Práctica 2: Limpieza y análisis de datos

Andoni Zengotitabengoa Fernandez & Lucas Farris

23 de mayo de 2020

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

El dataset elegido se llama `Wine Quality Data Set` (fuente: https://archive.ics.uci.edu/ml/datasets/wine+quality) y contiene datos de vinos rojos y blancos del tipo *Vinho Verde* portugués. Los datos incluyen variables fisicoquímicas y una variable objetivo sensorial, que representa la calidad del vino.

El dataset es importante porque permite que, a través del análisis estadístico, se pueda estudiar las relaciones entre la calidad percibida del vino y sus propiedades químicas y físicas. La pregunta que se pretende responder con los datos es si es posible predecir la calidad de un vino, teniendo en cuenta sus propiedades.

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

```
# importamos los datos de los csv descargados
red_wine_data <- read.csv('winequality-red.csv', sep = ";",  quote = "\"")
white_wine_data <- read.csv('winequality-white.csv', sep = ";",  quote = "\"")
# añadimos el tipo de vino como una nueva variable categórica
red_wine_data$type <- "red"
white_wine_data$type <- "white"
# juntamos los datos
dataset <- rbind(red_wine_data,white_wine_data)
dataset$type <- as.factor(dataset$type)
# nombres de las columnas disponibles
colnames(dataset)
```

```
##  [1] "fixed.acidity"        "volatile.acidity"     "citric.acid"
##  [4] "residual.sugar"       "chlorides"            "free.sulfur.dioxide"
##  [7] "total.sulfur.dioxide" "density"              "pH"
## [10] "sulphates"            "alcohol"              "quality"
## [13] "type"
```

```
# comprobaremos cuantos registros duplicados hay
sum(duplicated(dataset))
```

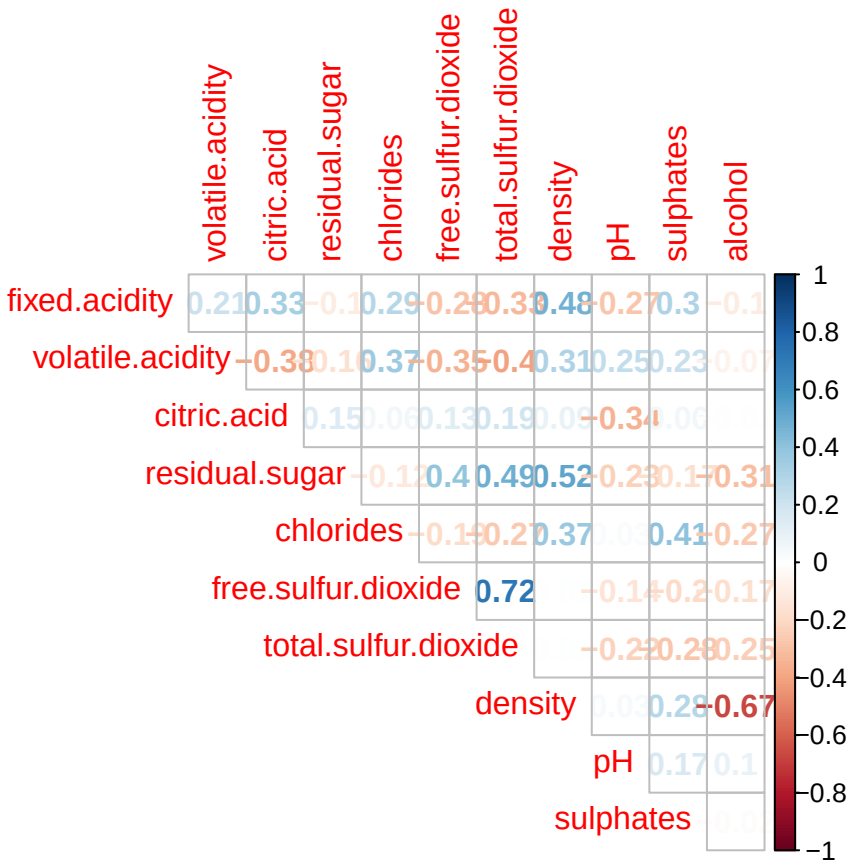
```
## [1] 1177
```

```
# eliminaremos los registros duplicados
dataset <- dataset[!duplicated(dataset),]
```

```
# cantidad de registros disponibles
nrow(dataset)
```

```
## [1] 5320
```

```
# miraremos ahora si hay variables numericas en los datos que tienen alta correlación
corrplot(cor(dataset[,0:11], method='pearson'), type="upper", method='number', diag=FALSE)
```



