



PROYECTO FIN DE MÁSTER

Análisis de la calidad del vino blanco y tinto



Tatán Rufino

2019-2020



ÍNDICE

1. INTRODUCCIÓN Y OBJETIVO DEL ANÁLISIS
2. LIMPIEZA Y ANÁLISIS DESCRIPTIVO
3. ANÁLISIS EXPLORATORIO
4. MODELIZACIÓN MATEMÁTICA
 1. MODELOS DE REGRESIÓN
 2. MODELOS DE CLASIFICACIÓN
 3. MODELO DE RESPUESTA BINARIA
5. CONCLUSIÓN Y PROPUESTAS
6. REFERENCIAS

1. INTRODUCCIÓN Y OBJETIVO DEL ANÁLISIS

1.1. Información del conjunto de datos^{*1}

Los dos conjuntos de datos están relacionados con variantes rojas y blancas del vino portugués "Vinho Verde". Debido a cuestiones de privacidad y logística, solo están disponibles las variables fisicoquímicas (entradas) y sensoriales (la salida) (por ejemplo, no hay datos sobre tipos de uvas, marcas de vinos, precios de venta de vinos, etc.).



Estos conjuntos de datos se pueden ver como tareas de clasificación o regresión. Las clases son ordenadas y no equilibradas (por ejemplo, hay muchos más vinos normales que excelentes o pobres). Se podrían utilizar algoritmos de detección de valores atípicos para detectar los pocos vinos excelentes o pobres. Además, no estamos seguros de si todas las variables de entrada son relevantes. Por lo tanto, podría ser interesante probar los métodos de selección de características.

1.2. Información de los atributos del dataset:

Variables de entrada (basadas en pruebas fisicoquímicas):

- Acidez fija
- Acidez volátil
- Ácido cítrico
- Azúcar residual
- Cloruros
- Dióxido de azufre libre
- Dióxido de azufre total
- Densidad
- pH
- Sulfatos
- alcohol

^{*1}(extraído del enlace del dataset)



Variable de salida (basada en datos sensoriales):

- Calidad (puntaje entre 3 y 9)

1.3. Antecedentes

Aquí expongo, tras investigar sobre el dataset y su contenido, datos de interés, que no sólo pueden aportar una visión más amplia al análisis si no también un plus de aprendizaje significativo al proyecto:

- **Acidez del vino**

Suma de los diferentes ácidos orgánicos que se encuentran en el mosto o en el vino. Se representa por el PH que en general puede variar entre 1 (acidez máxima) y 7 (mínima).

En el caso de los vinos este factor varía entre 2,7 - 3,9.

La acidez del vino no suele expresarse como el contenido de cada ácido, sino como la suma de todos los ácidos y referida al más importante, que es el tartárico; se mide, por tanto, en gramos de ácido tartárico por litro.

Durante la crianza del vino los ácidos se combinan con los alcoholes formando ésteres aromáticos.

La acidez puede ser fija o volátil:

- **Acidez Fija** - Conjunto de los ácidos naturales procedentes de la uva (tartárico, málico, cítrico y succínico) o formados en la fermentación maloláctica (láctico).

En general, los ácidos (acidez fija) son preservante naturales del vino y ayuda a mantener el color y cualidades aromáticas.

- **Acidez Volátil** - Conjunto de ácidos formados durante la fermentación o como consecuencia de alteraciones microbianas. Estos ácidos son, principalmente: ácido Acético, ácido Propionico, ácido Butírico y ácido Sulfúrico.

Si la acidez volátil, presente en todos los vinos, es muy elevada el vino se picará y avinagrará con el paso del tiempo. *Es conveniente que la acidez volátil de un vino sea lo más baja posible.*

- Ácido cítrico

El ácido cítrico puede ser utilizado para la acidificación química de los vinos o por su acción estabilizante particularmente para limitar los riesgos de quiebras férricas o para el prelavado de placas filtrantes. El contenido máximo en los vinos puede estar sometido a límites reglamentarios.

- Azúcar residual

Azúcar que queda en el vino después de la fermentación. Durante la fermentación, y por acción de las levaduras, se transforman en alcohol etílico, anhídrido carbónico y otras muchas sustancias que caracterizan al vino. Cuando esta transformación es prácticamente total se dice que el vino es seco, pero lo normal es que en todo vino quede cierta cantidad de azúcares sin fermentar, denominados azúcares reductores. *En los vinos jóvenes existe una relación entre la presencia de azúcares residuales y la intensidad aromática.*

- Cloruros

Objetivamente, la presencia de cloruros hace referencia a la gran mayoría de sales (*cloruro potásico y sódico*) que contiene el vino. Representa la base de la fuente de minerales presente en el vino, y generalmente no debe exceder los 50 mg/l. *Un alto contenido en cloruros puede ser un indicador de un mal estado contaminación de la tierra, así como un factor que podría afectar negativamente a su sabor.*

- Sulfitos o dióxidos de azufre

El dióxido de azufre es un gas incoloro, no inflamable, de olor picante, sofocante. Se conserva y transporta en estado líquido en recipientes de acero resistentes. Esas soluciones son inestables y no deben contener menos de 50 g/l de SO₂. La etiqueta debe mencionar el contenido de SO₂ en el momento de la puesta en venta y las condiciones de conservación y seguridad.

Es un producto comprendido en la categoría de los agentes conservantes con una acción antiséptica y antioxidante.

- Densidad

La densidad del vino es muy cercana al agua, debido a que el vino está compuesto principalmente de:

- 85% de agua
- 8-15% alcohol
- 1-3% resto de componentes

Su valor se suele comprender entre 0,97 – 1,1 g/ml. La densidad

Es un parámetro que en cata se percibe como estructura del vino o espesor en boca. Como criterio de calidad consideramos como buenos vinos aquellos que son ligeros, pero con cuerpo. La densidad es, en esencia, *un factor dependiente en el vino*, ya que varía en función (principalmente) del contenido en alcohol (inversamente proporcional) y los sólidos en suspensión y disueltos (sales y sulfatos) (directamente proporcional). *De modo que, hablar de cómo afecte un cambio de la densidad en el vino es como referirse el cómo afecta el cambio en contenido de alcohol y/o la variación de sales en el mismo.*

- pH

El pH es la medida que indica la acidez de los líquidos, generando así tres tipos de variantes con las que se los puede identificar: ácido, básico o neutro. Es un indicador que engloba el contenido en componentes ácidos del vino. *Su valor reviste singular importancia en la fermentación, conservación y carácter final de un vino.* Los vinos presentan un pH variable entre 2,9 y 4.

- Sulfatos

Están presentes en la uva de forma natural y también pueden ser adicionados en el proceso de fermentación del vino. *Son estos mismos sulfatos los que se "reducen" de manera natural durante el proceso de fermentación del vino, para producir sulfitos o dióxidos de azufre en los mismos.* Su presencia en el vino en equilibrio con los

sulfitos es equivalente a la cantidad de azúcar residual en equilibrio con la cantidad de alcohol producido.

El sulfato de amonio es un producto empleado como activador de la fermentación, reservado a las operaciones de fermento. Aporta ion amonio directamente asimilable por las levaduras. Los sulfatos aportados son totalmente solubles en el vino.

Se presenta como cristales anhidros, transparentes, de sabor picante y amargo, similares a los cristales de sulfato de potasio, con el que esta sal es isomorfa

- Alcohol

El alcohol etílico o etanol ($\text{CH}_3\text{-CH}_2\text{OH}$) presente en el vino se debe a la fermentación de la uva (fermentación alcohólica) que transforma los azúcares del mosto. Es la segunda sustancia, tras el agua, más abundante en el vino, entre un 8% y un 15%.

Además del etanol se encuentran en el vino y con presencia residual otros tipos de alcoholes, que a pesar de su pequeña proporción *juegan un papel muy importante en el aroma y sabor final.*



1.4. Objetivos del proyecto

- Analizar y comprender la medida en la que los atributos (características) químicas afectan a la calidad del vino, tanto de forma individual como en proporción colectiva.
- Crear modelos matemáticos que nos permitan, en base al conjunto de los atributos cuyos valores recoge el presente estudio, lo siguiente:
 - Predecir la calidad del vino en base a las variables de entrada
 - Clasificar los vinos según sus calidades, haciendo énfasis en los de mayor calidad
 - Evaluar la calidad de dichos modelos y predicciones con las estimaciones y análisis realizados en la fase inicial del proyecto, así como comparar con las características recogidas durante el estudio previo de las mismas

2. LIMPIEZA Y ANÁLISIS DESCRIPTIVO

La limpieza de datos y el análisis descriptivo preliminar se realiza con la herramienta R. Es fundamental, previo a conocer a fondo la naturaleza de las variables y su relación con la respuesta sensorial, **la calidad**, conocer las correlaciones existentes entre los predictores o variables continuas disponibles, a fin de alcanzar dos objetivos principalmente:

- Escoger los tipos y modelos matemáticos de aprendizaje adecuados para la predicción o clasificación
- Comprender en profundidad *la naturaleza de los atributos del producto* que se estudia, en este caso **la calidad del vino**

2.1. LIMPIEZA Y PREPARACIÓN DE DATOS

Carga de datos

```
library(xlsx)
library(tidyverse)
library(lattice)

library(knitr)
library(corrplot)
library(car)
library(xlsx)
library(visdat)
library(data.table)
library(GGally)
library(corrgram)
library(kableExtra)
library(psych)

#Creo una función que me permite remover outliers en función de los percentiles en los que se encuentran
remove_outliers <- function(x, quant) {

  require("dplyr")

  for(i in 1:ncol(x)) {
```

```

x = mutate(x,outliers=ifelse(x[[i]] < quantile(x[[i]],quant),0,1))

x = filter(x, outliers==0)

x = select(x, -outliers)

}

return(x)
}

setwd("D:/MAIN/MASTER/M11/WINE")

wine = data.frame(read.xlsx("SOURCE/SOURCE.xlsx", sheetIndex = 1))

NewNames = c("clase","acidez_fija","acidez_volatile","acidez_citrica","azucar_res","cloruros","sulfitos_libres","sulfitos_totales","densidad","ph","sulfatos","alcohol","calidad")

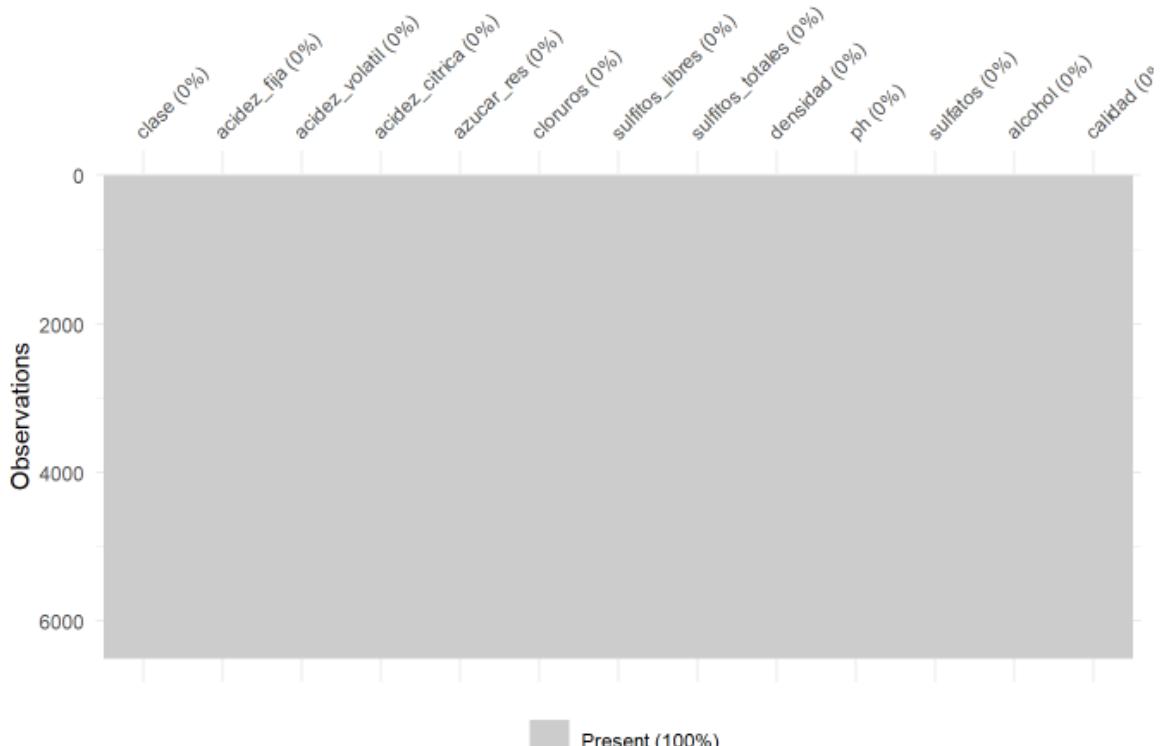
names(wine)=NewNames
    
```

Limpieza y visualización

```

# Chequeamos los NA del dataset

vis_miss(wine)
    
```



```
###Ninguno
```

```
kable(head(wine)) %>% kable_styling()
```

clase	acidez_fija	acidez_volatile	acidez_citrica	azucar_res	cloruros	sulfitos_libres	sulfitos_totales	densidad	ph	sulfatos	alcohol	calidad
WHITE	7.0	0.27	0.36	20.7	0.045	45		170	1.0010	3.00	0.45	8.8
WHITE	6.3	0.30	0.34	1.6	0.049	14		132	0.9940	3.30	0.49	9.5
WHITE	8.1	0.28	0.40	6.9	0.050	30		97	0.9951	3.26	0.44	10.1
WHITE	7.2	0.23	0.32	8.5	0.058	47		186	0.9956	3.19	0.40	9.9
WHITE	7.2	0.23	0.32	8.5	0.058	47		186	0.9956	3.19	0.40	9.9
WHITE	8.1	0.28	0.40	6.9	0.050	30		97	0.9951	3.26	0.44	10.1

```
wine %>% summary() %>% kable() %>% kable_styling()
```

clase	acidez_fija	acidez_volatile	acidez_citrica	azucar_res	cloruros	sulfitos_libres	sulfitos_totales	densidad	ph	sulfatos	alcohol	calidad
RED :1599	Min. : 3.800	Min. :0.0800	Min. :0.0000	Min. : 0.600	Min. :-0.00900	Min. : 1.00	Min. : 6.0	Min. : 0.9871	Min. :2.720	Min. :0.2200	Min. : 8.00	Min. :3.000
WHITE:4898	1st Qu.: 6.400	1st Qu.:0.2300	1st Qu.:0.2500	1st Qu.: 1.800	1st Qu.:0.03800	1st Qu.:17.00	1st Qu.: 77.0	1st Qu.: 0.9923	1st Qu.:3.110	1st Qu.:0.4300	9.50	1st Qu.:5.000
NA	Median : 7.000	Median :0.2900	Median :0.3100	Median : 3.000	Median :0.04700	Median :29.00	Median :118.0	Median : 0.9949	Median :3.210	Median :0.5100	Median :10.30	Median :6.000
NA	Mean : 7.215	Mean :0.3397	Mean :0.3186	Mean : 5.443	Mean :0.05603	Mean :30.53	Mean :115.7	Mean : 0.9947	Mean :3.219	Mean :0.5313	Mean :10.49	Mean :5.818
NA	3rd Qu.: 7.700	3rd Qu.:0.4000	3rd Qu.:0.3900	3rd Qu.: 8.100	3rd Qu.:0.06500	3rd Qu.:41.00	3rd Qu.:156.0	3rd Qu.: 0.9970	3rd Qu.:3.320	3rd Qu.:0.6000	3rd Qu.:11.30	3rd Qu.:6.000
NA	Max. :15.900	Max. :1.5800	Max. :1.6600	Max. :65.800	Max. :0.61100	Max. :289.00	Max. :440.0	Max. : 1.0390	Max. :4.010	Max. :2.0000	Max. :14.90	Max. :9.000

```
str(wine)
```

```
## 'data.frame': 6497 obs. of 13 variables:
## $ clase : Factor w/ 2 levels "RED","WHITE": 2 2 2 2 2 2 2 2 2 2 ...
## $ acidez_fija : num 7 6.3 8.1 7.2 7.2 8.1 6.2 7 6.3 8.1 ...
## $ acidez_volatile : num 0.27 0.3 0.28 0.23 0.23 0.28 0.32 0.27 0.3 0.22 ...
## $ acidez_citrica : num 0.36 0.34 0.4 0.32 0.32 0.4 0.16 0.36 0.34 0.43 ...
## $ azucar_res : num 20.7 1.6 6.9 8.5 8.5 6.9 7 20.7 1.6 1.5 ...
## $ cloruros : num 0.045 0.049 0.05 0.058 0.058 0.05 0.045 0.045 0.049 0.044 ...
## $ sulfitos_libres : num 45 14 30 47 47 30 30 45 14 28 ...
## $ sulfitos_totales: num 170 132 97 186 186 97 136 170 132 129 ...
## $ densidad : num 1.001 0.994 0.995 0.996 0.996 ...
## $ ph : num 3 3.3 3.26 3.19 3.19 3.26 3.18 3 3.3 3.22 ...
## $ sulfatos : num 0.45 0.49 0.44 0.4 0.4 0.44 0.47 0.45 0.49 0.45 ...
## $ alcohol : num 8.8 9.5 10.1 9.9 9.9 10.1 9.6 8.8 9.5 11 ...
## $ calidad : num 6 6 6 6 6 6 6 6 6 6 ...
```

#Se confirman que sólo el atributo clase, agregado al datos et original como factor

Se observan valores relativamente esperables, según los valores comprendidos como "usuales" según el estudio preliminar que hemos realizado. Valores importantes como la densidad, alcohol, pH, sulfitos totales, cloruros o acidez se encuentran dentro del rango de valores esperado.

También puede observarse que hay máximos que muestran valores fuera de los percentiles altos y que se candidatan como outliers, como es el caso de los sulfitos libres, totales, y el azúcar residual.

En general, el set de datos está bastante limpio y apenas requiere de limpieza, aparte de los outliers, que no se eliminarán aun hasta hacer el análisis exploratorio, a excepción de los más fuertes. Se procederá a la división por tipo de vino.

División por clase

Se crean los datasets de las dos clases de vino por separado, eliminando el atributo clase y eliminando outliers fuertes:

```
white = wine %>% filter(clase=="WHITE") %>% select(-clase)

white = remove_outliers(white, 0.999)

##Eliminamos los outliers más fuertes (por encima del percentil 99.9)

red = wine %>% filter(clase=="RED") %>% select(-clase)

red = remove_outliers(red, 0.999)
```

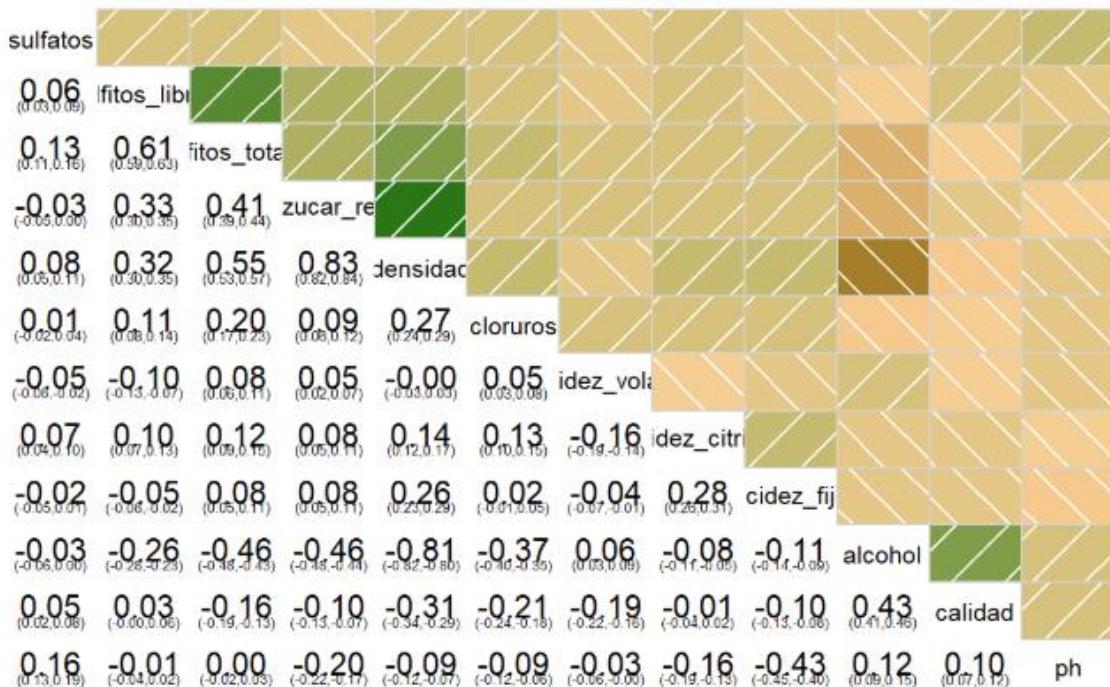
2.2. ANÁLISIS DESCRIPTIVO

Mapa de correlaciones de ambos datasets:

- Vino Blanco

```
cor_white = corrgram(white, type="data",
                     lower.panel=panel.conf,
                     upper.panel=panel.shade,
                     main= "Mapa de correlaciones del vino blanco",
                     order=T,
                     cex.labels=1.2,
                     col.regions = colorRampPalette(c("darkgoldenrod4", "burlywood1", "darkkhaki", "darkgreen")))
```

Mapa de correlaciones del vino blanco



```
correlation_white = round(as.data.frame(cor(as.matrix(white))), 2)
```

```

correlation_white[abs(correlation_white)<0.4] = "★"

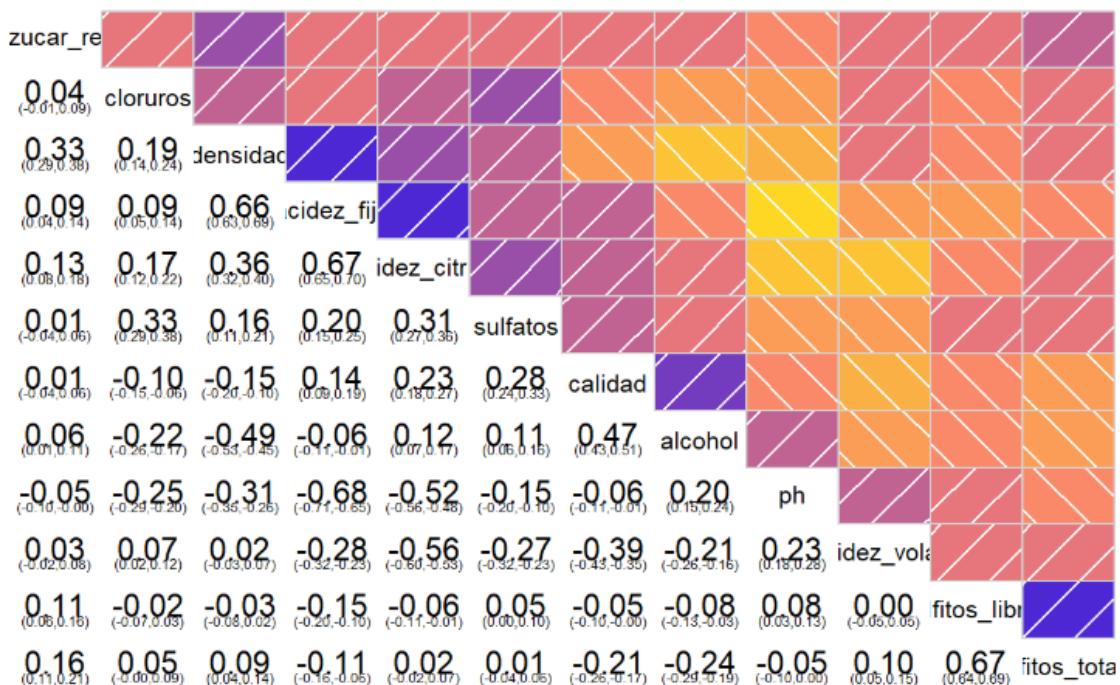
correlation_white %>%
  kable() %>% kable_styling(bootstrap_options = "striped", full_width =
F) %>%
  add_header_above(c(" ", "Vino blanco" = 12))
    
```

	Vino blanco											
	acidez_fija	acidez_volatil	acidez_citrica	azucar_res	cloruros	sulfitos_liberes	sulfitos_totales	densidad	ph	sulfatos	alcohol	calidad
acidez_fija	1	•	•	•	•	•	•	•	-0.43	•	•	•
acidez_volatil	•	1	•	•	•	•	•	•	•	•	•	•
acidez_citrica	•	•	1	•	•	•	•	•	•	•	•	•
azucar_res	•	•	•	1	•	•	0.41	0.83	•	•	-0.46	•
cloruros	•	•	•	•	1	•	•	•	•	•	•	•
sulfitos_liberes	•	•	•	•	•	1	0.61	•	•	•	•	•
sulfitos_totales	•	•	•	0.41	•	0.61	1	0.55	•	•	-0.46	•
densidad	•	•	•	0.83	•	•	0.55	1	•	•	-0.81	•
ph	-0.43	•	•	•	•	•	•	•	1	•	•	•
sulfatos	•	•	•	•	•	•	•	•	•	1	•	•
alcohol	•	•	•	-0.46	•	•	-0.46	-0.81	•	•	1	0.43
calidad	•	•	•	•	•	•	•	•	•	•	•	1

- Los predictores de azúcar, pH y ácido cítrico no juegan un papel aparentemente relevante en la calidad del vino. Las correlaciones son débiles entre la calidad y el ácido cítrico, los sulfitos libres y sulfatos.
- La densidad tiene una correlación de 0.83 con el azúcar residual y una correlación de -0.81 con alcohol. El alcohol es el único predictor que está considerablemente relacionado con la calidad del vino, y probablemente el más independiente.
- Existen correlaciones fuertes entre el alcohol-azucar_res, alcohol-dens, alcohol-sulfitos_totales, dens-azucar_res, dens-sulfitos_totales, sulfitos_totales-libres, y sulfitos_totales-azucar_res
- Vino Tinto

```
cor_red = corrgram(red, type="data",
                    lower.panel=panel.conf,
                    upper.panel=panel.shade,
                    main= "Mapa de correlaciones del vino tinto",
                    order=T,
                    cex.labels=1.2,
                    col.regions = colorRampPalette(c("yellow", "salmon", "blue")))
```

Mapa de correlaciones del vino tinto



```
correlation_red = round(as.data.frame(cor(as.matrix(red))), 2)

correlation_red[abs(correlation_red)<0.4] = "***

correlation_red %>%
  kable() %>% kable_styling(bootstrap_options = "striped", full_width =
F) %>%
  add_header_above(c(" ", "Vino blanco" = 12))
```



	Vino tinto											
	acidez_fija	acidez_volatil	acidez_citrica	azucar_res	cloruros	sulfitos_liberes	sulfitos_totales	densidad	ph	sulfatos	alcohol	calidad
acidez_fija	1	-	0.67	-	-	-	-	0.66	-0.68	-	-	-
acidez_volatil	-	1	-0.56	-	-	-	-	-	-	-	-	-
acidez_citrica	0.67	-0.56	1	-	-	-	-	-	-0.52	-	-	-
azucar_res	-	-	-	1	-	-	-	-	-	-	-	-
cloruros	-	-	-	-	1	-	-	-	-	-	-	-
sulfitos_liberes	-	-	-	-	-	1	0.67	-	-	-	-	-
sulfitos_totales	-	-	-	-	-	-	1	-	-	-	-	-
densidad	0.66	-	-	-	-	-	-	1	-	-	-0.49	-
ph	-0.68	-	-0.52	-	-	-	-	-	1	-	-	-
sulfatos	-	-	-	-	-	-	-	-	-	1	-	-
alcohol	-	-	-	-	-	-	-	-0.49	-	-	1	0.47
calidad	-	-	-	-	-	-	-	-	-	-	0.47	1

- Entre los dos vinos se dan algunas similitudes importantes, como es la correlación calidad-alcohol, alcohol-densidad (aunque con bastante menos fuerza), acidez-fija-pH, pero es de esperar pues se trata, como se explicaba en la introducción, de variables físico-químicamente dependientes.
- Sin embargo en el vino tinto se dan correlaciones únicas, como son la relación entre la acidez fija y volátil con la acidez cítrica. De hecho, la acidez cítrica tiene especial relevancia en el vino tinto, ya que está también estrechamente ligada al pH. Los sulfitos, en cambio, pierden importancia, y la acidez fija se correlaciona con fuerza con la densidad.

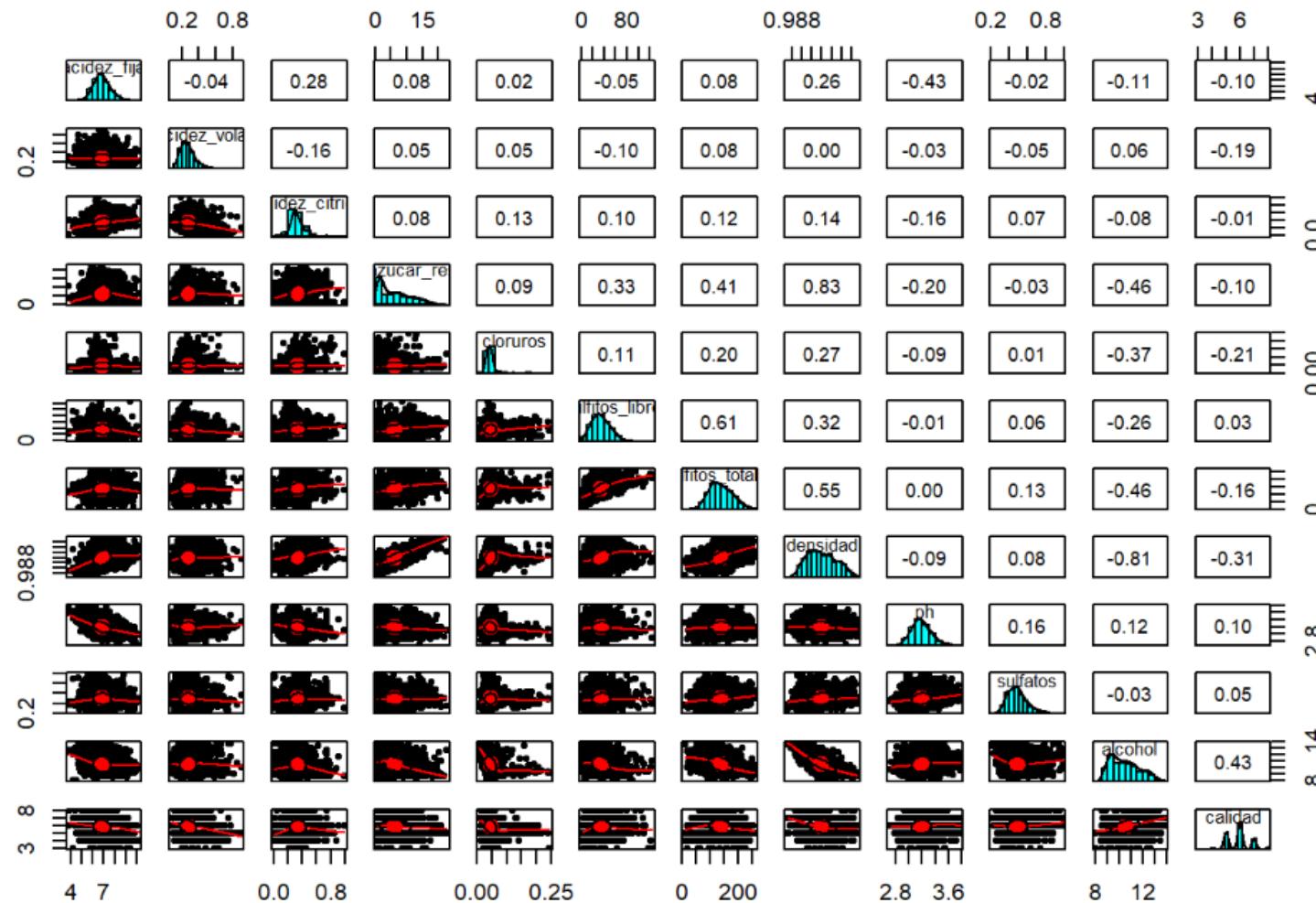
En el análisis exploratorio se ahondará en estas cuestiones.

Se usa, como antecedente para el análisis exploratorio, la magnífica función pairs.panel de la librería psych, que genera un plot con un mapa completo de SPLOM, correlaciones e histogramas de las variables continuas, aunque lo veremos más adelante en el análisis exploratorio con Tableau.



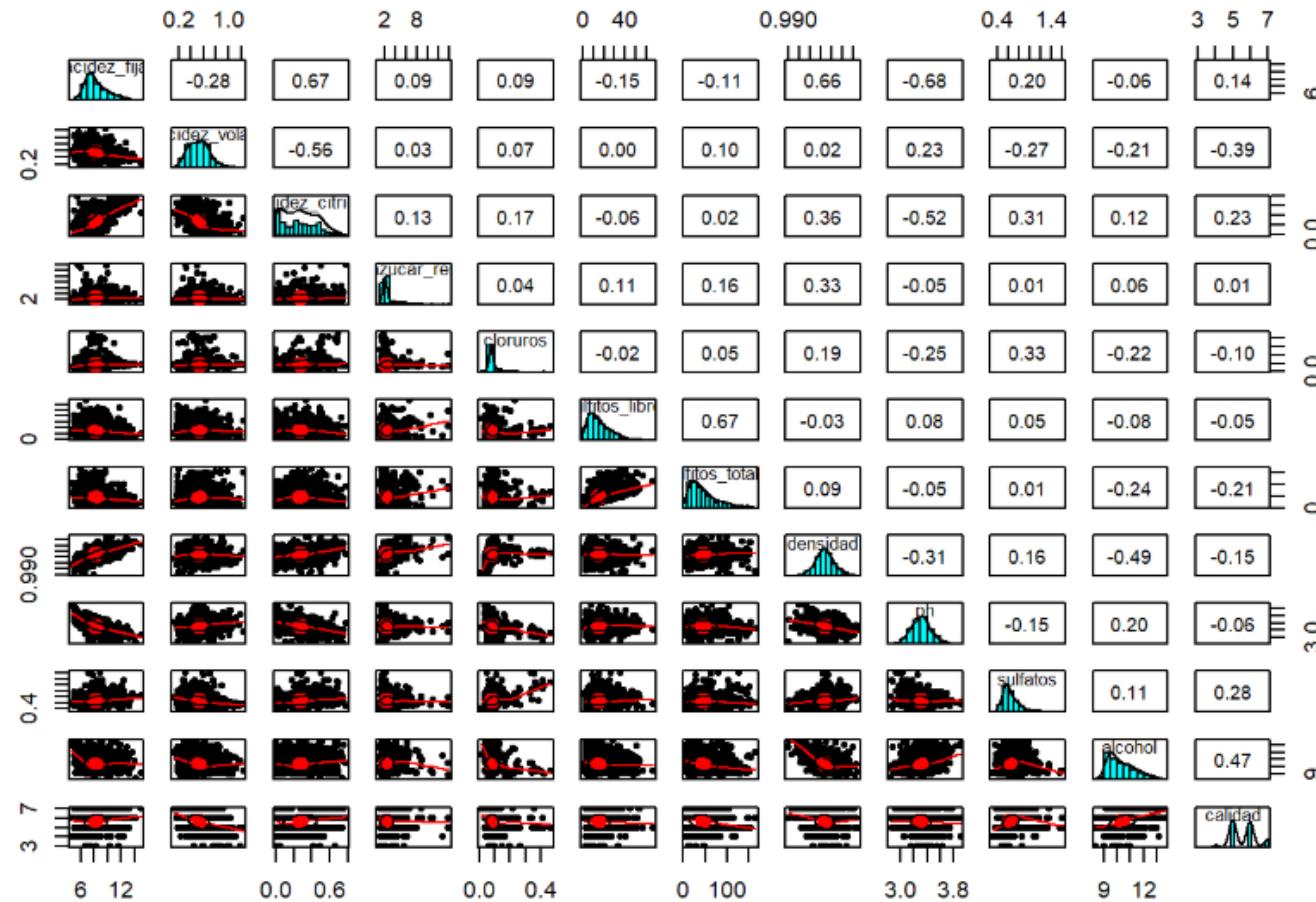
- Mapa completo de vino blanco

```
pairs.panels(white)
```



- Mapa completo de vino tinto

```
pairs.panels(red)
```



Dado que se tratan de muchos atributos cruzados, no es fácil contemplar las tendencias de todos los predictores.

