Master Universitario en Ciencia de datos

Asignatura: M2.851 - Tipología y ciclo de vida de los datos

Práctica 2: Limpieza y validación de los datos

Estudiante: José Ahias Lopez Portillo

1. Descripción del dataset. ¿Por qué es importante y qué pregunta/problema pretende responder?

Información del conjunto de datos

- DataSet: Red Wine Quality (UCI Machine Learning)
- Url DataSet: https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009
- Descripción: Los dos conjuntos de datos están relacionados con variantes rojas y blancas del vino portugués "Vinho Verde".

  Debido a problemas de privacidad y logística, solo variables fisicoquímicas (entradas) y sensoriales (el resultado) están disponibles.

Metadata

Columnas
fixed acidity
volatile acidity
citric acid
residual sugar
chlorides
free sulfur dioxide
total sulfur dioxide
density
pH
sulphates
alcohol
quality (score between 0 and 10)

URL GitHub: https://github.com/Ahias/Practica02_RedWineQuality

Objetivo del análisis científico

El objetivo principal de este análisis científico es poder crear un modelo predictivo de clasificación que permitan identificar si un vino es bueno o malo, para ello tomaremos 11 columnas de entrada predictiva y adicionaremos una columna de predicción binaria en base a la condición **“quality>5”,** que al crear, entrenar y validar el modelo por medio de un algoritmo de **“Random Forests”** podremos predecir fácilmente si los vinos son buenos o malos.

2. Integración y selección de los datos de interés a analizar

Lectura del archivo para la creación del modelo predictivo.

```
##Para poder hacer la lectura es necesario configurar
#una carpeta de DataSet en el directorio "C:" y copiar el archivo
#que vamos utilizar para generar nuestro modelo de clasificación.
df_wine<-read.csv("C:/dataset/winequality-red.csv")
```

Visualizar un resumen de tipos de datos y valores contenidos en las columnas.

```
#ver un resumen de metadatos
str(df_wine)
```

```
'data.frame':   1599 obs. of  13 variables:
 $ fixed.acidity       : num  7.4 7.8 7.8 11.2 7.4 7.4 7.9 7.3 7.8 7.5 ...
 $ volatile.acidity    : num  0.7 0.88 0.76 0.28 0.7 0.66 0.6 0.65 0.58 0.5 ...
 $ citric.acid         : num  0 0 0.04 0.56 0 0 0.06 0 0.02 0.36 ...
 $ residual.sugar      : num  1.9 2.6 2.3 1.9 1.9 1.8 1.6 1.2 2 6.1 ...
 $ chlorides           : num  0.076 0.098 0.092 0.075 0.076 0.075 0.069 0.065 0.073 0.071 ...
 $ free.sulfur.dioxide : num  11 25 15 17 11 13 15 15 9 17 ...
 $ total.sulfur.dioxide: num  34 67 54 60 34 40 59 21 18 102 ...
 $ density             : num  0.998 0.997 0.997 0.998 0.998 ...
 $ pH                  : num  3.51 3.2 3.26 3.16 3.51 3.51 3.3 3.39 3.36 3.35 ...
 $ sulphates           : num  0.56 0.68 0.65 0.58 0.56 0.56 0.46 0.47 0.57 0.8 ...
 $ alcohol             : num  9.4 9.8 9.8 9.8 9.4 9.4 9.4 10 9.5 10.5 ...
 $ quality             : int  5 5 5 6 5 5 5 7 7 5 ...
 $ IsGood              : num  0 0 0 1 0 0 0 1 1 0 ...
```

El análisis que vamos a realizar será por medio de un algoritmo de clasificación, en este caso vamos adicionar una nueva columna "**IsGood**", que nos permitirá identificar, si un vino es de buena calidad o no, los vinos de buena calidad serán aquellos que tenga una calidad mayor a "**5**"

```
##Adicionar una nueva columna que clasifique el vino en bueno o malo
df_wine$IsGood <- ifelse(df_wine$quality>5,1,0)
```

Algo importante que se debe mencionar es que en base a los análisis realizados mas adelante dentro del documento, no se eliminaran columnas, porque al analizar el contenido se puede observar que las columnas presentan información útil para el modelo.

3. Limpieza de los datos.

Validar si existen valores perdidos o no definidos.

```
# analizar si existen valores no validos
sapply(df_wine, function(x) sum(x=="NULL" || is.na(x) || length((as.character(x)))==0))
```

```
> sapply(df_wine, function(x) sum(x=="NULL" || is.na(x) || length((as.character(x)))==0))
       fixed.acidity     volatile.acidity          citric.acid       residual.sugar            chlorides
                   0                    0                    0                    0                    0
 free.sulfur.dioxide total.sulfur.dioxide              density                   pH            sulphates
                   0                    0                    0                    0                    0
             alcohol              quality
                   0                    0
> |
```

Tratamiento de valores extremos

```
#Luego de analizar visualmente las distribuciones vamos a eliminar
#los valores atípicos de las siguientes variables:
  #alcohol
  #fixed.acidity
  #residual.sugar
  #chlorides
  #total.sulfur.dioxide
  #sulphates


remove_outliers <- function(x, limit = 3) {
  mn <- mean(x, na.rm = T)
  out <- limit * sd(x, na.rm = T)
  x < (mn - out) | x > (mn + out)
}

df_wine<-df_wine[remove_outliers(df_wine$alcohol,3)==FALSE,]
df_wine<-df_wine[remove_outliers(df_wine$fixed.acidity,3)==FALSE,]
df_wine<-df_wine[remove_outliers(df_wine$residual.sugar,3)==FALSE,]
df_wine<-df_wine[remove_outliers(df_wine$chlorides,3)==FALSE,]
df_wine<-df_wine[remove_outliers(df_wine$total.sulfur.dioxide,3)==FALSE,]
df_wine<-df_wine[remove_outliers(df_wine$sulphates,3)==FALSE,]

count(df_wine)
```

Se ha tomado la decisión de eliminar los valores extremos, para ello se han seleccionado una serie de variables, las cuales se consideran que son variable con alto ruido generado por valores extremos.

Se tomo como base los diagramas de distribución en que se puede ver claramente que todas las distribuciones no son normales, en la mayoría presenta asimetrías positivas, pero en muchos casos los valores extremos se pueden ver claramente que están afectando directamente a los estadísticos entre los cuales podemos mencionar la media.

% valores atípicos eliminados = (1-1477/1599) *100=7.6% aunque puede ser un porcentaje alto en algunos escenarios, para fines de esta práctica consideramos que es un porcentaje aceptable.

4. Análisis de los datos

Iniciaremos el análisis, utilizando visualizaciones de histogramas por cada una de las variables para poder clasificar las distribuciones como normales o no normales, ver si hay valores extremos, si estos generan sesgos significativos y analizar cómo se comportan en las distribuciones individuales de vinos buenos y malos.

Validando intuitivamente las distribuciones de datos

Funciones genéricas para la generación de los histogramas.

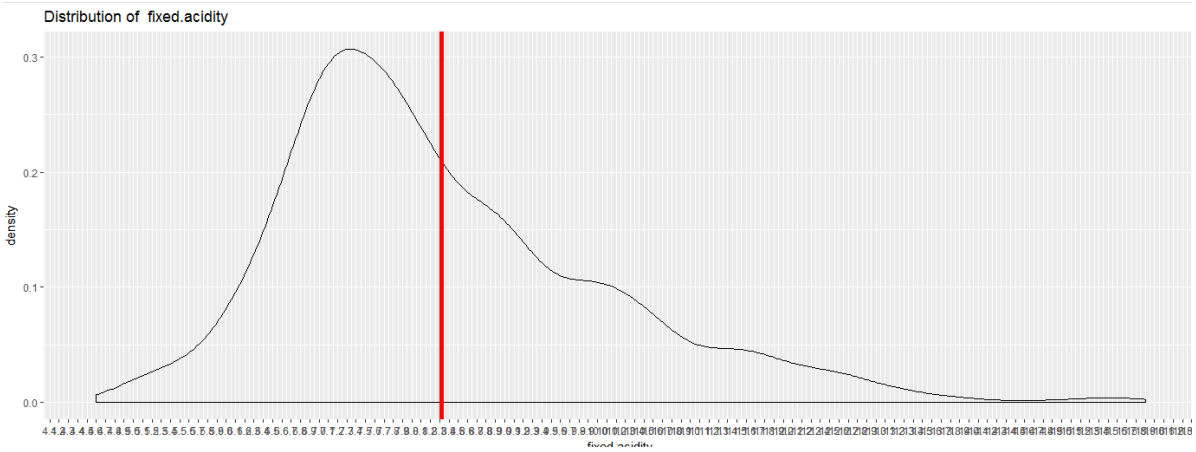
```
#Funcion de histograma global
GetHist <-function(df,x){
  if(is.numeric(df[, x])){

    ggplot(df, aes_string(x)) +
      geom_density(alpha=0.05)+
      scale_x_continuous(breaks = seq(0,20,0.1))+
      geom_vline(aes(xintercept = mean(df[, x])),col='red',size=2)+
      ggtitle(paste('Distribution of ', x))

  }
}
```

