M2.851 · Tipología y ciclo de vida de los datos · Practica2
2020-2 · Máster universitario en Ciencia de datos (Data science)
Estudios de Informática, Multimedia y Telecomunicación

Práctica 2: Limpieza y análisis de datos

En esta práctica se elabora un caso práctico orientado a aprender a identificar los datos relevantes para un proyecto analítico y usar las herramientas de integración, limpieza, validación y análisis de las mismas. Para hacer esta práctica tendréis que trabajar en grupos de 2 personas. Tendréis que entregar un solo archivo con el enlace Github (https://github.com (https://github.com)) donde se encuentren las soluciones incluyendo los nombres de los componentes del equipo. Podéis utilizar laWiki de Github para describir vuestro equipo y los diferentes archivos que corresponden a vuestra entrega. Cada miembro del equipo tendrá que contribuir con su usuario Github. Aunque no se trata del mismo enunciado, los siguientes ejemplos de ediciones anteriores os pueden servir como guía:

- Ejemplo:https://github.com/Bengis/nba-gap-cleaning (https://github.com/Bengis/nba-gap-cleaning)
- Ejemplo complejo (archivo adjunto).

Competencias

En esta práctica se desarrollan las siguientes competencias del Máster de Data Science:

- Capacidad de analizar un problema en el nivel de abstracción adecuado a cada situación y aplicar las habilidades y conocimientos adquiridos para abordarlo y resolverlo.
- Capacidad para aplicar las técnicas específicas de tratamiento de datos (integración, transformación, limpieza y validación) para su posterior análisis.

Objetivos

Los objetivos concretos de esta práctica son:

- Aprender a aplicar los conocimientos adquiridos y su capacidad de resolución de problemas en entornos nuevos o poco conocidos dentro de contextos más amplios o multidisciplinares.
- Saber identificar los datos relevantes y los tratamientos necesarios (integración, limpieza y validación) para llevar a cabo un proyecto analítico.
- Aprender a analizar los datos adecuadamente para abordar la información contenida en los datos.
- Identificar la mejor representación de los resultados para aportar conclusiones sobre el problema planteado en el proceso analítico.
- Actuar con los principios éticos y legales relacionados con la manipulación de datos en Tipología y ciclo de vida de los datosPráctica 2pág 2función del ámbito de aplicación.
- Desarrollar las habilidades de aprendizaje que les permitan continuar estudiando de un modo que tendrá que ser en gran medida autodirigido o autónomo.
- Desarrollar la capacidad de búsqueda, gestión y uso de información y recursos en el ámbito de la ciencia de datos.

Descripción de la Práctica a realizar

El objetivo de esta actividad será el tratamiento de un dataset, que puede ser el creado en la práctica 1 o bien cualquier dataset libre disponible en Kaggle (https://www.kaggle.com).Algunos (https://www.kaggle.com).Algunos) ejemplos de dataset con los que podéis trabajar son:

- Red Wine Quality (https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009 (https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009))
- Titanic: Machine Learning from Disaster (https://www.kaggle.com/c/titanic (https://www.kaggle.com/c/titanic))

  El último ejemplo corresponde a una competición activa de Kaggle de manera que, opcionalmente, podéis aprovechar el trabajo realizado durante la práctica para entrar en esta competición.

  Siguiendo las principales etapas de un proyecto analítico, las diferentes tareas a realizar (y justificar) son las siguientes:

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

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

3.Limpieza de los datos. - 3.1.¿Los datos contienen ceros o elementos vacíos? ¿Cómo gestionarías cada uno de estos casos? - 3.2.Identificación y tratamiento de valores extremos.

4.Análisis de los datos. - 4.1.Selección de los grupos de datos que se quieren analizar/comparar (planificación de los análisis a aplicar). - 4.2.Comprobación de la normalidad y homogeneidad de la varianza. - 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?

7.Código: Hay que adjuntar el código, preferiblemente en R, con el que se ha realizado la limpieza, análisis y representación de los datos. Si lo preferís, también podéis trabajar en Python.

Recursos

Los siguientes recursos son de utilidad para la realización de la práctica:

- Calvo M., SubiratsL., Pérez D. (2019). Introducción a la limpieza y análisis de los datos. Editorial UOC.
- Megan Squire (2015). Clean Data. Packt Publishing Ltd.
- Jiawei Han, Micheine Kamber, Jian Pei (2012). Data mining: concepts and techniques. Morgan Kaufmann.
- Jason W. Osborne (2010).Data Cleaning Basics: Best Practices in Dealing with Extreme Scores.Newborn and Infant Nursing Reviews; 10 (1): pp. 1527-3369.
- Peter Dalgaard (2008). Introductory statistics with R. Springer Science & Business Media.
- Wes McKinney (2012). Python for Data Analysis. O'Reilley Media, Inc.
- Tutorial de Githubhttps://guides.github.com/activities/hello-world (https://guides.github.com/activities/hello-world).

Criterios de valoración

Todos los apartados son obligatorios. La ponderación de los ejercicios es la siguiente:

- Los apartados 1, 2 y 6 valen 0,5 puntos.

- Los apartados 3, 5 y 7 valen 2 puntos.
- El apartado 4 vale 2,5 puntos.