3. ANÁLISIS EXPLORATORIO

En el análisis exploratorio se ahonda de manera meticulosa en el comportamiento de los predictores / clasificadores, a fin de comprender en mayor profundidad y visualizar anomalías y tendencias imperceptibles en los análisis descriptivos y en los conjuntos de datos.

Para esta parte se va a utilizar la herramienta de Tableau (desde la plataforma de Tableau Public, y se embeberán los gráficos y resultados con los correspondientes comentarios, para facilitar en lo posible su comprensión, así como su síntesis.

Se partirá de la base de conocimiento adquirida en el análisis descriptivo^{*1}:

SÍNTESIS DE LAS CORRELACIONES MÁS DESTACABLES

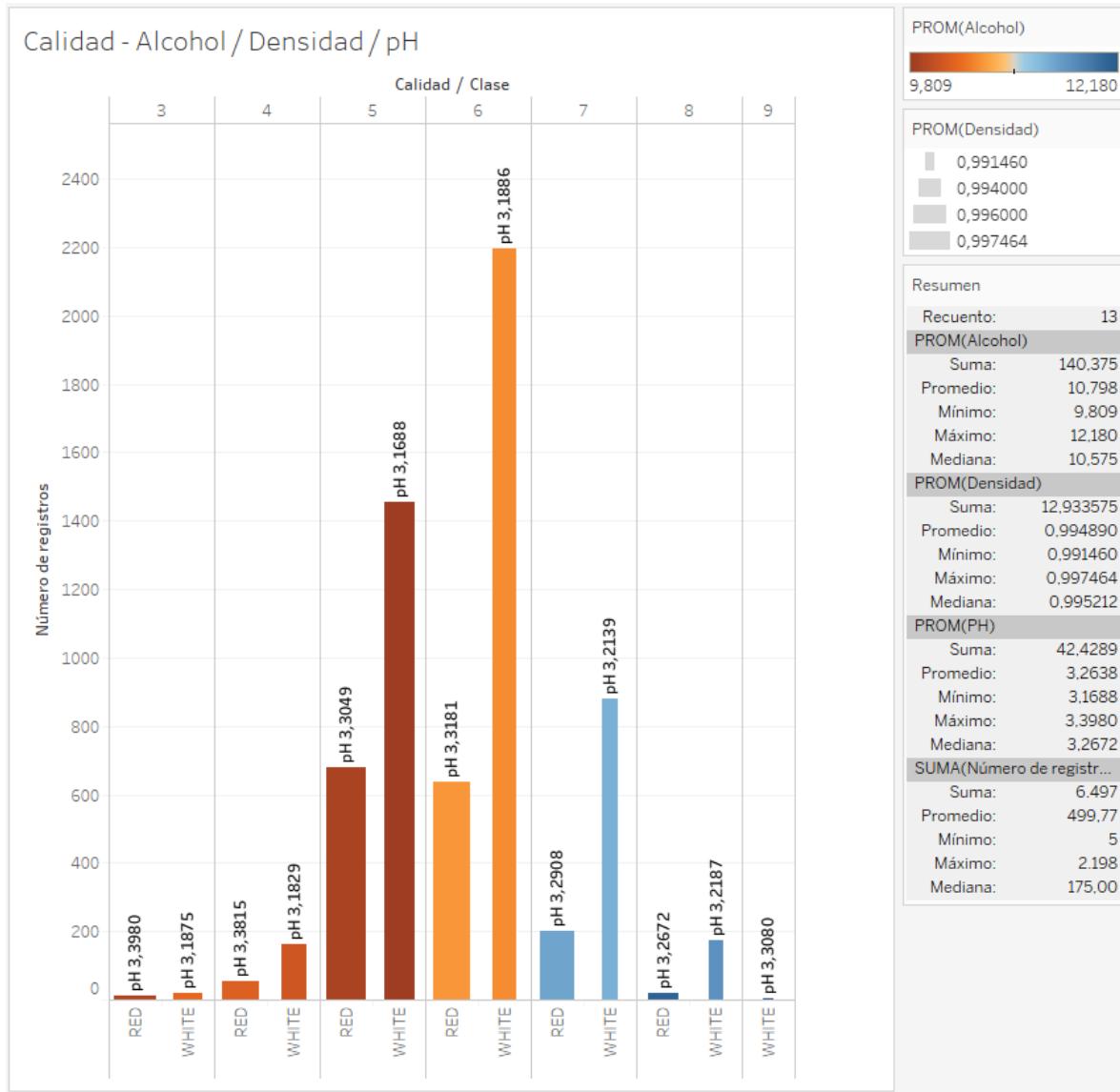
Vino Blanco		Vino Tinto	
Calidad-Alcohol	0,43	Calidad-Alcohol	0,47
Alcohol-Densidad	-0,81	Alcohol-Densidad	-0,49
pH-Acidez fija	-0,43	pH-Acidez Fija	-0,68
Alcohol-Azúcar res	-0,46	Densidad-Acidez fija	0,66
Alcohol-Sulfitos totales	-0,46	Acidez cítrica-Acidez fija	0,67
Densidad-Azúcar residual	0,83	Acidez cítrica-Acidez volátil	-0,56
Densidad-Sulfitos totales	0,55	Acidez cítrica-pH	-0,52
Sulfitos totales-libres	0,61		
Sulfitos totales – Azúcar res	0,41		

Nótese el hecho de que, en el vino blanco posee un promedio más elevado del valor absoluto de las correlaciones destacables (0,55) frente a las del tinto (0,50). La Acidez general en el tinto es mucho más prominente que en el blanco, donde el alcohol posee un papel muy relevante.

^{*1} Las correlaciones subrayadas corresponden a correlaciones compartidas entre ambos vinos

3.1. ANÁLISIS DE HISTOGRAMAS Y SCATTERS

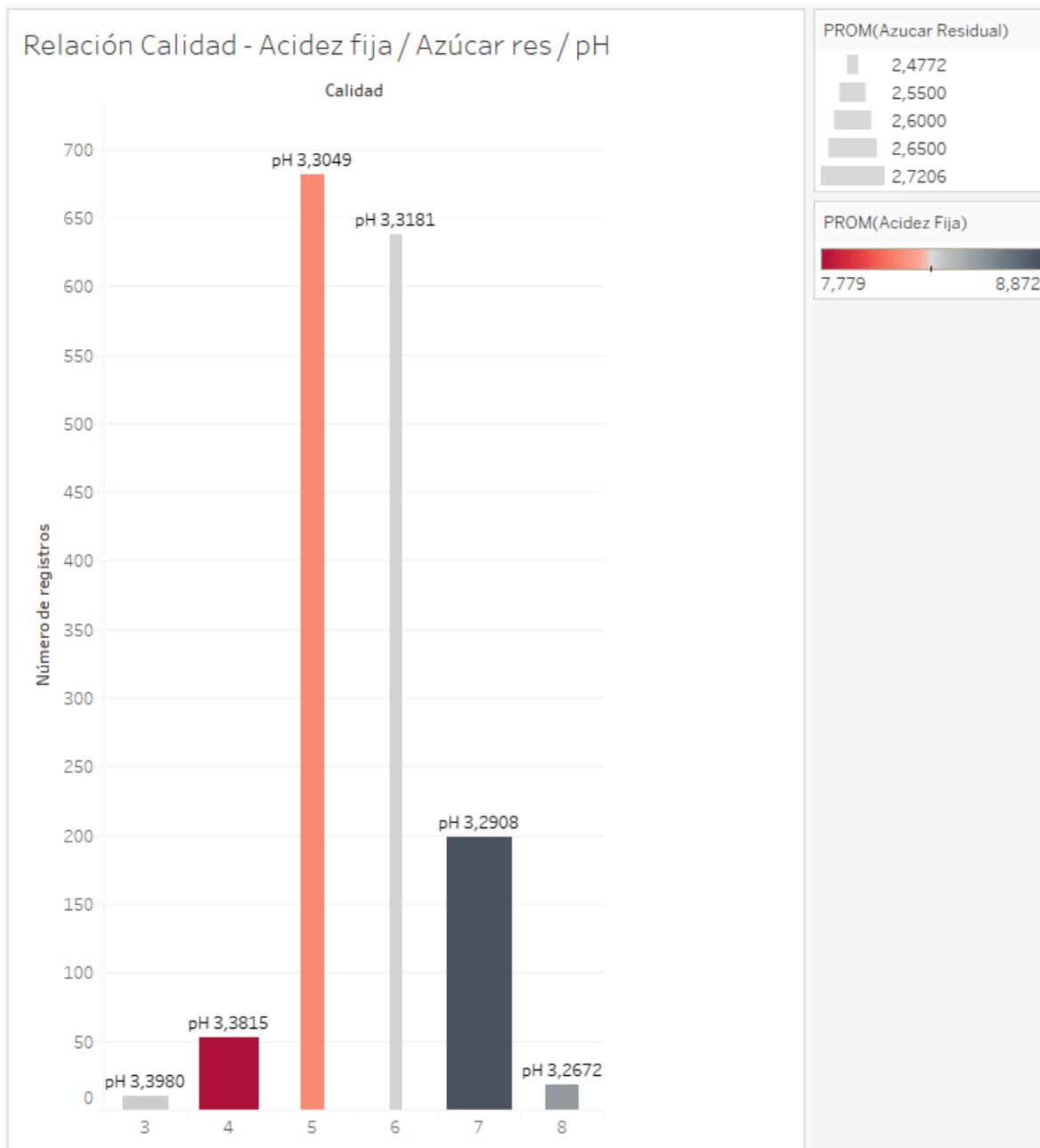
Relación Blanco-Tinto Calidad – Alcohol / Densidad / pH (Corr compartidas)



- Los vinos de alta calidad (rojo y blanco) poseen mayores proporciones de alcohol
- La densidad, como ya se vaticinaba en la introducción, evoluciona *inversamente proporcional* al alcohol. Este dato se verá mejor reflejado en un diagrama de cajas.
- El pH es mayor en las regiones de calidad media del vino, haciéndose menor en los extremos

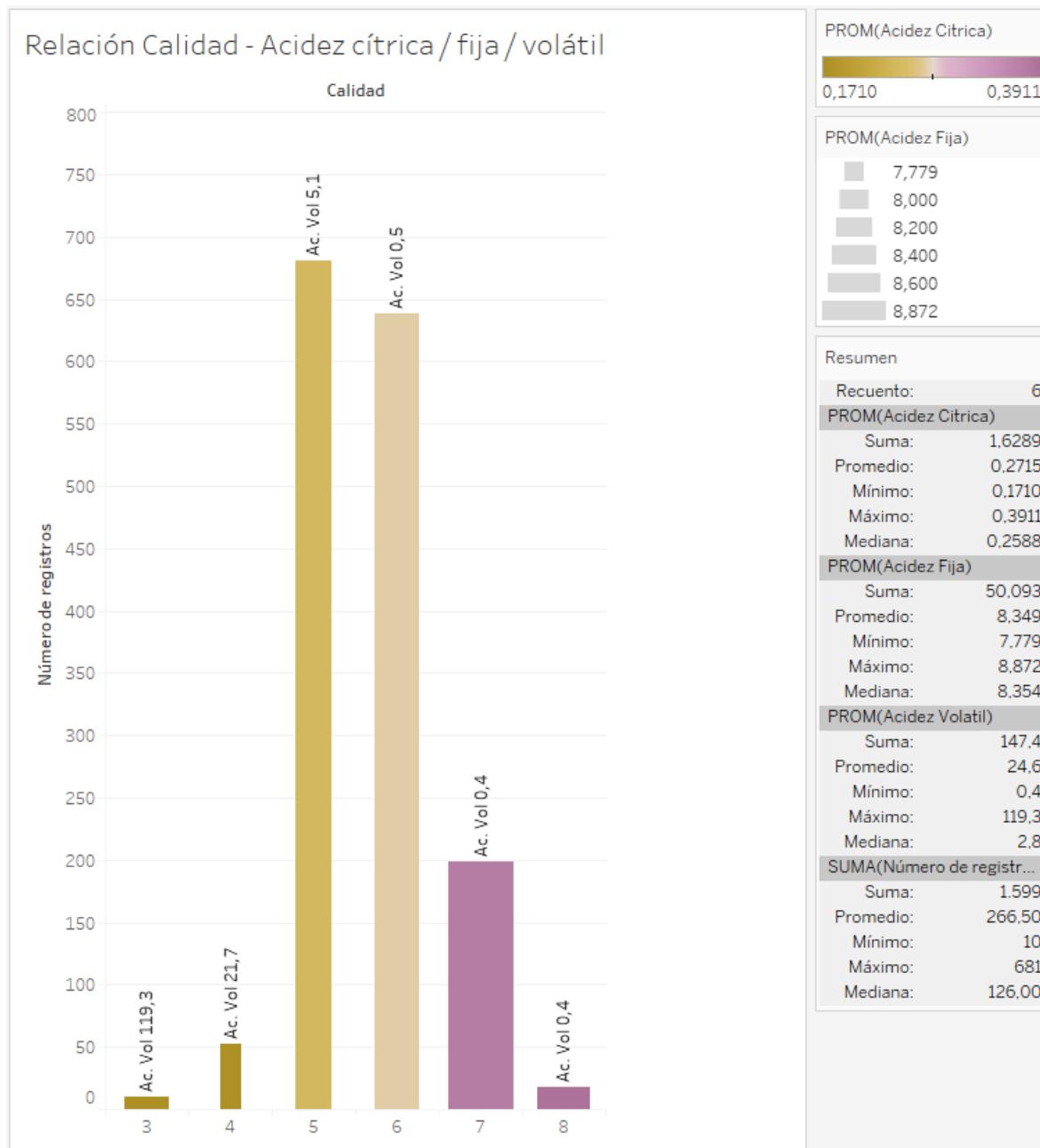
Tinto

Relación Calidad – Acidez fija / Azúcar res / pH

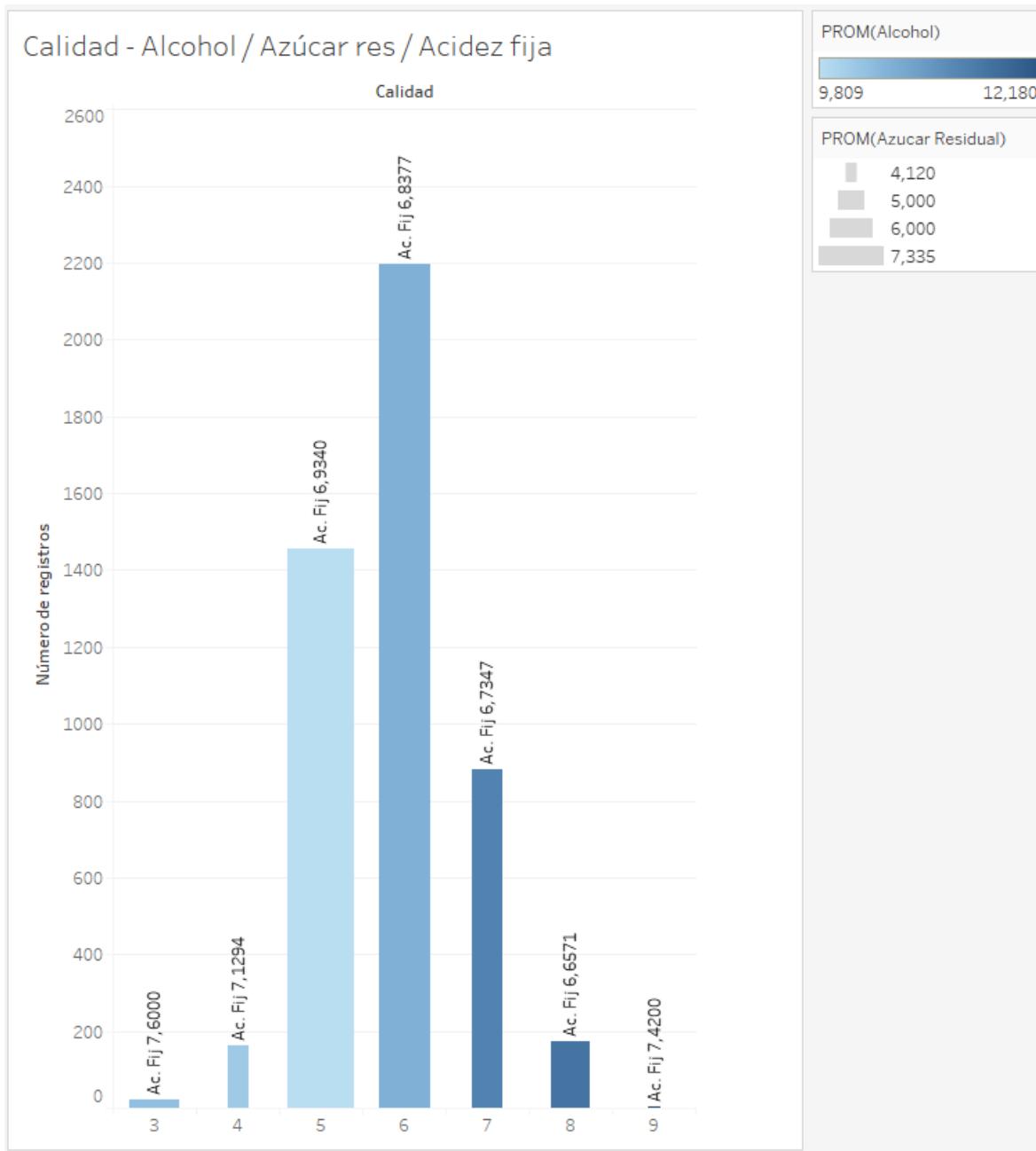


- Los resultados son bastante sugerentes: La acidez fija aumenta conforme aumenta la calidad del vino. Es probable, ya que se trata de un atributo *fuerte* del vino tinto, que sea altamente responsable de su calidad, y por tanto un predictor importante
- El azúcar residual no juega un papel especialmente importante en el tinto, aunque, al inversamente proporcional al pH, parece aumentar en las regiones extremas de la calidad.

Relación Calidad – Acidez cítrica / fija / volátil



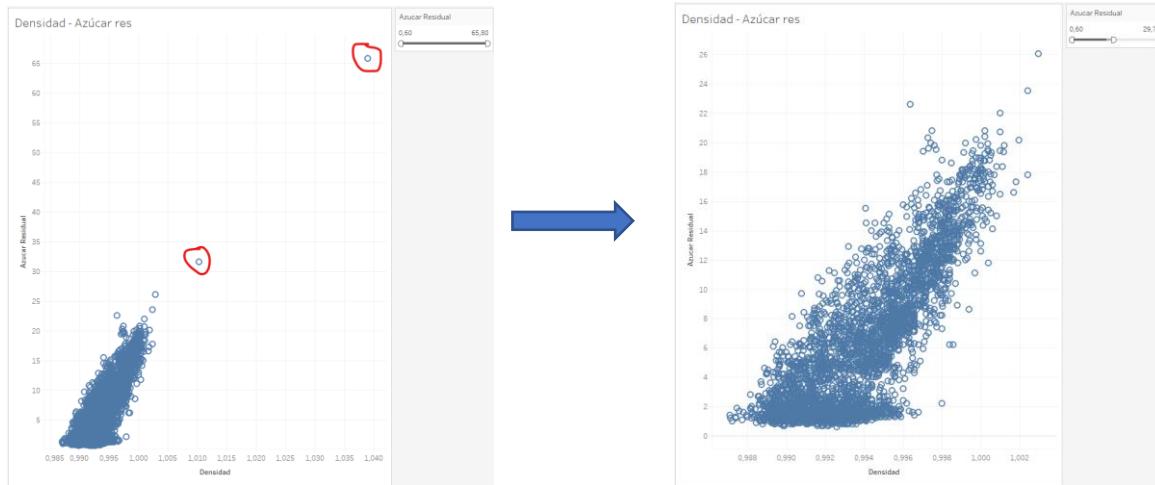
- La calidad del vino se ve favorecida fuertemente por el aumento del ácido cítrico y la acidez fija, mientras que ocurre lo contrario con la acidez volátil
- De hecho, la acidez volátil decrece dramáticamente conforme vamos mejorando de calidad (aunque existe una alta probabilidad de haber outliers)

Blanco
Relación Calidad – Alcohol / Azúcar res / pH


- De nuevo el alcohol se alza como atributo / predictor principal ligado a la calidad, mientras que el contenido en azúcar residual y la acidez parecen evolucionar de forma inversa:
 - El azúcar residual se concentra en las regiones de calidad media
 - La acidez fija parece ser mayor en las regiones periféricas de la calidad del vino, de manera que es probable que la acidez fija en el vino blanco

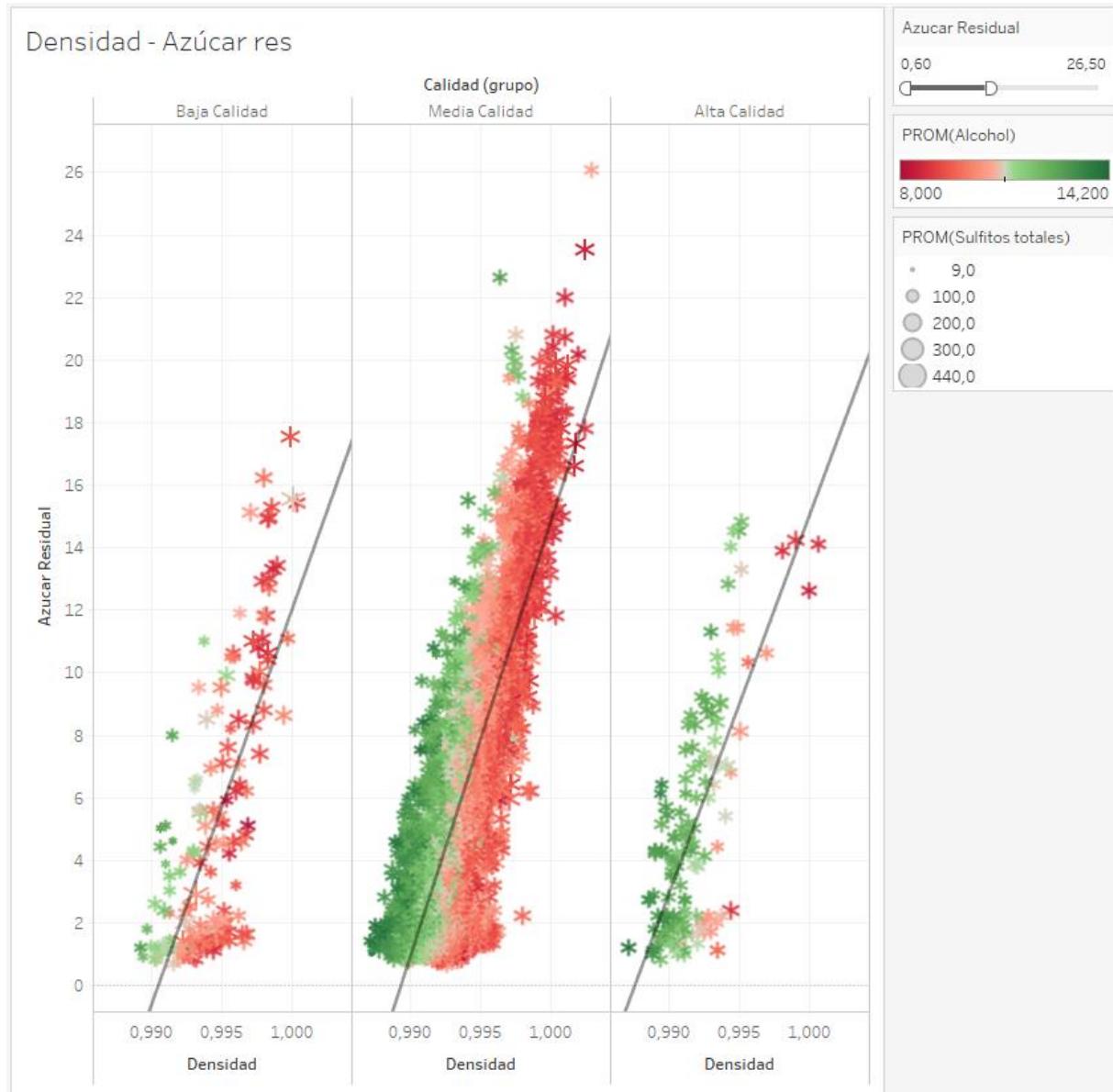
no cumplía un papel tan importante como predictor como si lo hacía en el vino tinto

Relación Densidad –Azúcar res



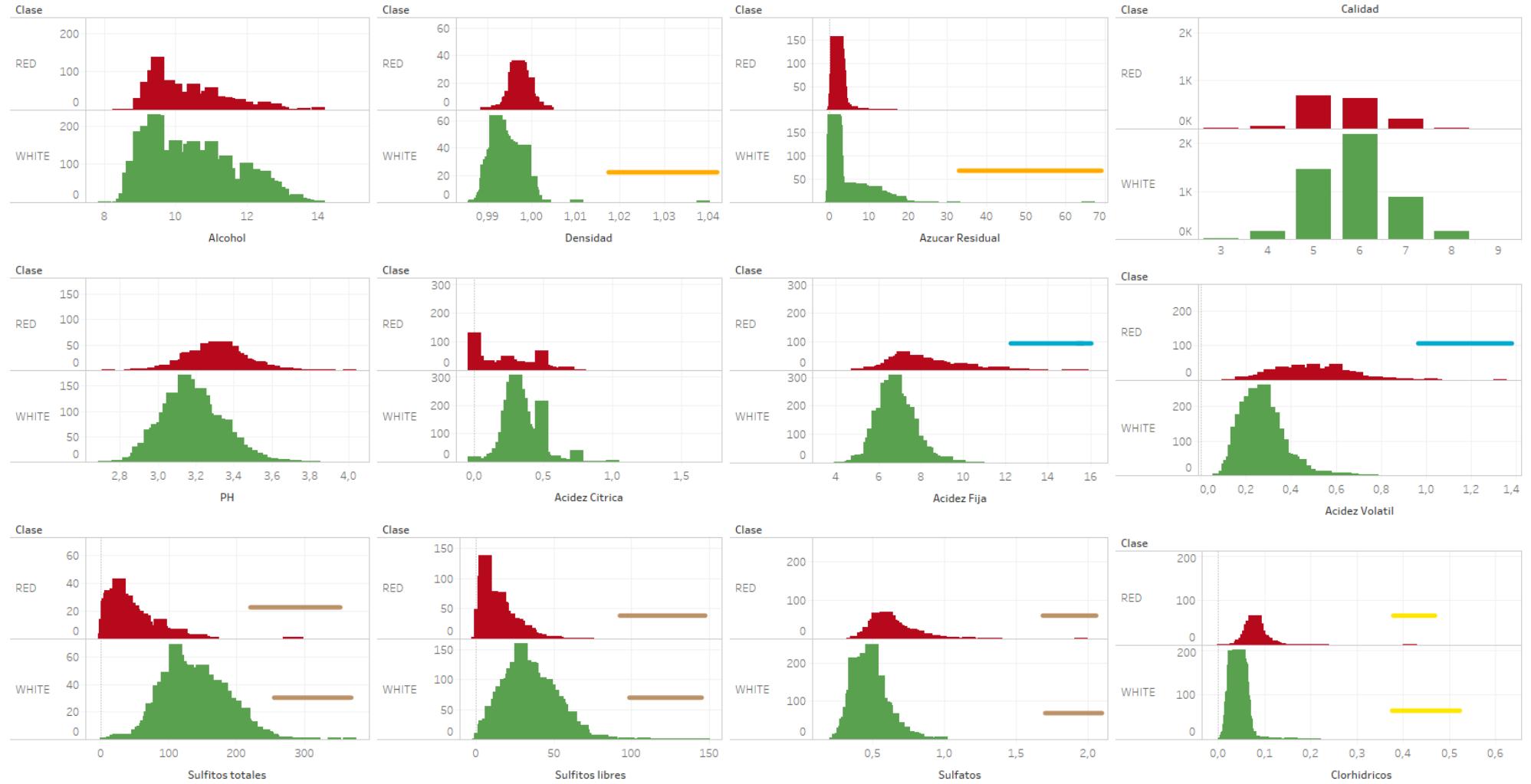
Las anomalías o incongruencias observadas anteriormente acerca del comportamiento del azúcar residual en el vino blanco se estaban viendo alteradas por dos outliers (marcados en rojo) que parecían estar afectando a la medida promedio con relativa severidad.