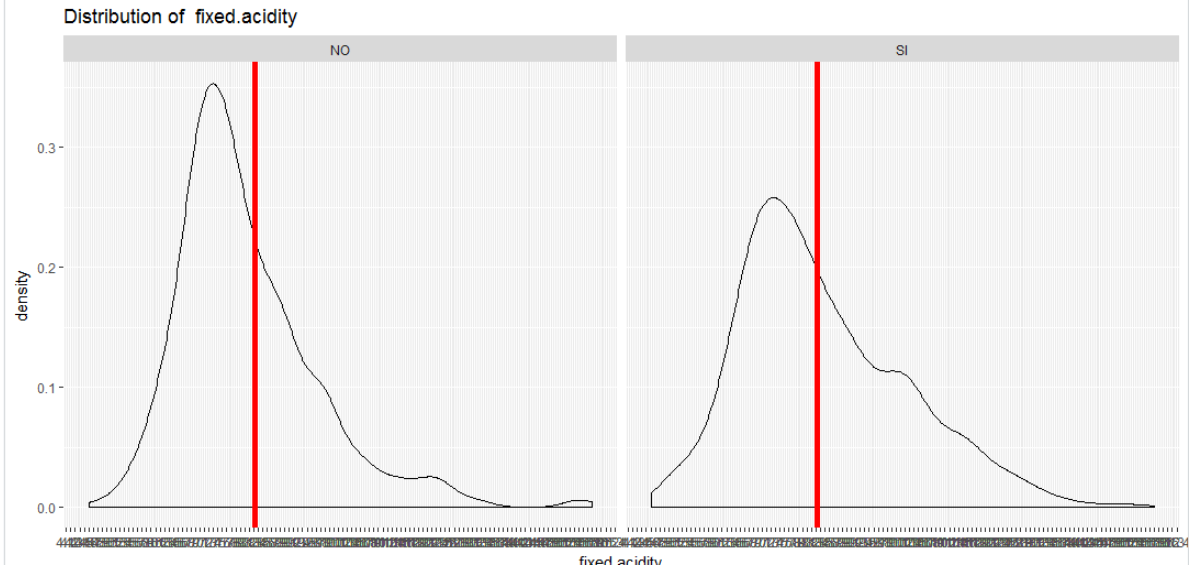
```
#Funcion para crear histogramas en base IsGood SI/NO
GetHist2Class <-function(df,x){
  if(is.numeric(df[, x])){

    ggplot(df, aes_string(x)) +
      geom_density(alpha=0.05)+
      facet_grid(.~ ifelse(IsGood==1,"SI","NO") ) +
      scale_x_continuous(breaks = seq(0,20,0.1))+
      geom_vline(aes(xintercept = mean(df[, x])),col='red',size=2)+
      ggtitle(paste('Distribution of ', x))

  }
}
```

Generación de histogramas por cada variable.

```
#fixed.acidity
GetHist(df_wine,"fixed.acidity")
GetHist2Class(df_wine,"fixed.acidity")
```

La línea roja es la media aritmética.

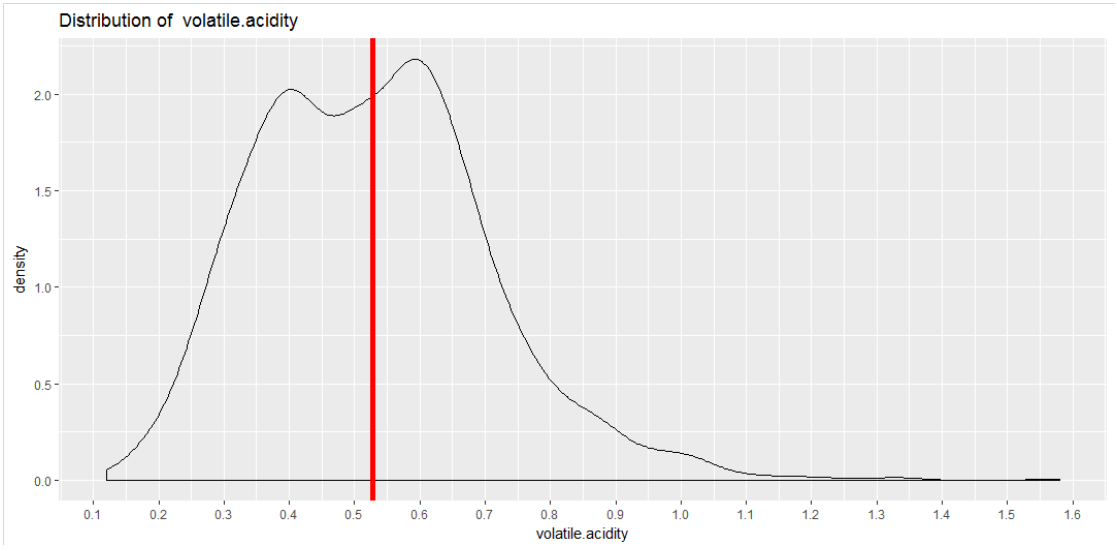


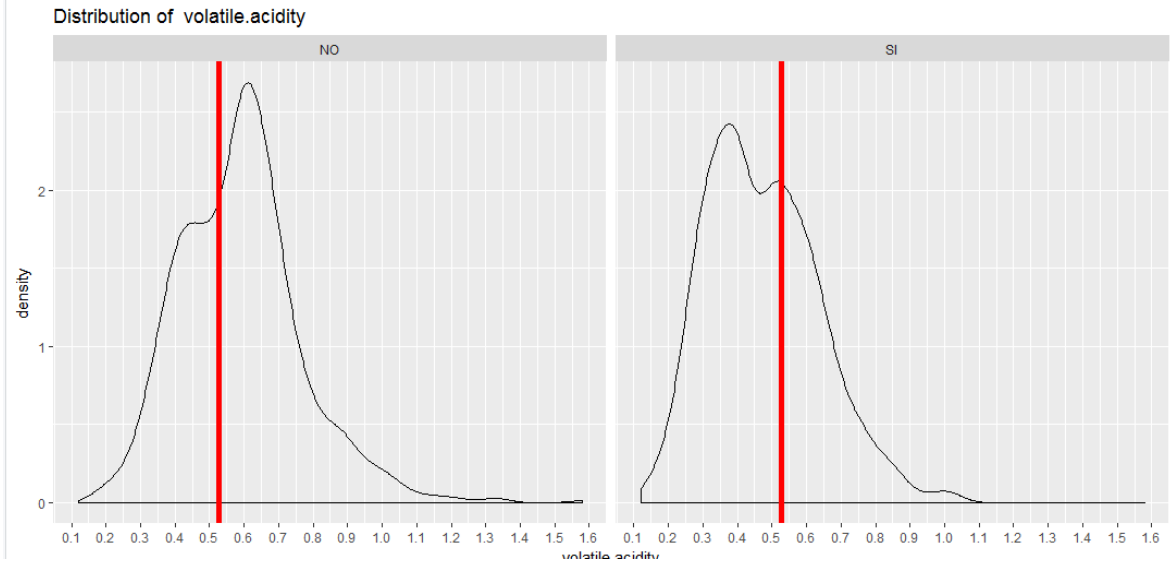


- No es una distribución normal.
- Posee asimetría positiva.

##volatile.acidity
GetHist(df_wine,"fixed.acidity")
GetHist2Class(df_wine,"fixed.acidity")

La línea roja es la media aritmética.