Se valorará la idoneidad de las respuestas, que deberán ser claras y completas. Las diferentes etapas deberán justificarse y acompañarse del código correspondiente. También se valorará la síntesis y claridad, a través del uso de comentarios, del código resultante, así como la calidad de los datos finales analizados.

Formato y fecha de entrega

Durante la semana del 24 al 28 de mayo el grupo podrá entregar al profesor una entrega parcial opcional. Esta entrega parcial es muy recomendable para recibir asesoramiento sobre la práctica y verificar que la dirección tomada es la correcta. Se entregarán comentarios a los estudiantes que hayan efectuado la entrega parcial pero no contará para la nota de la práctica. En la entrega parcial los estudiantes deberán entregar por correo electrónico,al profesor encargado del aula, el enlace al repositorio Github con el que hayan avanzado.

En referente a la entrega final, hay que entregar un único fichero que contenga el enlace Github, el cual no se podrá modificar posteriormente a la fecha de entrega, donde haya:

1.Una Wiki con los nombres de los componentes del grupo y una descripción de los ficheros.

2.Un documento PDF con las respuestas a las preguntas y los nombres de los componentes del grupo. Además, al final del documento, deberá aparecer la siguiente tabla de contribuciones al trabajo, la cual debe firmar cada integrante del grupo con sus iniciales. Las iniciales representan la confirmación de que el integrante ha participado en dicho apartado. Todos los integrantes deben participar en cada apartado, por lo que, idealmente, los apartados deberían estar firmados por todos los integrantes.

3.Una carpeta con el código generado para analizar los datos.

4.El fichero CSV con los datos originales.

5.El fichero CSV con los datos finales analizados.

Este documento de entrega final de la Práctica 2 se debe entregar en el espacio de Entrega y Registro de AC del aula antes de las 23:59 del día8 de junio. No se aceptarán entregas fuera de plazo.

> **Nombre y apellidos:**

> ***Rosa María Miranda Castro:**

Solución

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

Dataset

Red Wine Quality

Autor

Creado por: Paulo Cortez (Univ. Minho), Antonio Cerdeira, Fernando Almeida, Telmo Matos and Jose Reis (CVRVV) @ 2009

Descripción del dataset

El conjunto de datos se ha obtenido desde la dirección https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009 (https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009), el cual contiene información de está relacionado con variantes del vino tinto portugués "Vinho Verde", dicho dataset contiene 1599 registros (observaciones) y 12 variables, las cuales tenemos a continuación:

Variables de entrada (basadas en pruebas fisicoquímicas):

1 - fixed acidity: acidez fija

2 - volatile acidity: acidez volátil

3 - citric acid: ácido cítrico

4 - residual sugar: azúcar residual

5 - chlorides: cloruros

6 - free sulfur dioxide: dióxido de azufre libre

7 - total sulfur dioxide: dióxido de azufre total

8 - density: densidad

9 - pH: pH

10 - sulphates: sulfatos

11 - alcohol: alcohol

Variable de salida (basada en datos sensoriales):

12 - quality calidad (puntuación entre 0 y 10)

Importancia

Lo importante de esta fuente es que contiene datos de las características que sirven para clasificar a un vino tinto dependiendo del rango de calidad.

A partir de este conjunto de datos la pregunta que se pretende resolver es que variables son importantes para predecir la calidad del vino basado en datos fisicoquímicos.

Cita

P. Cortez, A. Cerdeira, F. Almeida, T. Matos y J. Reis. Modelar las preferencias del vino mediante la extracción de datos a partir de las propiedades fisicoquímicas. En Decision Support Systems, Elsevier, 47 (4): 547-553, 2009.

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

Librerias

In [94]:

```
#Librerias
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
from sklearn import decomposition            # Módulo de reducción de dimensionalidad.
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import GradientBoostingClassifier, RandomForestClassifier
from sklearn.svm import SVC
from scipy import stats
from sklearn.metrics import accuracy_score,confusion_matrix
```

2.1 Carga del dataset

Se procede a a realizar la lectura del archvivo **winequality-red.csv**, el cual se encuentra en formato csv, el mismo que se ha obtenido desde la dirección https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009 (https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009), que contiene información de variantes del vino tinto portugués "Vinho Verde", este archivo será almacenado en un dataframe red_wine_data

In [95]:

```
# Carga Conjunto de Datos "Wine Recognition"
red_wine_data = pd.read_csv("winequality-red.csv")
# Presentación de una muestra de los datos del dataframe
red_wine_data.head()
```

Out[95]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcoh
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9

Se presenta la estructura y tipo de dato de cada variable

In [96]:

```
# Tipo de dato asignado a cada campo
red_wine_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1599 entries, 0 to 1598
Data columns (total 12 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   fixed acidity         1599 non-null   float64
 1   volatile acidity      1599 non-null   float64
 2   citric acid           1599 non-null   float64
 3   residual sugar        1599 non-null   float64
 4   chlorides             1599 non-null   float64
 5   free sulfur dioxide   1599 non-null   float64
 6   total sulfur dioxide  1599 non-null   float64
 7   density               1599 non-null   float64
 8   pH                    1599 non-null   float64
 9   sulphates             1599 non-null   float64
 10  alcohol               1599 non-null   float64
 11  quality               1599 non-null   int64  
dtypes: float64(11), int64(1)
memory usage: 150.0 KB
```

> **Respuesta:**
>
> ---
>
> - **El número de atributos que podrían ser usadas para predecir la respuesta "quality":** 11
> - **El número filas:** 1599
> - **No existen columnas vacias:**

2.2 Análisis estadístico básico.

Se visualiza los estadísticos básicos de las variables del dataframe.

