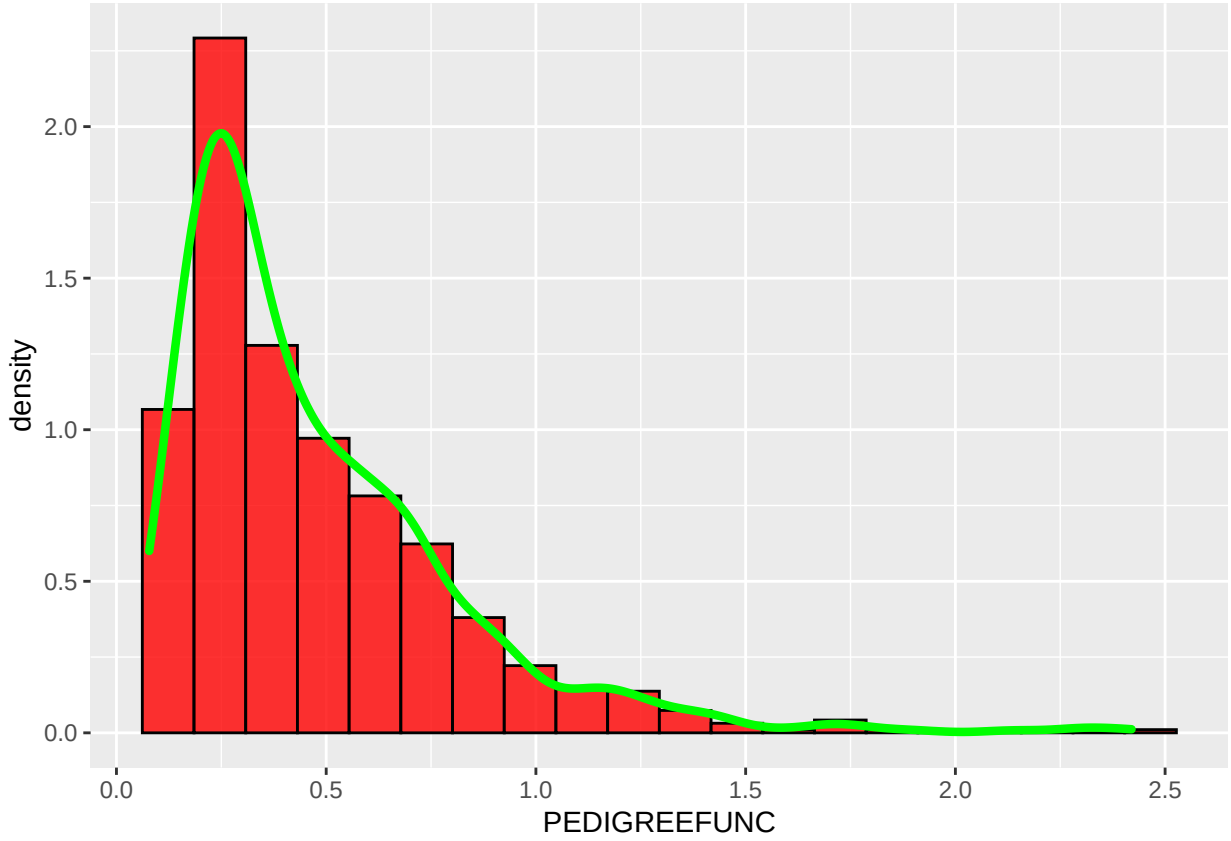
Vemos la variable BODYMASSINDEX

```
ggplot(datos, aes(x = BODYMASSINDEX)) +
  geom_histogram(aes(y = stat(density)), color = "black", fill = "red", bins = 20, alpha = 0.8) +
  geom_density(aes(BODYMASSINDEX), color = "green", size = 1.5)
```