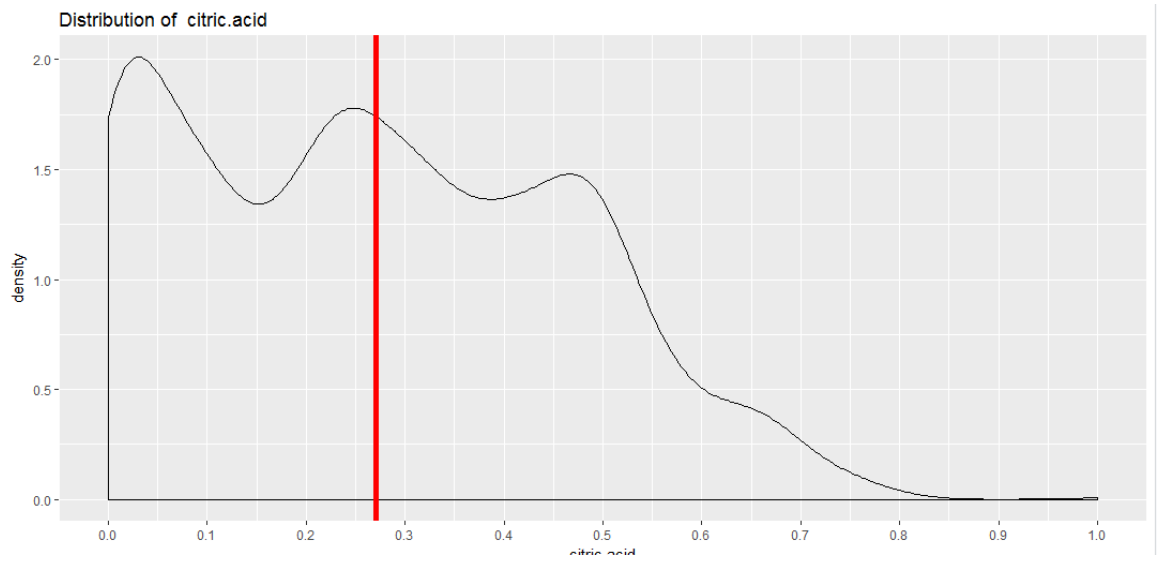


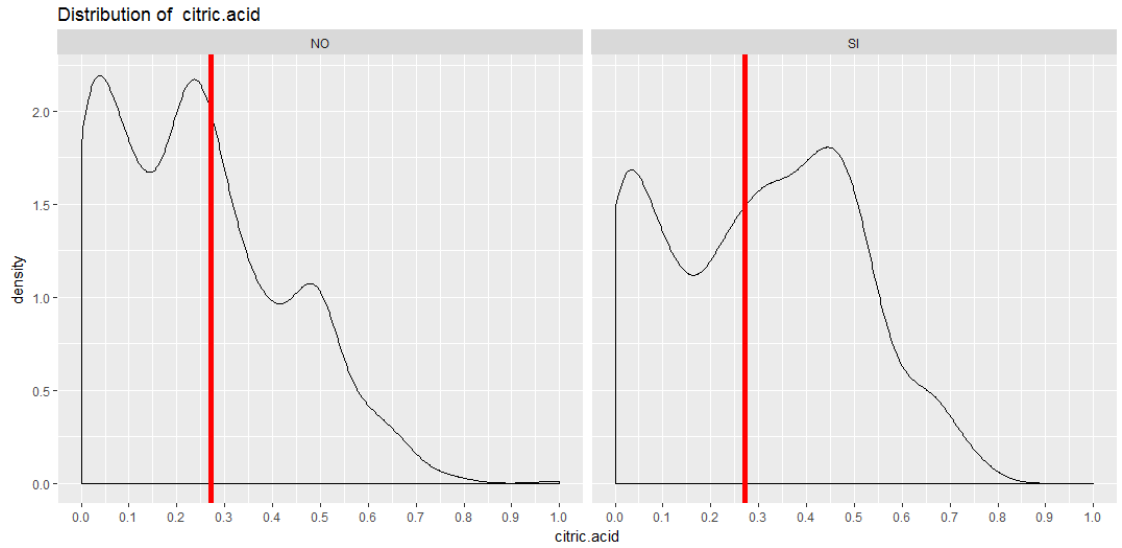


- No es una distribución normal.
- Puede existir una correlación inversa.
- Tiende a ser una distribución asimétrica positiva.

#citric.acid
GetHist(df_wine,"citric.acid")
GetHist2Class(df_wine,"citric.acid")

La línea roja es la media aritmética.

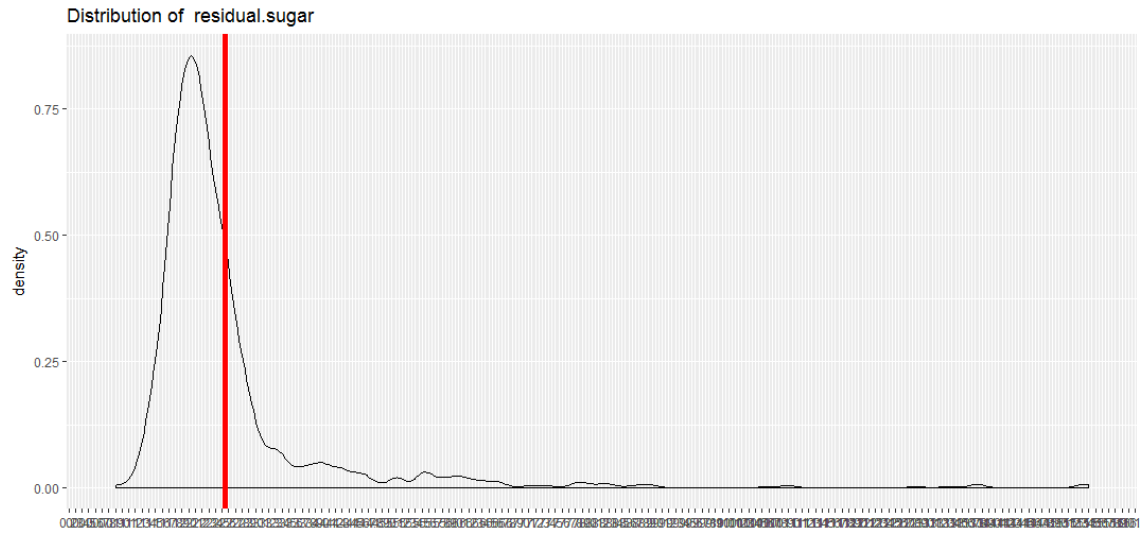


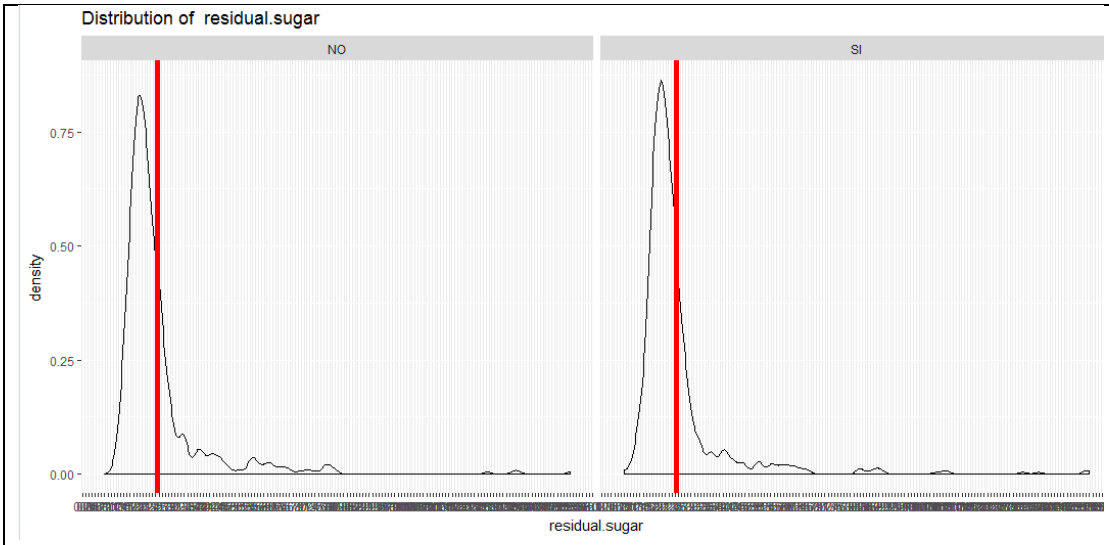


- No es una distribución normal.

#residual.sugar
GetHist(df_wine,"residual.sugar")
GetHist2Class(df_wine,"residual.sugar")

La línea roja es la media aritmética.