La correlación más alta (0.72) fue entre las variables *free.sulfur.dioxide* y *total.sulfur.dioxide* pero no es suficientemente alta para retirar una de las variables con confianza. La correlación más baja (-0.67) fue entre las variables *density* y *alcohol* pero tampoco es suficientemente baja para eliminar una de las variables.

3. Limpieza de los datos.

3.1. ¿Los datos contienen ceros o elementos vacíos? ¿Cómo gestionarías cada uno de estos casos?

```
# comprobaremos si algun valor de nuestro dataset es vacío
any(is.na(dataset))
```

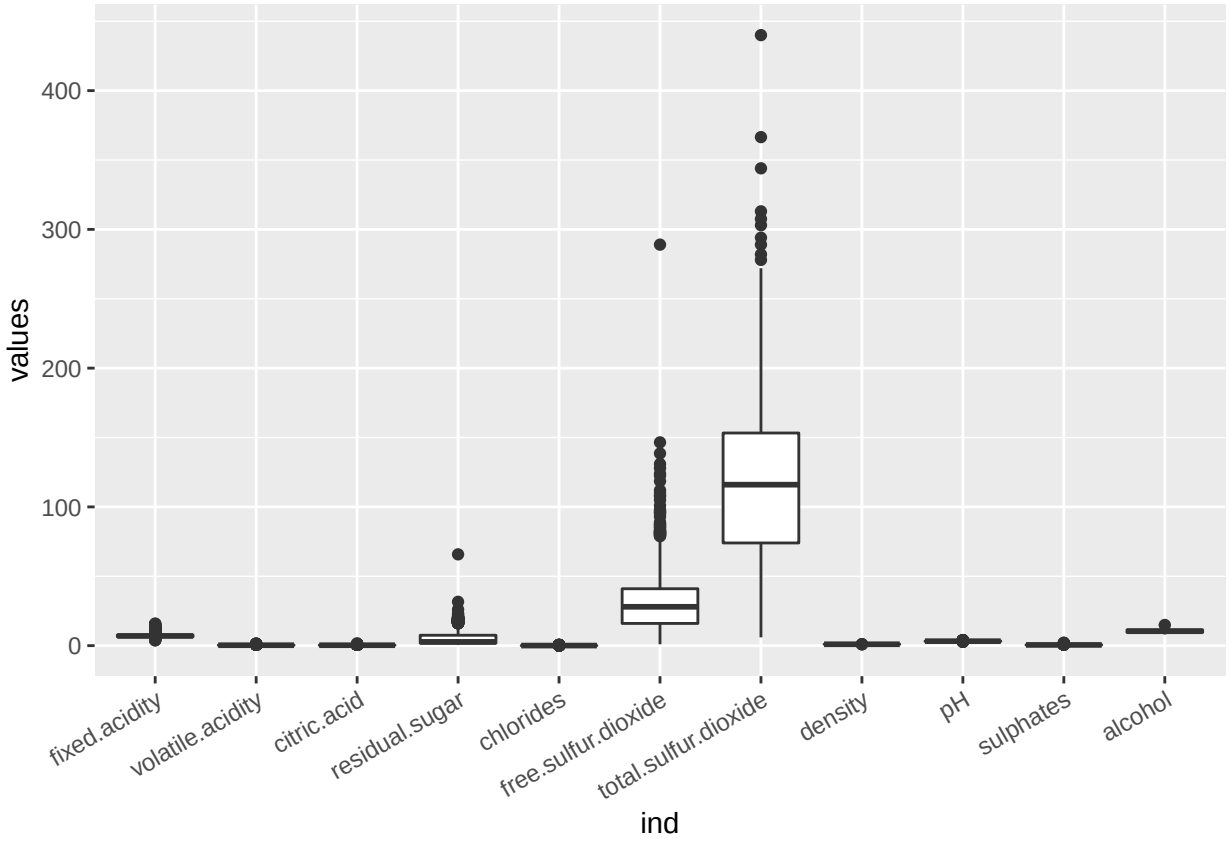
```
## [1] FALSE
```

En nuestro dataset no tenemos ningún caso de valores vacíos. Tenemos casos de valores ceros en la variable *citric.acid*, pero era esperado que algunos vinos no tendrían niguna cantidad de ácido cítrico. Si tuvieramos valores vacíos en nuestras variables numéricas, los podríamos reeplazarlos por la media de la variable), o predecir con valores que tengan la máxima probabilidad de ser correctos (por ejemplo *miss forest*).