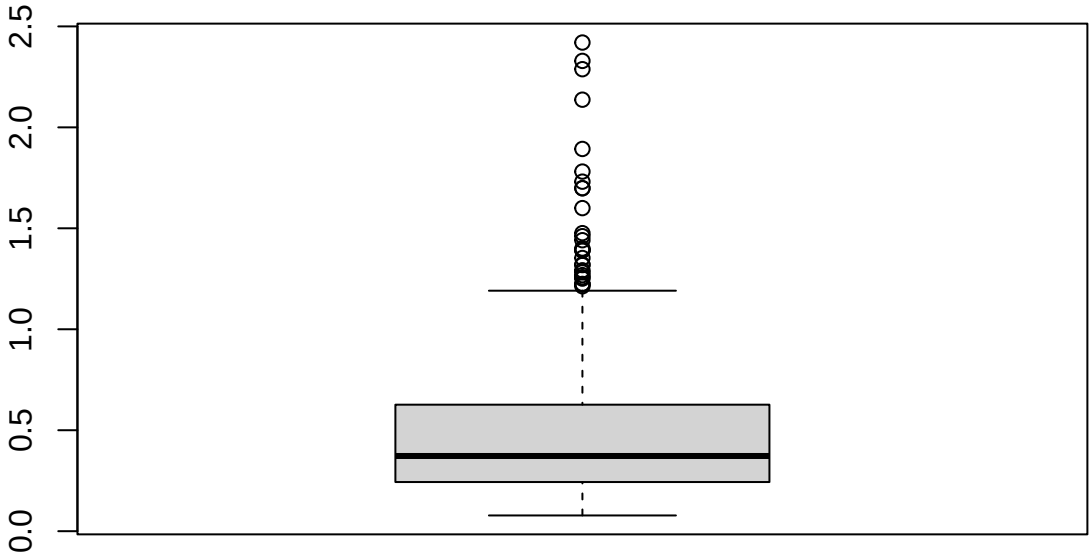


Analizamos por último la variable PEDIGREEFUNC

```
ggplot(datos) +
  geom_histogram(aes(x = PEDIGREEFUNC, y = stat(density)), color = "black", fill = "red", bins = 20, al
  geom_density(aes(PEDIGREEFUNC), color = "green", size = 1.5)
```

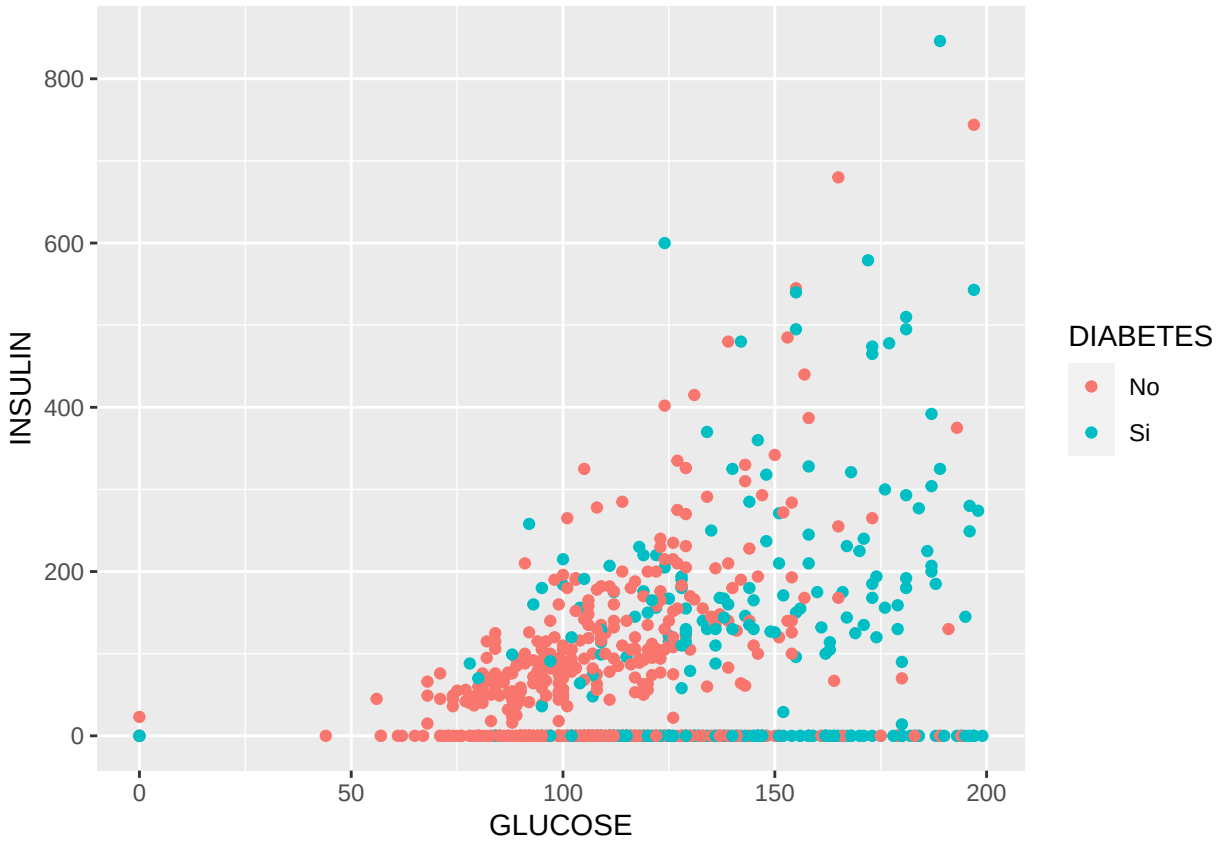



```
boxplot(datos$PEDIGREEFUNC)
```




```
ggplot(datos,aes(x = GLUCOSE , y = BLOODPRESS)) +
  geom_point(aes(color = DIABETES))
```


```
ggplot(datos,aes(x = GLUCOSE , y = INSULIN)) +
  geom_point(aes(color = DIABETES))
```