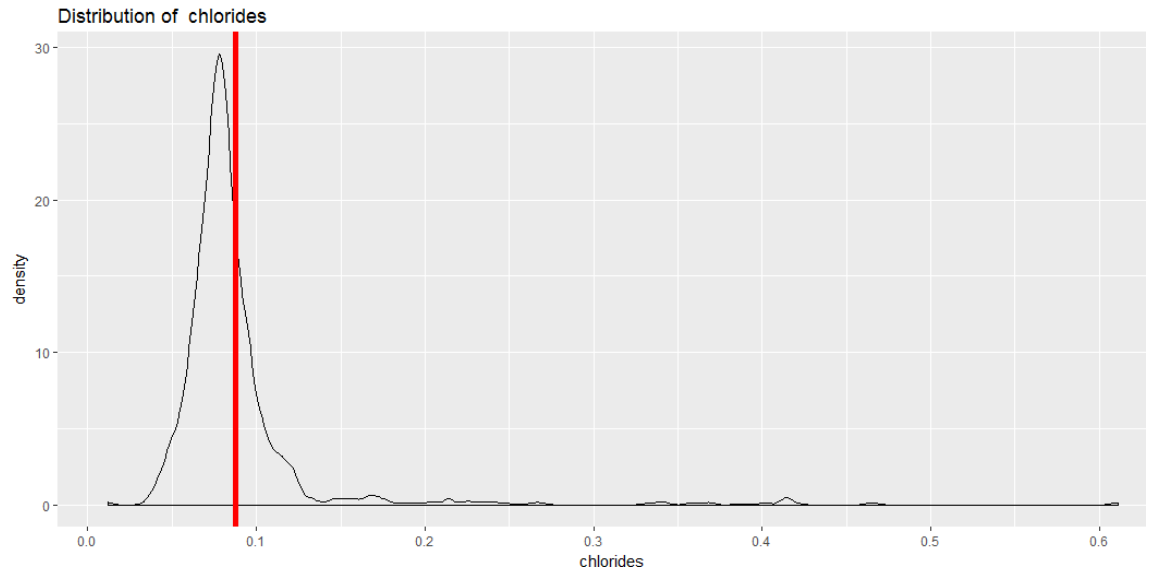


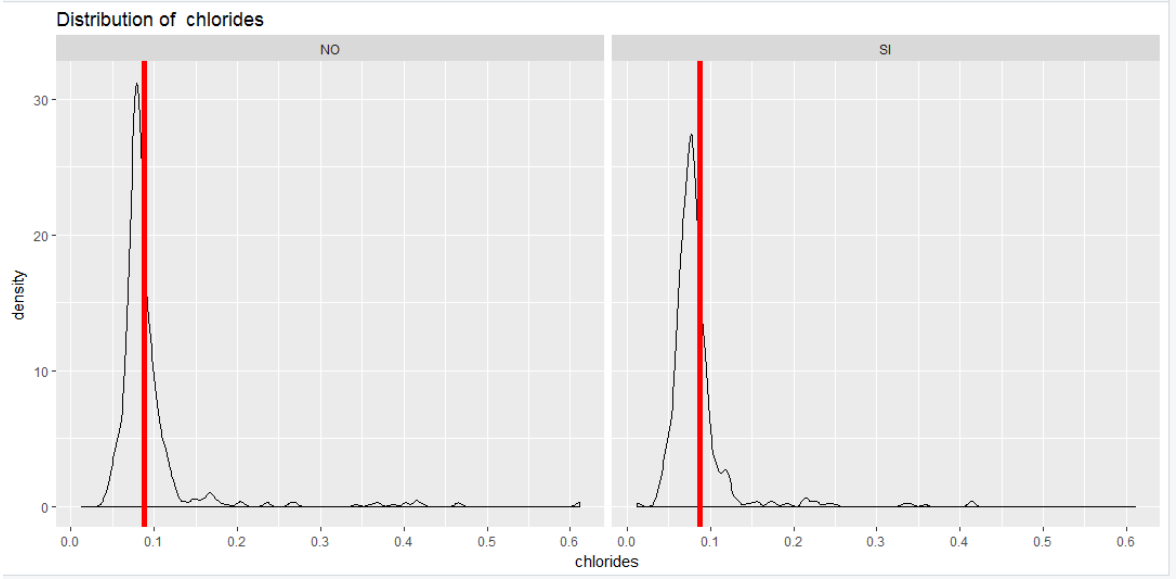


- No es una distribución normal.
- Posee asimetría positiva.
- Los valores extremos están afectando los estadísticos en especial la media.

#chlorides
GetHist(df_wine,"chlorides")
GetHist2Class(df_wine,"chlorides")

La línea roja es la media aritmética.

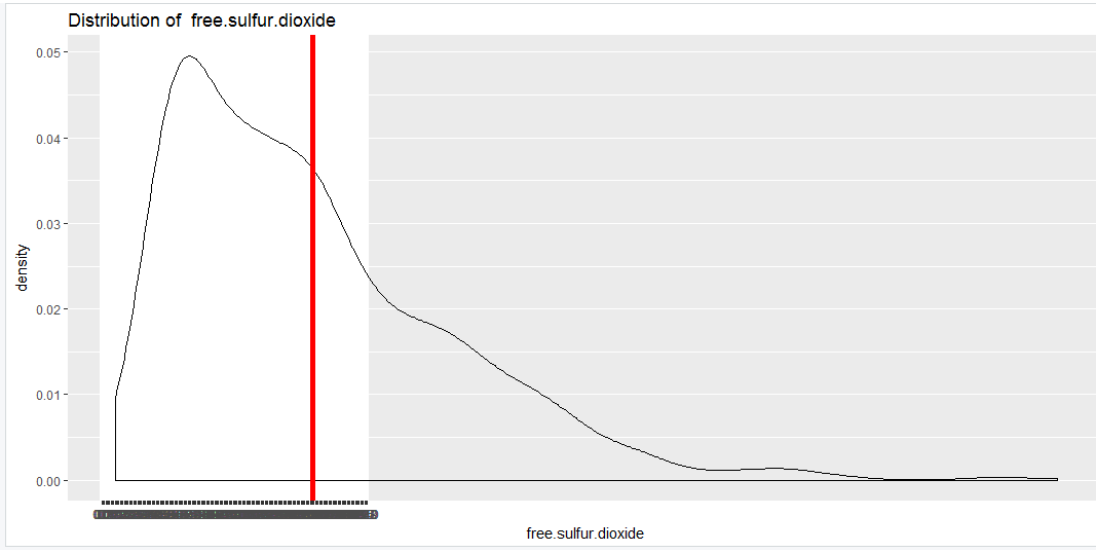


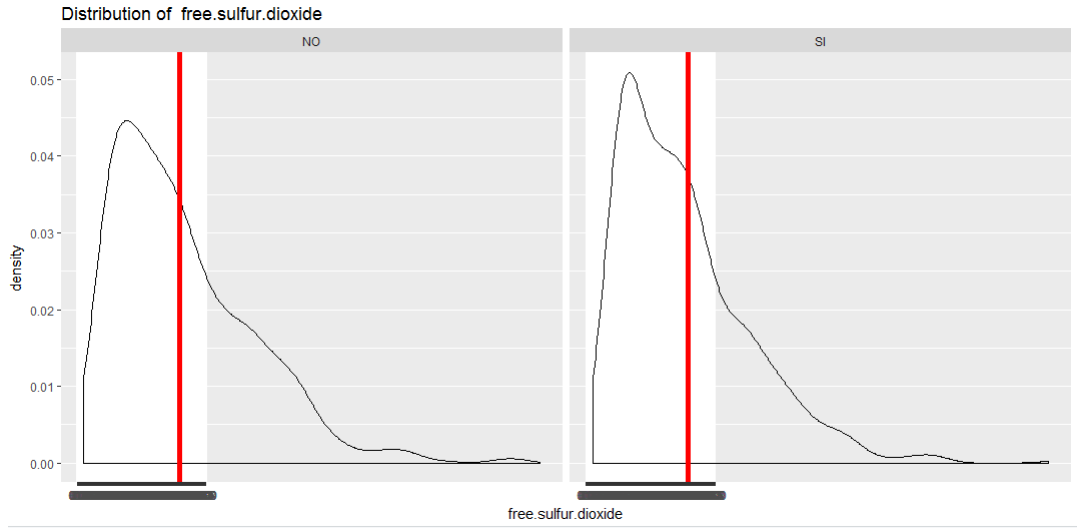


- No es una distribución normal.
- Posee asimetría positiva.
- Los valores extremos están afectando los estadísticos en especial la media.

#free.sulfur.dioxide
GetHist(df_wine,"free.sulfur.dioxide")
GetHist2Class(df_wine,"free.sulfur.dioxide")

La línea roja es la media aritmética.

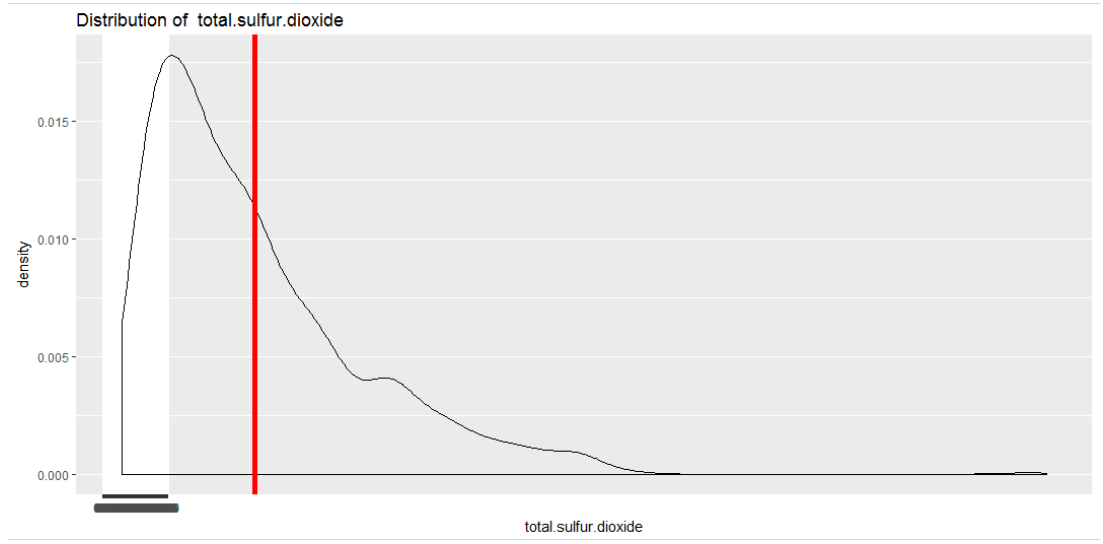


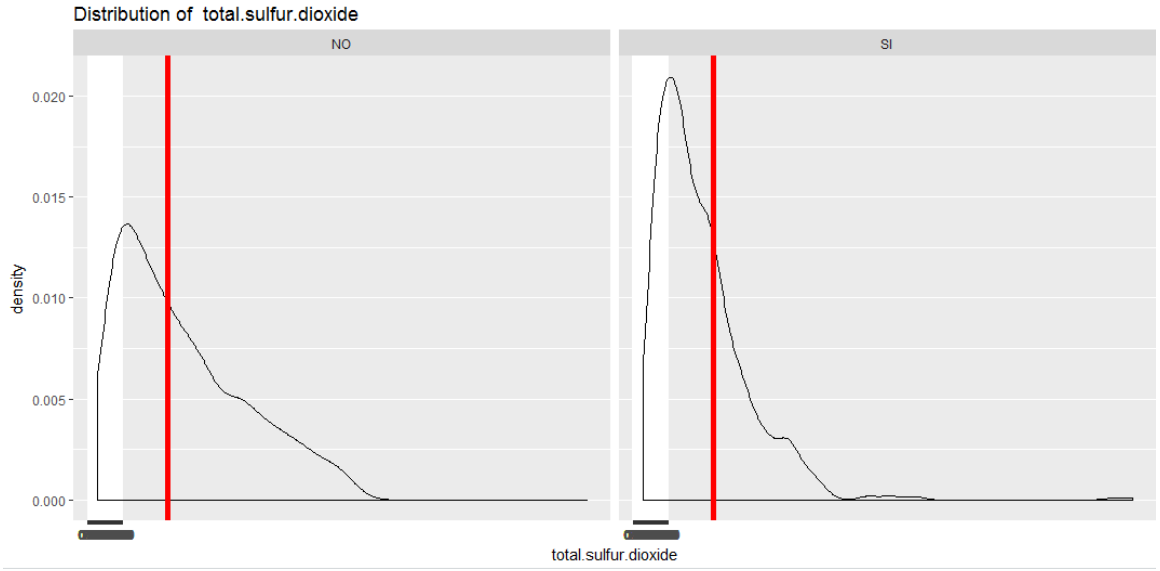


- No es una distribución normal.
- Posee asimetría positiva.