Se escoge estudiar la relación entre Densidad-Azúcar residual porque poseen una correlación fuerte. Este análisis, incorporándole la variable alcohol (debido a las interdependencias o colinealidades, y los sulfitos totales. El resultado es el siguiente:

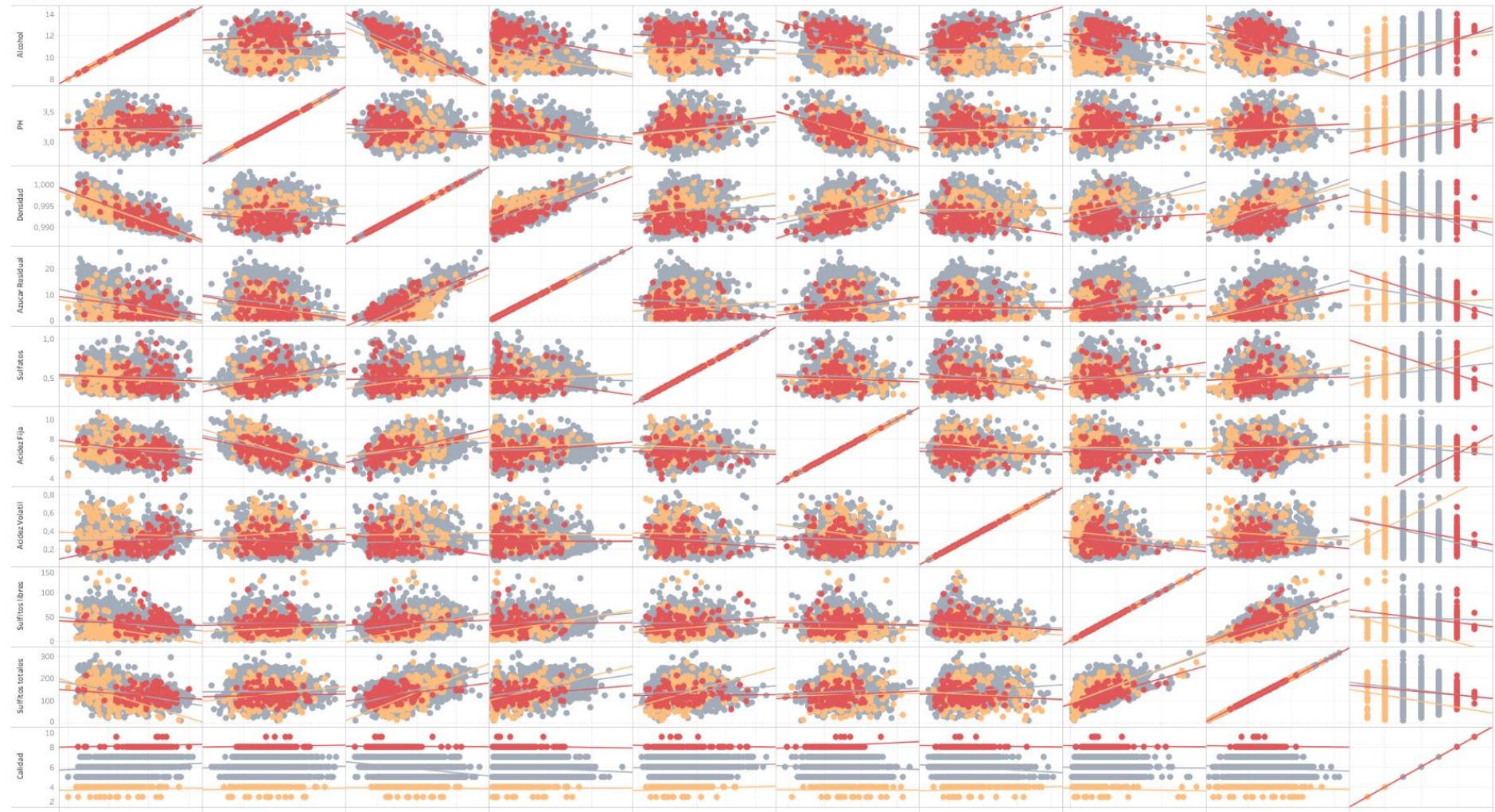


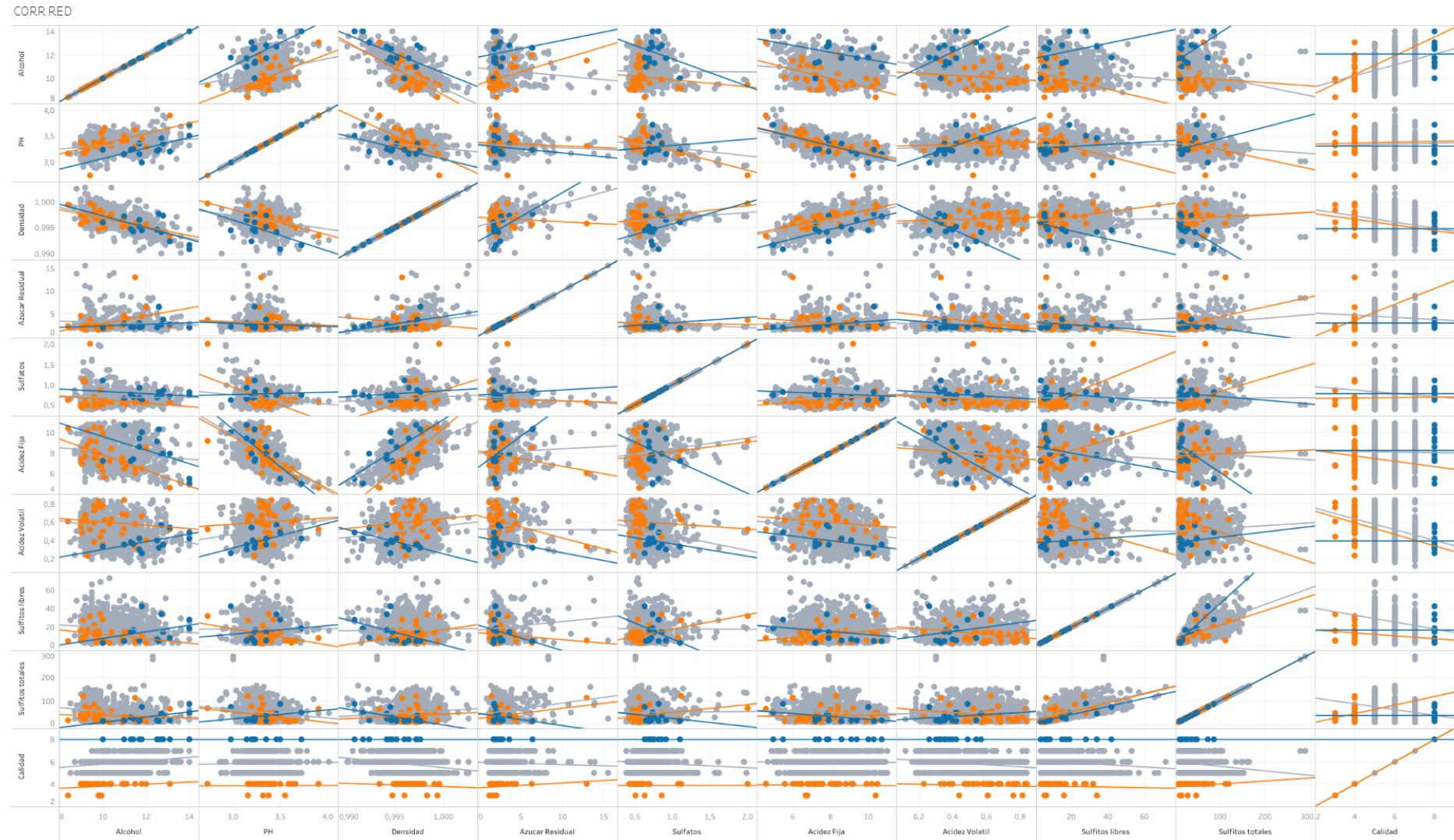
- La densidad crece conforme aumenta el azúcar residual aumenta
- El nivel de alcohol es mayor conforme menor es la densidad, como sabemos, y se concentra en las regiones de baja densidad y azúcar, lo cual confirma la relación natural química de dichos atributos
- Los sulfitos totales, están correlacionados con cierta consideración (en torno al 0,4) con el alcohol y el azúcar res; el tamaño de las partículas aumenta al aumentar el azúcar res, y es menor en colores verdes (alcohol alto)

3.2. ANÁLISIS EXHAUSTIVO DE CORRELACIONES, DISTRIBUCIONES Y TENDENCIAS



CORR WHITE





- **Outliers o valores atípicos**

Se destaca en la presencia de valores atípicos para la mayoría de los predictores. Para la representación de estos gráficos se han filtrado esos valores a fin de obtener gráficas que cualitativamente arrojen claridad y certeza en su lectura, pero el conjunto de muchos de esos valores, destacando principalmente la **acidez en general** (principalmente la acidez fija y cítrica en el vino tinto) y el **azúcar residual** generan algo de incertidumbre, ya que, según la información obtenida en la fuente, estos datos ya fueron previamente limpiados, hecho que se puede constatar en la amplia claridad y uniformidad descrita hasta el momento.

Concretamente, el outlier más fuerte analizado hasta ahora es el valor de 65,8 g/l del azúcar residual en el vino blanco, que dobla a la siguiente muestra en valor mayor por debajo de él en cantidad (31,6 g/l). En la Unión Europea, un vino con más de 45 g / l de azúcar se considera un vino dulce. El valor atípico tiene un nivel residual de azúcar de 65.8.

Los sulfitos libres tienen una muestra periférica mayor que 2 veces la siguiente más grande, pero como se ve en los mapas de dispersión y en los histogramas individuales, podrían estar relacionados con los vinos de baja calidad.

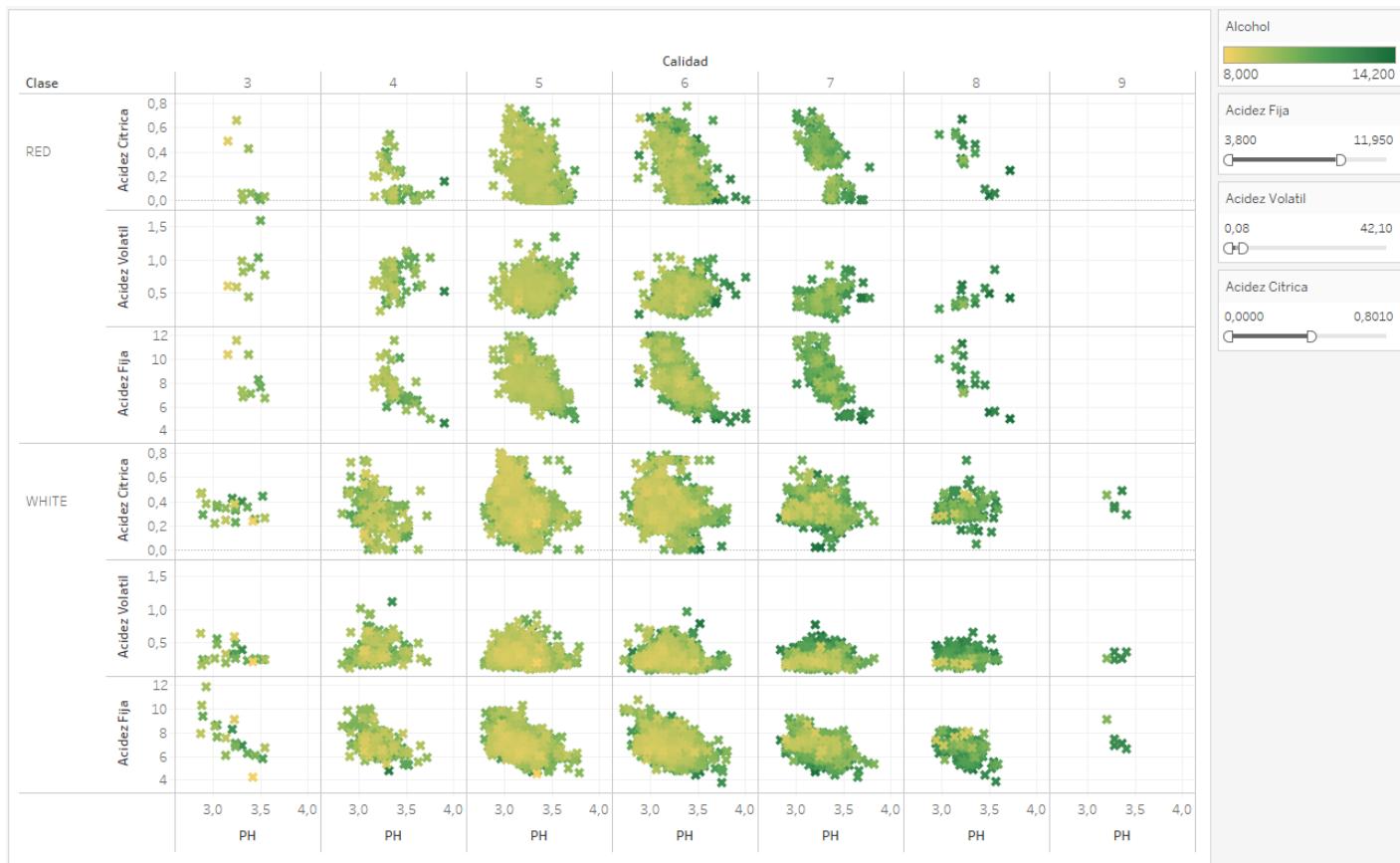
Se producen fenómenos similares con los predictores de la acidez fija, volátil, sulfitos totales, acidez cítrica... etc.

TRATAMIENTO DE LOS OUTLIERS

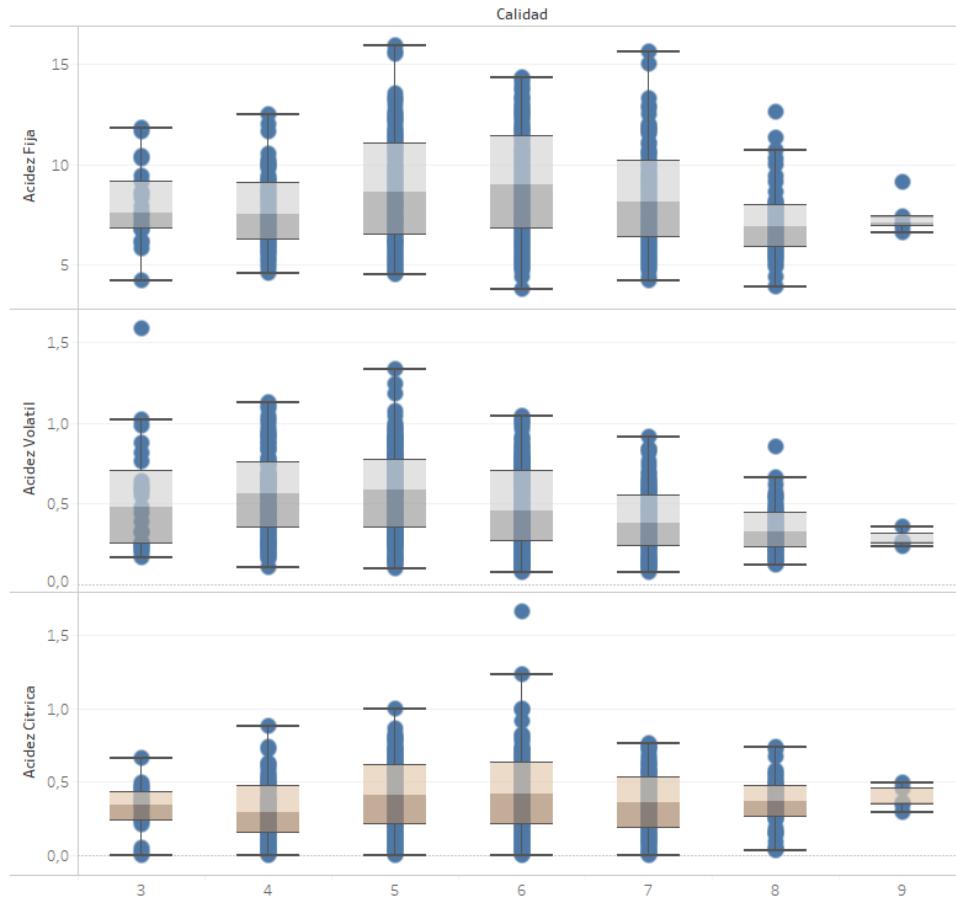
Predictor	Vino Blanco		Vino Tinto	
	Nº (<95%)	Eliminación	Nº (<95%)	Eliminación
Azúcar Residual	1	✓	-	-
Densidad	1	✓	-	-
Sulfitos libres	6	X	4	X
Sulfitos totales				
Sulfatos	-	X	4	X
Acidez volátil	-	X	2	✓
Acidez cítrica	2	✓	-	X
Acidez Fija	-	X	-	X
Cloruros	1	X	1	✓

El criterio principal escogido para eliminar o no los outliers se resumen en que dichos valores se encuentran por encima de un percentil muy alto ($> \sim 99\%$), aunque existen técnicas más exhaustivas y precisas que usaremos en la programación

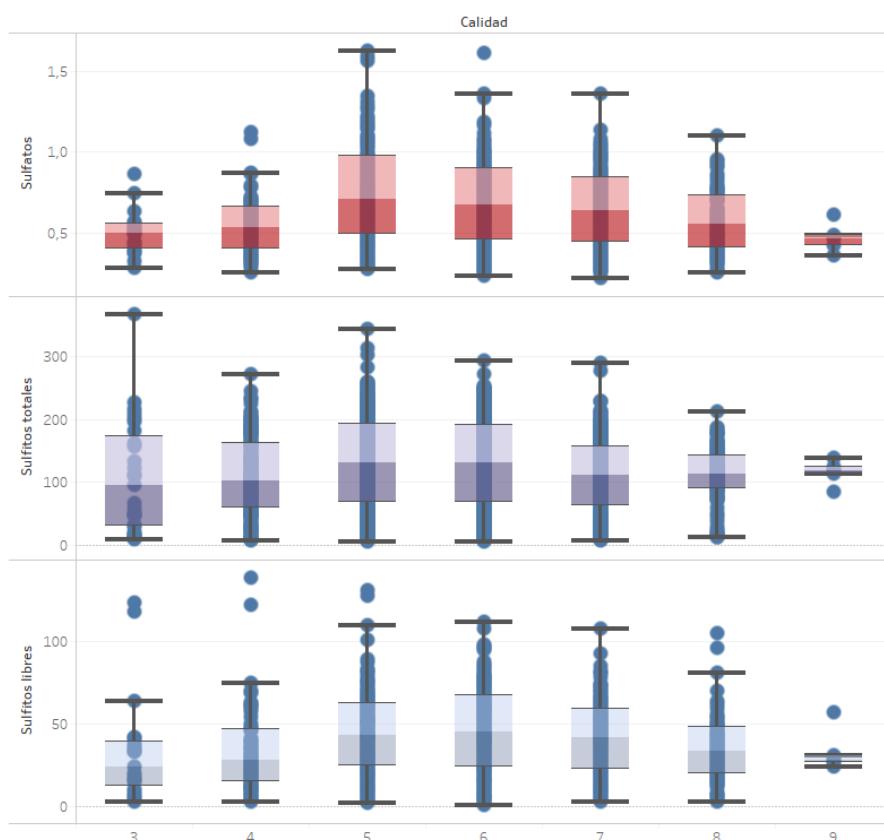
3.3. ANÁLISIS DE PREDICTORES DE MENOR CORRELACION *¹



*¹ Los diagramas de cajas se han considerado para los dos vinos juntos en los predictores de correlación baja, por simplificación y evitar reiteraciones



Se observa una concentración en torno al promedio del pH de la mayoría de los predictores, y una distribución equilibrada y equiparable a una distribución normal a lo largo de su rango de valores, como ya hemos comprobado en sus respectivos histogramas.



*¹ Los diagramas de cajas se han considerado para los dos vinos juntos

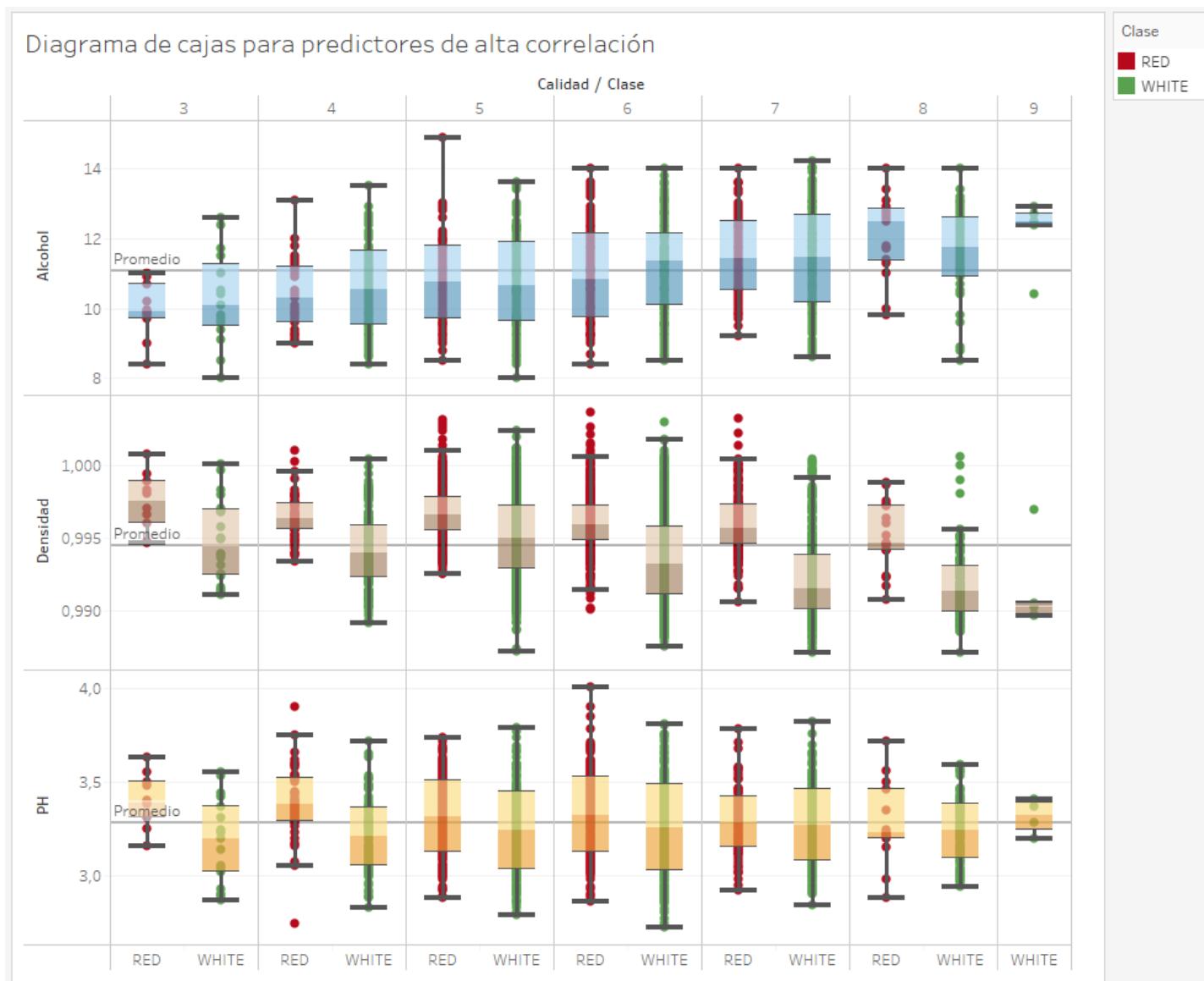
Se observa una distribución tendiente a normal, al igual que las variables ácidas en las variables para ambos vinos en general, y una fuerte presencia del alcohol en los vinos de mejor calidad.

Por otro lado, los gráficos de cajas muestran un comportamiento similar en todos los predictores de menor correlación:

El promedio aumenta en las regiones centrales y se modera en las periferias, y están presentes, como bien se ha analizado, múltiples outliers o valores atípicos.

3.4. ANÁLISIS FINALES

- **Diagramas de cajas**





Los predictores son claros:

- **Alcohol**

El alcohol tiende a aumentar conforme mejora la calidad. Un dato curioso que muestra el diagrama de cajas es que en las calidades periféricas, aunque notablemente en las regiones de alta calidad del vino, el rango de valores de alcohol se estrecha, esto quiere decir, **a mayores calidades, la graduación de alcohol es alta y de poca variabilidad.**

- En el caso del tinto, como puede observarse, las distribuciones de alcohol son más uniformes

- **Densidad**

La densidad es, como se ha dilucidado y considerado, una variable colineal al alcohol, de manera que su comportamiento es similar aunque en sentido inverso al alcohol.

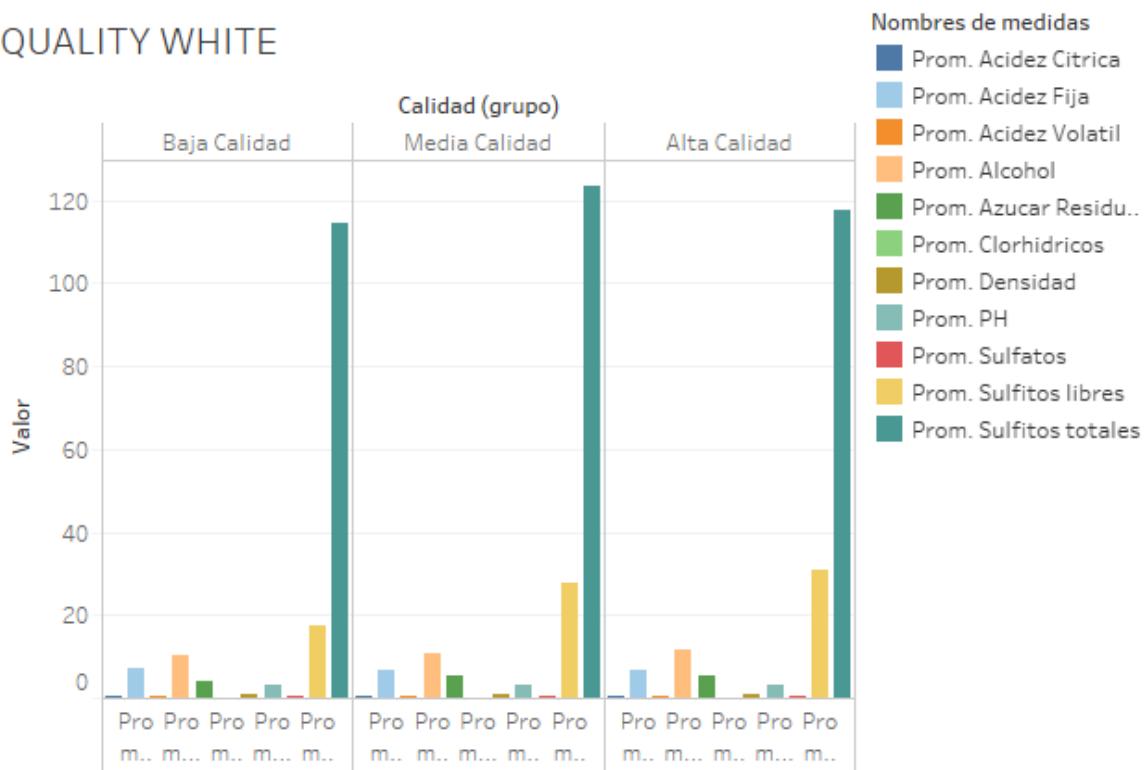
- En el caso del tinto, su decrecimiento es más suave, esto es, **tiene una distribución de densidad más uniforme y amplia a lo largo de las diferentes calidades, y es, probablemente, menos colineal que en el caso del vino blanco.**

- **pH**

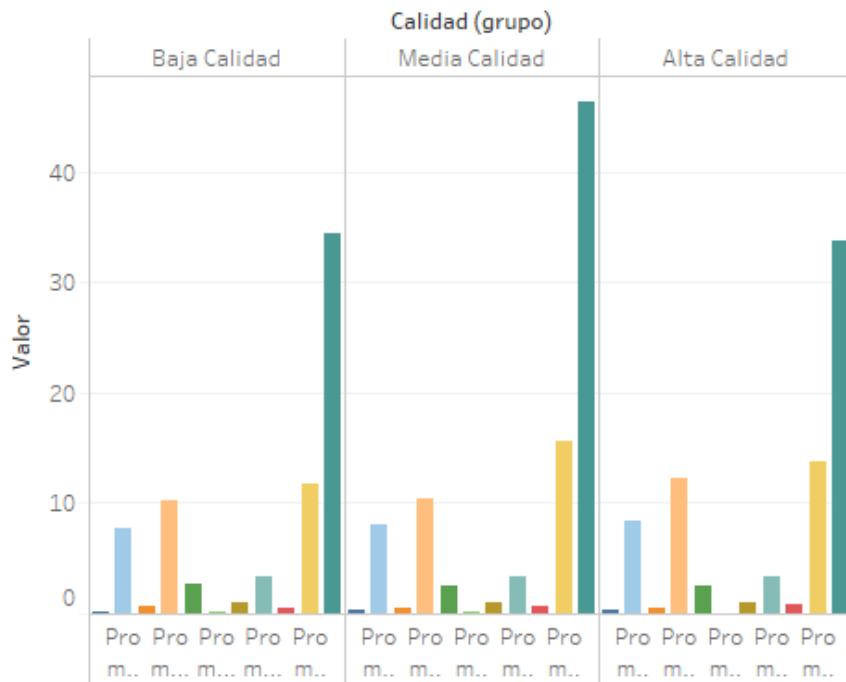
El pH es más alto en el tinto que en el blanco, pero su evolución es similar, mayor en el centro, menor en las periferias.

- Histogramas generales

QUALITY WHITE



QUALITY RED



En conclusión, para vinos de buena calidad, se prefiere:

- Un nivel de alcohol *moderadamente* alto
 - Un nivel de sulfitos libres *ligeramente* alto para el vino blanco y *ligeramente* bajo para el vino tinto
 - Un nivel de sulfitos totales *ligeramente* bajo para el vino blanco y *moderadamente* bajo para el vino tinto
 - El resto de variables pueden considerarse colineales de éstas, cumpliéndose, como se ha dicho, una proporción inversa en la densidad respecto al alcohol, y niveles más bajos en general para el resto de atributos en comparación con sus homólogos de menor calidad.
-
- **Preparación de los modelos matemáticos**

Los resultados del análisis presentan indicios de colinealidad entre las variables, además de correlaciones bajas y difusas entre sí. Del mismo modo, existen variables como el alcohol que están inequívocamente ligadas a la variable de respuesta

Las líneas de regresión y correlaciones de un buena parte de los predictores no son muy buenas lo que sugiere que los modelos de regresión / predicción quizás no sean lo suficientemente apropiados para generar modelos que sean fidedignos a la realidad.

La relación entre la respuesta (calidad) y los predictores es compleja y probablemente funcionen mejor modelos de clasificación no lineal. En cualquier caso se van a crear modelos de ambos tipos y se evaluará a posteriori su efectividad, así como las razones de por qué ha sido así.

Los modelos se van a generar en RStudio y se generarán predictores de diferentes clases, hasta que se alcance el modelo más adecuado.

4. MODELIZACIÓN MATEMÁTICA

La modelización se va a aplicar al vino tinto y blanco por separados. Para este proyecto se van a utilizar criterios y vías diferentes para cada vino, aprovechando que se tratan de datasets independientes, y que, tal como se ha dilucidado en el análisis exploratorio, poseen atributos con diferentes grados de importancia, lo cual enriquece el análisis. Finalmente se escogerá la vía, modelo y criterio que mejor se ajuste a las características del dataset y se aplicará por igual a los dos vinos. En resumen, este apartado se compondrá en 3 partes fundamentales:

1. PREDICCIÓN - ANÁLISIS DE VINO BLANCO

En este primer apartado se van a preparar los datos y a estudiar un número considerable de modelos de *regresión o predictivos*, también modelos usados en clasificación (como el modelo Random Forest), **pero con el objetivo de predecir la calidad**. Las condiciones en este punto son las siguientes:

- Variable respuesta: Calidad / Variable continua
- Criterio de evaluación: Se usará una función propia que generará dos plots de visualización:
 - Scatterplot comparando la nube de puntos del vector predicción vs vector real (calidad)
 - Histograma de la distribución que lleva el vector diferencia (predicción-real) para visualizar la frecuencia con que la predicción se diferencia en mayor o menor medida con la realidad.

Además, se generarán los indicadores siguientes (que son medidas de dispersión continuas):

- RMSE – Error cuadrático medio – Es deseable que tienda a 0
- MAE – Error medio absoluto – Es deseable que tienda a 0
- Corr – Correlación entre variables – Es deseable que tienda a 1

Estos indicadores servirán para evaluar los modelos generados entre sí en el objetivo de búsqueda de modelos predictivos.

2. CLASIFICACIÓN - ANÁLISIS DE VINO TINTO

En este segundo apartado se estudiará el vino tinto categorizando la variable calidad para la creación de modelos **con objetivo de clasificar**. Las condiciones en este punto son las siguientes:

- Variable respuesta: Calidad / Variable factor
- Criterio de evaluación: Se usará una Matriz de Confusión, y se evaluarán los modelos en base a los siguientes indicadores:
 - Precisión – Es deseable que tienda al 100%
 - Kappa – Es deseable que tienda al 100%

3. MODELO DE RESPUESTA BINARIA

Se modificará el dataset para dividir la variable respuesta bajo la etiqueta de **excelencia**, que comprenderá aquellos valores de la calidad que se encuentren igual o mayores a 7, y los que se encuentren por debajo, usando el modelo matemático que mejor haya satisfecho en ambos casos anteriores a la predicción / clasificación de la calidad del vino. Se aplicará a ambos tipos de vino.

RMarkDown

```
# Instalación de paquetes y declaración de librerías

packages = c("tidyverse", "RCurl", "psych", "stats",
           "randomForest", "glmnet", "caret", "kernlab",
           "rpart", "rpart.plot", "neuralnet", "C50",
           "doParallel", "AUC", "ggfortify", "rmdformats",
           "ggplot2", "naniar", "e1071",
           "lattice", "caret", "knitr", "corrplot",
           "kknn", "randomForest", "kernlab", "car", "xlsx",
           "data.table", "GGally", "gplots", "kableExtra")

if (length(setdiff(packages, rownames(installed.packages()))) > 0) {
  install.packages(setdiff(packages, rownames(installed.packages())))
}
```

```

invisible(lapply(packages, require, character.only = TRUE))

# Creacion de una funcion que elimina outliers en función de los valores
# que se encuentran dentro de los percentiles altos

remove_outliers <- function(x,quant) {

    require("dplyr")

    for(i in 1:11) {

        x = mutate(x,outliers=ifelse(x[[i]] < quantile(x[[i]],quant),0,1))
        x = filter(x, outliers==0)
        x = select(x, -outliers)

    }

    return(x)
}

eval = function(pred, true, plot = F, title = "") {
    rmse = sqrt(mean((pred - true)^2))
    mae = mean(abs(pred - true))
    cor = cor(pred, true)
    if (plot == TRUE) {
        par(mfrow = c(1,2), oma = c(0, 0, 2, 0))
        diff = pred - true
        plot(jitter(true, factor = 1),
             jitter(pred, factor = 0.5),
             pch = 3, asp = 1,
             xlab = "Real", ylab = "Prediccion")
        abline(0,1, lty = 2)
        hist(diff, breaks = 20, main = NULL)
        mtext(paste0(title, "Prediccion vs Real"), outer = TRUE)
        par(mfrow = c(1,1))}

    return(list(RMSE = rmse,

```

```

        MAE = mae,
        CORR = cor))
}

# ##### Recarga de los datos:

setwd("D:/MAIN/MASTER/M11/WINE")

wine = data.frame(read.xlsx("SOURCE/SOURCE.xlsx", sheetIndex = 1))

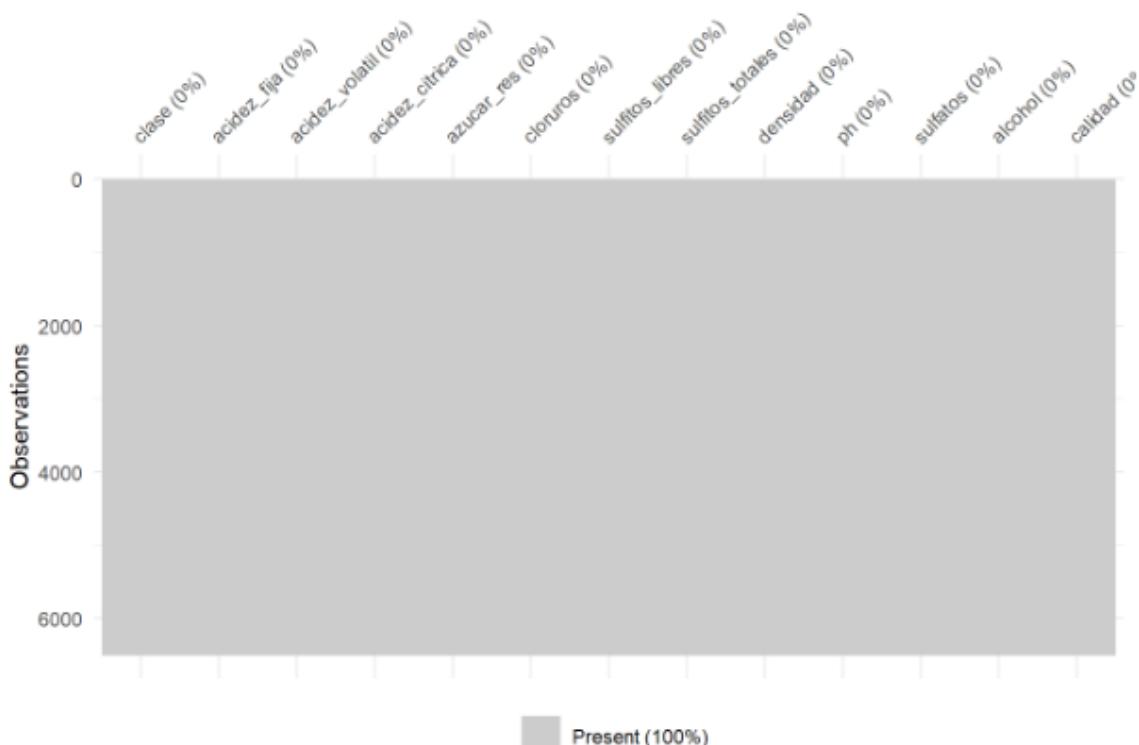
NewNames = c("clase", "acidez_fija", "acidez_volatile", "acidez_citrica", "azucar_res", "cloruros", "sulfitos_libres", "sulfitos_totales", "densidad", "ph", "sulfatos", "alcohol", "calidad")

names(wine)=NewNames
    
```

Limpieza y preparación de datos

Chequeamos los NA del dataset

```
vis_miss(wine)
```



```
###Ninguno
```

```
head(wine) %>% kable() %>% kable_styling()
```

clase	acidez_fija	acidez_volatile	acidez_citrica	azucar_res	cloruros	sulfitos_liberados	sulfitos_totales	densidad	ph	sulfatos	alcohol	calidad
WHITE	7.0	0.27	0.36	20.7	0.045	45	170	1.0010	3.00	0.45	8.8	6
WHITE	6.3	0.30	0.34	1.6	0.049	14	132	0.9940	3.30	0.49	9.5	6
WHITE	8.1	0.28	0.40	6.9	0.050	30	97	0.9951	3.26	0.44	10.1	6
WHITE	7.2	0.23	0.32	8.5	0.058	47	186	0.9956	3.19	0.40	9.9	6
WHITE	7.2	0.23	0.32	8.5	0.058	47	186	0.9956	3.19	0.40	9.9	6
WHITE	8.1	0.28	0.40	6.9	0.050	30	97	0.9951	3.26	0.44	10.1	6

```
summary(wine)
```

```
##      clase        acidez_fija        acidez_volatile        acidez_citrica
##  RED   :1599   Min.   : 3.800   Min.   :0.0800   Min.   :0.0000
##  WHITE:4898   1st Qu.: 6.400   1st Qu.:0.2300   1st Qu.:0.2500
##              Median : 7.000   Median :0.2900   Median :0.3100
##              Mean   : 7.215   Mean   :0.3397   Mean   :0.3186
##              3rd Qu.: 7.700   3rd Qu.:0.4000   3rd Qu.:0.3900
##              Max.   :15.900   Max.   :1.5800   Max.   :1.6600
##      azucar_res       cloruros       sulfitos_liberados       sulfitos_totales
##  Min.   : 0.600   Min.   :0.00900   Min.   :  1.00   Min.   :  6.0
##  1st Qu.: 1.800   1st Qu.:0.03800   1st Qu.: 17.00   1st Qu.: 77.0
##  Median : 3.000   Median :0.04700   Median : 29.00   Median :118.0
##  Mean   : 5.443   Mean   :0.05603   Mean   : 30.53   Mean   :115.7
##  3rd Qu.: 8.100   3rd Qu.:0.06500   3rd Qu.: 41.00   3rd Qu.:156.0
##  Max.   :65.800   Max.   :0.61100   Max.   :289.00   Max.   :440.0
##      densidad          ph          sulfatos          alcohol
##  Min.   :0.9871   Min.   :2.720   Min.   :0.2200   Min.   :  8.00
##  1st Qu.:0.9923   1st Qu.:3.110   1st Qu.:0.4300   1st Qu.: 9.50
##  Median :0.9949   Median :3.210   Median :0.5100   Median :10.30
##  Mean   :0.9947   Mean   :3.219   Mean   :0.5313   Mean   :10.49
##  3rd Qu.:0.9970   3rd Qu.:3.320   3rd Qu.:0.6000   3rd Qu.:11.30
##  Max.   :1.0390   Max.   :4.010   Max.   :2.0000   Max.   :14.90
##      calidad
##  Min.   :3.000
##  1st Qu.:5.000
```

```
## Median :6.000
## Mean   :5.818
## 3rd Qu.:6.000
## Max.   :9.000
```

Se observan valores relativamente esperables, según los valores comprendidos como “usuales” según el análisis exploratorio que se ha realizado. Valores importantes como la densidad, alcohol, pH, sulfitos totales, cloruros o acidez se encuentran dentro del rango de valores esperado.