3.2. Identificación y tratamiento de valores extremos.

Para examinar visualmente las distribuciones de las variables numéricas crearemos boxplots de cada una.

```
ggplot(stack(dataset[,0:11]), aes(x = ind, y = values)) +
    geom_boxplot() + theme(axis.text.x = element_text(angle = 30, hjust = 1))
```



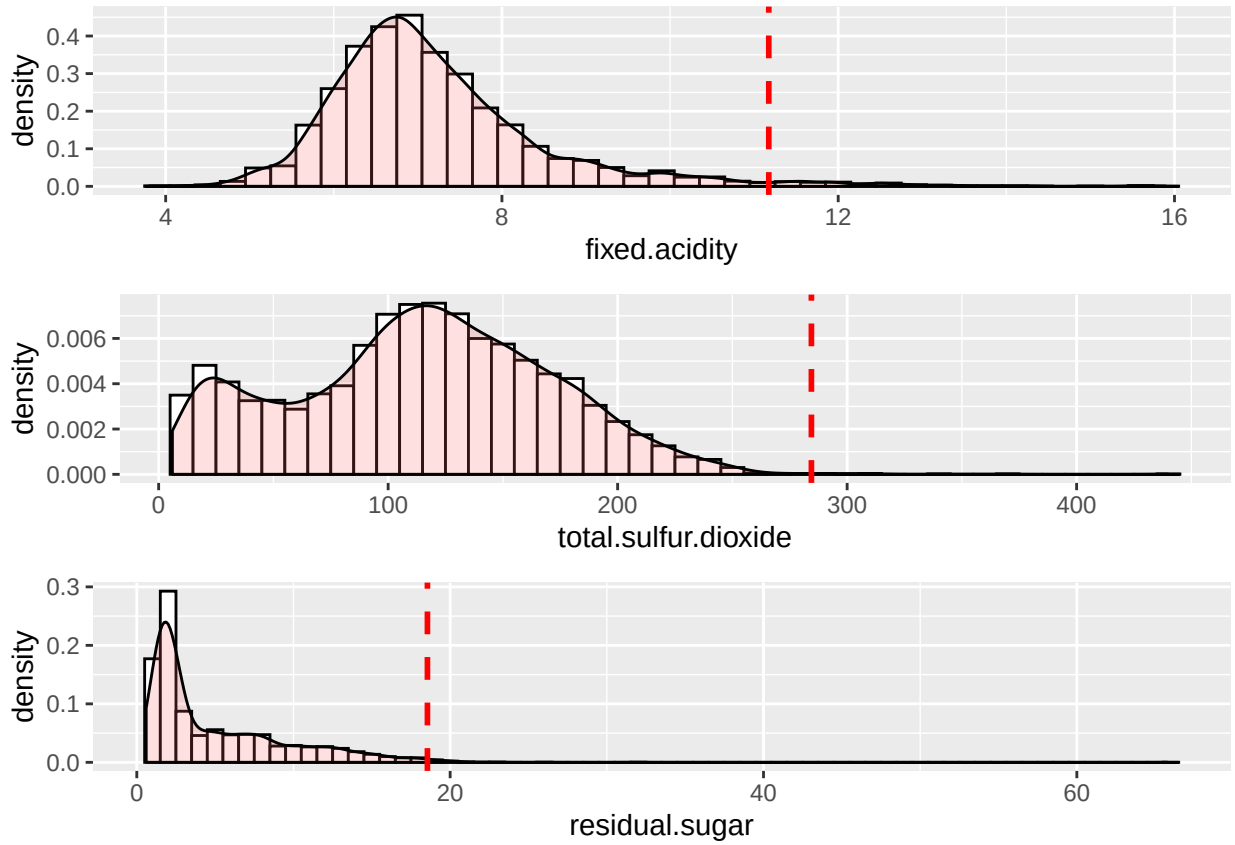
Casi todas las variables contienen valores extremos. En este caso hay muchas posibles explicaciones: es posible que sean errores en las mediciones de los vinos, puede ser que tenemos suposiciones incorrectas sobre nuestros datos, o otros errores. Investigaremos las distribuciones de algunas de las variables.

```
p1 <- ggplot(dataset, aes(x=fixed.acidity)) + geom_histogram(aes(y=..density..), binwidth=0.3, colour="

p2 <- ggplot(dataset, aes(x=total.sulfur.dioxide)) + geom_histogram(aes(y=..density..), binwidth=10, co

p3 <- ggplot(dataset, aes(x=residual.sugar)) + geom_histogram(aes(y=..density..), binwidth=1, colour="b

grid.arrange(p1, p2, p3, nrow = 3)
```