In [97]:

```
1 red_wine_data.describe()
```

Out[97]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total su dio
count	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000
mean	8.319637	0.527821	0.270976	2.538806	0.087467	15.874922	46.467
std	1.741096	0.179060	0.194801	1.409928	0.047065	10.460157	32.895
min	4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	6.000
25%	7.100000	0.390000	0.090000	1.900000	0.070000	7.000000	22.000
50%	7.900000	0.520000	0.260000	2.200000	0.079000	14.000000	38.000
75%	9.200000	0.640000	0.420000	2.600000	0.090000	21.000000	62.000
max	15.900000	1.580000	1.000000	15.500000	0.611000	72.000000	289.000

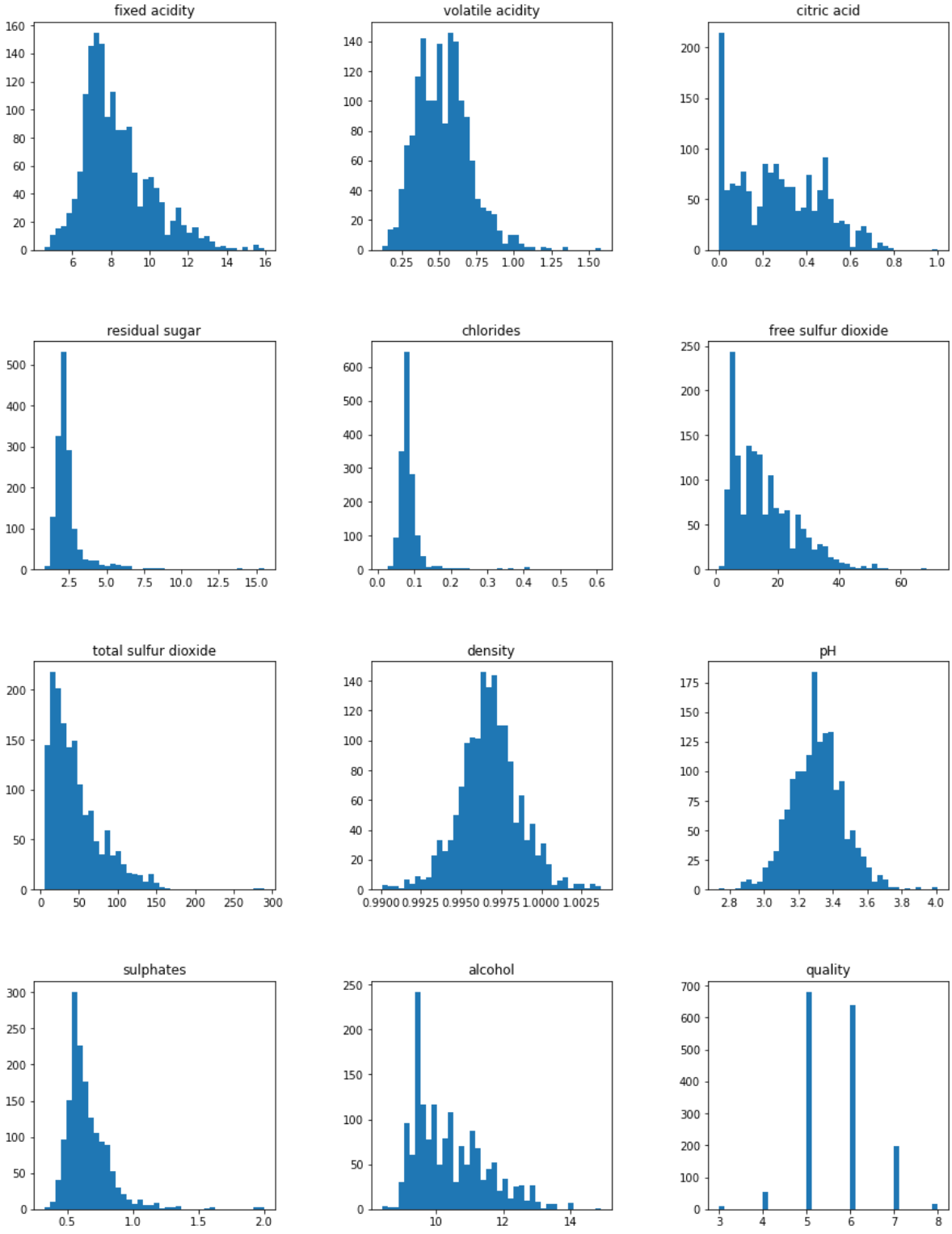
Se visualiza las gráficas de distribución de las variables del comjunto de datos.