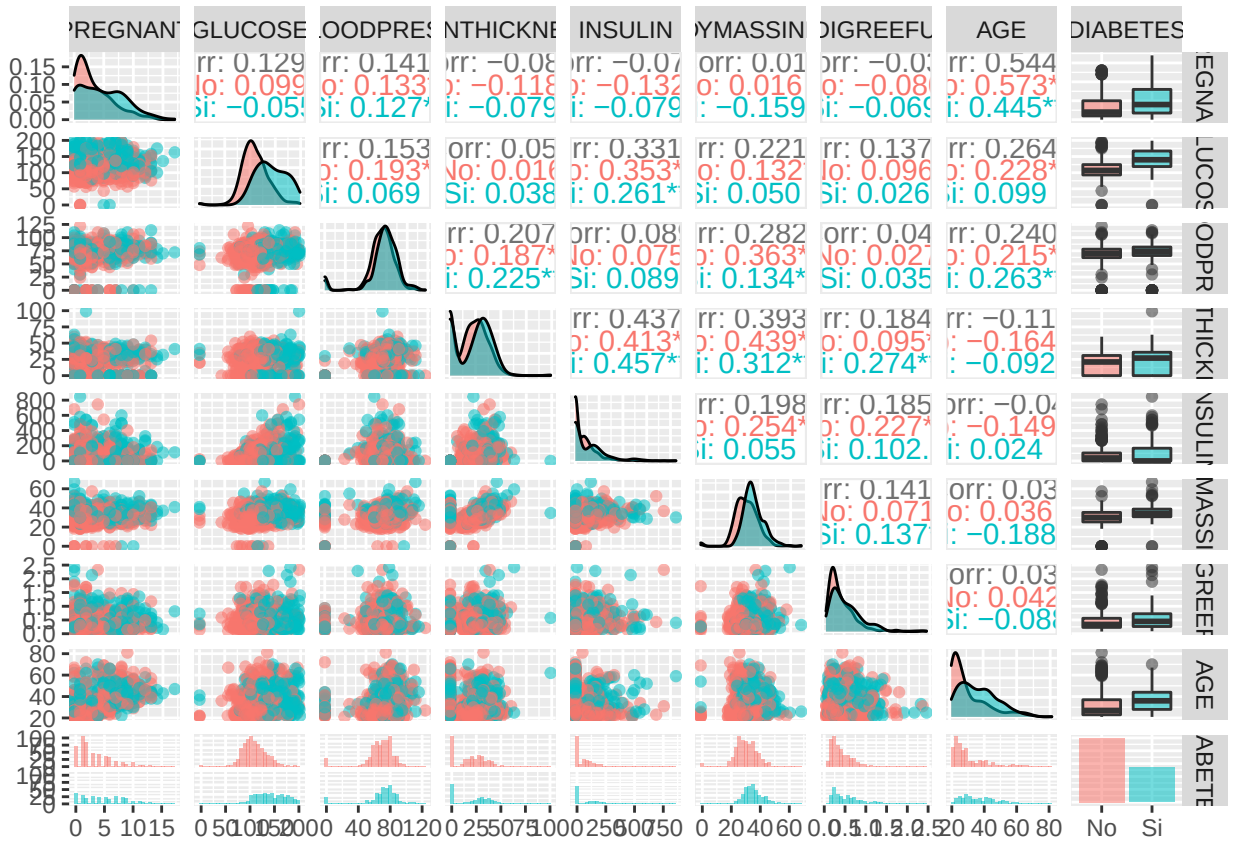

```
ggplot(datos,aes(x = GLUCOSE , y = BODYMASSINDEX)) +
  geom_point(aes(color = DIABETES))
```


```
ggplot(datos,aes(y = BODYMASSINDEX , x = BLOODPRESS)) +
  geom_point(aes(color = DIABETES))
```