También puede observarse que hay máximos que muestran valores fuera de los percentiles altos y que se candidatan como outliers, como es el caso de los sulfitos libres, totales, y el azúcar residual.

En general, el set de datos está bastante limpio y apenas requiere de limpieza, aparte de los outliers, que no se eliminarán aun hasta hacer el análisis exploratorio, a excepción de los más fuertes. Se procederá a la división por tipo de vino.

```
str(wine)
```

```
## 'data.frame': 6497 obs. of 13 variables:
## $ clase           : Factor w/ 2 levels "RED","WHITE": 2 2 2 2 2 2 2 2 2 2 ...
## $ acidez_fija     : num  7 6.3 8.1 7.2 7.2 8.1 6.2 7 6.3 8.1 ...
## $ acidez_volatil  : num  0.27 0.3 0.28 0.23 0.23 0.28 0.32 0.27 0.3 0.22 ...
## $ acidez_citrica  : num  0.36 0.34 0.4 0.32 0.32 0.4 0.16 0.36 0.34 0.43 ...
## $ azucar_res       : num  20.7 1.6 6.9 8.5 8.5 6.9 7 20.7 1.6 1.5 ...
## $ cloruros        : num  0.045 0.049 0.05 0.058 0.058 0.05 0.045 0.045 0.049 0.044 ...
## $ sulfitos_libres : num  45 14 30 47 47 30 30 45 14 28 ...
## $ sulfitos_totales: num  170 132 97 186 186 97 136 170 132 129 ...
## $ densidad         : num  1.001 0.994 0.995 0.996 0.996 ...
## $ ph               : num  3 3.3 3.26 3.19 3.19 3.26 3.18 3 3.3 3.22 ...
## $ sulfatos         : num  0.45 0.49 0.44 0.4 0.4 0.44 0.47 0.45 0.49 0.45 ...
## $ alcohol          : num  8.8 9.5 10.1 9.9 9.9 10.1 9.6 8.8 9.5 11 ...
## $ calidad          : num  6 6 6 6 6 6 6 6 6 6 ...
```

#Se confirman que sólo el atributo clase, agregado al dataset original como factor

Se crean pues, los datasets de las dos clases de vino por separado, eliminando el atributo clase y eliminando outliers fuertes:

```
# unloadNamespace("seriation")
```

```
white = wine %>% filter(clase=="WHITE") %>% select(-clase)
```

```
white = remove_outliers(white, 0.999)
```

##Eliminamos los outliers más fuertes (por encima del percentil 99.9)

```
red = wine %>% filter(clase=="RED") %>% select(-clase)
```

```
red = remove_outliers(red, 0.9999)
```

```
summary(white)
```

```
## acidez_fija      acidez_volatil      acidez_citrica      azucar_res
## Min.   : 3.800   Min.   :0.0800   Min.   :0.0000   Min.   : 0.600
## 1st Qu.: 6.300   1st Qu.:0.2100   1st Qu.:0.2700   1st Qu.: 1.700
## Median : 6.800   Median :0.2600   Median :0.3200   Median : 5.200
## Mean    : 6.846   Mean    :0.2768   Mean    :0.3329   Mean    : 6.346
## 3rd Qu.: 7.300   3rd Qu.:0.3200   3rd Qu.:0.3800   3rd Qu.: 9.838
## Max.   :10.200   Max.   :0.9050   Max.   :0.9900   Max.   :22.600
## cloruros        sulfitos_libres    sulfitos_totales    densidad
## Min.   :0.00900   Min.   : 2.00   Min.   : 9.0    Min.   :0.9871
## 1st Qu.:0.03600   1st Qu.: 23.00  1st Qu.:108.0   1st Qu.:0.9917
## Median :0.04300   Median : 34.00  Median :134.0   Median :0.9937
## Mean   :0.04551   Mean   : 35.17  Mean   :137.8   Mean   :0.9940
## 3rd Qu.:0.05000   3rd Qu.: 46.00  3rd Qu.:167.0   3rd Qu.:0.9961
## Max.   :0.24400   Max.   :124.00  Max.   :260.0   Max.   :1.0012
## ph                sulfatos       alcohol          calidad
## Min.   :2.720     Min.   :0.2200   Min.   : 8.00   Min.   :3.000
```

```
## 1st Qu.:3.090   1st Qu.:0.4100   1st Qu.: 9.50   1st Qu.:5.000
## Median :3.180   Median :0.4700   Median :10.40   Median :6.000
## Mean    :3.188   Mean     :0.4888   Mean     :10.51   Mean     :5.883
## 3rd Qu.:3.280   3rd Qu.:0.5500   3rd Qu.:11.40   3rd Qu.:6.000
## Max.    :3.770   Max.     :0.9800   Max.     :13.90   Max.     :9.000
```

4.1. PREDICCIÓN – ANÁLISIS DEL VINO BLANCO

0. Preparación de datos para modelaje

Recarga del dataset y normalización

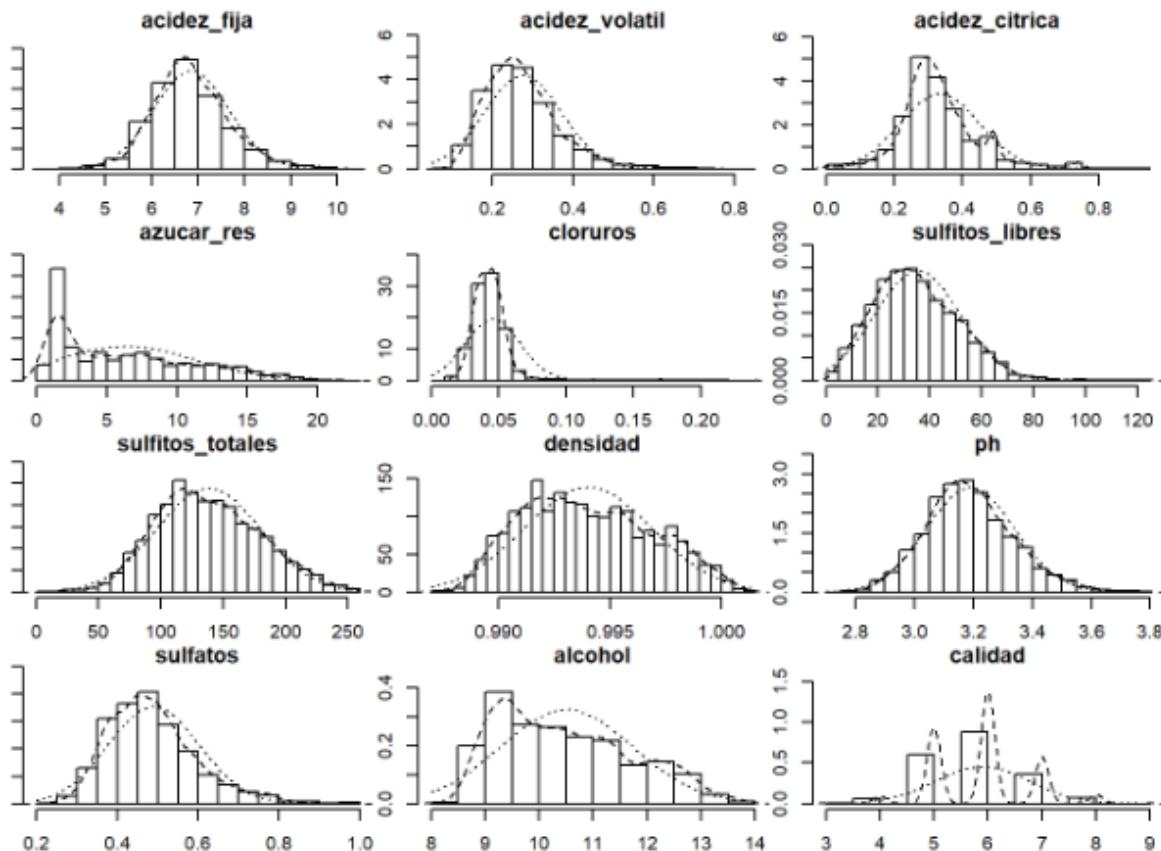
```
set.seed(1)

normalize = function(x) { (x-min(x)) / (max(x)-min(x)) } # Función para normalizar datasets
```

Creación y normalización de train y test para predicción (Vino Blanco)

```
Index <- sample(1:nrow(white), 0.8*nrow(white))
whitetrain <- white[Index, ]
whitetest  <- white[-Index, ]

multi.hist(whitetrain)
```



```

whitetrainN = data.frame(apply(whitetrain[,-12], 2, normalize),
                         calidad = whitetrain[,12])

whitetrain.min = apply(whitetrain[,-12], 2, min)
whitetrain.max = apply(whitetrain[,-12], 2, max)
whitetestN = data.frame(sweep(whitetest, 2, c(whitetrain.min, 0)) %>%
                        sweep(2, c(whitetrain.max-whitetrain.min, 1), FU
N = "/"))
summary(whitetrainN)

```

## acidez_fija	acidez_volatile	acidez_citrica	azucar_res
## Min. :0.0000	Min. :0.0000	Min. :0.0000	Min. :0.0000
## 1st Qu.:0.3906	1st Qu.:0.1769	1st Qu.:0.2967	1st Qu.:0.0500
## Median :0.4688	Median :0.2449	Median :0.3516	Median :0.2045
## Mean :0.4760	Mean :0.2672	Mean :0.3661	Mean :0.2613

```

## 3rd Qu.:0.5469   3rd Qu.:0.3265   3rd Qu.:0.4176   3rd Qu.:0.422
7
## Max. :1.0000   Max. :1.0000   Max. :1.0000   Max. :1.0000
## cloruros      sulfitos_libres sulfitos_totales densidad
## Min. :0.0000   Min. :0.0000   Min. :0.0000   Min. :0.0000
## 1st Qu.:0.1169  1st Qu.:0.1653  1st Qu.:0.3944  1st Qu.:0.3223
## Median :0.1472  Median :0.2562  Median :0.4980  Median :0.4703
## Mean   :0.1579  Mean   :0.2666  Mean   :0.5147  Mean   :0.4863
## 3rd Qu.:0.1775  3rd Qu.:0.3554  3rd Qu.:0.6295  3rd Qu.:0.6375
## Max. :1.0000   Max. :1.0000   Max. :1.0000   Max. :1.0000
## ph            sulfatos      alcohol      calidad
## Min. :0.0000   Min. :0.0000   Min. :0.0000   Min. :3.000
## 1st Qu.:0.3524 1st Qu.:0.2500  1st Qu.:0.2542  1st Qu.:5.000
## Median :0.4381  Median :0.3289  Median :0.3898  Median :6.000
## Mean   :0.4471  Mean   :0.3547  Mean   :0.4247  Mean   :5.887
## 3rd Qu.:0.5333  3rd Qu.:0.4342  3rd Qu.:0.5763  3rd Qu.:6.000
## Max. :1.0000   Max. :1.0000   Max. :1.0000   Max. :9.000

```

```
summary(whitetestN)
```

```

## acidez_fija      acidez_volatile    acidez_citrica      azucar_res
## Min. :0.01562   Min. :0.0000   Min. :0.0000   Min. :0.004546
## 1st Qu.:0.39062  1st Qu.:0.1769  1st Qu.:0.2967  1st Qu.:0.050000
## Median :0.46875  Median :0.2449  Median :0.3407  Median :0.214773
## Mean   :0.47558  Mean   :0.2701  Mean   :0.3647  Mean   :0.260626
## 3rd Qu.:0.54688  3rd Qu.:0.3265  3rd Qu.:0.4286  3rd Qu.:0.418182
## Max. :0.96875   Max. :1.1224   Max. :1.0879  Max. :0.972727
## cloruros      sulfitos_libres sulfitos_totales
## Min. :0.02597   Min. :-0.008264  Min. :0.03984
## 1st Qu.:0.11688  1st Qu.: 0.165289  1st Qu.:0.39044
## Median :0.14719  Median : 0.256198  Median :0.49402
## Mean   :0.15872  Mean   : 0.262913  Mean   :0.50726
## 3rd Qu.:0.17749  3rd Qu.: 0.347107  3rd Qu.:0.62948
## Max. :1.01732   Max. : 0.867769  Max. :0.96813
## densidad          ph            sulfatos      alcohol
## Min. :-0.00788   Min. :0.01905   Min. :0.03947  Min. :0.0678
## 1st Qu.: 0.32360  1st Qu.:0.34286  1st Qu.:0.23684  1st Qu.:0.2542

```

```

## Median : 0.46418   Median :0.42857   Median :0.32895   Median :0
.4068

## Mean    : 0.48052   Mean    :0.44065   Mean    :0.34921   Mean    :0.4302
## 3rd Qu.: 0.61855   3rd Qu.:0.53333   3rd Qu.:0.42105   3rd Qu.:0.5763
## Max.    : 0.98711   Max.    :0.98095   Max.    :0.97368   Max.    :1.0000
##      calidad
## Min.    :3.000
## 1st Qu.:5.000
## Median :6.000
## Mean    :5.868
## 3rd Qu.:6.000
## Max.    :9.000

```

Test Shapiro, para testear la normalidad de nuestra distribución

```
shapiro.test(white$calidad)
```

```

##
##  Shapiro-Wilk normality test
##
## data: white$calidad
## W = 0.8877, p-value < 2.2e-16

```

Se rechaza la hipótesis nula, ya que p es muy inferior a 0,05. Esto quiere decir claramente que uno (o más de un) de los predictores/clasificadores está altamente relacionado con la respuesta.

1. Regresión lineal

1.1. Regresión lineal simple

```

linealtrain <- lm(calidad ~ ., whitetrainN)
summary(linealtrain)

```

```

## 
## Call:
## lm(formula = calidad ~ ., data = whitetrainN)
## 
## Residuals:
##       Min     1Q   Median     3Q    Max 
## -3.6688 -0.5019 -0.0459  0.4616  3.0587 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 5.713399  0.134211 42.570 < 2e-16 ***
## acidez_fija 1.002245  0.171499  5.844 5.52e-09 ***
## acidez_volatil -1.301519  0.098075 -13.271 < 2e-16 *** 
## acidez_citrica 0.048559  0.100905  0.481  0.6304  
## azucar_res    2.401659  0.219292  10.952 < 2e-16 *** 
## cloruros      0.022851  0.151465  0.151  0.8801  
## sulfitos_libres 0.571237  0.117898  4.845 1.31e-06 *** 
## sulfitos_totales -0.004766  0.109094 -0.044  0.9652  
## densidad      -3.388315  0.376399 -9.002 < 2e-16 *** 
## ph            1.110247  0.134512  8.254 < 2e-16 *** 
## sulfatos      0.550028  0.087727  6.270 4.01e-10 *** 
## alcohol        0.473982  0.198037  2.393  0.0167 *  
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Residual standard error: 0.7462 on 3858 degrees of freedom
## Multiple R-squared:  0.2838, Adjusted R-squared:  0.2818 
## F-statistic: 139 on 11 and 3858 DF, p-value: < 2.2e-16

```

- La R² del modelo es de 0.2874, de forma que este modelo no se ajusta especialmente a la realidad, ni puede contemplar gran parte de la dispersión real de los atributos.

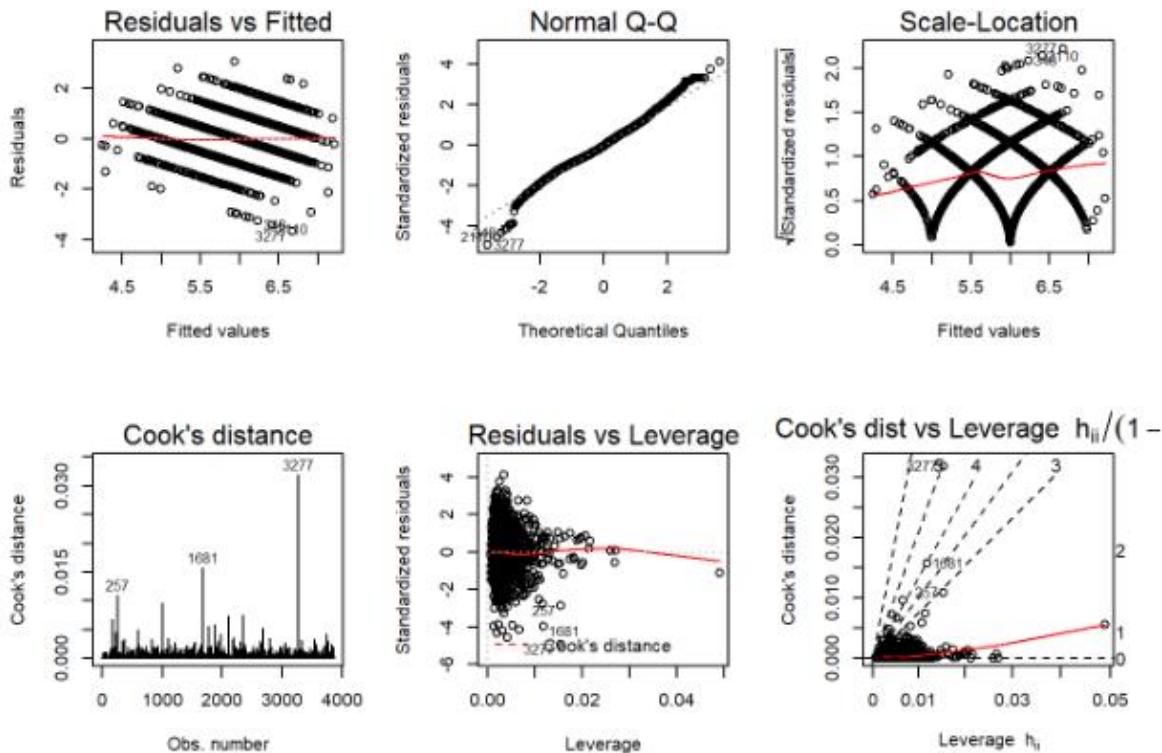
A continuación se hace un análisis de varianza a fin de encontrar outliers que puedan eliminarse para mejorarlo:

```

par(mfrow=c(2,3))
lapply(1:6, function(x) plot(linealtrain, which=x,

```

```
%>% invisible()
```



- Los gráficos Residuals vs Fitted muestran si los residuos tienen patrones no lineales. Los residuos alrededor de una línea horizontal sin patrones distintos, es una buena indicación de que no tenemos relaciones no lineales.
- La gráfica QQ normal muestra los residuos que se ajustan a la línea. Por lo tanto, puede llamarlo residuos distribuidos normalmente.
- El gráfico de Scale-location muestra si los residuos se distribuyen por igual a lo largo de los rangos de los predictores. Así es como puede verificar el supuesto de varianza igual (homocedasticidad). Es bueno si ve una línea horizontal con puntos de dispersión iguales (al azar).
- La trama Residuals vs Leverage tiene un aspecto típico cuando hay algún caso influyente. Apenas puede ver las líneas de distancia de Cook (una línea discontinua roja) porque todos los casos están dentro de la distancia de Cook.

- El gráfico de Cook's distance resalta las observaciones más atípicas del modelo, se contemplan 3 observaciones que se encuentran visiblemente dispersas del resto del conjunto, así que los eliminamos

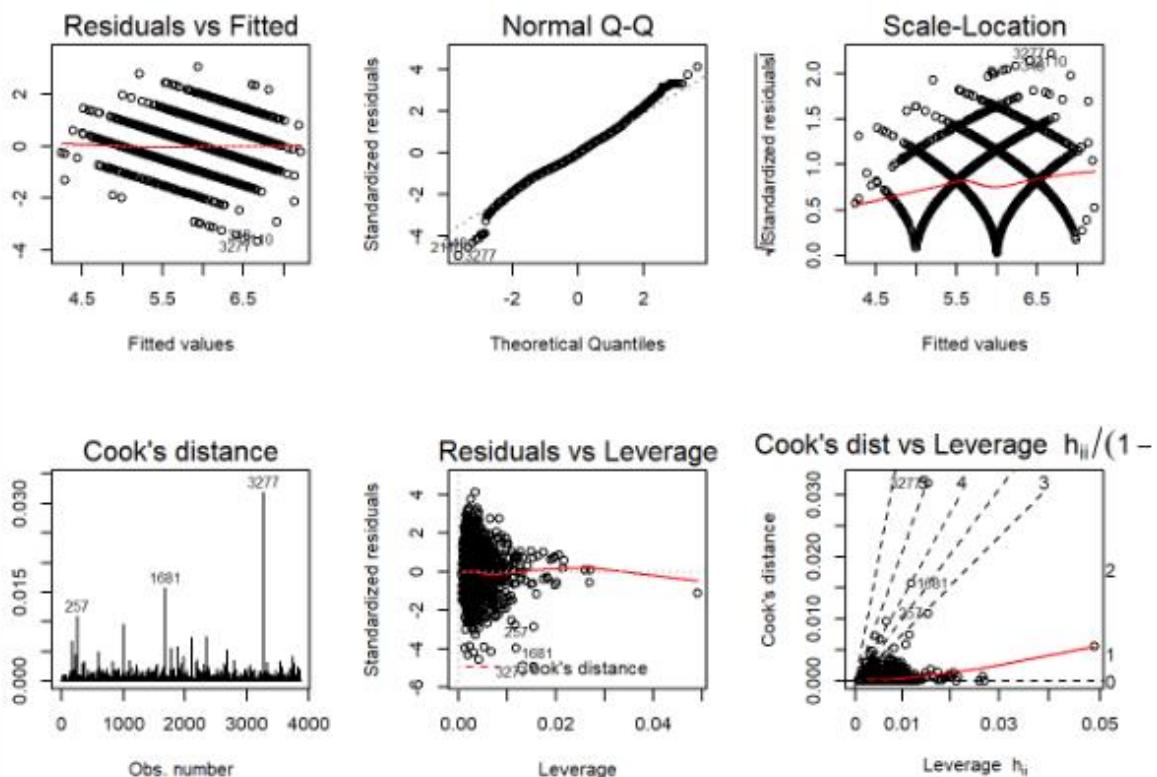
```
whitetrainN=whitetrainN[-c(257,1681,3277),]

linealwhite <- lm(calidad ~ ., whitetrainN)
summary(linealwhite)
```

```
##
## Call:
## lm(formula = calidad ~ ., data = whitetrainN)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.3862 -0.5029 -0.0468  0.4607  3.0498
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 5.68146   0.13353  42.548 < 2e-16 ***
## acidez_fija 1.05166   0.17082   6.156 8.20e-10 ***
## acidez_volatil -1.29856   0.09752 -13.316 < 2e-16 ***
## acidez_citrica 0.03221   0.10035   0.321   0.7482
## azucar_res 2.42000   0.21811  11.095 < 2e-16 ***
## cloruros 0.05254   0.15106   0.348   0.7280
## sulfitos_libres 0.67136   0.11816   5.682 1.43e-08 ***
## sulfitos_totales -0.03047   0.10851  -0.281   0.7789
## densidad -3.41902   0.37435  -9.133 < 2e-16 ***
## ph 1.12822   0.13375   8.435 < 2e-16 ***
## sulfatos 0.54782   0.08722   6.281 3.74e-10 ***
## alcohol 0.47553   0.19705   2.413   0.0159 *
## ---
## Signif. codes: 0 '****' 0.001 '***' 0.01 '**' 0.05 '*' 0.1 '.' 1
##
## Residual standard error: 0.7417 on 3855 degrees of freedom
## Multiple R-squared: 0.2888, Adjusted R-squared: 0.2867
```

```
## F-statistic: 142.3 on 11 and 3855 DF, p-value: < 2.2e-16
```

```
par(mfrow=c(2, 3))
lapply(1:6, function(x) plot(linealtrain, which=x,
                                labels.id= 1:nrow(whitetrainN))) %>% invisible()
```

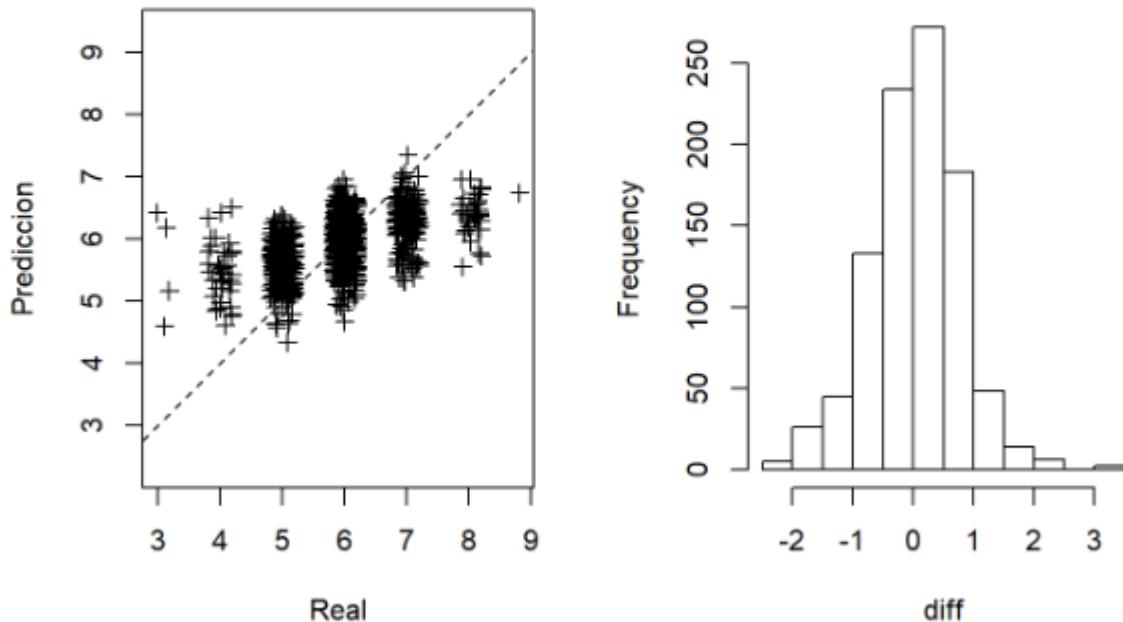


```
# La R2 no ha variado, de manera que se deja el modelo como está
```

```
linealwhitePredictor = predict(linealwhite, whitetestN[,-12])
linealwhiteEvaluator = eval(linealwhitePredictor,
                           whitetestN[,12], plot = T, title = "lm: ")
```



Im: Predicción vs Real



```
unlist(linealwhiteEvaluator)
```

```
##      RMSE      MAE      CORR
## 0.7422834 0.5750728 0.5411154
```

- El RMSE o error cuadrático medio es de 0,7, y tratándose de un dataset con más de 1000 unidades, no se trata de un valor especialmente bajo.
- El MAE es de 0,57, que en la escala de 3-9 que nos encontramos, es algo mayor al 10% del orden de magnitud, lo cual no es tampoco un error fatal, pero es inseguro.
- La correlación es de 0,52, que es aceptable si se trata de evaluar la dependencia de ambas, pero no suficiente para considerarlo óptimo el modelo

- El histograma sirve para mostrar que este modelo es bueno en las regiones centrales, donde es lógico, entre otras cosas, porque se dispone de mayor cantidad de datos, pero también por la baja correlación existente entre mucho de sus predictores. El modelo, por tanto sería relativamente aceptable para predecir calidad de vinos medios.

Para solucionar el problema de este modelo, se puede hacer lo siguiente:

- Estudiar la colinealidad y asimetría del dataset, y mejorar el modelo eliminando predictores de alta colinealidad
- Probar modelos polinomiales, regresión cuadrática
- Probar a categorizar variables

1.2 - Regresión lineal mejorada:

Se estudia la multicolinealidad, y se mejora el modelo

En multicolinealidad (colinealidad entre tres o más variables) implica que hay redundancia entre las variables predictoras, y el modelo se vuelve inestable.

La multicolinealidad se va a evaluar calculando un puntaje llamado factor de inflación de la varianza (o VIF), que mide cuánto se infla la varianza de un coeficiente de regresión debido a la multicolinealidad en el modelo.