Comprobando visualmente se puede observar que los valores son de hecho extremos, luego procederemos a excluir sus registros del conjunto de datos.

```
soft_outlier_detection <- function(data) {
   lowerq = quantile(data, na.rm = TRUE)[2]
   upperq = quantile(data, na.rm = TRUE)[4]
   iqr = upperq - lowerq
   threshold_upper = (iqr) + upperq
   threshold_lower = lowerq - (iqr)
   data > threshold_upper | data <  threshold_lower
}
clean_dataset <- dataset[rowSums(sapply(dataset[,0:11], soft_outlier_detection), na.rm = TRUE) > 0, ]
nrow(clean_dataset)
```

```
## [1] 2116
```

4. Análisis de los datos.

4.1. Selección de los grupos de datos que se quieren analizar/comparar (planificación de los análisis a aplicar).

En nuestro caso nos gustaría analizar todos los datos que tenemos. Los análisis que nos gustaría aplicar son:

1. Si hay diferencia entre la calidad de vinos rojos y blancos
2. Si hay variables que tienen alta correlación con la calidad (separa por tipo de vino)
3. Si es posible crear una regresión lineal para explicar la relación entre calidad y las variables independientes.

4.2. Comprobación de la normalidad y homogeneidad de la varianza.

```
# usaremos el test Shapiro-Wilk
for (col in colnames(clean_dataset)[0:12]) {
  test_data = clean_dataset[,col]
  if (shapiro.test(test_data)$p < 0.05) {
    print(paste('La variable', col, 'no es normal'))
  } else {
    print(paste('La variable', col, 'es normal'))
  }
}
```

```
## [1] "La variable fixed.acidity no es normal"
## [1] "La variable volatile.acidity no es normal"
## [1] "La variable citric.acid no es normal"
## [1] "La variable residual.sugar no es normal"
## [1] "La variable chlorides no es normal"
## [1] "La variable free.sulfur.dioxide no es normal"
## [1] "La variable total.sulfur.dioxide no es normal"
## [1] "La variable density no es normal"
## [1] "La variable pH no es normal"
## [1] "La variable sulphates no es normal"
## [1] "La variable alcohol no es normal"
## [1] "La variable quality no es normal"
```

Según el test ninguna de las variables es normal. Como ninguna de las variables es normal, para comprobar la homocedasticidad usaremos el test de Fligner-Killeen.

```
if (fligner.test(clean_dataset[,0:12])$p.value < 0.05) {
  print(paste('Las variables', 'no presentan homocedasticidad'))
} else {
  print(paste('La variable', 'presentan homocedasticidad'))
}
```

```
## [1] "Las variables no presentan homocedasticidad"
```

4.3. Aplicación de pruebas estadísticas para comparar los grupos de datos. En función de los datos y el objetivo del estudio, aplicar pruebas de contraste de hipótesis, correlaciones, regresiones, etc. Aplicar al menos tres métodos de análisis diferentes.

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

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