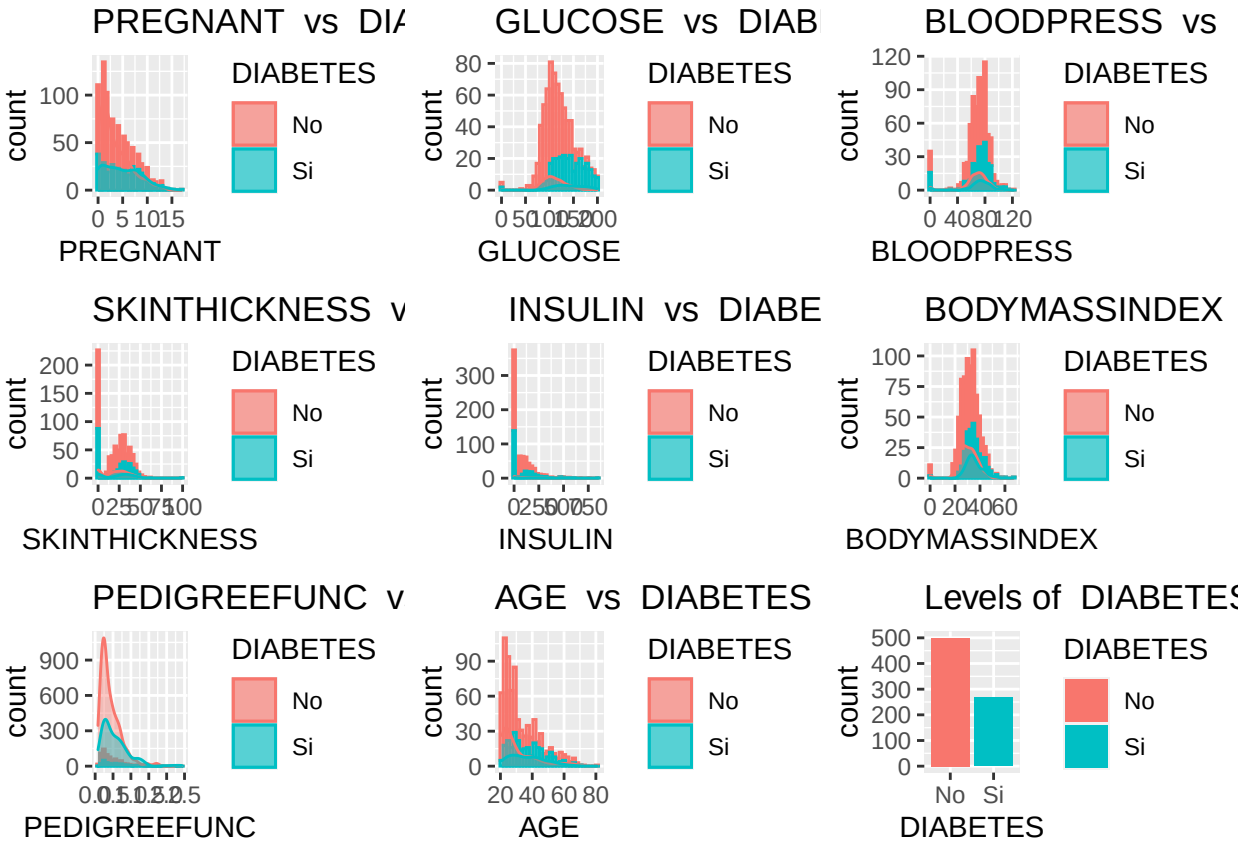
Veamos como se comportan todas juntas:

```
ggpairs(datos,aes(color = DIABETES, alpha = 0.3))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
PlotDataframe(fdata = datos,
              output.name = "DIABETES")
```