El VIF de un predictor es una medida de la inflación de la varianza que se predice a partir de una regresión lineal utilizando los otros predictores.

```
vif(linealtrain)
```

##	acidez_fija	acidez_volatile	acidez_citrica	azucar_res
##	3.318183	1.126612	1.161478	16.735265
##	cloruros	sulfitos_libres	sulfitos_totales	densidad
##	1.250269	1.781134	2.283445	41.789577

##	ph	sulfatos	alcohol
##	2.573579	1.175578	11.812092

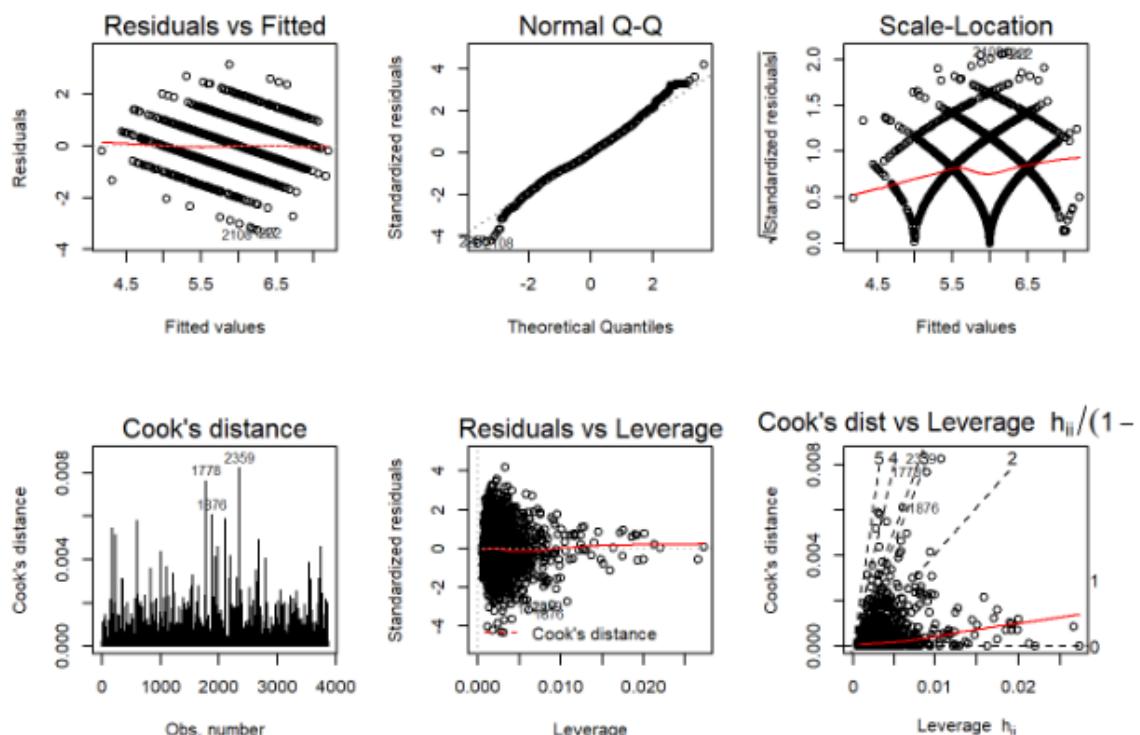
```
# La densidad es el predictor con mayor colinealidad, ya que conocemos la
# relacion de dependencia que tiene con el alcohol, asi que va a retirarse
linealwhite1 <- lm(calidad ~ . - acidez_citrica -densidad, whitetrainN)
summary(linealwhite1)
```

```
##
## Call:
## lm(formula = calidad ~ . - acidez_citrica - densidad, data = whitetrainN)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.2525 -0.5028 -0.0394  0.4617  3.1261
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4.99460   0.11009  45.368 < 2e-16 ***
## acidez_fija -0.15401   0.10717 -1.437 0.150774
## acidez_volatil -1.33249   0.09713 -13.718 < 2e-16 ***
## azucar_res     0.51793   0.06564   7.891 3.88e-15 ***
## cloruros      -0.19912   0.14930 -1.334 0.182376
## sulfitos_libres 0.83846   0.11787   7.114 1.34e-12 ***
## sulfitos_totales -0.28306   0.10591 -2.673 0.007559 **
## ph             0.28401   0.09743   2.915 0.003579 **
## sulfatos       0.29780   0.08348   3.567 0.000365 ***
## alcohol        2.14095   0.07505  28.527 < 2e-16 ***
##
## ---
## Signif. codes:  0 '****' 0.001 '***' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7495 on 3857 degrees of freedom
## Multiple R-squared:  0.2734, Adjusted R-squared:  0.2717
## F-statistic: 161.2 on 9 and 3857 DF,  p-value: < 2.2e-16
```

```
vif(linealwhite1) # Ahora ha mejorado mucho el análisis de variación, se procede a visualizarlo
```

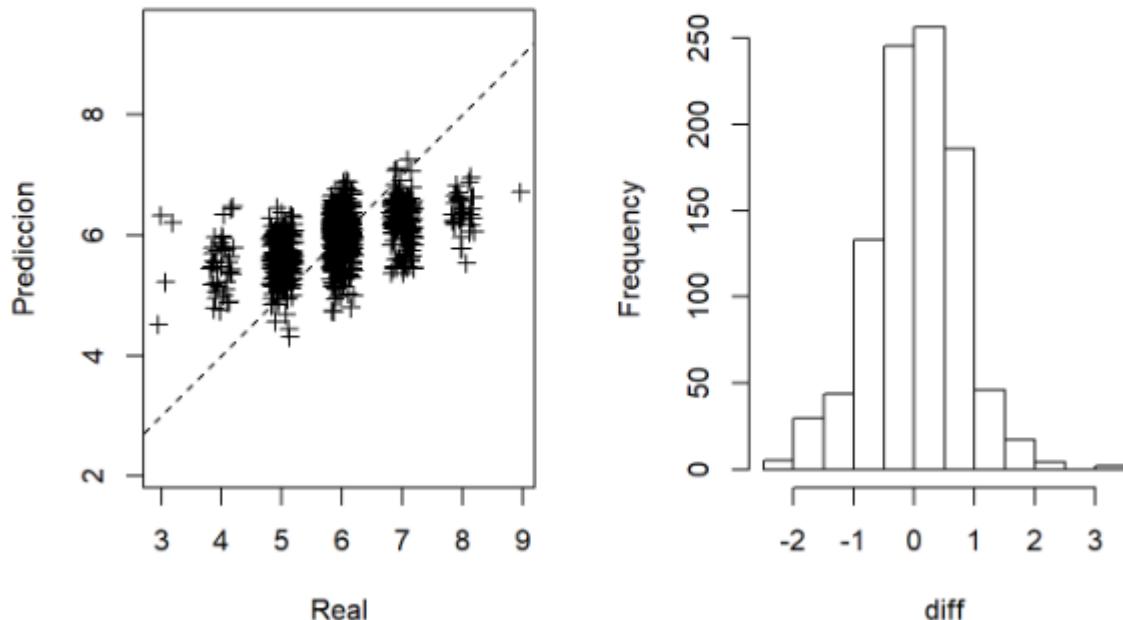
```
##      acidez_fija     acidez_volatile     azucar_res      cloruros
##      1.280498       1.095152       1.481352       1.199820
##      sulfitos_libres sulfitos_totales      ph      sulfatos
##      1.738808       2.129526       1.334991       1.053913
##      alcohol
##      1.679461
```

```
par(mfrow=c(2,3))
lapply(1:6, function(x) plot(linealwhite1, which=x,
                                labels.id= 1:nrow(whitetrainN))) %>% invisible()
```



```
linealwhitePredictor1 = predict(linealwhite1, whitetestN[,-12])
linealwhiteEvaluator1 = eval(linealwhitePredictor1,
                             whitetestN[,12], plot = T, title = "lm: ")
```

Im: Predicción vs Real



```
unlist(linealwhiteEvaluator)
```

```
##          RMSE         MAE        CORR
## 0.7422834 0.5750728 0.5411154
```

```
unlist(linealwhiteEvaluator1)
```

```
##          RMSE         MAE        CORR
## 0.7470187 0.5800930 0.5329971
```

Se ha mejorado la inflación de la varianza sin obtener cambios en los errores del modelo, sin embargo, los indicadores escogidos para evaluar su criterio han empeorado, de manera que estos cambios no han servido.

2. Regresión Múltiple (cuadrática-binomial)

```

quadraticWhite = lm(calidad ~ poly(acidez_fija, 2) +
  poly(acidez_volatile, 2) +
  poly(acidez_citrica, 2) +
  poly(cloruros, 2) +
  poly(sulfitos_libres, 2) +
  poly(sulfitos_totales, 2) +
  poly(azucar_res, 2) +
  poly(densidad, 2) +
  poly(ph, 2) +
  poly(sulfatos, 2) +
  poly(alcohol, 2),
  data = whitetrainN)

summary(quadraticWhite)

```

```

## 
## Call:
## lm(formula = calidad ~ poly(acidez_fija, 2) + poly(acidez_volatile,
##     2) + poly(acidez_citrica, 2) + poly(cloruros, 2) + poly(sulfitos_libres,
##     2) + poly(sulfitos_totales, 2) + poly(azucar_res, 2) + poly(densidad,
##     2) + poly(ph, 2) + poly(sulfatos, 2) + poly(alcohol, 2),
##     data = whitetrainN)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -3.2530 -0.4888 -0.0262  0.4383  3.1179 
## 
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)               5.88829   0.01163 506.085 < 2e-16 ***
## poly(acidez_fija, 2)1    7.99333   1.34137  5.959 2.76e-09 ***
## poly(acidez_fija, 2)2   -2.38563   0.76902 -3.102 0.001935 ** 
## poly(acidez_volatile, 2)1 -9.27854   0.83499 -11.112 < 2e-16 ***
## poly(acidez_volatile, 2)2  3.02703   0.75207  4.025 5.81e-05 *** 
## poly(acidez_citrica, 2)1  0.42113   0.79003  0.533 0.594029
## 
```

```

## poly(acidez_citrica, 2)2      -4.42387   0.79266  -5.581 2.56e-08
***
```

```

## poly(cloruros, 2)1           -0.79357   0.82987  -0.956 0.339004
## poly(cloruros, 2)2           2.50372   0.81245  3.082 0.002073 **
## poly(sulfitos_libres, 2)1    5.46205   0.99426  5.494 4.19e-08 ***
## poly(sulfitos_libres, 2)2    -5.39726   0.78118  -6.909 5.68e-12 ***
## poly(sulfitos_totales, 2)1   -1.31594   1.13945  -1.155 0.248207
## poly(sulfitos_totales, 2)2   -3.94524   0.78046  -5.055 4.50e-07 ***
## poly(azucar_res, 2)1          30.22277  3.05550  9.891 < 2e-16 ***
## poly(azucar_res, 2)2          -3.35840  0.95253  -3.526 0.000427 ***
## poly(densidad, 2)1            -42.24931 4.76313  -8.870 < 2e-16 ***
## poly(densidad, 2)2            1.22171   1.20994  1.010 0.312686
## poly(ph, 2)1                 9.54286   1.18016  8.086 8.18e-16 ***
## poly(ph, 2)2                 1.53154   0.75279  2.034 0.041972 *
## poly(sulfatos, 2)1            5.10574   0.79361  6.434 1.40e-10 ***
## poly(sulfatos, 2)2            -0.25207  0.75188  -0.335 0.737449
## poly(alcohol, 2)1             3.17760   2.54712  1.248 0.212281
## poly(alcohol, 2)2             3.41952   0.94237  3.629 0.000289 ***
## ---
## Signif. codes:  0 '****' 0.001 '***' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7235 on 3844 degrees of freedom
## Multiple R-squared:  0.3251, Adjusted R-squared:  0.3213
## F-statistic: 84.18 on 22 and 3844 DF,  p-value: < 2.2e-16

```

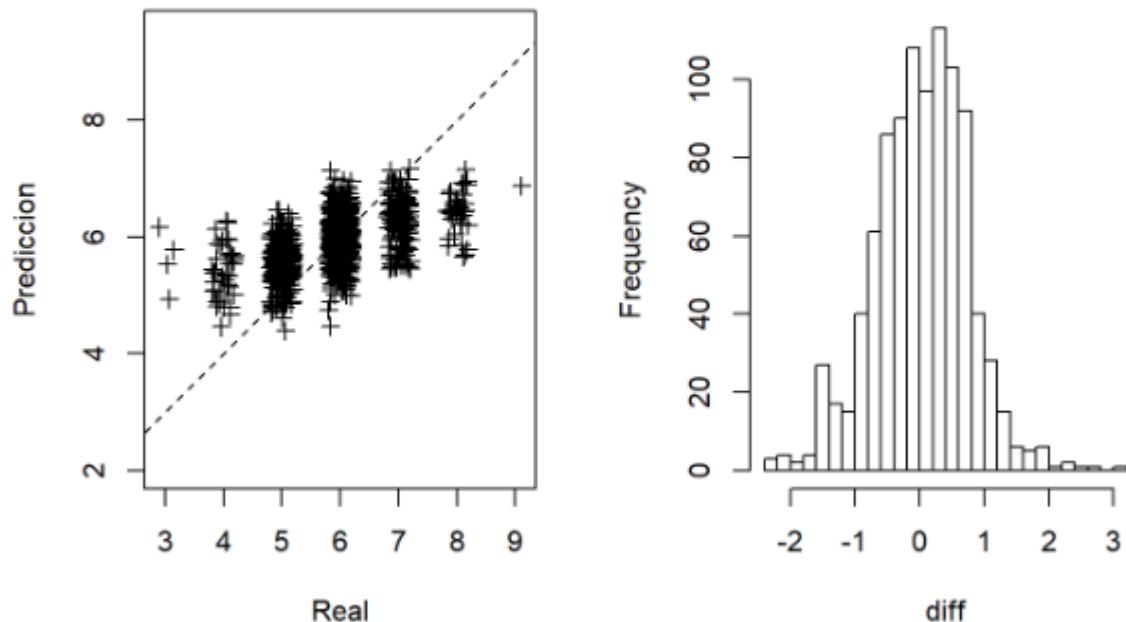
```

quadraticWhitePredictor = predict(quadraticWhite, whitetestN[,-12])
quadraticWhiteEvaluator = eval(quadraticWhitePredictor,
                               whitetestN$calidad, plot = T, title = "qm: "
)

```



qm: Predicción vs Real



```
unlist(quadraticWhiteEvaluator)
```

```
##          RMSE        MAE        CORR
## 0.7268407 0.5681213 0.5673510
```

El modelo mejora, pero no es suficiente

2A. Regresión Poisson

```
set.seed(143)
PoissonWhite = glm(calidad~, data = whitetrainN,
                    family = "poisson")
summary(PoissonWhite)
```

```
## Call:
glm(formula = calidad ~ ., family = "poisson", data = whitetrainN)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-1.49744	-0.21262	-0.01602	0.18780	1.16465

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	1.737963	0.074863	23.215	< 2e-16 ***
acidez_fija	0.183944	0.095810	1.920	0.054873 .
acidez_volatil	-0.231881	0.055457	-4.181	2.9e-05 ***
acidez_citrica	0.004783	0.056712	0.084	0.932787
azucar_res	0.416698	0.122858	3.392	0.000695 ***
cloruros	0.004349	0.086871	0.050	0.960072
sulfitos_libres	0.113744	0.065786	1.729	0.083810 .
sulfitos_totales	-0.003637	0.060972	-0.060	0.952428
densidad	-0.586978	0.210683	-2.786	0.005335 **
ph	0.194764	0.074591	2.611	0.009025 **
sulfatos	0.092579	0.047808	1.936	0.052810 .
alcohol	0.075426	0.110413	0.683	0.494528

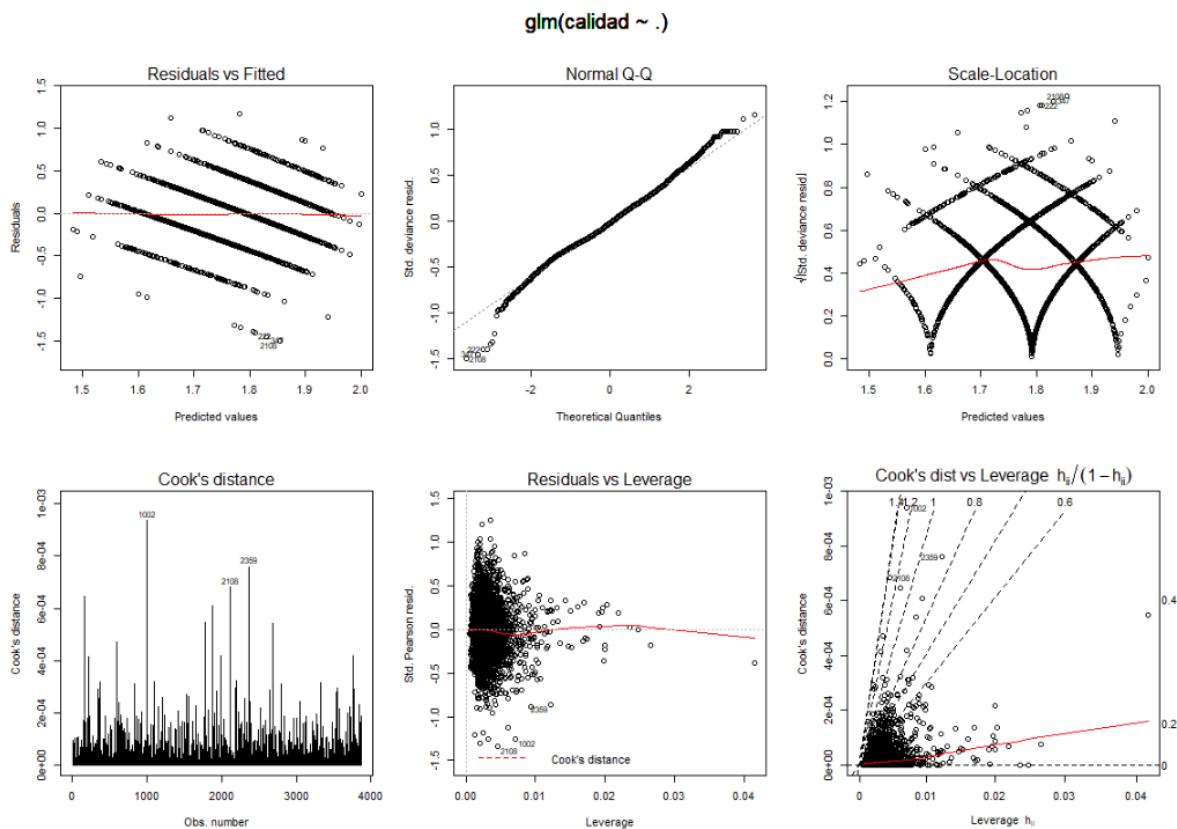
Signif. codes:	0 '***'	0.001 '**'	0.01 '*'	0.05 '.'
	0.1 ' '	1		

(Dispersion parameter for poisson family taken to be 1)

Null deviance: 506.64 on 3866 degrees of freedom
 Residual deviance: 360.24 on 3855 degrees of freedom
 AIC: 14416

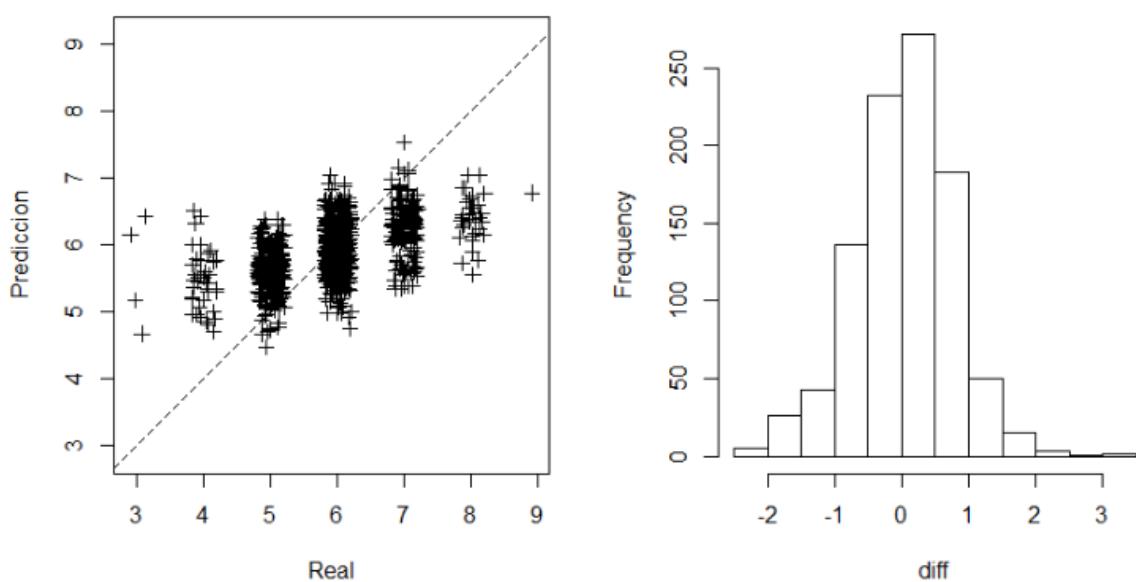
Number of Fisher Scoring iterations: 4

```
par(mfrow=c(2,3))
lapply(1:6, function(x) plot(PoissonWhite, which=x,
                             labels.id= 1:nrow(whitetrainN))) %>% invisible()
```



```
PoissonWhitePredictor = predict(PoissonWhite, whitetestN, type = "response")
PoissonWhiteEvaluator = eval(PoissonWhitePredictor,
                             whitetest[,12], plot = T, title = "glm: ")
```

Poisson: Predicción vs Real



```
unlist(PoissonWhiteEvaluator)
```

```
##          RMSE        MAE       CORR
## 0.7413665 0.5746920 0.5427551
```

3. Modelo ANOVA, categorizando las variables

Se categorizan los predictores con las siguientes reglas:

- 0 - 10%
- 10 - 30%
- 30 - 50%
- 50% - 75%
- 90 - 100%

```
c1 = apply(whitetrainN, 2, function(x) quantile(x, 0.10))
c2 = apply(whitetrainN, 2, function(x) quantile(x, 0.30))
c3 = apply(whitetrainN, 2, function(x) quantile(x, 0.50))
c4 = apply(whitetrainN, 2, function(x) quantile(x, 0.75))
c5 = apply(whitetrainN, 2, function(x) quantile(x, 0.9))

categorize = function(dataset = whitetrainN) {
  df.cat = dataset
  for (i in 1:(ncol(dataset)-1)) {
    col = dataset[,i]
    cat = case_when(col<c1[i] ~ "0",
                    col>=c1[i] & col< c2[i] ~ "1",
                    col>=c2[i] & col< c3[i] ~ "2",
                    col>=c3[i] & col< c4[i] ~ "3",
                    col>=c4[i] & col< c5[i] ~ "4",
                    col>=c5[i] ~ "5")
    df.cat[,i] = cat
  }
  return(df.cat)
}
```

```

}

whitetrainCat = categorize(whitetrainN)

whitetestCat = categorize(whitetestN)

summary(whitetrainCat)

```

```

## acidez_fija      acidez_volatile      acidez_citrica
## Length:3867      Length:3867      Length:3867
## Class :character  Class :character  Class :character
## Mode  :character  Mode  :character  Mode  :character

##
## azucar_res      cloruros      sulfitos_libres
## Length:3867      Length:3867      Length:3867
## Class :character  Class :character  Class :character
## Mode  :character  Mode  :character  Mode  :character

##
## sulfitos_totales      densidad      ph
## Length:3867      Length:3867      Length:3867
## Class :character  Class :character  Class :character
## Mode  :character  Mode  :character  Mode  :character

##
## sulfatos      alcohol      calidad
## Length:3867      Length:3867      Min.   :3.000
## Class :character  Class :character  1st Qu.:5.000
## Mode  :character  Mode  :character  Median :6.000
##                           Mean   :5.888
##                           3rd Qu.:6.000
##                           Max.   :9.000
## 
```

```
head(whitetrainCat)
```

	acidez_fija	acidez_volatile	acidez_citrica	azucar_res	cloruros
## 1017	4	1	4	4	5
## 4775	0	4	2	1	1

```

## 2177      3      3      2      2
1          1      0      5      1      1
## 1533      5      0      5      1      1
## 4567      2      5      1      2      1
## 2347      3      0      2      2      0
##      sulfitos_libres sulfitos_totales densidad ph sulfatos alcohol calidad
## 1017        4          4      4 2      1      3      6
## 4775        2          1      1 5      4      3      6
## 2177        0          2      1 3      3      4      5
## 1533        1          2      2 1      1      3      6
## 4567        1          1      0 3      3      5      6
## 2347        0          1      2 3      2      3      8

```

```

anovaWhite = lm(calidad ~ ., data = whitetrainCat)

summary(anovaWhite)

```

```

## Call:
## lm(formula = calidad ~ ., data = whitetrainCat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.2124 -0.4832 -0.0248  0.4609  3.1083
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)            5.2008398  0.1249307 41.630 < 2e-16 ***
## acidez_fija1         -0.0139502  0.0491111 -0.284  0.776384
## acidez_fija2          0.0559274  0.0493839  1.133  0.257495
## acidez_fija3          0.0738974  0.0504188  1.466  0.142820
## acidez_fija4          0.1545815  0.0560894  2.756  0.005879 **
## acidez_fija5          0.0457549  0.0646783  0.707  0.479347
## acidez_volatile1     -0.1253489  0.0490603 -2.555  0.010657 *
## acidez_volatile2     -0.2404612  0.0493882 -4.869  1.17e-06 ***
## acidez_volatile3     -0.4005763  0.0485191 -8.256 < 2e-16 ***
## acidez_volatile4     -0.4175823  0.0519220 -8.042  1.16e-15 ***
## acidez_volatile5     -0.4674920  0.0584224 -8.002  1.61e-15 ***

```

## acidez_citrica1	0.1935164	0.0481650	4.018	5.99e-05	***
## acidez_citrica2	0.3517596	0.0501933	7.008	2.84e-12	***
## acidez_citrica3	0.2633287	0.0497384	5.294	1.26e-07	***
## acidez_citrica4	0.2451736	0.0518926	4.725	2.39e-06	***
## acidez_citrica5	0.1915950	0.0549885	3.484	0.000499	***
## azucar_res1	0.1684007	0.0537222	3.135	0.001734	**
## azucar_res2	0.4056329	0.0564467	7.186	7.98e-13	***
## azucar_res3	0.5161044	0.0638199	8.087	8.15e-16	***
## azucar_res4	0.7408020	0.0810891	9.136	< 2e-16	***
## azucar_res5	0.9197667	0.0969100	9.491	< 2e-16	***
## cloruros1	0.0157415	0.0474202	0.332	0.739939	
## cloruros2	-0.0966250	0.0489197	-1.975	0.048320	*
## cloruros3	-0.1226116	0.0505178	-2.427	0.015266	*
## cloruros4	-0.2047859	0.0549948	-3.724	0.000199	***
## cloruros5	-0.1506506	0.0586340	-2.569	0.010227	*
## sulfitos_libres1	0.3177607	0.0488677	6.502	8.93e-11	***
## sulfitos_libres2	0.5029439	0.0505294	9.953	< 2e-16	***
## sulfitos_libres3	0.5053564	0.0517182	9.771	< 2e-16	***
## sulfitos_libres4	0.5236874	0.0584529	8.959	< 2e-16	***
## sulfitos_libres5	0.4688212	0.0635669	7.375	2.00e-13	***
## sulfitos_totales1	0.1273972	0.0488285	2.609	0.009114	**
## sulfitos_totales2	0.1990851	0.0508848	3.912	9.30e-05	***
## sulfitos_totales3	0.1058462	0.0537974	1.967	0.049198	*
## sulfitos_totales4	0.0356534	0.0614716	0.580	0.561950	
## sulfitos_totales5	-0.0005214	0.0688695	-0.008	0.993959	
## densidad1	-0.1288990	0.0554423	-2.325	0.020128	*
## densidad2	-0.2760391	0.0694684	-3.974	7.21e-05	***
## densidad3	-0.6106912	0.0864404	-7.065	1.90e-12	***
## densidad4	-0.8531502	0.1114939	-7.652	2.49e-14	***
## densidad5	-0.9133940	0.1337102	-6.831	9.76e-12	***
## ph1	-0.0590882	0.0483707	-1.222	0.221946	
## ph2	0.0030978	0.0500582	0.062	0.950658	
## ph3	-0.0089691	0.0498808	-0.180	0.857310	
## ph4	0.1373054	0.0556478	2.467	0.013653	*
## ph5	0.2514563	0.0617042	4.075	4.69e-05	***
## sulfatos1	0.0613950	0.0496498	1.237	0.216326	
## sulfatos2	0.0420074	0.0503793	0.834	0.404432	
## sulfatos3	0.1084603	0.0487445	2.225	0.026134	*
## sulfatos4	0.1896843	0.0528370	3.590	0.000335	***

```

## sulfatos5      0.1395994  0.0573379  2.435 0.014951 *
## alcohol11     -0.0455785  0.0544180 -0.838 0.402329
## alcohol12     -0.0143301  0.0608651 -0.235 0.813880
## alcohol13      0.1620276  0.0663800  2.441 0.014696 *
## alcohol14      0.3092320  0.0787411  3.927 8.75e-05 ***
## alcohol15      0.5797185  0.0925099  6.267 4.10e-10 ***
## ---
## Signif. codes:  0 '****' 0.001 '***' 0.01 '**' 0.05 '*' 0.1 '.' 1
##
## Residual standard error: 0.7208 on 3811 degrees of freedom
## Multiple R-squared:  0.3359, Adjusted R-squared:  0.3263
## F-statistic: 35.05 on 55 and 3811 DF,  p-value: < 2.2e-16

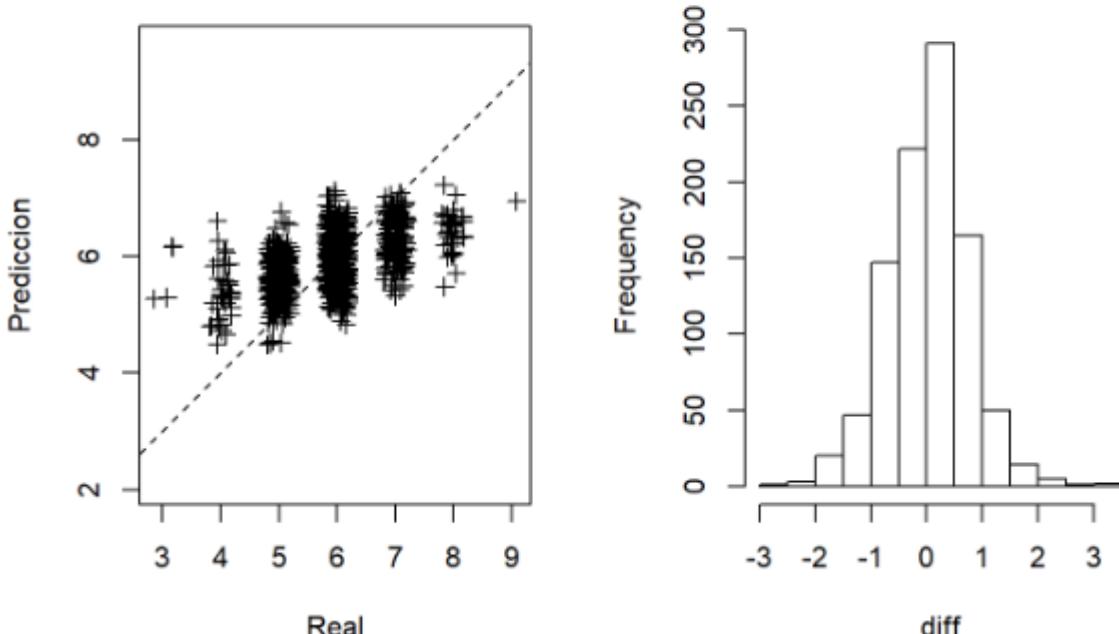
```

```

anovaWhitePredictor = predict(anovaWhite, whitetestCat[,-12])
anovaWhiteEvaluator = eval(anovaWhitePredictor,
                           whitetestCat$calidad, plot = T, title = "cm: ")

```

cm: Predicción vs Real



```
unlist(anovaWhiteEvaluator)
```

```
##      RMSE      MAE      CORR
## 0.7324826 0.5751814 0.5584532
```

Se puede comprobar que los resultados son similares a los conseguidos en el modelo binomial.

4. Interacción y selección de variables (paso a paso)

```
InteractWhite = lm(calidad~ .^2, data = whitetrain)
summary(InteractWhite)
```

```
##
## Call:
## lm(formula = calidad ~ .^2, data = whitetrain)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.3971 -0.4897 -0.0108  0.4353  3.0243
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 -4.563e+02  4.388e+02 -1.040  0.29846
## acidez_fija                  1.397e+01  1.875e+01  0.745  0.45643
## acidez_volatil                -2.223e+02  2.385e+02 -0.932  0.35136
## acidez_citrica                 1.392e+02  2.513e+02  0.554  0.57965
## azucar_res                     7.068e+00  1.653e+00  4.275 1.96e-05
## cloruros                      -1.899e+03  1.463e+03 -1.298  0.19438
## sulfitos_libres                -3.429e+00  1.893e+00 -1.811  0.07018
## sulfitos_totales                -1.697e+00  7.403e-01 -2.293  0.02193
## densidad                       4.631e+02  4.401e+02  1.052  0.29273
## ph                             2.328e+02  1.100e+02  2.116  0.03445
## sulfatos                        1.102e+02  2.033e+02  0.542  0.58799
## alcohol                         1.473e+01  8.571e+00  1.718  0.08582
## acidez_fija:acidez_volatil     -2.862e-01  2.639e-01 -1.085  0.27810
## acidez_fija:acidez_citrica     -1.789e-01  2.371e-01 -0.755  0.45058
## acidez_fija:azucar_res          7.237e-03  7.513e-03  0.963  0.33549
```

## acidez_fija:cloruros .23126	-1.868e+00	1.560e+00	-1.197	0
## acidez_fija:sulfitos_libres	-2.481e-05	1.873e-03	-0.013	0.98943
## acidez_fija:sulfitos_totales	-1.031e-03	8.110e-04	-1.272	0.20352
## acidez_fija:densidad	-1.440e+01	1.870e+01	-0.770	0.44125
## acidez_fija:ph	3.419e-01	1.055e-01	3.242	0.00120
## acidez_fija:sulfatos	2.059e-01	2.158e-01	0.954	0.34000
## acidez_fija:alcohol	-3.421e-02	3.080e-02	-1.111	0.26666
## acidez_volatile:acidez_citrica	1.247e+00	1.014e+00	1.230	0.21874
## acidez_volatile:azucar_res	-8.739e-02	9.068e-02	-0.964	0.33525
## acidez_volatile:cloruros	-2.034e+00	7.308e+00	-0.278	0.78073
## acidez_volatile:sulfitos_libres	1.111e-02	1.035e-02	1.073	0.28327
## acidez_volatile:sulfitos_totales	4.350e-03	4.334e-03	1.004	0.31556
## acidez_volatile:densidad	2.088e+02	2.416e+02	0.864	0.38762
## acidez_volatile:ph	1.594e+00	1.337e+00	1.192	0.23319
## acidez_volatile:sulfatos	-1.192e+00	1.333e+00	-0.894	0.37114
## acidez_volatile:alcohol	9.503e-01	2.964e-01	3.206	0.00136
## acidez_citrica:azucar_res	1.576e-02	9.264e-02	0.170	0.86497
## acidez_citrica:cloruros	2.760e+00	4.738e+00	0.583	0.56019
## acidez_citrica:sulfitos_libres	9.997e-03	8.134e-03	1.229	0.21913
## acidez_citrica:sulfitos_totales	-2.838e-03	3.821e-03	-0.743	0.45773
## acidez_citrica:densidad	-1.434e+02	2.544e+02	-0.564	0.57309
## acidez_citrica:ph	1.193e+00	1.182e+00	1.010	0.31275
## acidez_citrica:sulfatos	-1.006e-01	1.134e+00	-0.089	0.92932
## acidez_citrica:alcohol	3.405e-02	3.161e-01	0.108	0.91423
## azucar_res:cloruros	-1.090e+00	5.610e-01	-1.942	0.05220
## azucar_res:sulfitos_libres	-1.833e-03	7.309e-04	-2.508	0.01217
## azucar_res:sulfitos_totales	-2.928e-04	2.962e-04	-0.988	0.32307
## azucar_res:densidad	-6.870e+00	1.640e+00	-4.188	2.88e-05
## azucar_res:ph	1.354e-02	4.065e-02	0.333	0.73897
## azucar_res:sulfatos	-1.008e-02	7.831e-02	-0.129	0.89754
## azucar_res:alcohol	-6.578e-03	5.618e-03	-1.171	0.24175
## cloruros:sulfitos_libres	6.445e-04	5.112e-02	0.013	0.98994
## cloruros:sulfitos_totales	8.407e-03	2.641e-02	0.318	0.75030
## cloruros:densidad	1.982e+03	1.478e+03	1.341	0.17996
## cloruros:ph	-1.856e+01	7.607e+00	-2.440	0.01474
## cloruros:sulfatos	-1.213e+01	8.902e+00	-1.363	0.17295
## cloruros:alcohol	7.574e-01	1.946e+00	0.389	0.69713
## sulfitos_libres:sulfitos_totales	-1.782e-04	2.067e-05	-8.620	< 2e-16

## sulfitos_libres:densidad	3.429e+00	1.916e+00	1.789	0
.07366				
## sulfitos_libres:ph	-7.616e-04	9.772e-03	-0.078	0.93788
## sulfitos_libres:sulfatos	2.406e-02	8.744e-03	2.751	0.00597
## sulfitos_libres:alcohol	4.677e-03	2.540e-03	1.842	0.06560
## sulfitos_totales:densidad	1.727e+00	7.510e-01	2.299	0.02155
## sulfitos_totales:ph	-7.362e-03	4.499e-03	-1.636	0.10187
## sulfitos_totales:sulfatos	-1.297e-02	4.225e-03	-3.069	0.00216
## sulfitos_totales:alcohol	2.307e-03	1.001e-03	2.305	0.02122
## densidad:ph	-2.344e+02	1.101e+02	-2.129	0.03330
## densidad:sulfatos	-1.145e+02	2.059e+02	-0.556	0.57821
## densidad:alcohol	-1.463e+01	8.789e+00	-1.665	0.09607
## ph:sulfatos	2.549e+00	1.054e+00	2.419	0.01560
## ph:alcohol	-1.442e-01	1.686e-01	-0.855	0.39264
## sulfatos:alcohol	-3.266e-01	2.571e-01	-1.270	0.20404
##				
## (Intercept)				
## acidez_fija				
## acidez_volatile				
## acidez_citrica				
## azucar_res	***			
## cloruros	.			
## sulfitos_libres	.			
## sulfitos_totales	*			
## densidad				
## ph	*			
## sulfatos	.			
## alcohol	.			
## acidez_fija:acidez_volatile				
## acidez_fija:acidez_citrica				
## acidez_fija:azucar_res				
## acidez_fija:cloruros				
## acidez_fija:sulfitos_libres				
## acidez_fija:sulfitos_totales				
## acidez_fija:densidad				
## acidez_fija:ph	**			
## acidez_fija:sulfatos				
## acidez_fija:alcohol				
## acidez_volatile:acidez_citrica				