In [98]:

```
fig = plt.figure(figsize=(15,20))
fig.subplots_adjust(hspace=0.4, wspace=0.4)
i=1
for variable in red_wine_data:
    ax = fig.add_subplot(4, 3, i)
    ax.hist(red_wine_data[variable],bins=40)
    ax.set_title(variable)
    i=i+1

plt.show()

```



Se puede observar que la variable **quality**, toma valores entre 3 y 8, así mismo se puede evidenciar que en las variables **residual sugar** y **chlorides** tanto en sus estadísticos básico como en los diagramas, que existe mucha dispersión en sus datos

2.3 selección de los datos de interés a analizar.

Para poder cumplir el objetivo del análisis el mismo que consiste en determinar un modelo que permita clasificar los vinos en función de sus datos fisicoquímicos. se debera hacer:

Para las variables independientes se consideraran a los atributos que estan basadas en las pruebas de laboratorio fisicoquímicas las que por lo general sirven para caracterizar al vino, a dichas variables se les almacenará en un dataframe **X**, mientras que en un dataframe **y** se procederá a almacenar la variable independiente u objetivo, dicha variable resulta de las pruebas sensoriales realizada por los expertos humanos para determinar su calidad. Esta última variable resultará de la discretización de la variable quality en dos grupos: cuando quality esta entre 7 y 8 será categorizada como **vino de alta calidad (1)**, mientra que si no pertenece a este grupo será categorizada como **vino de baja calidad (0)**.

In [99]:

```
X=red_wine_data.drop(columns=['quality']) #Variables Independientes
y=np.where(red_wine_data['quality']>=7,1,0) #alta calidad (1) baja calidad(0) #Variable
red_wine_data['y']=y
red_wine_data.tail()

```

Out[99]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	al
1594	6.2	0.600	0.08	2.0	0.090	32.0	44.0	0.99490	3.45	0.58	
1595	5.9	0.550	0.10	2.2	0.062	39.0	51.0	0.99512	3.52	0.76	
1596	6.3	0.510	0.13	2.3	0.076	29.0	40.0	0.99574	3.42	0.75	
1597	5.9	0.645	0.12	2.0	0.075	32.0	44.0	0.99547	3.57	0.71	
1598	6.0	0.310	0.47	3.6	0.067	18.0	42.0	0.99549	3.39	0.66	

3.Limpieza de los datos

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

Se procederá a determinar si los campos del dataframe contiene datos vacios o ceros y posterior gestionarlos en el caso de hallarlos y hacer el tratamiento de outlier.

Para poder entender si existen valores su utilizará la función isna().sum()

In [100]:

```
# identificación de valores nulos
red_wine_data.isna().sum()
```

Out[100]:

```
fixed acidity           0
volatile acidity        0
citric acid             0
residual sugar          0
chlorides               0
free sulfur dioxide     0
total sulfur dioxide    0
density                 0
pH                      0
sulphates               0
alcohol                 0
quality                 0
y                       0
dtype: int64
```

In [101]:

```
# identificación de valores ceros
np.sum(red_wine_data==0)
```

Out[101]:

```
fixed acidity             0
volatile acidity          0
citric acid             132
residual sugar            0
chlorides                 0
free sulfur dioxide       0
total sulfur dioxide      0
density                   0
pH                        0
sulphates                 0
alcohol                   0
quality                   0
y                      1382
dtype: int64
```

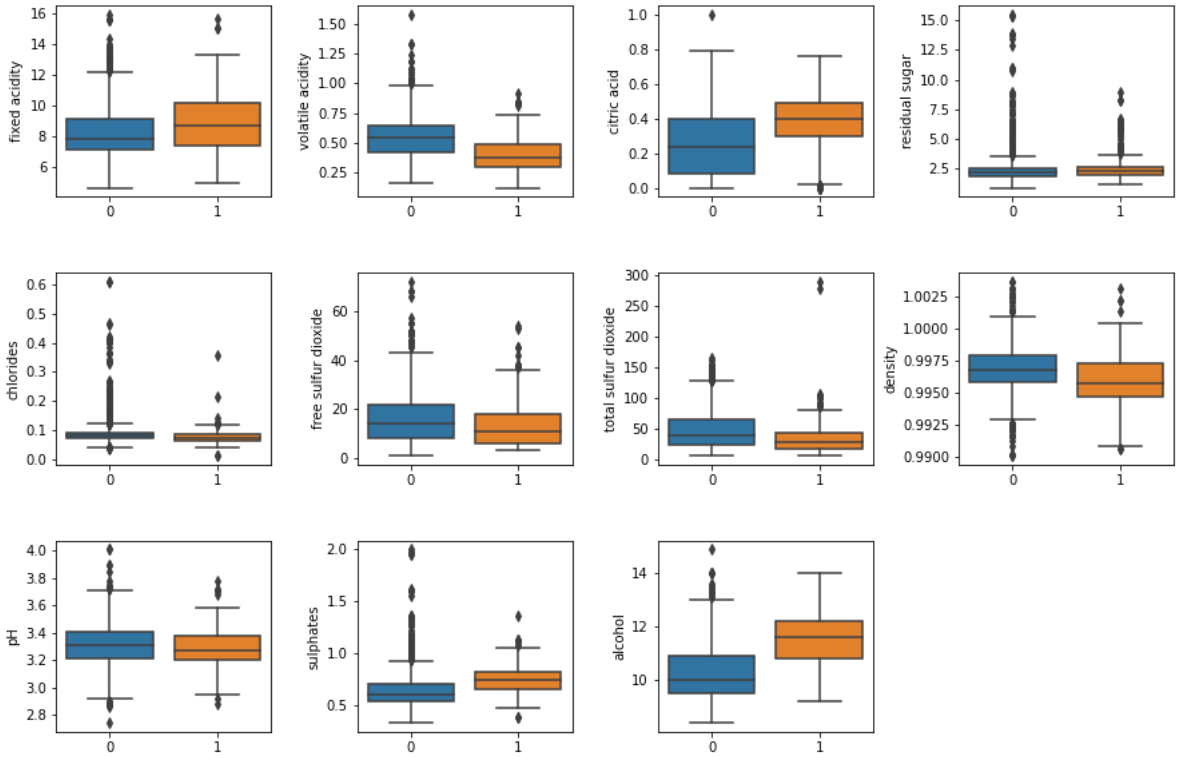
> **Respuesta:**
>
> - **Los datos no contiene valores o elementos vacios, en ninguna variable del conjunto de datos, en cambio la variable citric acid presenta 132 valores en cero, pero revisando sus estadísticas básicas determinamos que esta variable se encuentra en una escala de 0 a 1 con media de 0.27, por lo que se determina que son valores reales de los datos.**
> - **Si existiera estos datos se deberian imputar, utilizaríamos un sencillo método de imputación, en este caso imputando por el valor de la mediana en lugar del la media, ya que la media introduce valores atípicos o ruido en el análisis.**