#total.sulfur.dioxide
GetHist(df_wine,"total.sulfur.dioxide")
GetHist2Class(df_wine,"total.sulfur.dioxide")

La línea roja es la media aritmética.

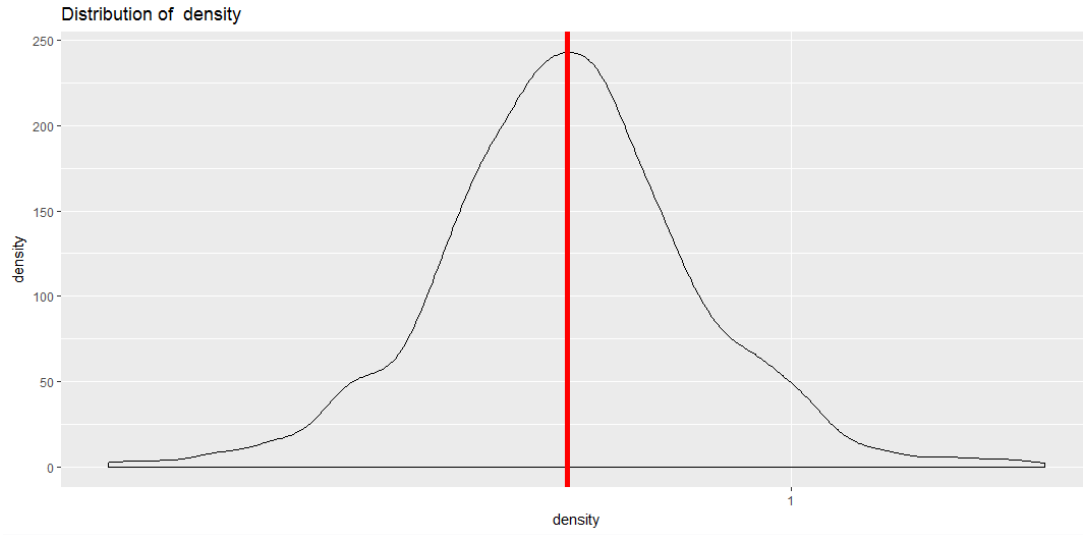


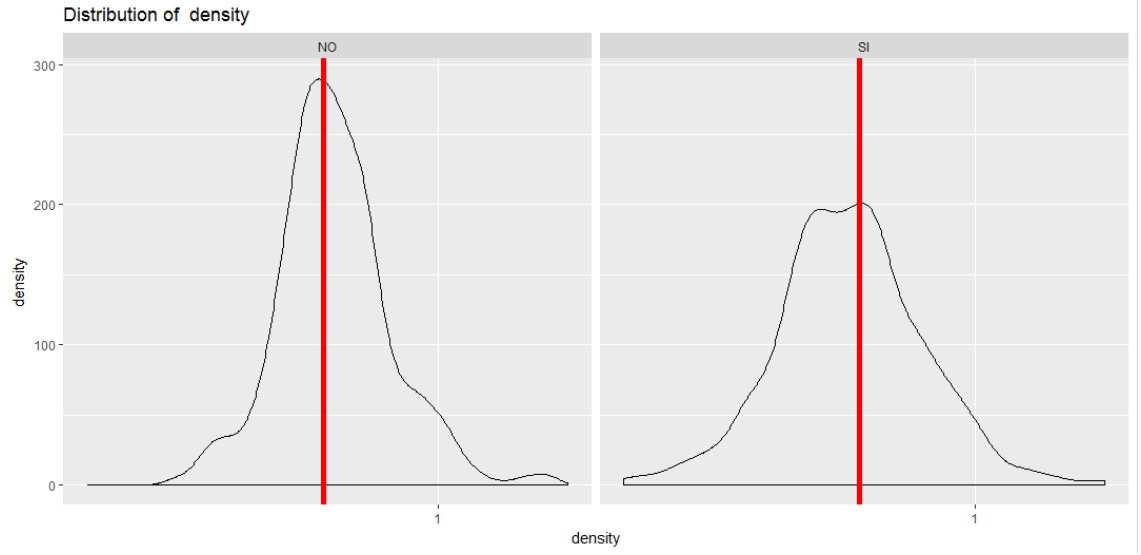


- No es una distribución normal.
- Posee asimetría positiva.
- Los valores extremos están afectando los estadísticos en especial la media.

```
#density
GetHist(df_wine,"density")
GetHist2Class(df_wine,"density")
```

La línea roja es la media aritmética.

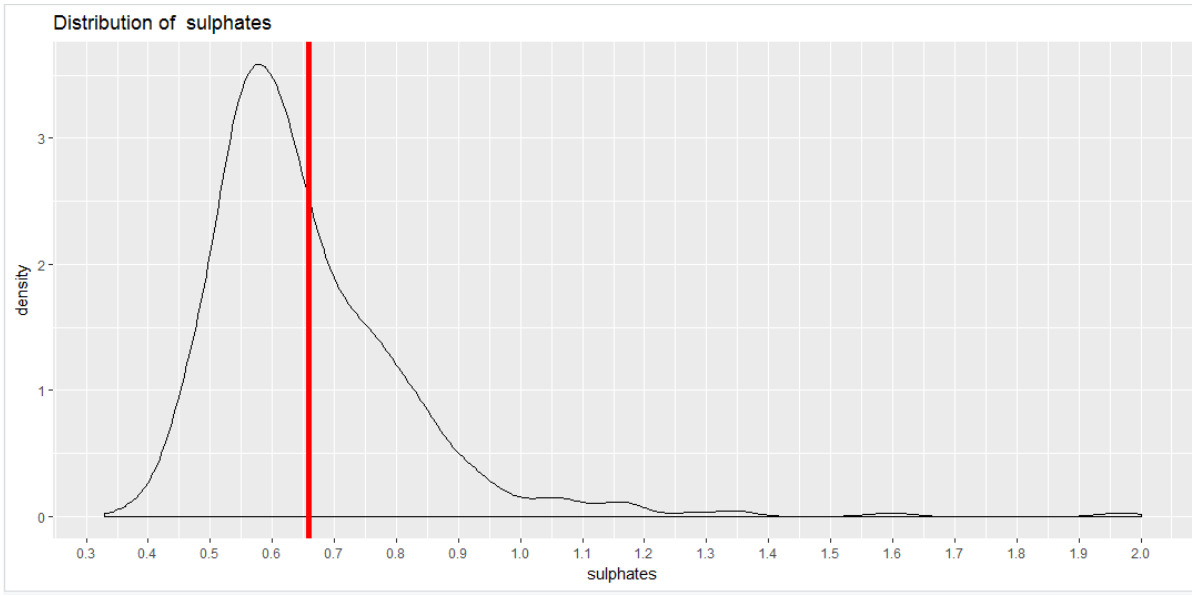


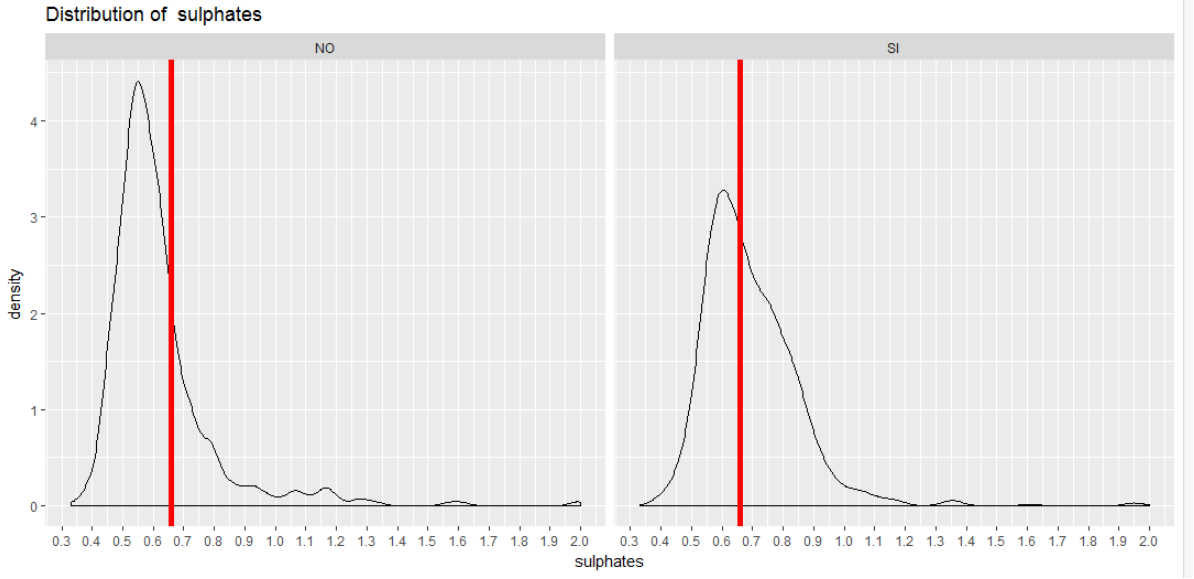


- Tiene una distribución muy similar a una normal.