```

## acidez_volatile:azucar_res
## acidez_volatile:cloruros
## acidez_volatile:sulfitos_libres
## acidez_volatile:sulfitos_totales
## acidez_volatile:densidad
## acidez_volatile:ph
## acidez_volatile:sulfatos
## acidez_volatile:alcohol      **
## acidez_citrica:azucar_res
## acidez_citrica:cloruros
## acidez_citrica:sulfitos_libres
## acidez_citrica:sulfitos_totales
## acidez_citrica:densidad
## acidez_citrica:ph
## acidez_citrica:sulfatos
## acidez_citrica:alcohol
## azucar_res:cloruros          .
## azucar_res:sulfitos_libres   *
## azucar_res:sulfitos_totales
## azucar_res:densidad         ***
## azucar_res:ph
## azucar_res:sulfatos
## azucar_res:alcohol
## cloruros:sulfitos_libres
## cloruros:sulfitos_totales
## cloruros:densidad
## cloruros:ph                  *
## cloruros:sulfatos
## cloruros:alcohol
## sulfitos_libres:sulfitos_totales ***
## sulfitos_libres:densidad     .
## sulfitos_libres:ph
## sulfitos_libres:sulfatos    **
## sulfitos_libres:alcohol     .
## sulfitos_totales:densidad   *
## sulfitos_totales:ph
## sulfitos_totales:sulfatos   **
## sulfitos_totales:alcohol    *
## densidad:ph                 *

```

```
## densidad:sulfatos
## densidad:alcohol
## ph:sulfatos *
## ph:alcohol
## sulfatos:alcohol
## ---
## Signif. codes: 0 '***' 0.001 ' '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7122 on 3803 degrees of freedom
## Multiple R-squared: 0.3569, Adjusted R-squared: 0.3457
## F-statistic: 31.97 on 66 and 3803 DF, p-value: < 2.2e-16
```

Se realiza el método paso a paso:

```

StepWhite = lm(calidad ~ 1, data = whitetrainN)

InteractWhiteStep = step(StepWhite,
                        ~ (acidez_fija + acidez_volatile +
                           acidez_citrica +
                           azucar_res +
                           cloruros +
                           sulfitos_libres +
                           sulfitos_totales +
                           densidad +
                           ph +
                           sulfatos +
                           alcohol)^2,
                        direction = "both", trace = 0)

summary(InteractWhiteStep)

```

```
##  
## Call:  
  
## lm(formula = calidad ~ alcohol + acidez_volatile + sulfitos_libres +  
##      azucar_res + sulfatos + densidad + ph + acidez_fija + alcohol:acidez_volatile +  
##      sulfitos_libres:azucar_res + densidad:ph + ph:acidez_fija +  
##      sulfitos_libres:acidez_fija + alcohol:sulfitos_libres + sulfatos:ph +
```

```

##      alcohol:densidad + azucar_res:densidad + sulfitos_libres:dens
##      idad +
##      azucar_res:sulfatos + acidez_volatile:ph + acidez_volatile:sulfitos_libres,
##      data = whitetrainN)
##
## Residuals:
##      Min      1Q Median      3Q      Max
## -3.2140 -0.4889 -0.0215  0.4447  3.0215
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)               6.99211   0.33717  20.738 < 2e-16 ***
## alcohol                  -0.98540   0.37533  -2.625  0.008688 **
## acidez_volatile          -3.32333   0.39210  -8.476 < 2e-16 ***
## sulfitos_libres          -4.45612   0.95225  -4.680 2.97e-06 ***
## azucar_res                 4.63867   0.42475  10.921 < 2e-16 ***
## sulfatos                 -0.04546   0.28167  -0.161  0.871783
## densidad                 -2.20801   0.60362  -3.658  0.000258 ***
## ph                        0.26864   0.40750   0.659  0.509780
## acidez_fija              -0.46531   0.33200  -1.402  0.161133
## alcohol:acidez_volatile  2.89530   0.43274   6.691 2.54e-11 ***
## sulfitos_libres:azucar_res -5.04135   1.13746  -4.432 9.59e-06 ***
## densidad:ph                -3.05465   0.44115  -6.924 5.11e-12 ***
## ph:acidez_fija            2.98236   0.60998   4.889 1.05e-06 ***
## sulfitos_libres:acidez_fija 1.55007   0.80537   1.925  0.054345 .
## alcohol:sulfitos_libres     5.81931   1.06216   5.479 4.56e-08 ***
## sulfatos:ph                 1.79314   0.53252   3.367  0.000767 ***
## alcohol:densidad            -2.08501   0.42061  -4.957 7.46e-07 ***
## azucar_res:densidad         -1.50542   0.38733  -3.887  0.000103 ***
## sulfitos_libres:densidad     6.19548   1.87453   3.305  0.000958 ***
## azucar_res:sulfatos        -0.93074   0.38077  -2.444  0.014557 *
## acidez_volatile:ph          1.17082   0.63689   1.838  0.066091 .
## acidez_volatile:sulfitos_libres 1.11199   0.68348   1.627  0.103829
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7189 on 3845 degrees of freedom
## Multiple R-squared:  0.3336, Adjusted R-squared:  0.3299
## F-statistic: 91.65 on 21 and 3845 DF,  p-value: < 2.2e-16

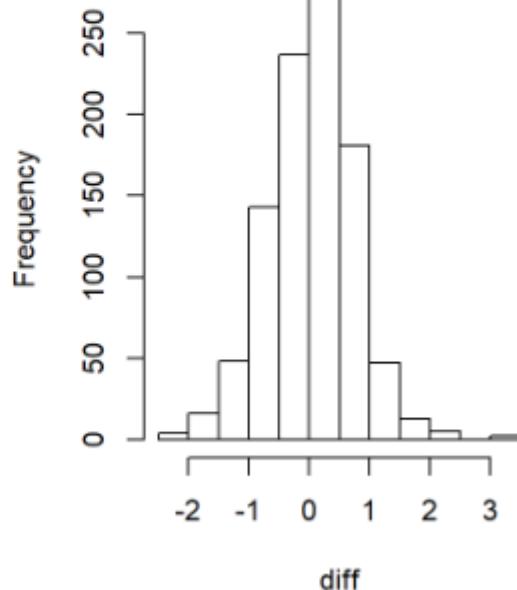
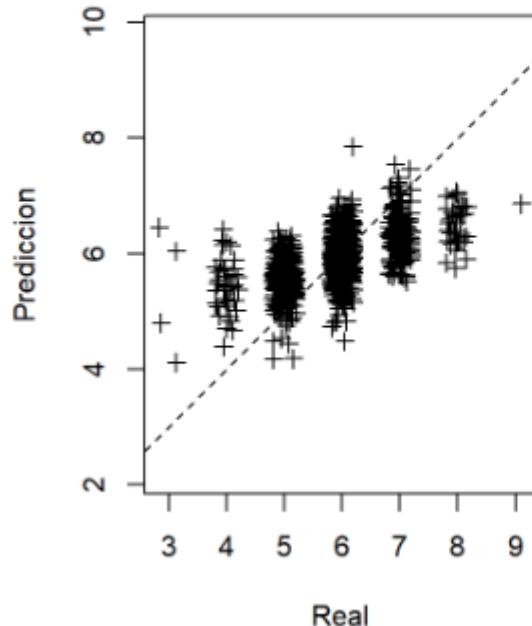
```

```

InteractWhiteStepPredictor = predict(InteractWhiteStep, white
testN[,-12])

InteractWhiteStepEvaluator = eval(InteractWhiteStepPredictor, whitetestN$calidad, plot=T, title="sm: ")
    
```

sm: Predicción vs Real



```
unlist(InteractWhiteStepEvaluator)
```

```

##          RMSE        MAE       CORR
## 0.7202138 0.5581140 0.5789019
    
```

Con este modelo se ha mejorado notablemente la correlación, el error también ha disminuido.

5. Random Forest

5.1. Random Forest con librería RandomForest

Se va a construir un modelo de Árboles de decisión, agrupando 1000 árboles de decisión, cada árbol con 3 variables.

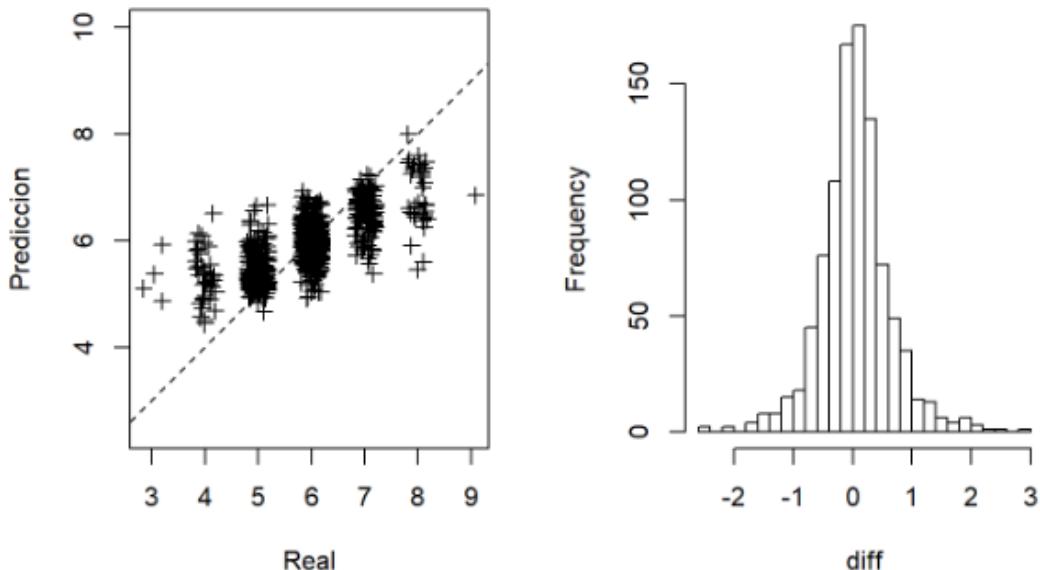
```
RandForestWhite = randomForest(calidad~., data = whitetrainN, ntree = 100
0, mtry = sqrt(12))

RandForestWhite
## Call:
##   randomForest(formula = calidad ~ ., data = whitetrainN, ntree = 1000,
##   mtry = sqrt(12))

##           Type of random forest: regression
##   Number of trees: 1000
##   No. of variables tried at each split: 3
##
##   Mean of squared residuals: 0.3575264
##   % Var explained: 53.63

RandForestWhitePredictor = predict(RandForestWhite, whitetestN[,-12])
RandForestWhiteEvaluator = eval(RandForestWhitePredictor, whitetestN$calidad, plot = T, title = "rfm: ")
```

rfm: Predicción vs Real



```
RandForestWhiteEvaluator
```

```
## $RMSE
## [1] 0.6024253
##
## $MAE
## [1] 0.4301487
##
## $CORR
## [1] 0.7404821
```

Con este cambio de modelo, y tal como se puede observar, los errores han descendido en torno al 8-10% y la correlación ha subido en torno a .15 puntos.

5.2. Random Forest con librería Caret

Se usará la función train del package “caret” para aplicar una validación cruzada con las siguientes características:

- De ahora en adelante, los modelos creados con la librería caret se realizarán con el dataset sin normalizar, ya que se usará la función preproceso donde se centra y escala previamente el dataset
- 10 veces con 3 repeticiones
- Se incluyen 2, 4-6 variables respectivamente por nivel de árbol.

A fin de que se optimicen los parámetros del modelo. Se evaluará posteriormente con el RMSE. No se aplicarán muchas combinaciones por cuestiones de memoria. Se usará el paquete doParallel para mejorar el rendimiento.

Se usará la función trainControl para ajustar los hiperparámetros. Tras diferentes entrenamientos, se escogen estos valores porque se encuentran dentro de los márgenes de éxito, sin que provoque un consumo computacional grande

```
controlRandForest = trainControl(method = "repeatedcv", number = 10, repeats = 3)

matrizRandForest = expand.grid(.mtry = c(2, 3, 6))
```

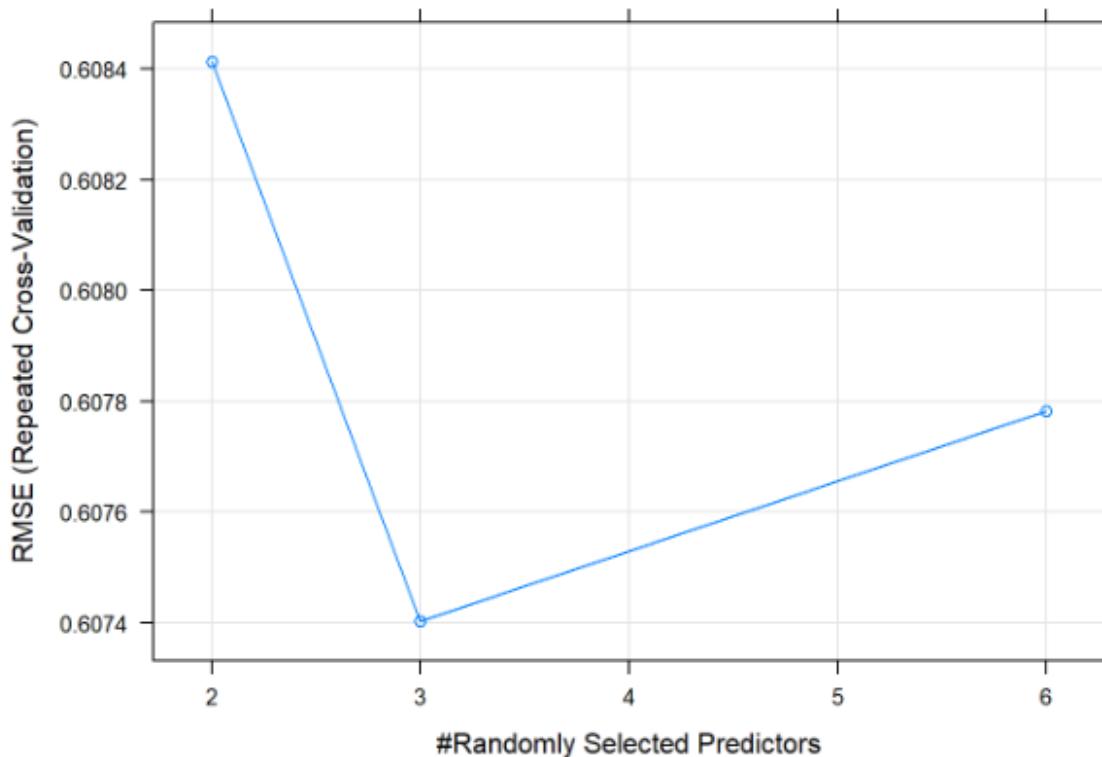
```

set.seed(1)

clusterRandForest = makePSOCKcluster(4)
registerDoParallel(clusterRandForest)

RandForestCrossValWhite = train(calidad~, data = whitetrain,
                                 method = 'rf',
                                 metric = "RMSE",
                                 trControl = controlRandForest,
                                 tuneGrid = matrizRandForest,
                                 preProcess = c("center", "scale"))

stopCluster(clusterRandForest)
plot(RandForestCrossValWhite)
    
```



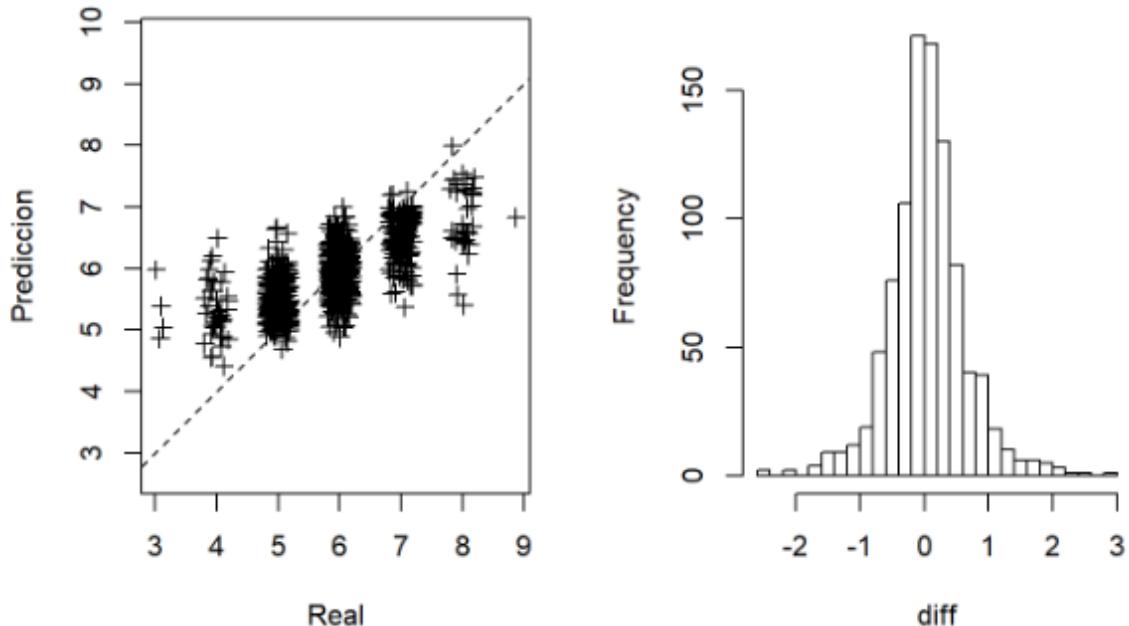
```
RandForestCrossValWhite$bestTune
```

```
##      mtry
```

```
## 2      3
```

```
RandForestCrossValWhitePredictor = predict(RandForestCrossValWhite, white
test[, -12])
RandForestCrossValWhiteEvaluator = eval(RandForestCrossValWhitePredictor,
whitetest[, 12], plot = T, title = "rfcvm: ")
```

rfcvm: Predicción vs Real



```
unlist(RandForestCrossValWhiteEvaluator)
```

```
##          RMSE        MAE       CORR
## 0.6040035 0.4321573 0.7387660
```

6. K-Nearest Neighbours

6.1. K-Nearest Neighbours con librería kknn

```
set.seed(12)

KKNNWhite <- train.kknn(calidad~.,
                         whitetrainN,
                         ks = c(3, 5, 7, 9, 11, 17),
                         distance = c(1, 2),
                         kernel =c("rectangular", "gaussian", "cos"))
```

```
## Warning in if (distance == 2) Euclid <- TRUE: la condición tiene longitud >
## 1 y sólo el primer elemento será usado
```

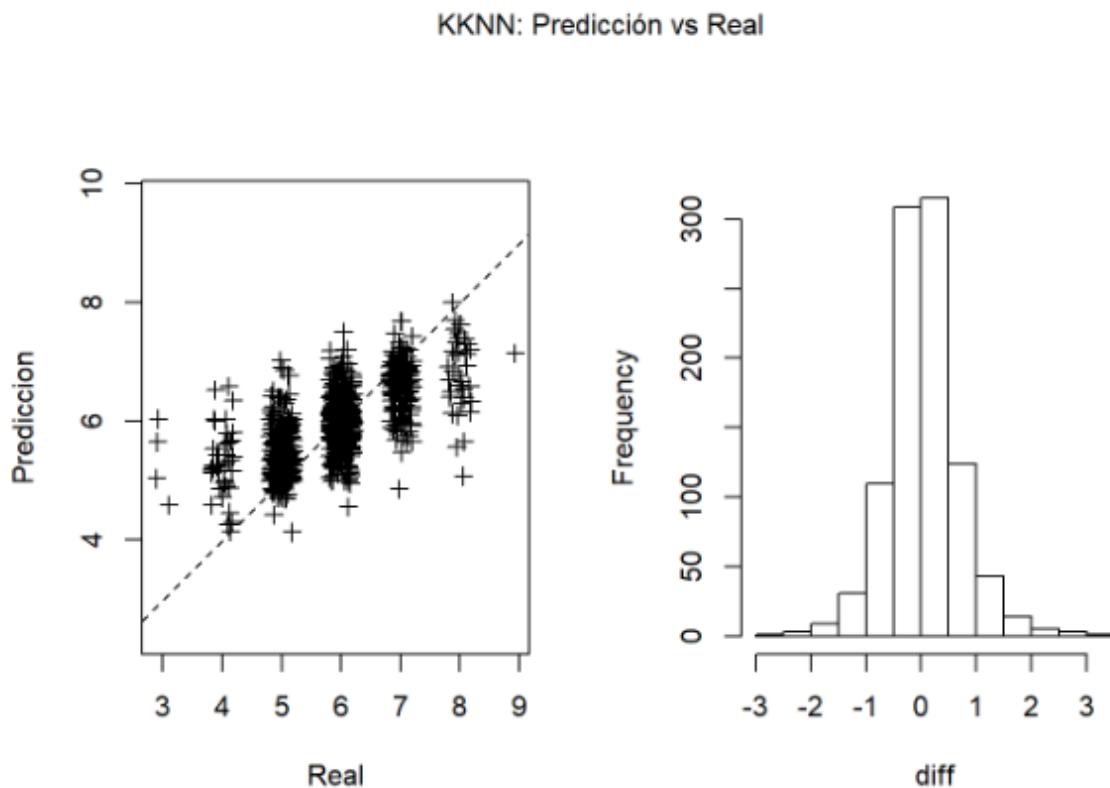
```
summary(KKNNWhite)
```

```
##
## Call:
## train.kknn(formula = calidad ~ ., data = whitetrainN, ks = c(3,      5, 7, 9, 1
1, 17), distance = c(1, 2), kernel = c("rectangular",      "gaussian", "cos"))
##
## Type of response variable: continuous
## minimal mean absolute error: 0.4530551
## Minimal mean squared error: 0.4256228
## Best kernel: cos
## Best k: 9
```

```
KKNNWhitePredictor = predict(KKNNWhite, whitetestN[,-12])
```

```
## Warning in if (distance <= 0) stop("distance must >0"): la condición t
iene
## longitud > 1 y sólo el primer elemento será usado
## Warning in if (distance == 2) Euclid <- TRUE: la condición tiene longi
tud >
## 1 y sólo el primer elemento será usado
```

```
KKNNWhiteEvaluator = eval(KKNNWhitePredictor, whitetestN[,12]
, plot = T, title = "KKNN: ")
```



```
unlist(KKNNWhiteEvaluator)
```

##	RMSE	MAE	CORR
##	0.6585388	0.4740561	0.6697706

6.2. K-Nearest Neighbors con librería Caret

Este modelo también es sugerido por el estudio considerado como fuente del proyecto. Este es un modelo de clasificación supervisada no paramétrico, de modo que es un modelo perfecto a entrenar para un set de datos de variables continuas como el nuestro.

Para KKNN, se utilizarán 5 kmáx, 2 distancias y 3 valores del núcleo.

Para el valor de la distancia, 1 es la distancia de Manhattan y 2 es la distancia euclíadiana (más favorable).

```
set.seed(1)

clusterKKNN = makePSOCKcluster(4)

registerDoParallel(clusterKKNN)

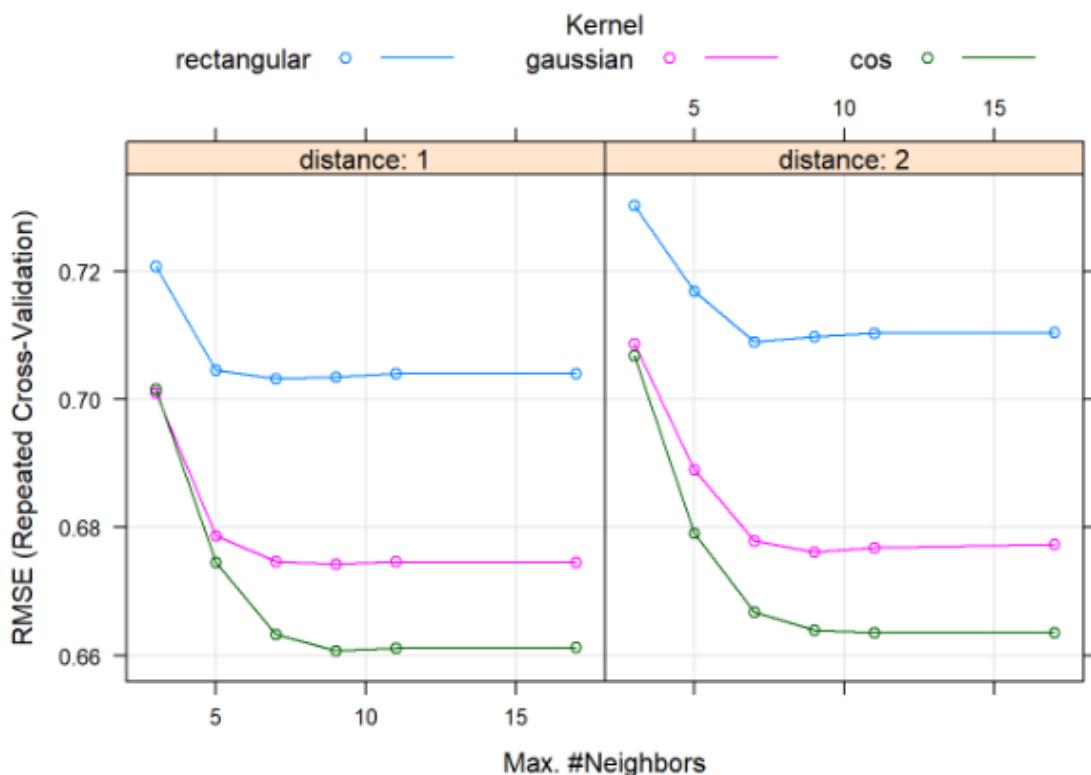
controlKKNN = trainControl(method = "repeatedcv", repeats = 5, classProbs = TRUE)

matrizKKNN = expand.grid(kmax = c(3, 5, 7, 9, 11, 17), distance = c(1, 2),
,
kernel = c("rectangular", "gaussian", "cos"))

KKNNCrossValWhite <- train(calidad ~ ., data = whitetrain,
method = "kknn",
trControl = controlKKNN,
tuneGrid = matrizKKNN,
metric = "RMSE",
preProcess = c("center", "scale"))
```

```
## Warning in train.default(x, y, weights = w, ...): cannot compute class
## probabilities for regression
```

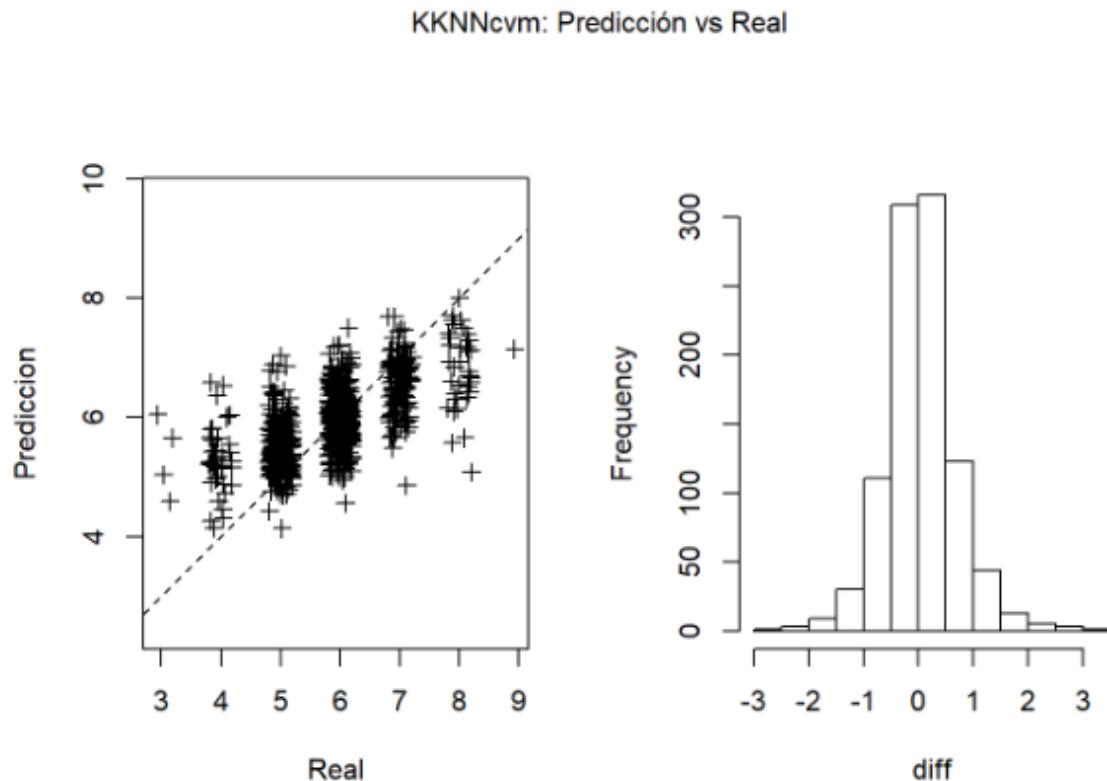
```
stopCluster(clusterKKNN)
plot(KKNNCrossValWhite)
```



```
KKNNCrossValWhite$bestTune
```

```
##      kmax distance kernel
## 21      9        1     cos
```

```
KKNNCrossValWhitePredictor = predict(KKNNCrossValWhite, whitetest[,-12])
KKNNCrossValWhiteEvaluator = eval(KKNNCrossValWhitePredictor, whitetest[,12], plot = T, title = "KKNNcvm: ")
```



```
unlist(KKNNCrossValWhiteEvaluator)
```

```
##          RMSE        MAE       CORR
## 0.6585970 0.4740692 0.6697000
```

7. Support Vector Machine

Este modelo es uno de los modelos principales considerados en el estudio (Modeling wine preferences by data mining), citado en la introducción y fuente principal del presente dataset:

Las SVM presentan ventajas teóricas sobre NN (vecinos cercanos), como la ausencia de mínimos locales en la fase de aprendizaje. En efecto, el SVM fue considerado recientemente uno de los algoritmos de Data mining esenciales. Si bien

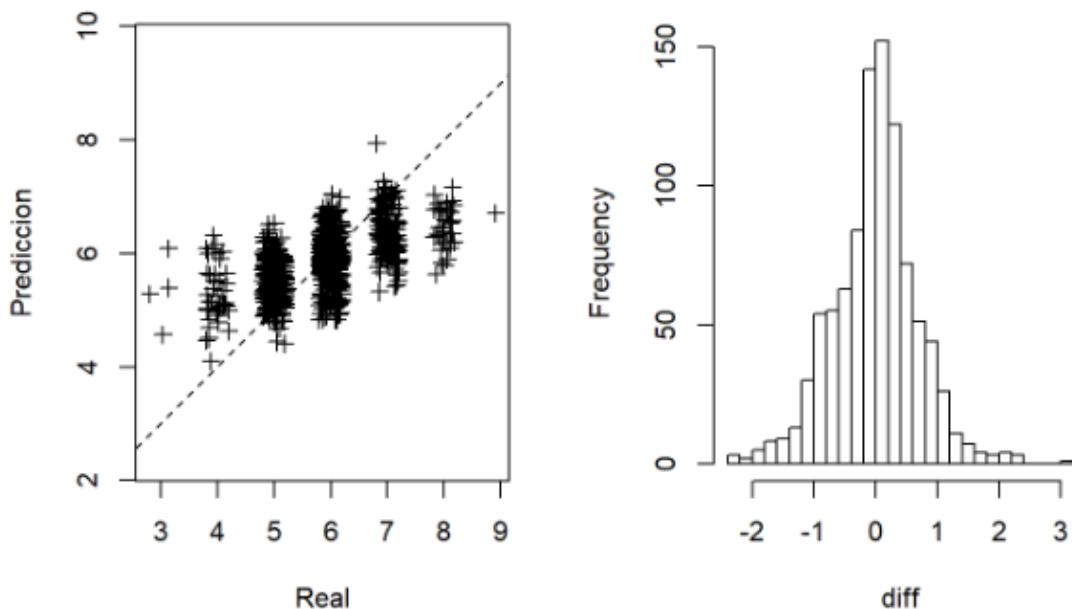
el modelo MR es más fácil de interpretar, aún es posible extraer más conocimiento de NN y SVM, dado en términos de importancia variable de entrada.

Se usará el model ksvm de la librería kernel, con la función kernel centrada en base radial gaussiana, que es la que mejor se ajusta al modelo que se quiere.