3.2.Identificación y tratamiento de valores extremos.

Los valores extremos u outliers son aquellos que parecen no ser congruentes, si los comparamos con el resto de los datos. Para identificarlos, en nuestro caso haremos uso de la representación por medio de diagrama de caja por cada variable en función de la calidad del vino y así poder determinar qué valores distan mucho del rango intercuartíl.

In [102]:

```
fig = plt.figure(figsize=(15,10))
fig.subplots_adjust(hspace=0.4, wspace=0.4)
i=1
for variable in X:
    ax = fig.add_subplot(3, 4, i)
    sns.boxplot(data=X,x=y,y=variable,ax=ax)
    i=i+1
plt.show()
```



Respuesta:

Se observa un gran número de observaciones que se podrian considerar como valores extremos especialmente en las variables **residual sugar** y **chlorides**, esto gracias a los gráficos de cajas, sin embargo al revisar sus estadísticos básicos se evidencia que existe un total del 25% de sus registros dentro de esta clasificación, por tal motivo se concluye que esos datos no son atípicos y es mas se debe tener en cuenta que estos datos corresponden a los registros de las pruebas fisicoquímicas, lo cual no es tan problable que existan errores.

In [103]:

```
red_wine_data.to_csv('red_wine_data_output.csv')
```

4. Análisis de los Datos

4.1 Selección de los grupos de Datos

Dado el objetivo del análisis el mismo que consiste en determinar un modelo que permita clasificar los vinos en función de sus datos fisicoquímicos, se procederá a realizar algunas pruebas para poder comparar los grupos de datos que se quieren analizar.

Seleccionamos los grupos definidos inicialmente cuando quality esta entre 7 y 8 será categorizada como **vino de alta calidad (1)**, mientra que si no pertenece a este grupo será categorizada como **vino de baja calidad (0)**.