#sulphates
GetHist(df_wine,"sulphates")
GetHist2Class(df_wine,"sulphates")

La línea roja es la media aritmética.

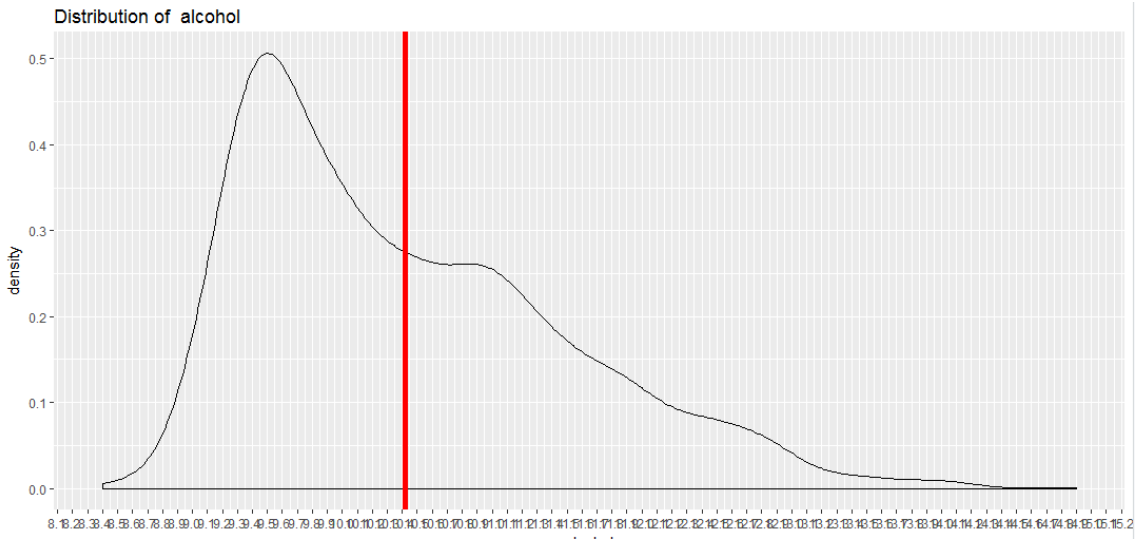


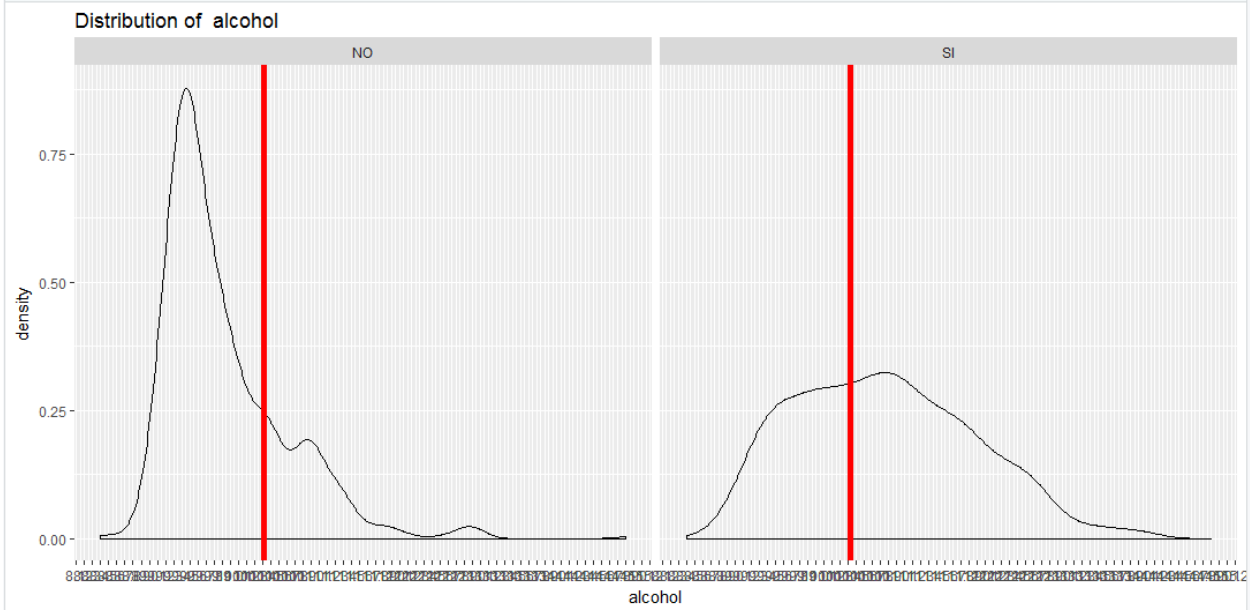


- No es una distribución normal.
- Posee asimetría positiva.
- Los valores extremos están afectando los estadísticos en especial la media.

#alcohol
GetHist(df_wine,"alcohol")
GetHist2Class(df_wine,"alcohol")

La línea roja es la media aritmética.

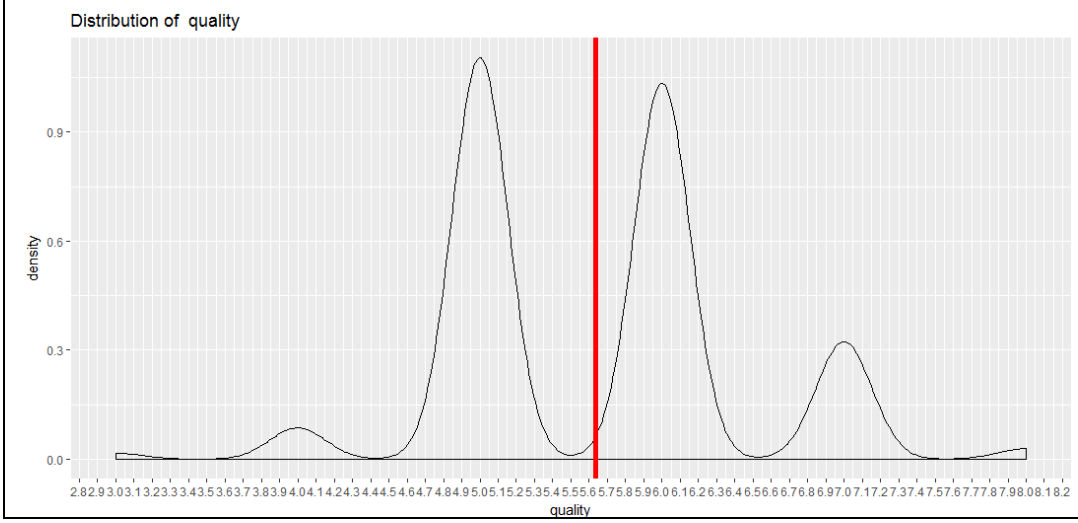


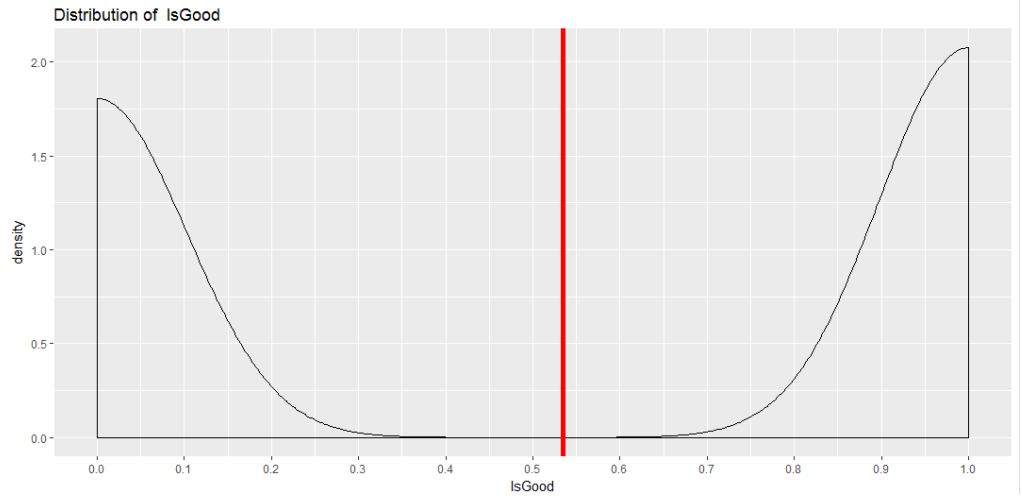


- No es una distribución normal.
- Posee asimetría positiva.
- Los valores extremos están afectando los estadísticos en especial la media.

```
#quality
#IsGood
GetHist(df_wine,"quality")
GetHist(df_wine,"IsGood")
```

La línea roja es la media aritmética.





- No es una distribucion normal.
- Los vinos o son muy bueno o son muy malos, los vinos de punto intermedio son pocos.