```
SVMWhite=ksvm(calidad ~ .,
               data = whitetrainN,
               scaled = F,
               kernel = "rbfdot",
               C = 1)

SVMWhitePredictor = predict(SVMWhite, whitetestN[,-12])
SVMWhiteEvaluator = eval(SVMWhitePredictor, whitetestN[,12], plot = T, title = "SVM: ")
```

SVM: Predicción vs Real



```
unlist(SVMWhiteEvaluator)
##      RMSE      MAE      CORR
```

```
## 0.6899487 0.5147250 0.6232339
```

Este modelo, y contra pronóstico, en relación a las suposiciones extraídas del estudio escogido como referencia principal de este proyecto, no parece ser más adecuado que el de árbol aleatorio. Se usará la metodología de CrossValidation y optimización de hiperparámetros para mejorar el obtenido hasta ahora.

Se utilizará la función de base Radial como núcleo del modelo SVM

Se hará exactamente lo mismo pero usando el método `svmRadial`" de la librería caret

```
set.seed(1)

clusterSVM = makePSOCKcluster(4)

registerDoParallel(clusterSVM)

controlSVM = trainControl(method = "repeatedcv", number = 5, repeats = 5)

# Para la matriz del modelo voy a utilizar los siguientes hiperparámetros :
# *Sigma = El ancho del inverso del núcleo, para suavizar. También conocida como distribución acumulativa inversa normal.
# *C = Coste de desclasificación

matrizSVM = expand.grid(C = 2^(1:3), sigma = seq(0.25, 2, length = 8))

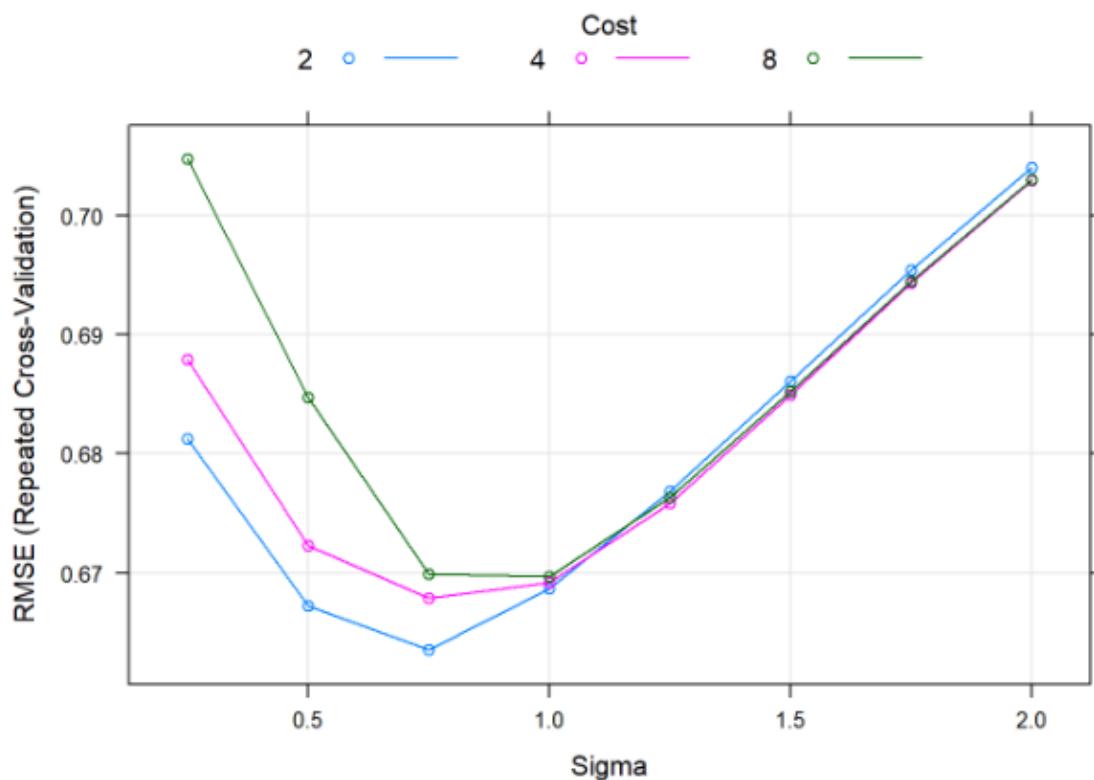
# Ambos parámetros han sido ajustados ya que en los valores de sigma existía un pico de precisión entre los valores escogidos y el valor de C, que más allá del intervalo escogido no parecían revelar más datos significativos en el entrenamiento.

SVMCrossValWhite = train(calidad~, data = whitetrainN,
                         method = 'svmRadial',
                         trControl = controlSVM,
```



```
tuneGrid = matrizSVM)

stopCluster(clusterSVM)
plot(SVMCrossValWhite)
```



```
SVMCrossValWhite$bestTune
```

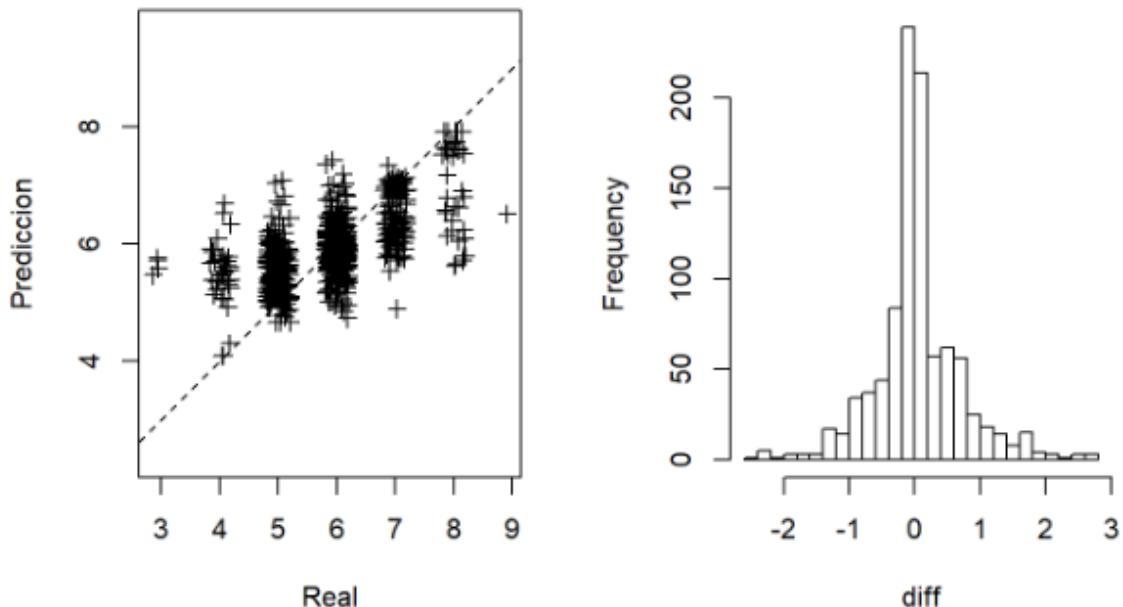
```
## sigma C
## 3 0.75 2
```

Se comprueba que los mejores valores de entrenamiento corresponden a sigma = 0,75 y C = 2. En el se encuentran los valores de RMSE más bajo, tal como se observa en el mínimo relativo.

```
SVMCrossValWhitePredictor = predict(SVMCrossValWhite, whitetestN[,-12])
```

```
SVMCrossValWhiteEvaluator = eval(SVMCrossValWhitePredictor, w  
hitetestN[,12], plot = T, title = "SVMcvm: ")
```

SVMcvm: Predicción vs Real



```
unlist(SVMCrossValWhiteEvaluator)
```

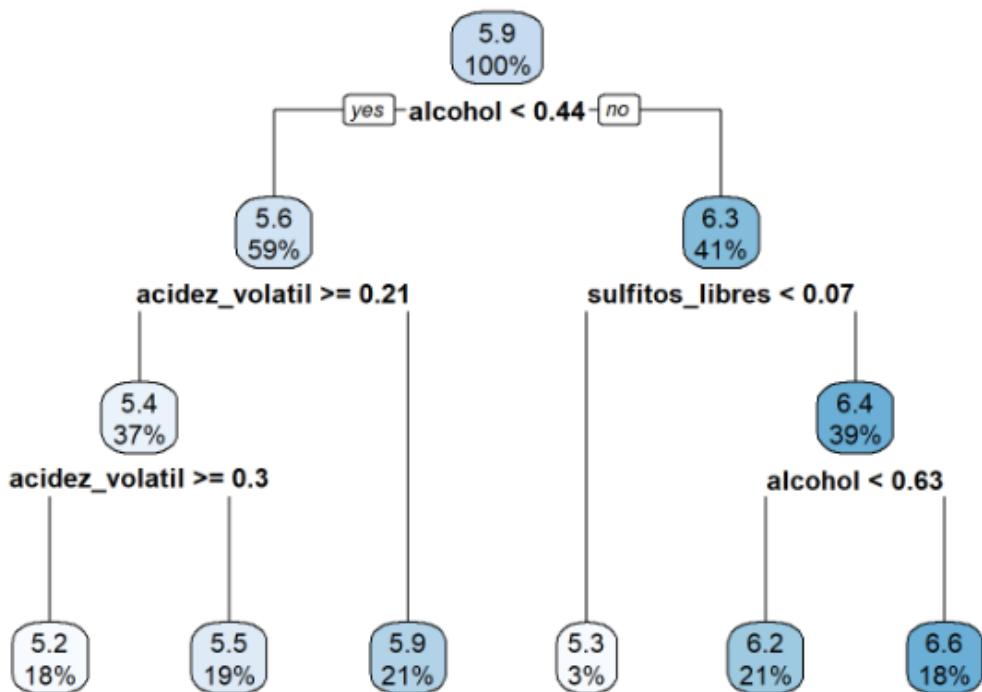
```
##          RMSE        MAE       CORR
## 0.6590361 0.4390023 0.6665344
```

8. Regression Tree

Se estudia, como último modelo a usar, el árbol de regresión con la librería rpart.

```
RTWhite = rpart(calidad~, data = whitetrainN)

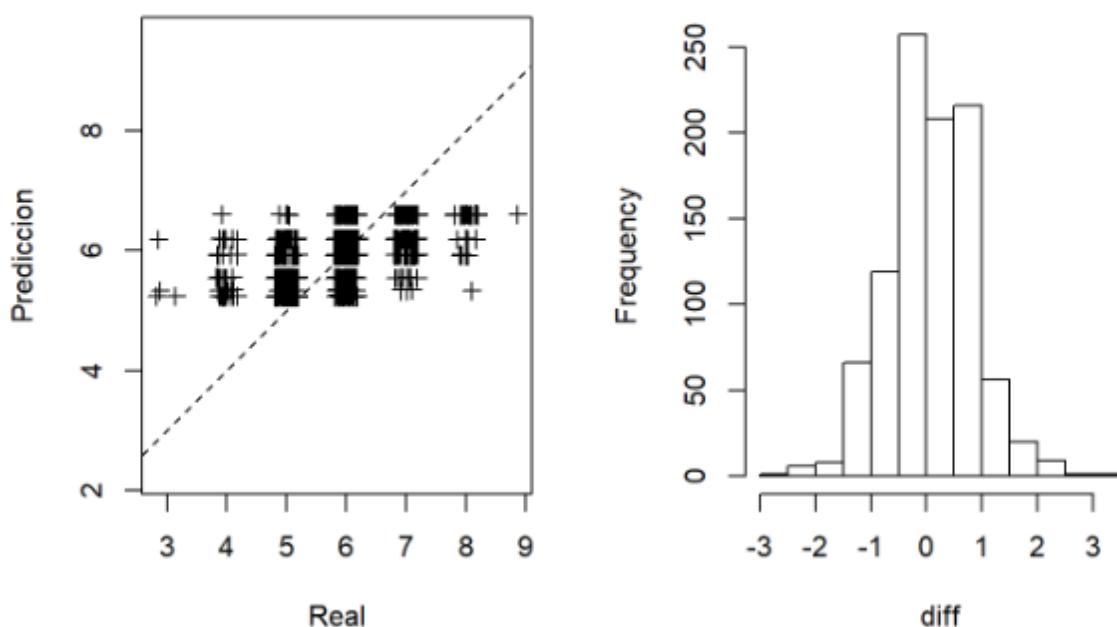
rpart.plot(RTWhite)
```



```

RTWhitePredictor = predict(RTWhite, whitetestN[,-12])
RTWhiteEvaluator = eval(RTWhitePredictor, whitetestN[,12], plot = T, title = "RT: ")
  
```

RT: Predicción vs Real



```
unlist(RTWhiteEvaluator)
```

```
##          RMSE        MAE       CORR
## 0.7661085 0.6047041 0.4979885
```

9. Sumario

```
cbind(Lineal = unlist(linealwhiteEvaluator1),
      Multiple = unlist(quadraticWhiteEvaluator),
      Anova = unlist(anovaWhiteEvaluator),
      Step = unlist(InteractWhiteStepEvaluator),
      RF = unlist(RandForestWhiteEvaluator),
      RFCV = unlist(RandForestCrossValWhiteEvaluator),
      KKNN = unlist(KKNNWhiteEvaluator),
      KKNNCV = unlist(KKNNCrossValWhiteEvaluator),
      SVM = unlist(SVMWhiteEvaluator),
      SVMCV = unlist(SVMCrossValWhiteEvaluator),
      RT = unlist(RTWhiteEvaluator) ) %>% round(3) %>% kable() %>% kable_styling()
```

	Lineal	Multiple	Poisson	Anova	Step	RandForest	RandForestCV	KKNN	KKNNCV	SVM	SVMCV	RT
RMSE	0.747	0.727	0.741	0.732	0.720	0.602	0.604	0.659	0.659	0.690	0.659	0.766
MAE	0.580	0.568	0.575	0.575	0.558	0.430	0.432	0.474	0.474	0.515	0.439	0.605
CORR	0.533	0.567	0.543	0.558	0.579	0.740	0.739	0.670	0.670	0.623	0.667	0.498

- El modelo de bosque aleatorio arroja sin lugar a dudas los mejores valores respecto a las variables usadas para su evaluación, principalmente en lo relativo al error absoluto de la media (MAE), que ha conseguido minimizarse con mayor distancia del resto de modelos.
- En los histogramas generados por la función eval() dejan de manifiesto que no hay ningún modelo que pueda satisfacer aceptablemente una predicción o clasificación en zonas de respuesta periféricas (calidad bajas o altas)

4.2. CLASIFICACIÓN – ANÁLISIS DEL VINO TINTO

0. Preparación de datos para modelaje

Creación de train y test para clasificación (Vino Tinto):

```
wineC = wine
whiteC = wineC %>% filter(clase=="WHITE") %>% select(-clase)
redC = wineC %>% filter(clase=="RED") %>% select(-clase)
redC$calidad = as.factor(redC$calidad)

summary(redC)
```

## acidez_fija	acidez_volatile	acidez_citrica	azucar_res
## Min. : 4.60	Min. :0.1200	Min. :0.000	Min. : 0.900
## 1st Qu.: 7.10	1st Qu.:0.3900	1st Qu.:0.090	1st Qu.: 1.900
## Median : 7.90	Median :0.5200	Median :0.260	Median : 2.200
## Mean : 8.32	Mean :0.5278	Mean :0.271	Mean : 2.539
## 3rd Qu.: 9.20	3rd Qu.:0.6400	3rd Qu.:0.420	3rd Qu.: 2.600
## Max. :15.90	Max. :1.5800	Max. :1.000	Max. :15.500
## cloruros	sulfitos_libres	sulfitos_totales	densidad
## Min. :0.01200	Min. : 1.00	Min. : 6.00	Min. :0.9901
## 1st Qu.:0.07000	1st Qu.: 7.00	1st Qu.: 22.00	1st Qu.:0.9956
## Median :0.07900	Median :14.00	Median : 38.00	Median :0.9968
## Mean :0.08747	Mean :15.87	Mean : 46.47	Mean :0.9967
## 3rd Qu.:0.09000	3rd Qu.:21.00	3rd Qu.: 62.00	3rd Qu.:0.9978
## Max. :0.61100	Max. :72.00	Max. :289.00	Max. :1.0037
## ph	sulfatos	alcohol	calidad
## Min. :2.740	Min. :0.3300	Min. : 8.40	3: 10
## 1st Qu.:3.210	1st Qu.:0.5500	1st Qu.: 9.50	4: 53
## Median :3.310	Median :0.6200	Median :10.20	5:681
## Mean :3.311	Mean :0.6581	Mean :10.42	6:638
## 3rd Qu.:3.400	3rd Qu.:0.7300	3rd Qu.:11.10	7:199
## Max. :4.010	Max. :2.0000	Max. :14.90	8: 18

```
IndexC = createDataPartition(redC$calidad, p = 0.8, list = F)
redtrain = redC[IndexC,]
redtest = redC[-IndexC,]
```

Para el análisis del vino tinto solo se usarán los 3 modelos de clasificación con mayor éxito creados para el análisis de vino blanco: K-Nearest Neighbors, RandomForest y SVM con la librería Caret.

1. Random Forest

```
set.seed(1)

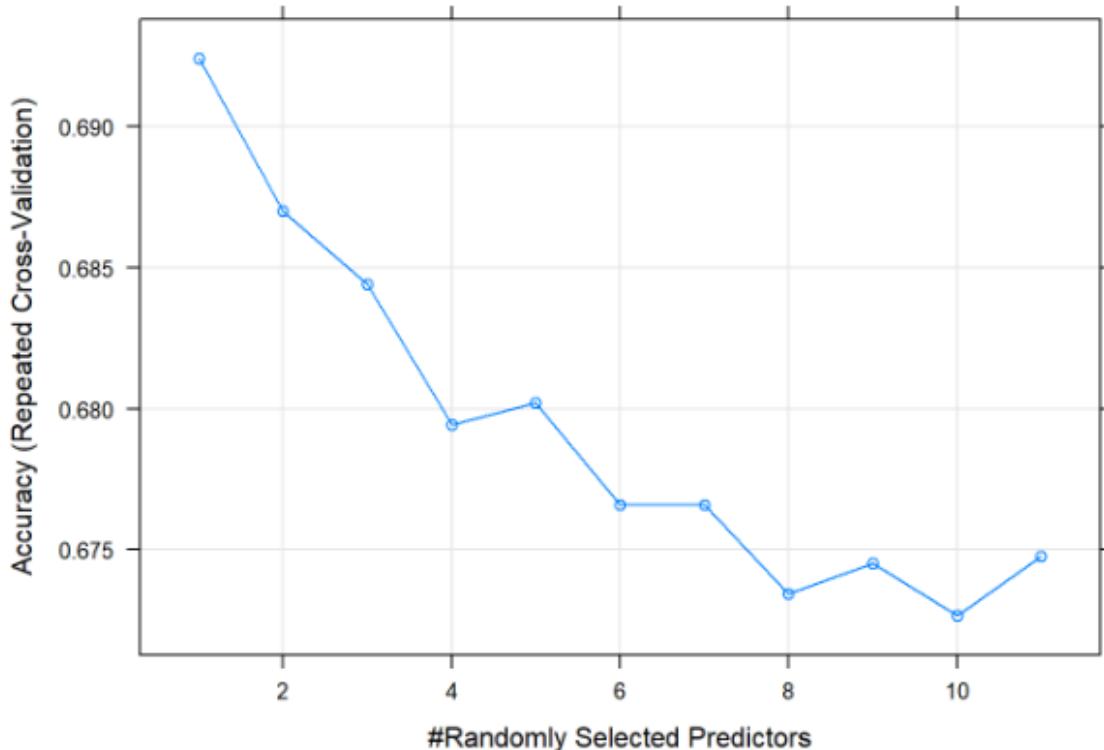
controlRandForest = trainControl(method = "repeatedcv", number = 10, repeats = 3)

matrizRandForest = expand.grid(.mtry = c(1:11)) ## Cambiaremos un poco respecto al calculo con el vino blanco

clusterRandForest = makePSOCKcluster(4)
registerDoParallel(clusterRandForest)

RandForestCrossValRed = train(calidad~, data = redtrain,
                               method = 'rf',
                               metric = "Accuracy",
                               trControl = controlRandForest,
                               tuneGrid = matrizRandForest,
                               preprocess = c("center", "scale"))

stopCluster(clusterRandForest)
plot(RandForestCrossValRed)
```



```
RandForestCrossValRed$bestTune
```

```
##     mtry
## 1      1
```

```
RandForestCrossValRedPredictor = predict(RandForestCrossValRed, redtest)
RandForestCrossValRedConfMatrix = confusionMatrix(RandForestCrossValRedPredictor, redtest$calidad)
RandForestCrossValRedConfMatrix
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   3   4   5   6   7   8
##           3   0   0   0   0   0   0
##           4   1   0   0   0   0   0
```

```

##      5   1   7 113   40   2   0
##      6   0   3  23   81   18   2
##      7   0   0   0   6  19   0
##      8   0   0   0   0   0   1
##
## Overall Statistics
##
##          Accuracy : 0.6751
##          95% CI : (0.6205, 0.7264)
##          No Information Rate : 0.429
##          P-Value [Acc > NIR] : < 2.2e-16
##
##          Kappa : 0.4665
##
## McNemar's Test P-Value : NA
##
## Statistics by Class:
##
##          Class: 3 Class: 4 Class: 5 Class: 6 Class: 7 Class: 8
## Sensitivity          0.000000 0.000000  0.8309  0.6378  0.48718 0.333333
## Specificity          1.000000 0.996743  0.7238  0.7579  0.97842 1.000000
## Pos Pred Value       NaN 0.000000  0.6933  0.6378  0.76000 1.000000
## Neg Pred Value       0.993691 0.968354  0.8506  0.7579  0.93151 0.993671
## Prevalence            0.006309 0.031546  0.4290  0.4006  0.12303 0.009464
## Detection Rate        0.000000 0.000000  0.3565  0.2555  0.05994 0.003155
## Detection Prevalence 0.000000 0.003155  0.5142  0.4006  0.07886 0.003155
## Balanced Accuracy     0.500000 0.498371  0.7773  0.6978  0.73280 0.666667

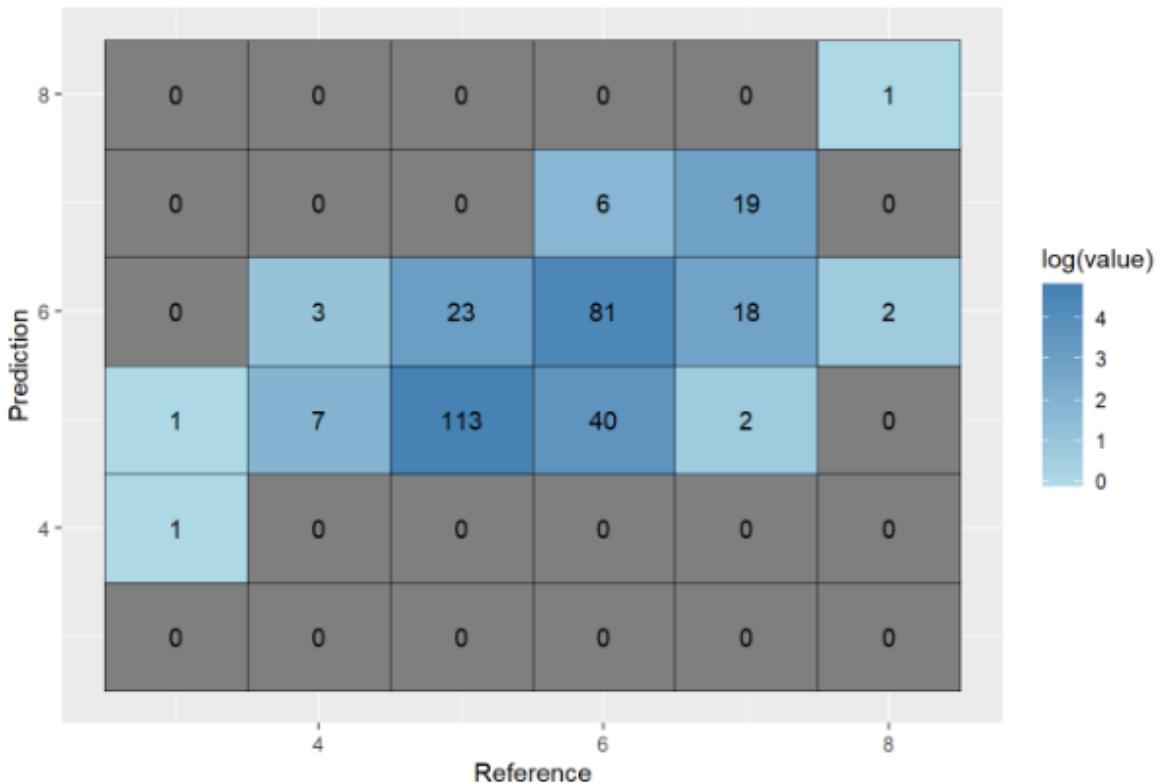
```

```

RandForestCrossValRedCF = RandForestCrossValRedConfMatrix$table %>% melt()
ggplot(RandForestCrossValRedCF, aes(Reference, y = Prediction)) +
  geom_tile(aes(fill = log(value)), color="black") +
  geom_text(aes(label = value)) +
  scale_fill_gradient(low = "lightblue", high = "steelblue") +
  labs(title = "Matriz de confusión RandForest")

```

Matriz de confusión RandForest



2. K-Nearest Neighbours

Se aprovechan las matrices y configuraciones de entrenamiento usadas anteriormente con el vino blanco.

```

set.seed(1)

clusterKKNN = makePSOCKcluster(4)

registerDoParallel(clusterKKNN)

controlKKNN = trainControl(method = "repeatedcv", repeats = 5, number = 7)

matrizKKNN = expand.grid(kmax = c(3, 5, 7, 9, 11, 17), distance = c(1, 2),
  kernel = c("rectangular", "gaussian", "cos"))

KKNNCrossValRed <- train(calidad ~ ., data = redtrain,
  
```

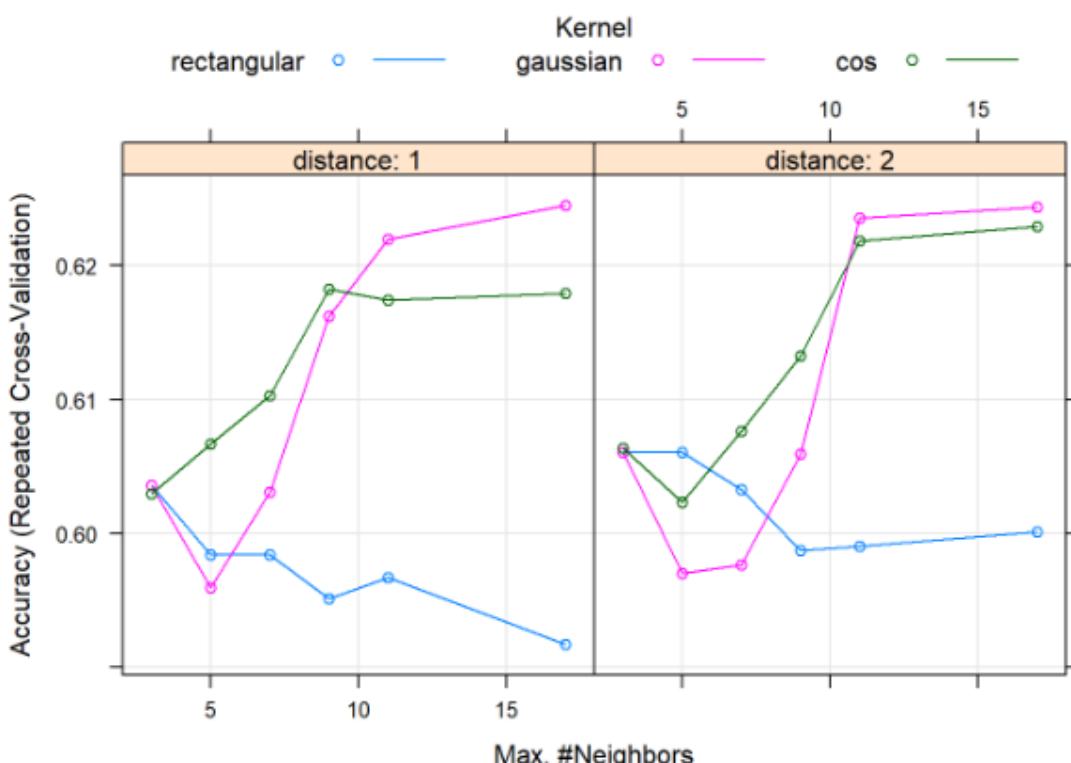
```

method = "kknn",
trControl = controlKKNN,
tuneGrid = matrizKKNN,
metric = "Accuracy",
preProcess = c("center", "scale"))

stopCluster(clusterKKNN)
summary(KKNNCrossValRed)
    
```

```

## 
## Call:
## kknn::train.kknn(formula = .outcome ~ ., data = dat, kmax = param$kmax
## , distance = param$distance, kernel = as.character(param$kernel))
## 
## Type of response variable: nominal
## Minimal misclassification: 0.3549142
## Best kernel: gaussian
## Best k: 15
plot(KKNNCrossValRed)
    
```



```
KKNNCrossValRed$bestTune
```

```
##      kmax distance   kernel
## 32      17        1 gaussian
```

```
KKNNCrossValRedPredictor <- predict(KKNNCrossValRed, redtest)
KKNNCrossValRedConfMatrix = confusionMatrix(KKNNCrossValRedPredictor, redtest$calidad)
KKNNCrossValRedConfMatrix
```

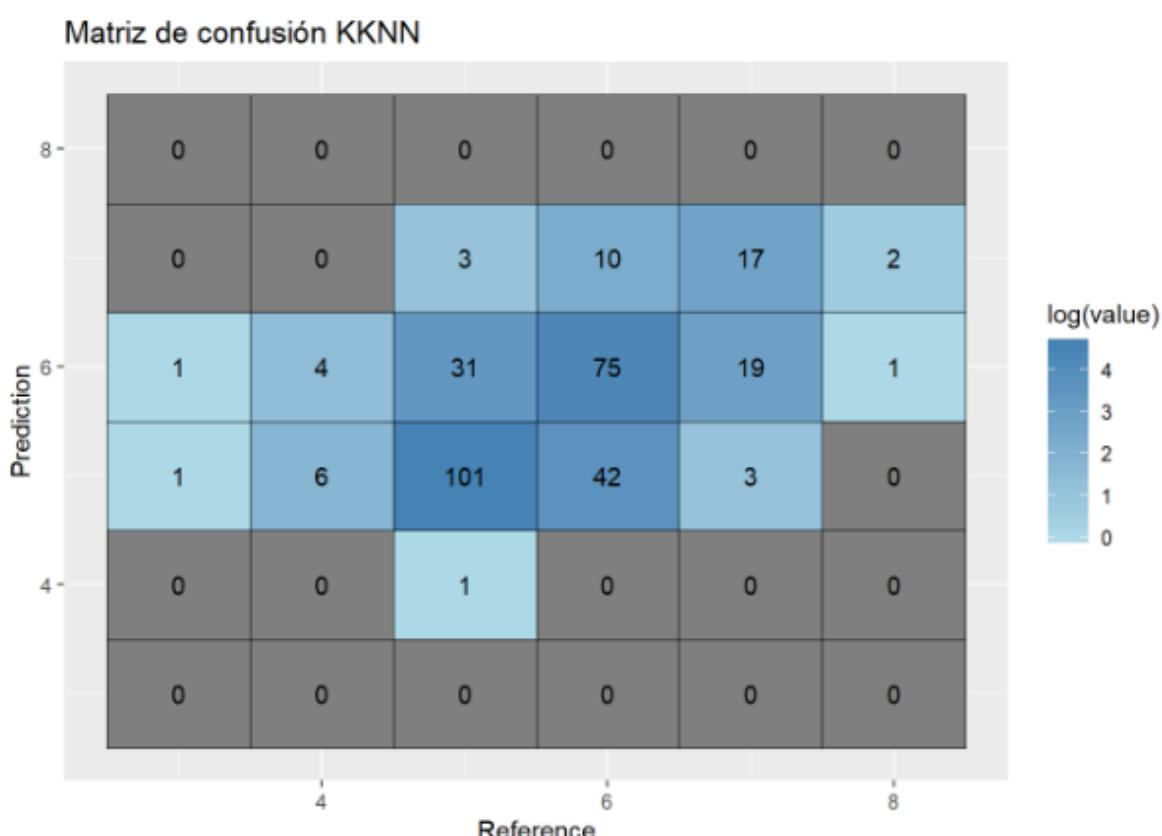
```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction 3 4 5 6 7 8
##           3 0 0 0 0 0 0
##           4 0 0 1 0 0 0
##           5 1 6 101 42 3 0
##           6 1 4 31 75 19 1
##           7 0 0 3 10 17 2
##           8 0 0 0 0 0 0
##
## Overall Statistics
##
##                 Accuracy : 0.6088
##                 95% CI : (0.5527, 0.6629)
## No Information Rate : 0.429
## P-Value [Acc > NIR] : 9.322e-11
##
##                 Kappa : 0.3638
##
## McNemar's Test P-Value : NA
##
## Statistics by Class:
##
##                 Class: 3 Class: 4 Class: 5 Class: 6 Class: 7 Class: 8
## Sensitivity          0.000000 0.000000 0.7426 0.5906 0.43590 0.000000
## Specificity          1.000000 0.996743 0.7127 0.7053 0.94604 1.000000
```

## Pos Pred Value	NaN	0.000000	0.6601	0.5725	0.53125
## Neg Pred Value	0.993691	0.968354	0.7866	0.7204	0.92281 0.990536
## Prevalence	0.006309	0.031546	0.4290	0.4006	0.12303 0.009464
## Detection Rate	0.000000	0.000000	0.3186	0.2366	0.05363 0.000000
## Detection Prevalence	0.000000	0.003155	0.4826	0.4132	0.10095 0.000000
## Balanced Accuracy	0.500000	0.498371	0.7277	0.6479	0.69097 0.500000

```

KKNNCrossValRedCF = KKNNCrossValRedConfMatrix$table %>% melt()
ggplot(KKNNCrossValRedCF, aes(Reference, y = Prediction))+
  geom_tile(aes(fill = log(value)), color="black")+
  geom_text(aes(label = value)) +
  scale_fill_gradient(low = "lightblue", high = "steelblue") +
  labs(title = "Matriz de confusión KKNN")

```



3. Support Vector Machine

Misma filosofía que usada con el Vino Blanco.