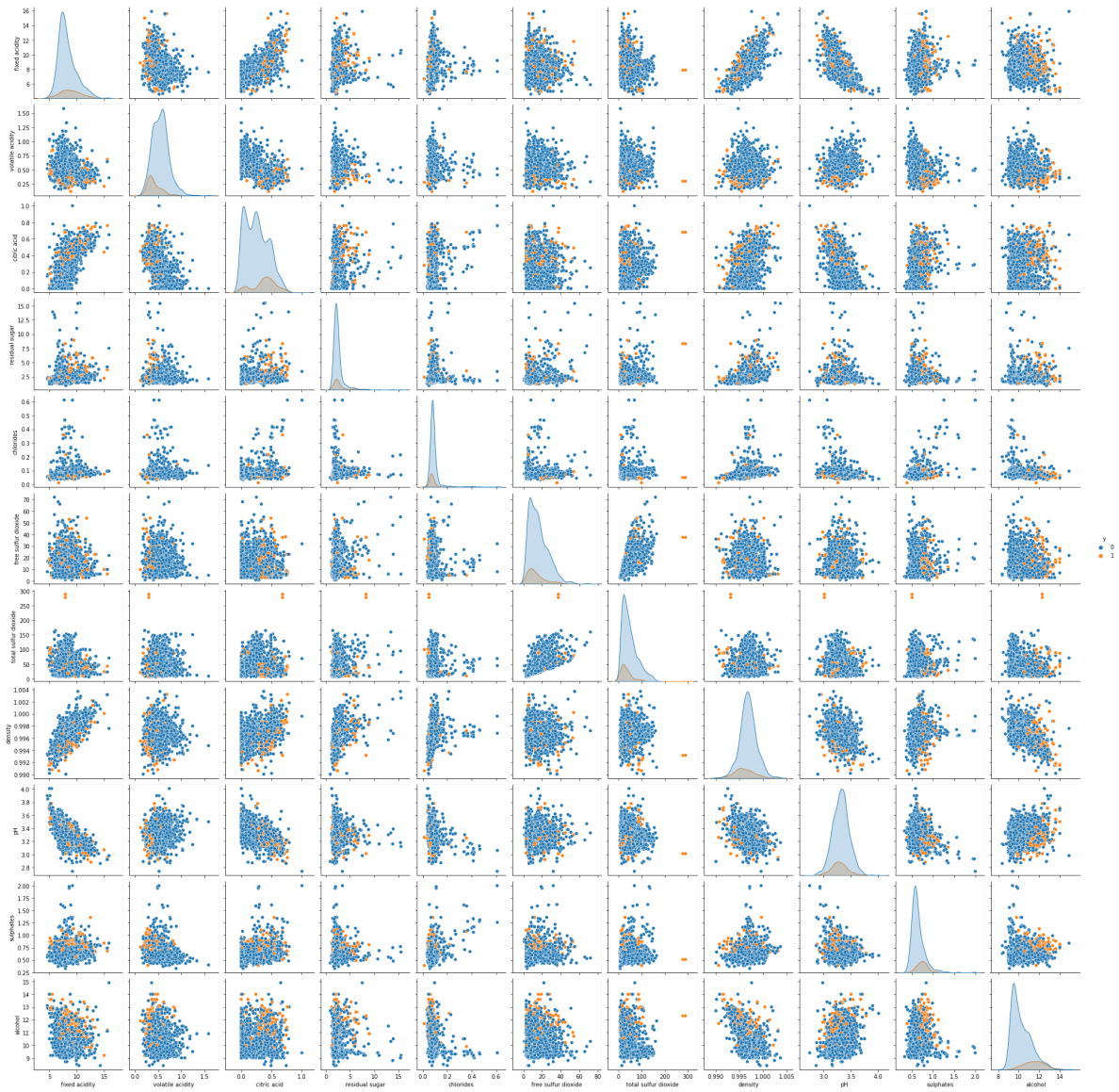
In [104]:

```
alta_calidad=red_wine_data[red_wine_data['y']== 1]
baja_calidad=red_wine_data[red_wine_data['y']== 0]
```

Con los grupos definidos visualizamos gráficas de distribución y de puntos para todo el conjunto de datos con relación a la calidad de los vinos.

In [105]:

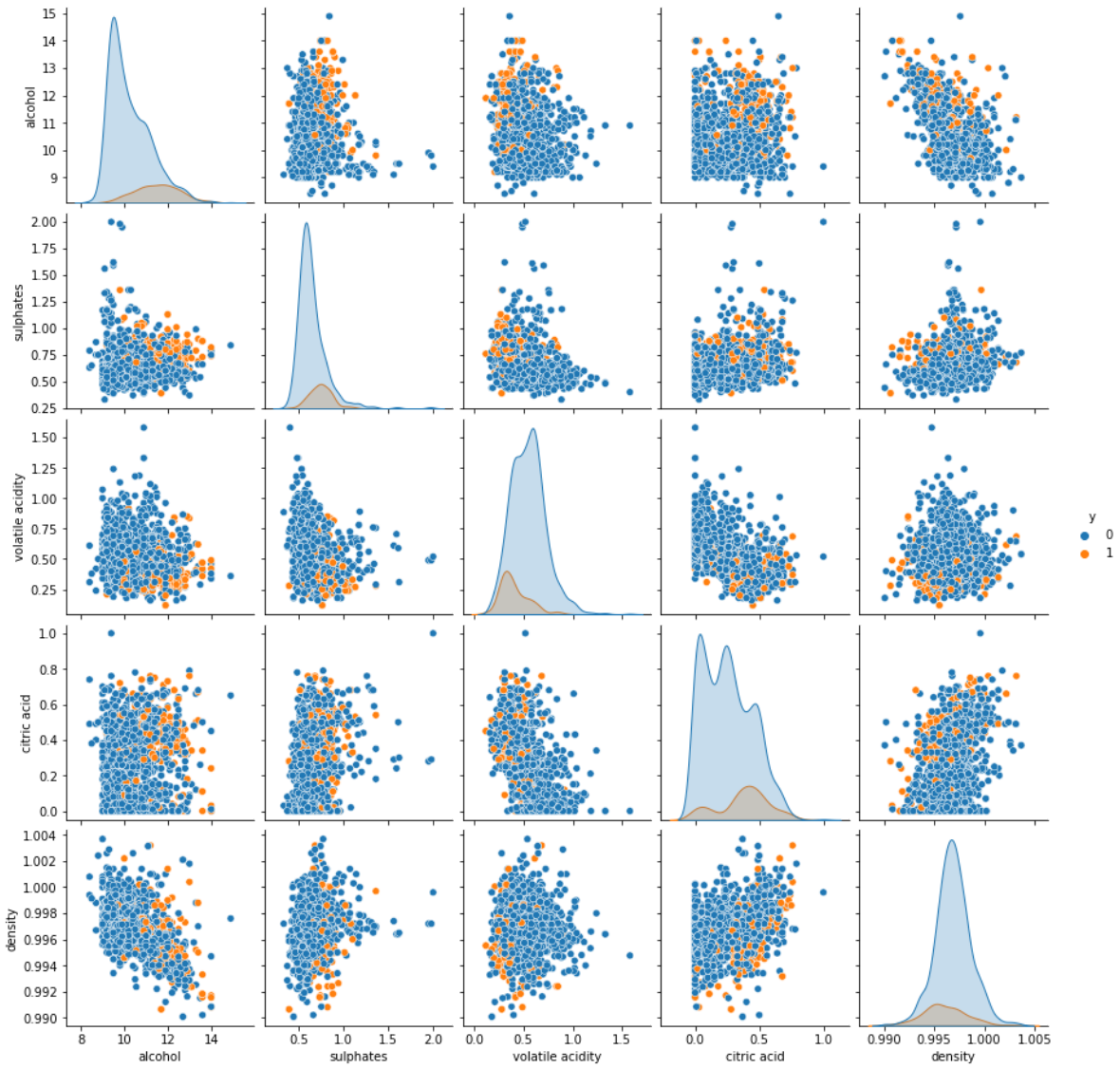
```
sns.pairplot(red_wine_data.drop(columns=['quality']),hue='y',markers='o')
plt.show()
```