Validando numéricamente las distribuciones de datos

Principales estadísticos

```
describe(df_wine)
```

```
> describe(df_wine)
                      vars    n  mean    sd median trimmed   mad  min    max  range  skew kurtosis   se
fixed.acidity            1 1477  8.28  1.66   7.90    8.13  1.33 4.60  13.50   8.90  0.80     0.22 0.04
volatile.acidity         2 1477  0.53  0.18   0.52    0.52  0.18 0.12   1.58   1.46  0.68     1.26 0.00
citric.acid              3 1477  0.26  0.19   0.25    0.25  0.25 0.00   0.79   0.79  0.31    -0.92 0.00
residual.sugar           4 1477  2.40  0.89   2.20    2.23  0.44 0.90   6.70   5.80  2.38     6.60 0.02
chlorides                5 1477  0.08  0.02   0.08    0.08  0.01 0.01   0.23   0.21  2.16     9.54 0.00
free.sulfur.dioxide      6 1477 15.30  9.80  13.00   14.12  8.90 1.00  57.00  56.00  1.07     0.95 0.25
total.sulfur.dioxide     7 1477 43.36 28.43  36.00   39.72 25.20 6.00 139.00 133.00  1.05     0.50 0.74
density                  8 1477  1.00  0.00   1.00    1.00  0.00 0.99   1.00   0.01 -0.03     0.53 0.00
pH                       9 1477  3.32  0.15   3.32    3.32  0.13 2.88   4.01   1.13  0.31     0.85 0.00
sulphates               10 1477  0.64  0.13   0.62    0.63  0.12 0.33   1.11   0.78  0.78     0.49 0.00
alcohol                 11 1477 10.44  1.04  10.20   10.34  1.04 8.40  13.60   5.20  0.74    -0.22 0.03
quality                 12 1477  5.65  0.81   6.00    5.60  1.48 3.00   8.00   5.00  0.19     0.25 0.02
IsGood                  13 1477  0.54  0.50   1.00    0.55  0.00 0.00   1.00   1.00 -0.17    -1.97 0.01
```

Comprobación numérica de la normalidad y homogeneidad de la varianza

```
#Aunque los histogramas y los estadísticos nos indican que nuestras variables
#no siguen una distribución normal, vamos a realizar una prueba de validación de normalidad
#utilizando Anderson-Darling, en la cual vamos a validar el p_valor vrs α=0.05,
#si es superior al nivel de significancia determinamos
#que la distribución es normal de lo contrario es una distribución libre.
alpha = 0.05
col.names = colnames(df_wine)
for (i in 1:ncol(df_wine)) {
  if (i == 1) cat("Variables que no siguen una distribución normal:\n")
  if (is.integer(df_wine[,i]) | is.numeric(df_wine[,i])) {
    p_val = ad.test(df_wine[,i])$p.value
    if (p_val < alpha) {
      cat(col.names[i])
```

```
    # Format output
    if (i < ncol(df_wine) - 1) cat(", ")
    if (i %% 3 == 0) cat("\n")
   }
  }
}
```

fligner.test(IsGood ~ fixed.acidity, data = df_wine)

```
> fligner.test(IsGood ~ fixed.acidity, data = df_wine)

        Fligner-Killeen test of homogeneity of variances

data:  IsGood by fixed.acidity
Fligner-Killeen:med chi-squared = 74.025, df = 85, p-value =
0.7965
```

R:/ Varianza homogénea p-value>0.05

fligner.test(IsGood ~ volatile.acidity, data = df_wine)

```
> fligner.test(IsGood ~ volatile.acidity, data = df_wine)

        Fligner-Killeen test of homogeneity of variances

data:  IsGood by volatile.acidity
Fligner-Killeen:med chi-squared = 172.37, df = 139, p-value =
0.02872
```

R:/ Varianza no homogénea p-value<0.05

fligner.test(IsGood ~ citric.acid, data = df_wine)

```
> fligner.test(IsGood ~ citric.acid, data = df_wine)

        Fligner-Killeen test of homogeneity of variances

data:  IsGood by citric.acid
Fligner-Killeen:med chi-squared = 78.454, df = 77, p-value =
0.4325
```

R:/ Varianza homogénea p-value>0.05

fligner.test(IsGood ~ residual.sugar, data = df_wine)

```
> fligner.test(IsGood ~ residual.sugar, data = df_wine)

        Fligner-Killeen test of homogeneity of variances

data:  IsGood by residual.sugar
Fligner-Killeen:med chi-squared = 65.47, df = 70, p-value =
0.6311
```

R:/ Varianza homogénea p-value>0.05

fligner.test(IsGood ~ chlorides, data = df_wine)

```
> fligner.test(IsGood ~ chlorides, data = df_wine)

        Fligner-Killeen test of homogeneity of variances

data:  IsGood by chlorides
Fligner-Killeen:med chi-squared = 93.634, df = 119, p-value =
0.9585
```

R:/ Varianza homogénea p-value>0.05

fligner.test(IsGood ~ free.sulfur.dioxide, data = df_wine)

```
> fligner.test(IsGood ~ free.sulfur.dioxide, data = df_wine)

        Fligner-Killeen test of homogeneity of variances

data:  IsGood by free.sulfur.dioxide
Fligner-Killeen:med chi-squared = 27.553, df = 52, p-value =
0.9979
```

R:/ Varianza homogénea p-value>0.05

fligner.test(IsGood ~ total.sulfur.dioxide, data = df_wine)

```
> fligner.test(IsGood ~ total.sulfur.dioxide, data = df_wine)

        Fligner-Killeen test of homogeneity of variances

data:  IsGood by total.sulfur.dioxide
Fligner-Killeen:med chi-squared = 122.56, df = 125, p-value =
0.545
```

R:/ Varianza homogénea p-value>0.05

fligner.test(IsGood ~ density, data = df_wine)

```
> fligner.test(IsGood ~ density, data = df_wine)

        Fligner-Killeen test of homogeneity of variances

data:  IsGood by density
Fligner-Killeen:med chi-squared = 317.11, df = 416, p-value =
0.9999
```

R:/ Varianza homogénea p-value>0.05

fligner.test(IsGood ~ pH, data = df_wine)

```
> fligner.test(IsGood ~ pH, data = df_wine)

        Fligner-Killeen test of homogeneity of variances

data:  IsGood by pH
Fligner-Killeen:med chi-squared = 51.451, df = 81, p-value =
0.9957
```

R:/ Varianza homogénea p-value>0.05

```
fligner.test(IsGood ~ sulphates, data = df_wine)
> fligner.test(IsGood ~ sulphates, data = df_wine)

        Fligner-Killeen test of homogeneity of variances

data:  IsGood by sulphates
Fligner-Killeen:med chi-squared = 130.6, df = 72, p-value =
2.936e-05
```

R:/ Varianza no homogénea p-value<0.05

```
fligner.test(IsGood ~ alcohol, data = df_wine)
> fligner.test(IsGood ~ alcohol, data = df_wine)

        Fligner-Killeen test of homogeneity of variances

data:  IsGood by alcohol
Fligner-Killeen:med chi-squared = 139.02, df = 61, p-value =
5.029e-08
```