```

set.seed(1)

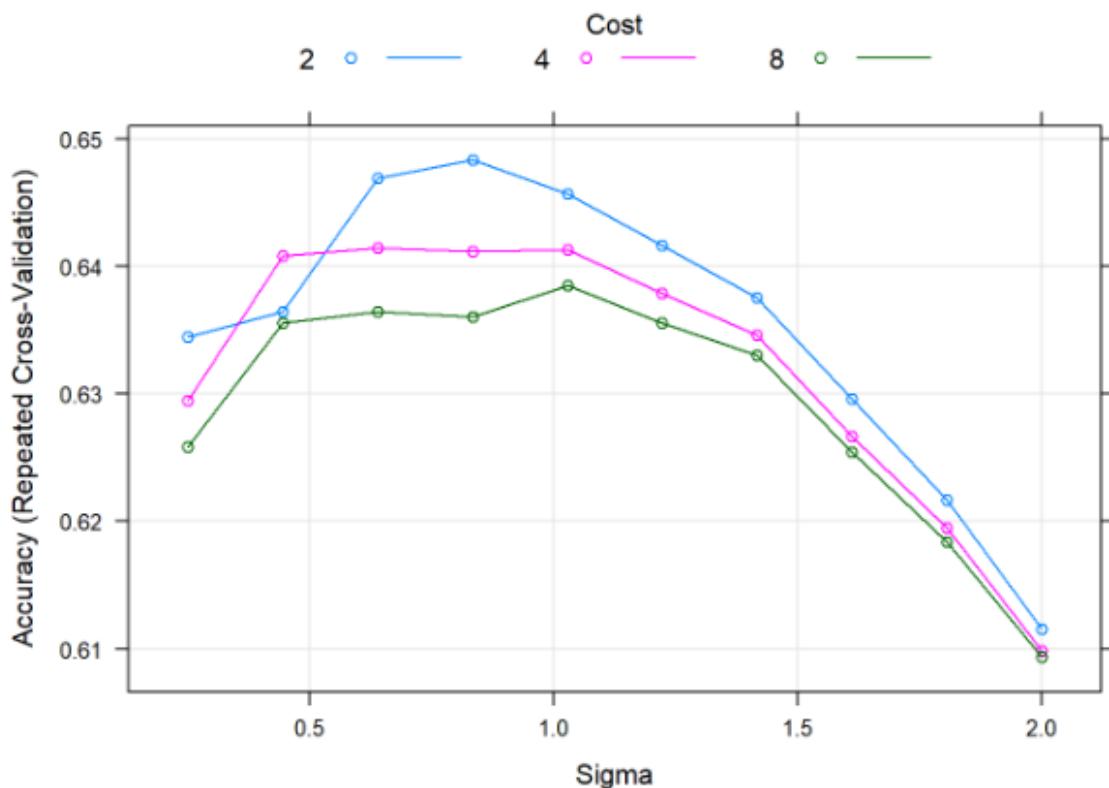
clusterSVM = makePSOCKcluster(4)
registerDoParallel(clusterSVM)
controlSVM = trainControl(method = "repeatedcv", number = 5, repeats = 5)
matrizSVM = expand.grid(C = 2^(1:3), sigma = seq(0.25, 2, length = 10))

# Ambos parámetros han sido ajustados ya que en los valores de sigma existía un pico de precisión entre los valores escogidos y el valor de C, que más allá del intervalo escogido no parecían revelar más datos significativos en el entrenamiento.

SVMCrossValRed = train(calidad~, data = redtrain,
                        method = 'svmRadial',
                        trControl = controlSVM,
                        tuneGrid = matrizSVM)

stopCluster(clusterSVM)
plot(SVMCrossValRed)

```



```
SVMCrossValRed$bestTune
```

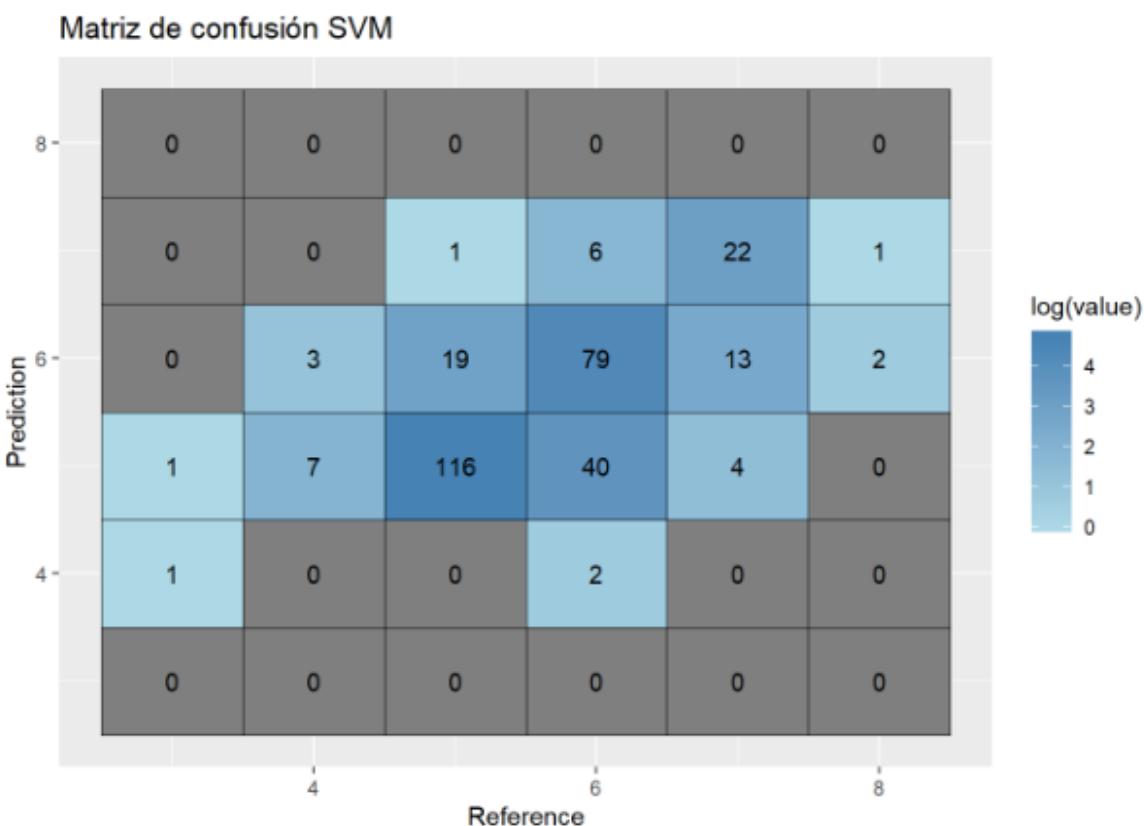
```
##           sigma C
## 4 0.8333333 2
```

```
SVMCrossValRedPredictor = predict(SVMCrossValRed, redtest)
SVMCrossValRedConfMatrix = confusionMatrix(SVMCrossValRedPredictor, redtest$calidad)
SVMCrossValRedConfMatrix
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   3   4   5   6   7   8
##   3          0   0   0   0   0   0
##   4          1   0   0   2   0   0
##   5          1   7 116  40   4   0
##   6          0   3  19  79  13   2
##   7          0   0   1   6  22   1
##   8          0   0   0   0   0   0
##
## Overall Statistics
##
##               Accuracy : 0.6845
##                 95% CI : (0.6303, 0.7353)
##      No Information Rate : 0.429
## P-Value [Acc > NIR] : < 2.2e-16
##
##               Kappa : 0.4863
##
## McNemar's Test P-Value : NA
##
## Statistics by Class:
##
##                   Class: 3 Class: 4 Class: 5 Class: 6 Class: 7 Class: 8
## Sensitivity       0.000000 0.000000  0.8529   0.6220  0.56410 0.000000
```

## Specificity	1.000000	0.990228	0.7127	0.8053	0.97122		
1.000000							
## Pos Pred Value		Nan	0.000000	0.6905	0.6810	0.73333	Nan
## Neg Pred Value	0.993691	0.968153	0.8658	0.7612	0.94077	0.990536	
## Prevalence	0.006309	0.031546	0.4290	0.4006	0.12303	0.009464	
## Detection Rate	0.000000	0.000000	0.3659	0.2492	0.06940	0.000000	
## Detection Prevalence	0.000000	0.009464	0.5300	0.3659	0.09464	0.000000	
## Balanced Accuracy	0.500000	0.495114	0.7828	0.7137	0.76766	0.500000	

```
SVMCrossValRedCF = SVMCrossValRedConfMatrix$table %>% melt()
ggplot(SVMCrossValRedCF, aes(Reference, y = Prediction))+
  geom_tile(aes(fill = log(value)), color="black")+
  geom_text(aes(label = value)) +
  scale_fill_gradient(low = "lightblue", high = "steelblue") +
  labs(title = "Matriz de confusión SVM")
```



4. Sumario

```

rbind(RandForest = RandForestCrossValRedConfMatrix$overall %>
% round(3) ,
      KKNN = KKNNCrossValRedConfMatrix$overall %>% round(3) ,
      SVM = SVMCrossValRedConfMatrix$overall%>% round(3) ) %>% kable() %>%
kable_styling()
    
```

	Accuracy	Kappa	AccuracyLower	AccuracyUpper	AccuracyNull	AccuracyPValue	McnemarPValue
RandForest	0.675	0.467	0.620	0.726	0.429	0	NaN
KKNN	0.609	0.364	0.553	0.663	0.429	0	NaN
SVM	0.685	0.486	0.630	0.735	0.429	0	NaN

- El modelo que mejor se ajusta a la realidad es, de nuevo, el modelo Random Forest, con una precisión del 69,7% y un Kappa por encima del 50% (que es una medida la precisión del sistema respecto a la precisión de un sistema aleatorio).
- Sin embargo, si se observa en la matriz de confusión completa de los modelos, y tal y como se había predicho en el análisis exploratorio y análisis del vino blanco, ninguno de los modelos tiene suficiente sensibilidad para clasificar los vinos de calidades bajas o altas (periféricas).

4.3. MODELO DE RESPUESTA BINARIA

RandomForest en Clasificación

Como ejercicio final del proyecto, se creará un modelo RandomForest de clasificación simplificado, orientado a un dataset de respuesta binaria (calidad >=7) que se creará a partir del original, para ambos tipos de vino y finalmente se evaluará con la matriz de confusión.

```

wineF = wine %>%
  mutate(excelencia=ifelse(calidad>=7, "SI", "NO") %>% as.factor()) %>%
  select(-calidad)

redF = wineF %>% filter(clase=="RED") %>% select(-clase)
whiteF = wineF %>% filter(clase=="WHITE") %>% select(-clase)
    
```

```
summary(redF)
```

```
## acidez_fija      acidez_volatil      acidez_citrica      azucar_res
## Min.   : 4.60   Min.   :0.1200   Min.   :0.000   Min.   : 0.900
## 1st Qu.: 7.10   1st Qu.:0.3900   1st Qu.:0.090   1st Qu.: 1.900
## Median : 7.90   Median :0.5200   Median :0.260   Median : 2.200
## Mean    : 8.32   Mean    :0.5278   Mean    :0.271   Mean    : 2.539
## 3rd Qu.: 9.20   3rd Qu.:0.6400   3rd Qu.:0.420   3rd Qu.: 2.600
## Max.    :15.90   Max.    :1.5800   Max.    :1.000   Max.    :15.500
## cloruros      sulfitos_libres      sulfitos_totales      densidad
## Min.   :0.01200   Min.   : 1.00   Min.   : 6.00   Min.   :0.9901
## 1st Qu.:0.07000   1st Qu.: 7.00   1st Qu.:22.00   1st Qu.:0.9956
## Median :0.07900   Median :14.00   Median : 38.00   Median :0.9968
## Mean    :0.08747   Mean    :15.87   Mean    : 46.47   Mean    :0.9967
## 3rd Qu.:0.09000   3rd Qu.:21.00   3rd Qu.: 62.00   3rd Qu.:0.9978
## Max.    :0.61100   Max.    :72.00   Max.    :289.00   Max.    :1.0037
## ph          sulfatos      alcohol      excelencia
## Min.   :2.740   Min.   :0.3300   Min.   : 8.40   NO:1382
## 1st Qu.:3.210   1st Qu.:0.5500   1st Qu.: 9.50   SI: 217
## Median :3.310   Median :0.6200   Median :10.20
## Mean    :3.311   Mean    :0.6581   Mean    :10.42
## 3rd Qu.:3.400   3rd Qu.:0.7300   3rd Qu.:11.10
## Max.    :4.010   Max.    :2.0000   Max.    :14.90
```

```
summary(whiteF)
```

```
## acidez_fija      acidez_volatil      acidez_citrica      azucar_res
## Min.   : 3.800   Min.   :0.0800   Min.   :0.0000   Min.   : 0.600
## 1st Qu.: 6.300   1st Qu.:0.2100   1st Qu.:0.2700   1st Qu.: 1.700
## Median : 6.800   Median :0.2600   Median :0.3200   Median : 5.200
## Mean    : 6.855   Mean    :0.2782   Mean    :0.3342   Mean    : 6.391
## 3rd Qu.: 7.300   3rd Qu.:0.3200   3rd Qu.:0.3900   3rd Qu.: 9.900
## Max.    :14.200   Max.    :1.1000   Max.    :1.6600   Max.    :65.800
## cloruros      sulfitos_libres      sulfitos_totales      densidad
## Min.   :0.00900   Min.   : 2.00   Min.   : 9.0    Min.   :0.9871
```

```

## 1st Qu.:0.03600 1st Qu.: 23.00 1st Qu.:108.0 1st Qu.:0.99
17
## Median :0.04300 Median : 34.00 Median :134.0 Median :0.9937
## Mean :0.04577 Mean : 35.31 Mean :138.4 Mean :0.9940
## 3rd Qu.:0.05000 3rd Qu.: 46.00 3rd Qu.:167.0 3rd Qu.:0.9961
## Max. :0.34600 Max. :289.00 Max. :440.0 Max. :1.0390
## ph sulfatos alcohol excelencia
## Min. :2.720 Min. :0.2200 Min. : 8.00 NO:3838
## 1st Qu.:3.090 1st Qu.:0.4100 1st Qu.: 9.50 SI:1060
## Median :3.180 Median :0.4700 Median :10.40
## Mean :3.188 Mean :0.4898 Mean :10.51
## 3rd Qu.:3.280 3rd Qu.:0.5500 3rd Qu.:11.40
## Max. :3.820 Max. :1.0800 Max. :14.20

```

```

IndexF = createDataPartition(redF$excelencia, p = 0.85, list = F)
redtrainF = redF[IndexF,]
redtestF = redF[-IndexF,]

IndexF = createDataPartition(whiteF$excelencia, p = 0.85, list = F)
whitetrainF = whiteF[IndexF,]
whitetestF = whiteF[-IndexF,]

```

1. Random Forest Vino blanco

```

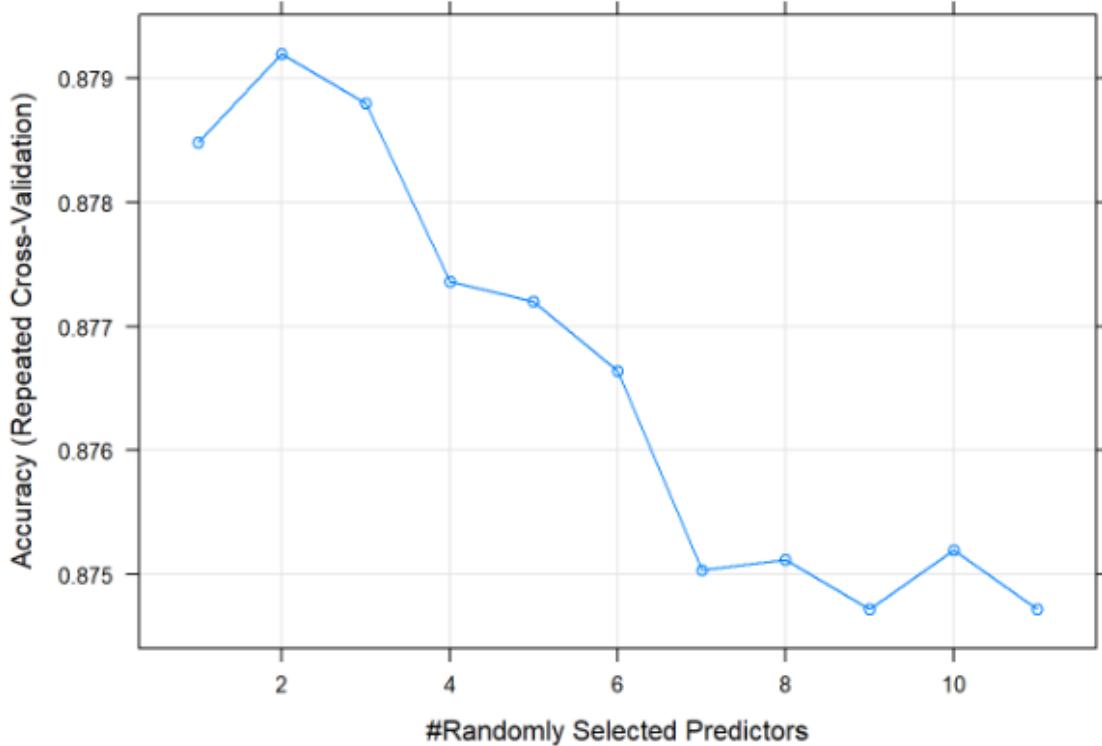
set.seed(10)

clusterRandForest = makePSOCKcluster(4)
registerDoParallel(clusterRandForest)

RandForestCrossValWhiteF = train(excelencia~, data = whitetrainF,
                                 method = 'rf',
                                 metric = "Accuracy",
                                 trControl = controlRandForest,
                                 tuneGrid = matrizRandForest,
                                 preprocess = c("center", "scale"))

```

```
stopCluster(clusterRandForest)
plot(RandForestCrossValWhiteF)
```



```
RandForestCrossValWhiteF$bestTune
```

```
##      mtry
## 2      2
```

```
RandForestCrossValWhiteFPredictor = predict(RandForestCrossValWhiteF, whitetestF)
RandForestCrossValWhiteFConfMatrix = confusionMatrix(RandForestCrossValWhiteFPredictor, whitetestF$excelencia)
RandForestCrossValWhiteFConfMatrix
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   NO   SI
##       NO 100  10
##       SI  10 100
```

```

##           NO 553 57
##           SI 22 102
##
##           Accuracy : 0.8924
##           95% CI : (0.8677, 0.9139)
##           No Information Rate : 0.7834
##           P-Value [Acc > NIR] : 7.302e-15
##
##           Kappa : 0.6554
##
##           McNemar's Test P-Value : 0.0001306
##
##           Sensitivity : 0.9617
##           Specificity : 0.6415
##           Pos Pred Value : 0.9066
##           Neg Pred Value : 0.8226
##           Prevalence : 0.7834
##           Detection Rate : 0.7534
##           Detection Prevalence : 0.8311
##           Balanced Accuracy : 0.8016
##
##           'Positive' Class : NO
##

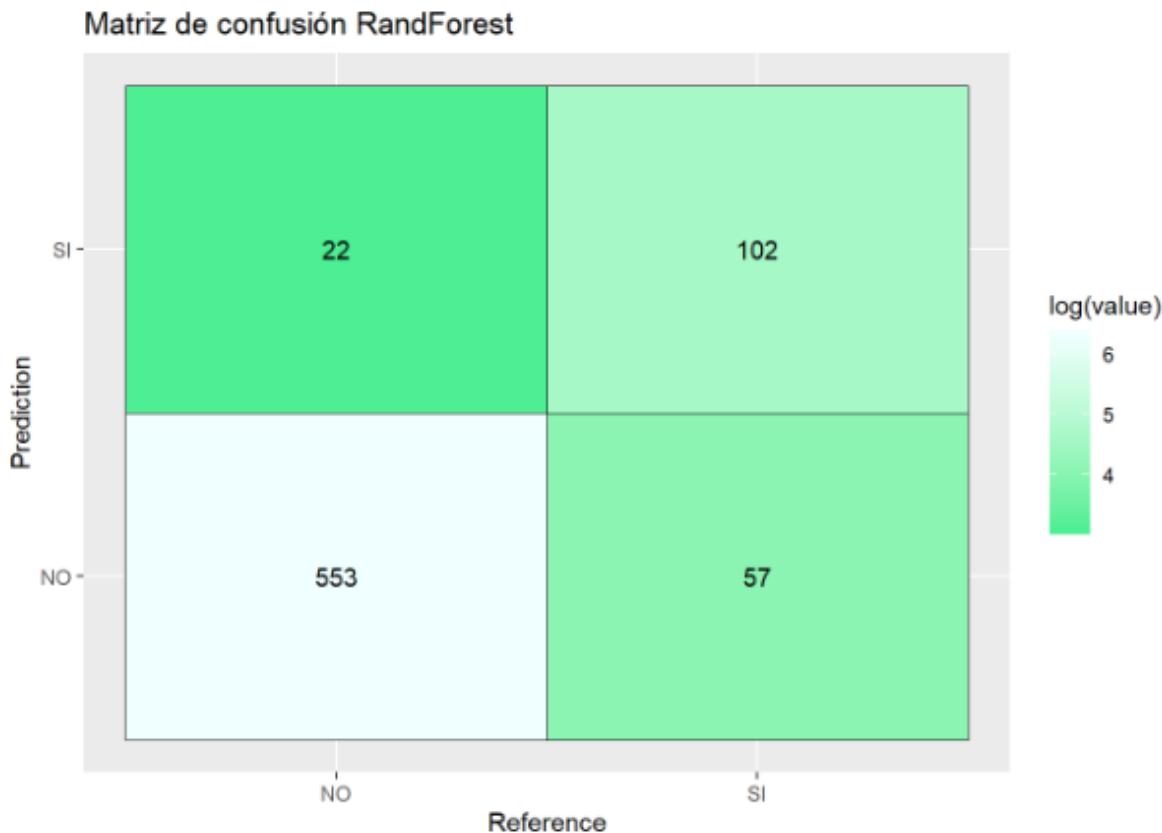
```

```

RandForestCrossValWhiteFCF = RandForestCrossValWhiteFConfMatrix$table %>%
  melt()

ggplot(RandForestCrossValWhiteFCF, aes(Reference, y = Prediction)) +
  geom_tile(aes(fill = log(value)), color="black") +
  geom_text(aes(label = value)) +
  scale_fill_gradient(low = "seagreen2", high = "azure") +
  labs(title = "Matriz de confusión RandForest")

```



2. Random Forest Vino tinto

```

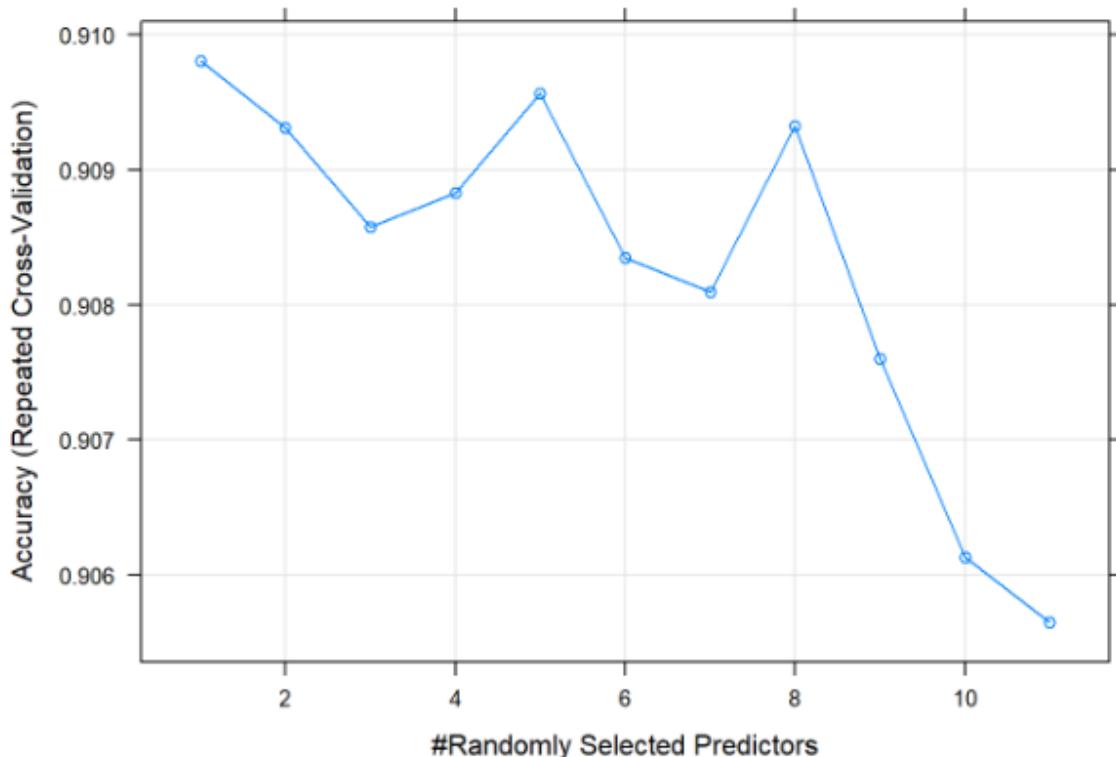
set.seed(10)

clusterRandForest = makePSOCKcluster(4)
registerDoParallel(clusterRandForest)

RandForestCrossValRedF = train(excelencia~, data = redtrainF,
                               method = 'rf',
                               metric = "Accuracy",
                               trControl = controlRandForest,
                               tuneGrid = matrizRandForest,
                               preProcess = c("center", "scale"))

stopCluster(clusterRandForest)
plot(RandForestCrossValRedF)

```



```
RandForestCrossValRedF$bestTune
```

```
##     mtry
## 1      1
```

```
RandForestCrossValRedFPredictor = predict(RandForestCrossValRedF, redtestF)
RandForestCrossValRedFConfMatrix = confusionMatrix(RandForestCrossValRedF$Predictor, redtestF$excelencia)
RandForestCrossValRedFConfMatrix
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   NO   SI
##           NO 201  17
##           SI   6  15
```

```

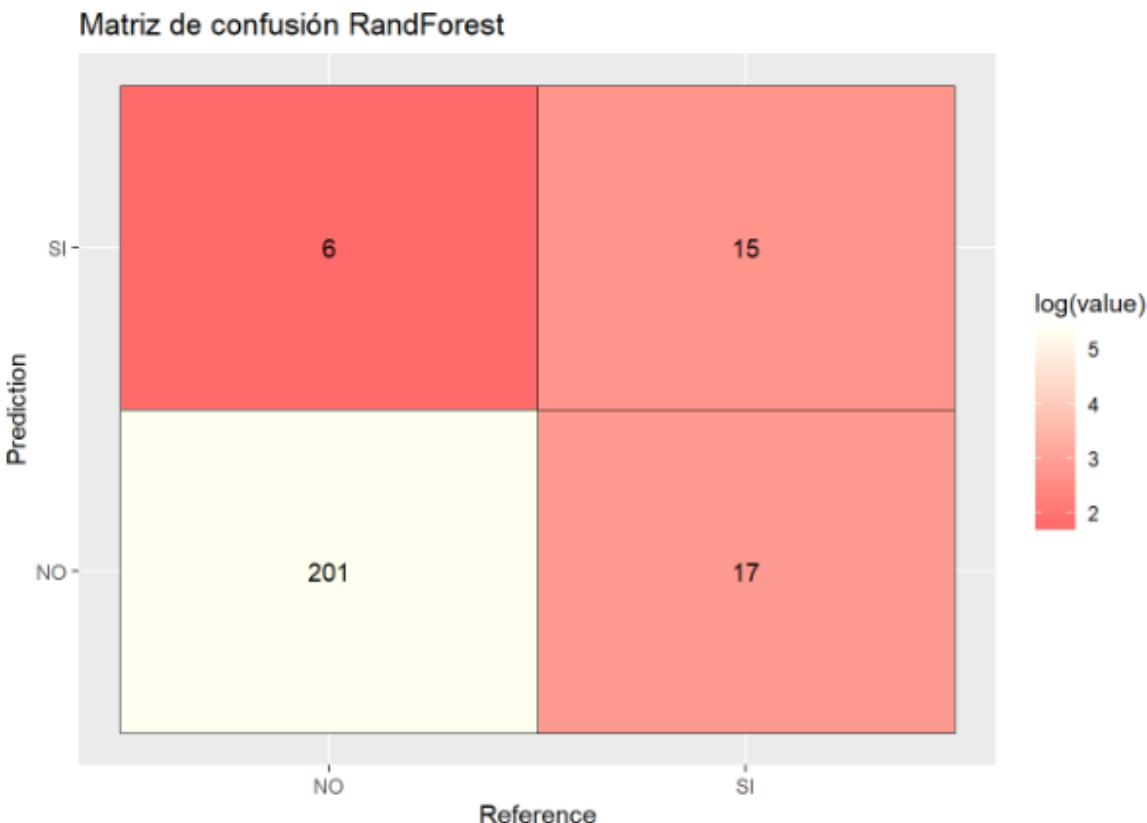
## 
##           Accuracy : 0.9038
##           95% CI : (0.8591, 0.938)
##   No Information Rate : 0.8661
##   P-Value [Acc > NIR] : 0.04861
##
##           Kappa : 0.5145
##
##   Mcnemar's Test P-Value : 0.03706
##
##           Sensitivity : 0.9710
##           Specificity : 0.4688
##           Pos Pred Value : 0.9220
##           Neg Pred Value : 0.7143
##           Prevalence : 0.8661
##           Detection Rate : 0.8410
##           Detection Prevalence : 0.9121
##           Balanced Accuracy : 0.7199
##
##           'Positive' Class : NO
##

```

```

RandForestCrossValRedFCF = RandForestCrossValRedFConfMatrix$table %>% melt()
ggplot(RandForestCrossValRedFCF, aes(Reference, y = Prediction)) +
  geom_tile(aes(fill = log(value)), color="black") +
  geom_text(aes(label = value)) +
  scale_fill_gradient(low = "indianred1", high = "ivory") +
  labs(title = "Matriz de confusión RandForest")

```



3. Sumario

```
rbind(`Vino Blanco` = RandForestCrossValWhiteFConfMatrix$overall %>% round(3),
      `Vino Tinto` = RandForestCrossValRedFConfMatrix$overall %>% round(3)
    ))
```

```
##          Accuracy Kappa AccuracyLower AccuracyUpper AccuracyNull
## Vino Blanco     0.892  0.655        0.868        0.914       0.783
## Vino Tinto      0.904  0.515        0.859        0.938       0.866
##          AccuracyPValue McNemarPValue
## Vino Blanco           0.000        0.000
## Vino Tinto            0.049        0.037
```

Los resultados obtenidos en este modelo simplificado de respuesta binaria han mejorado notablemente; aunque el modelo sigue presentando limitaciones y ha obligado a simplificarlo todo, pero es bastante fiable. Esta simplificación ‘ayuda’ a mejorar el problema con la predicción/clasificación en las regiones periféricas de la calidad.



5. CONCLUSIÓN Y PROPUESTAS

RMarkDown

5.1. DISCUSIÓN Y EXPOSICIÓN DE CONCLUSIONES

Las conclusiones tanto para el Vino Blanco como el Vinto Tinto dejan claro que el mejor modelo de aprendizaje para este dataset ha de ser de clasificación, y concretamente el algoritmo Random Forest. Esto es, se trata de un dataset que posee muchas variables con problemas de colinealidad y mucho ruido, baja correlación.

El mejor modelo que puede ajustarse a las necesidades de clasificación/regresión de un dataset de estas características porque el algoritmo del Random Forest aprovecha al máximo el llamado fenómeno “bagging”, que es la capacidad para promediar interacciones entre datos ruidosas, y poseen baja parcialidad (concretamente no es nuestro caso con todas las variables, pero sí con las más importantes).

El modelo de clasificación frente al de predicción parece más adecuado para este dataset, aunque en general se puede decir que los datos del Vino Blanco son más equilibrados, y muy probablemente esto se deba a que se posee una cantidad de datos bastante mayor de éste. Este modelo sólo se consideraría suficientemente válido para usarlo en toma de decisiones si sirviese exclusivamente a los vinos de calidad media.

Predicción - RandomForest para el Vino Blanco

```
unlist(RandForestCrossValWhiteEvaluator)
```

```
##          RMSE         MAE        CORR
## 0.6040035 0.4321573 0.7387660
```



Clasificación - RandomForest para el Vino Tinto

```
RandForestCrossValRedConfMatrix$overall %>% round(3)
```

	Accuracy	Kappa	AccuracyLower	AccuracyUpper	AccuracyNull
##	0.675	0.467	0.620	0.726	0.429
## AccuracyPValue		McnemarPValue			
##	0.000	NaN			

Clasificación - RandomForest para excelencia de ambos vinos

```
rbind(`Vino Blanco` = RandForestCrossValWhiteFConfMatrix$overall %>% round(3),
      `Vino Tinto` = RandForestCrossValRedFConfMatrix$overall %>% round(3)
    )
```

	Accuracy	Kappa	AccuracyLower	AccuracyUpper	AccuracyNull
## Vino Blanco	0.892	0.655	0.868	0.914	0.783
## Vino Tinto	0.904	0.515	0.859	0.938	0.866
##		AccuracyPValue	McnemarPValue		
## Vino Blanco		0.000	0.000		
## Vino Tinto		0.049	0.037		

5.2. PROPUESTAS DE MEJORA

1. TÉCNICAS

- Un estudio de los outliers más profundo (nótese que en los análisis de varianza, mejorando el modelo de regresión lineal, se presentaban observaciones muy dispersas de la nube de puntos), y una limpieza de variables colineales (se han intuido algunas durante el comienzo de esta parte del proyecto), podría haber ayudado significativamente a la mejora de los modelos

- Un estudio y análisis más meticuloso acerca de los predictores/clasificadores que presentaban mayor error hubiera ayudado a desentrañar el problema de la predicción/clasificación en las regiones periféricas.
- Un estudio más exhaustivo del balanceo de hiperparámetros. De hecho, durante el desarrollo del proyecto, la variación de los mismos ha producido cambios muy resaltables en el resultado final de algunos modelos. Un mayor conocimiento de todas las opciones que hay para computar podría contribuir de gran grado a su mejora
- Una mejora de la capacidad de computación o uso de más clusters

2. COMERCIALES

- Informarse mejor de todo el abanico de atributos / variables presentes en el vino, sus características e incidencias. Una recogida de datos basados en un estudio especializado y auditado por profesionales enólogos podría contribuir a una gran mejora iterativa del modelaje, porque no sólo serviría para obtener mejores datos si no para tener cómo y con qué contrastarlos a posteriori
- Consultar a los clientes sobre sus verdaderas necesidades en la venta. Podría darse la situación en la que la prioridad de la respuesta a clasificar/predecir no sea la calidad, o que ésta no se haya recogido/estimado bajo los parámetros precisos. Una mejor comunicación con el cliente podría ayudar notablemente a su mejora
- En general, solicitar más datos a los clientes / proveedores. Una mayor cantidad de datos permitiría la posibilidad de usar modelos más complejos (como modelos de Deep Learning) que permitiera un aprendizaje más profundo del comportamiento de los vinos



6. REFERENCIAS

6.1. BIBLIOGRAFÍA

- Dataset: <https://archive.ics.uci.edu/ml/datasets/Wine+Quality>
- Tableau public:
https://public.tableau.com/profile/tatan7904#/vizhome/Vinos_15826478993250/QUALITYW_HITE
- Datos sobre el vino:
 - <http://www.diccionariodelvino.com/>
 - <https://www.science.org.au/curious/earth-environment/chemistry-wine-part-1>
 - <https://www.sciencedirect.com/science/article/pii/S0167923609001377>
 - <https://www.scitepress.org/Papers/2015/55519/55519.pdf>
 - http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0101-20612015000100095
 - <http://www.morethanorganic.com/sulphur-in-the-bottle>
 - <http://gwi.missouri.edu/publications/2013spring.pdf>
 - <http://gwi.missouri.edu/publications/2013spring.pdf>