Ampliamos la visualización para las variables que pueden permitirmos clasificar de mejor manera la calidad de los vinos.

In [106]:

```
1 sns.pairplot(red_wine_data,vars=['alcohol','sulphates','volatile acidity','citric acid'
2 plt.show()
```



4.2 Normalidad y Homogeneidad de la Varianza de los Datos

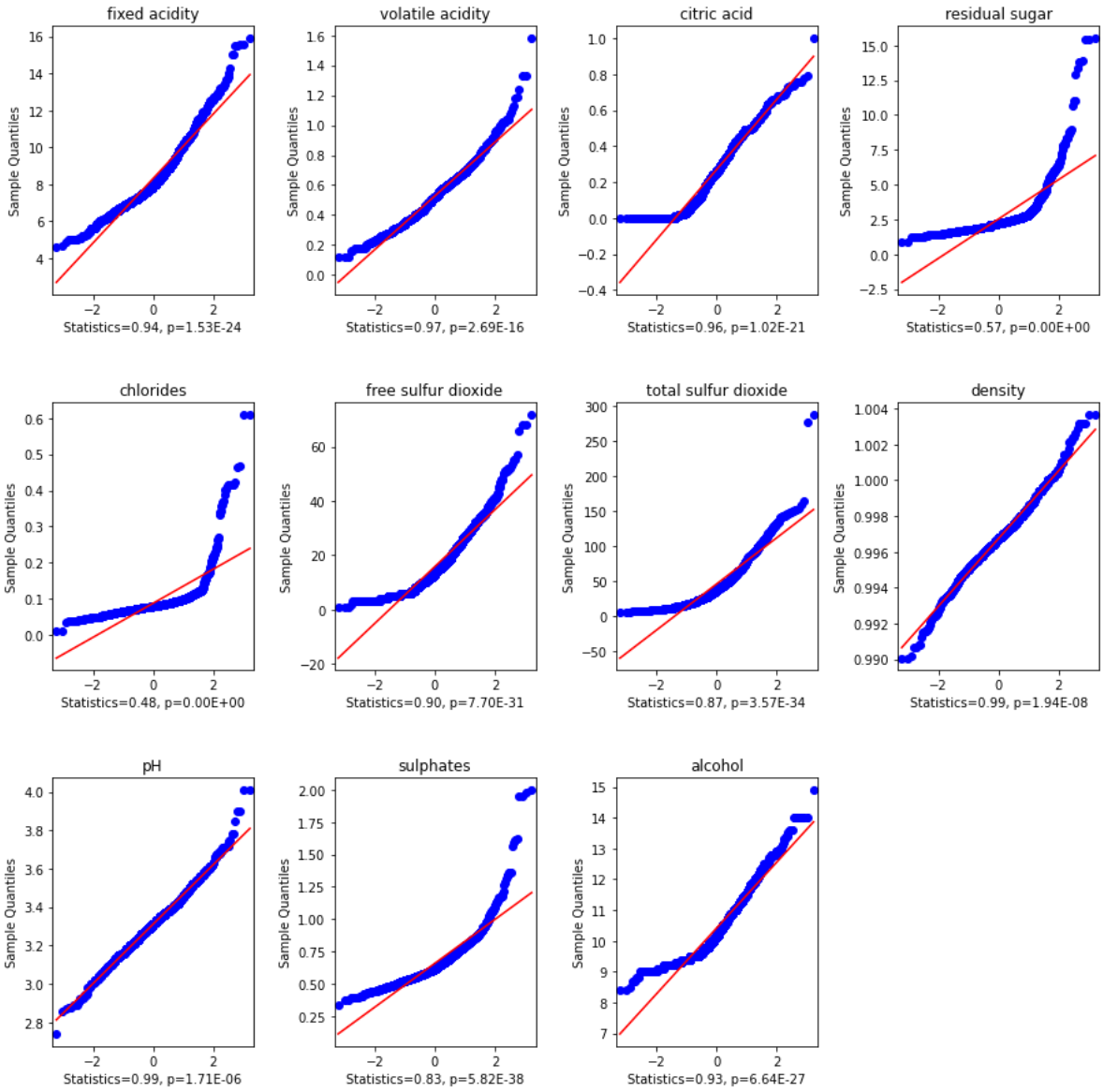
4.2.1 Normalidad

Para la comprobación de que los valores que toman nuestras variables cuantitativas provienen de una población distribuida normalmente, utilizaremos la prueba de normalidad de Shapiro_Wilk. Siendo la hipótesis nula que la población está distribuida normalmente, si el p-valor es menor a alfa= 0,05 (nivel de significancia) entonces la hipótesis nula es rechazada (se concluye que los datos no vienen de una distribución normal). Si el p-valor es mayor a alfa= 0,05, se concluye que no se puede rechazar dicha hipótesis.

In [107]:

```
from statsmodels.graphics.gofplots import qqplot
from scipy.stats import shapiro

fig = plt.figure(figsize=(15,15))
fig.subplots_adjust(hspace=0.4, wspace=0.4)
i=1
for variable in list(X):
    ax = fig.add_subplot(3, 4, i)
    ax.set_title(variable)
    qqplot(X[variable], line='s',ax=ax) # q-q plot
    stat, p = shapiro(X[variable])
    ax.set_xlabel('Statistics=%.2f, p=%.2E' % (stat, p))
    i=i+1
plt.show()
```



Respuesta:

> Analizando los p-valor de todas las variables aplicando el test de Shapiro_Wilk se observa que se obtienen valores inferiores a alfa= 0,05 (nivel de significancia) entonces la hipótesis nula es rechazada por lo que se concluye que los datos no vienen de una distribución normal.

4.2.2 Homogeneidad de la Varianza de los Datos

Se realizará el test de homogeneidad de la varianza de todas las variables dependientes con relación a la calidad del vino. Para esto utilizatremos la prueba de Levene, como las variables no siguen una distribución normal se utilizará la mediana como métrica de tendencia central, con esto, con un p-valor superior a 0,05, aceptamos la hipótesis de que las varianzas de las muestras son homogéneas.

In [108]:

```
for variable in list(X):
    statistic,pvalue = stats.levene(X.loc[baja_calidad.index,variable], X.loc[alta_cali