Identification and fitting process of classification models

Lo primero que hacemos antes de realizar ningún modelo será informarnos acerca del tema, con la finalidad de conocer cuales son las variables que más afectan a la diabetes.

Tras leer numerosos articulos podemos llegar a la conclusion que las variables que mas afectan a nuestra variable respuesta son tener obesidad, edad, presión arterial alta, antecedentes familiares y altos niveles de glucosa.

Para comenzar con el modelo lo que haremos será dividir nuestra muestra en 2 grupos, el de entrenamiento y el de test. La proporción con la que trabajaremos será de un 80-20.

```
trainIndex <- createDataPartition(datos$DIABETES,
                                  p = 0.8,
                                  list = FALSE,
                                  times = 1)
fTR <- datos[trainIndex,]
fTS <- datos[-trainIndex,]
fTR_eval <- fTR
fTS_eval <- fTS
```

Definimos el initialize trainControl:

```
ctrl <- trainControl(method = "cv",                          #k-fold cross-validation
                     number = 10,                            #Number of folds
                     summaryFunction = defaultSummary,       #Performance summary for comparing models in
                     classProbs = TRUE)
```

Empezamos con la regresión logística

```
LogReg.fit <- train(form = DIABETES ~ .,
                    data = fTR,
                    method = "glm",
                    preProcess = c("center","scale"),
                    metric = "Accuracy",
                    trControl = ctrl)
```

Observamos datos que han salido:

```
LogReg.fit
```

```
## Generalized Linear Model
##
## 615 samples
##   8 predictor
##   2 classes: 'No', 'Si'
##
## Pre-processing: centered (8), scaled (8)
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 554, 554, 554, 554, 553, 553, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.7740613  0.4761652
```

```
summary(LogReg.fit)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4237  -0.7378  -0.4244   0.7291   2.8753
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -0.8543     0.1070  -7.982 1.43e-15 ***
## PREGNANT        0.4838     0.1233   3.924 8.70e-05 ***
## GLUCOSE         1.0904     0.1294   8.428  < 2e-16 ***
## BLOODPRESS     -0.2319     0.1135  -2.043   0.0410 *
## SKINTHICKNESS   0.1206     0.1241   0.972   0.3313
## INSULIN        -0.1455     0.1189  -1.224   0.2209
## BODYMASSINDEX   0.6198     0.1280   4.844 1.27e-06 ***
## PEDIGREEFUNC    0.2712     0.1071   2.534   0.0113 *
## AGE             0.1617     0.1240   1.304   0.1921
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 796.05  on 614  degrees of freedom
## Residual deviance: 586.23  on 606  degrees of freedom
## AIC: 604.23
##
```

```
## Number of Fisher Scoring iterations: 5
```

Ahora evaluamos con nuestro modelo los datos:

```
fTR_eval$LRprob <- predict(LogReg.fit, type="prob", newdata = fTR) # predict probabilities
fTR_eval$LRpred <- predict(LogReg.fit, type="raw", newdata = fTR) # predict classes

fTS_eval$LRprob <- predict(LogReg.fit, type="prob", newdata = fTS) # predict probabilities
fTS_eval$LRpred <- predict(LogReg.fit, type="raw", newdata = fTS) # predict classes
```

Representamos:

```
Plot2DClass(fTR[,c("PREGNANT", "GLUCOSE", "BLOODPRESS", "SKINTHICKNESS", "INSULIN", "BODYMASSINDEX","PE
            fTR$DIABETES,     #Output variable
            LogReg.fit,#Fitted model with caret
            var1 = "BODYMASSINDEX", var2 = "BLOODPRESS", #variables to represent the plot
            selClass = "YES")     #Class output to be analyzed
```