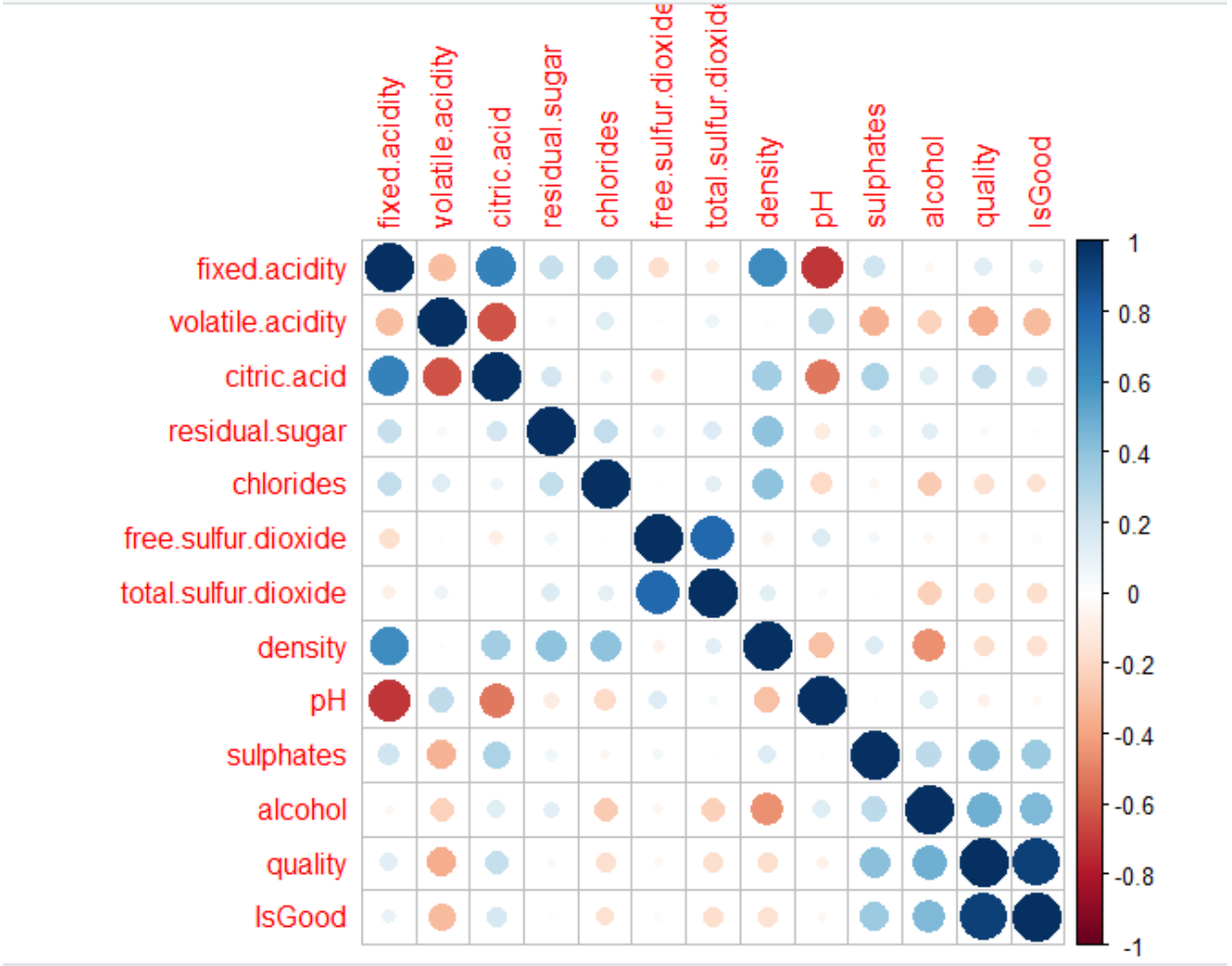
R:/ Varianza no homogénea p-value<0.05

En la sección anterior intuitivamente determinamos que ninguna variable sigue una distribución normal, al utilizar la prueba de normalidad de Anderson-Darling, comprobamos que ninguna variable tiene una distribución normal, por ello en el análisis de la correlación no podemos utilizar el análisis de Pearson.

Pruebas estadísticas

Análisis de correlaciones cualitativamente.

```
#Matrix de correlación (Heatmap)
corrplot(cor(df_wine,method = "spearman"))
```



Podemos observar en la matriz que las variables que más influyen en la calidad del vino son:

- Alcohol
- Sulphates
- Volatile.acidity
- Density
- Total.silfur.dioxide

Análisis de correlaciones Cuantitativamente

```
#calcular las correlaciones cuantitativamente
corr_matrix <- matrix(nc = 3, nr = 0)
colnames(corr_matrix) <- c("variable","estimate", "p-value")
for (i in 1:(ncol(df_wine) - 2)) {
  if (is.integer(df_wine[,i]) | is.numeric(df_wine[,i])) {
    spearman_test = cor.test(df_wine[,i],
                             df_wine[,length(df_wine)],
                             method = "spearman")
    corr_coef = spearman_test$estimate
    p_val = spearman_test$p.value
    xcol=col.names[i]
    # Add row to matrix
    newrow = matrix(ncol = 3, nrow = 1)
```

```
    newrow[1][1] = xcol
    newrow[2][1] = corr_coef
    newrow[3][1] = p_val
    corr_matrix <- rbind(corr_matrix, newrow)
  }
}

corr_matrix
```

```
> corr_matrix
       variable               estimate               p-value
  [1,] "fixed.acidity"        "0.0978200108704578"   "0.000166385816110881"
  [2,] "volatile.acidity"     "-0.311996379367018"   "1.04499427273983e-34"
  [3,] "citric.acid"          "0.174174723438212"    "1.5865733503243e-11"
  [4,] "residual.sugar"       "0.0122081810757202"   "0.639211099964514"
  [5,] "chlorides"            "-0.151688825566518"   "4.66603664992763e-09"
  [6,] "free.sulfur.dioxide"  "-0.0296628771490392"  "0.254585541417965"
  [7,] "total.sulfur.dioxide" "-0.175583706449209"   "1.08213350673969e-11"
  [8,] "density"              "-0.156000634094431"   "1.66784814830378e-09"
  [9,] "pH"                   "-0.0311738240498625"  "0.231173467991803"
 [10,] "sulphates"            "0.369958357343993"    "4.01782334038733e-49"
 [11,] "alcohol"              "0.442761424202224"    "6.04544714271511e-72"
```

Comprobamos numéricamente que las variables que más influyen son:

- Alcohol (correlación positiva)
- Sulphates (correlación positiva)
- Volatile.acidity (correlación negativa)
- Density (correlacion negativa)
- Total.silfur.dioxide (correlación negativa)

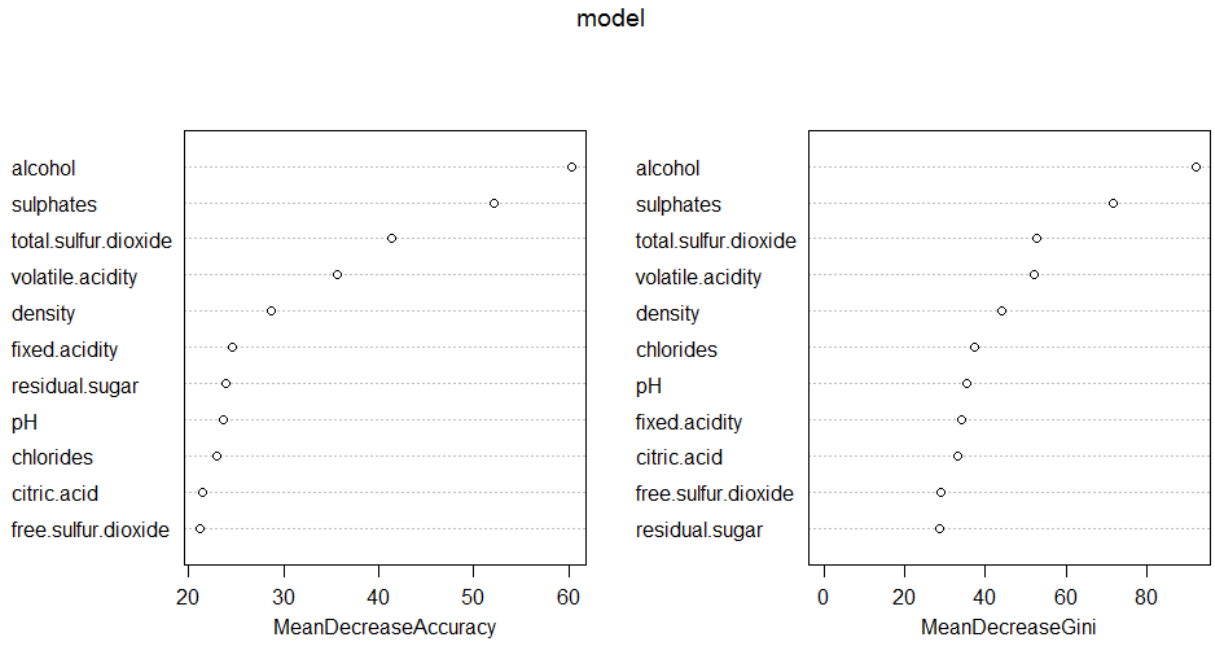
5. Representación de los resultados a partir de tablas y gráficas

Modelo predictivo

Generación del modelo

```
##-----------------------------------------------------
#Generar el modelo predictivo y test
#eliminar la variable Quality porque de ahi derivamos IsGood
df_wine <-df_wine[,-12]

#generar el modelo de entrenamiento y test
samp <- sample(nrow(df_wine), 0.7 * nrow(df_wine))
train <- df_wine[samp, ]
test <- df_wine[-samp, ]

#Generando el modelo de Random Forest
model <- randomForest(as.factor(IsGood) ~ ., data = train,  do.trace=T, importance=T)
model
```

```
# Validar que variables son más relevante en la predicción
varImpPlot(model)
```



Luego de aplicar Random Forest tenemos los siguientes resultados:

Variables que mas influyen en la predicción:

- Alcohol
- Sulphates
- Total.sulfur.dioxide
- Volatile.acidity
- Density
- Fixed.acidity
- Residual.sugar
- Ph

Las variables que menos influyen:

- Chlorides
- Citric.acid
- Free.sulfur.dioxide

Validación del modelo

```
#Consultar informacion del modelo con la validacion contra test
pred<-predict(model, newdata=test, type="class")
test$IsGood <- factor(test$IsGood)
confusionMatrix(pred, test$IsGood)
```

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 162  40
         1  47 195

               Accuracy : 0.8041
                 95% CI : (0.764, 0.84)
    No Information Rate : 0.5293
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.606
 Mcnemar's Test P-Value : 0.5201

            Sensitivity : 0.7751
            Specificity : 0.8298
         Pos Pred Value : 0.8020
         Neg Pred Value : 0.8058
             Prevalence : 0.4707
         Detection Rate : 0.3649
   Detection Prevalence : 0.4550
      Balanced Accuracy : 0.8025

       'Positive' Class : 0
```

6. Resolución del problema. A partir de los resultados obtenidos, ¿cuáles son las conclusiones? ¿Los resultados permiten responder al problema?

- La solución desarrollada permite predecir en un 80.41%.
- El modelo generado aun puede optimizarse, ya que no se ha realizado validación cruzada y la afinación de los hyper-parameter.
- Todas las variables no tienen una distribución normal.
- Existen modelos extremos que generar sesgos dentro la distribución y afectan a los estadísticos.
- Se tomó la decisión de eliminar los principales valores extremos que generaban ruido dentro de algunas distribuciones.
- El 80% de las variables tienen una varianza homogénea.
- Las variables que mejor predicen, si el vino es bueno o malo son:
    - Alcohol
    - Sulphates
    - Total.sulfur.dioxide
    - Volatile.acidity
    - Density
    - Fixed.acidity
    - Residual.sugar
    - Ph