    print(variable+' : Statistics=%.2f, p-value=%.2f' % (statistic,pvalue))
```

```
fixed acidity : Statistics=13.12, p-value=0.00
volatile acidity : Statistics=12.95, p-value=0.00
citric acid : Statistics=1.06, p-value=0.30
residual sugar : Statistics=2.29, p-value=0.13
chlorides : Statistics=1.81, p-value=0.18
free sulfur dioxide : Statistics=1.75, p-value=0.19
total sulfur dioxide : Statistics=15.20, p-value=0.00
density : Statistics=17.03, p-value=0.00
pH : Statistics=0.06, p-value=0.81
sulphates : Statistics=0.71, p-value=0.40
alcohol : Statistics=1.45, p-value=0.23
```

> **Respuesta:**
>
> **- Se observa en el siguiente test que la hipótesis nula consiste en que ambas varianzas son iguales es decir la Homogeneidad de la Varianza con relación a la calidad del vino, las variables que cumplen esta condición p-valor superior a 0,05 son:**
>
> - **residual sugar: p-value=0.41**
> - **citric acid : p-value=0.30**
> - **chlorides: p-value=0.12**
> - **free sulfur dioxide: p-value=0.14**
> - **pH: p-value=0.91**
> - **sulphates: p-value=0.95**
> - **alcohol : p-value=0.23**

4.3 Pruebas Estadisticas

Se procederá a aplicar pruebas estadísticas para comparar los grupos de datos para conocer cuales de las variables ejercen una mayor influencia sobre la calidad del vino

4.3.1 Test igualdad de medianas

Procedemos a realizar el test de igualdad de las medianas para estimar que variables son útiles para clasificar la calidad del vino.

In [109]:

```
for variable,var in zip(list(X),varianza):
    statistic,pvalue = stats.kruskal(X.loc[baja_calidad.index,variable], X.loc[alta_cal
    print(variable+' : Statistics=%.2f, p-value=%.2f' % (statistic,pvalue))
```

```
fixed acidity : Statistics=24.79, p-value=0.00
volatile acidity : Statistics=135.15, p-value=0.00
citric acid : Statistics=71.50, p-value=0.00
residual sugar : Statistics=5.82, p-value=0.02
chlorides : Statistics=39.36, p-value=0.00
free sulfur dioxide : Statistics=12.68, p-value=0.00
total sulfur dioxide : Statistics=47.25, p-value=0.00
density : Statistics=36.51, p-value=0.00
pH : Statistics=7.08, p-value=0.01
sulphates : Statistics=128.39, p-value=0.00
alcohol : Statistics=234.32, p-value=0.00
```

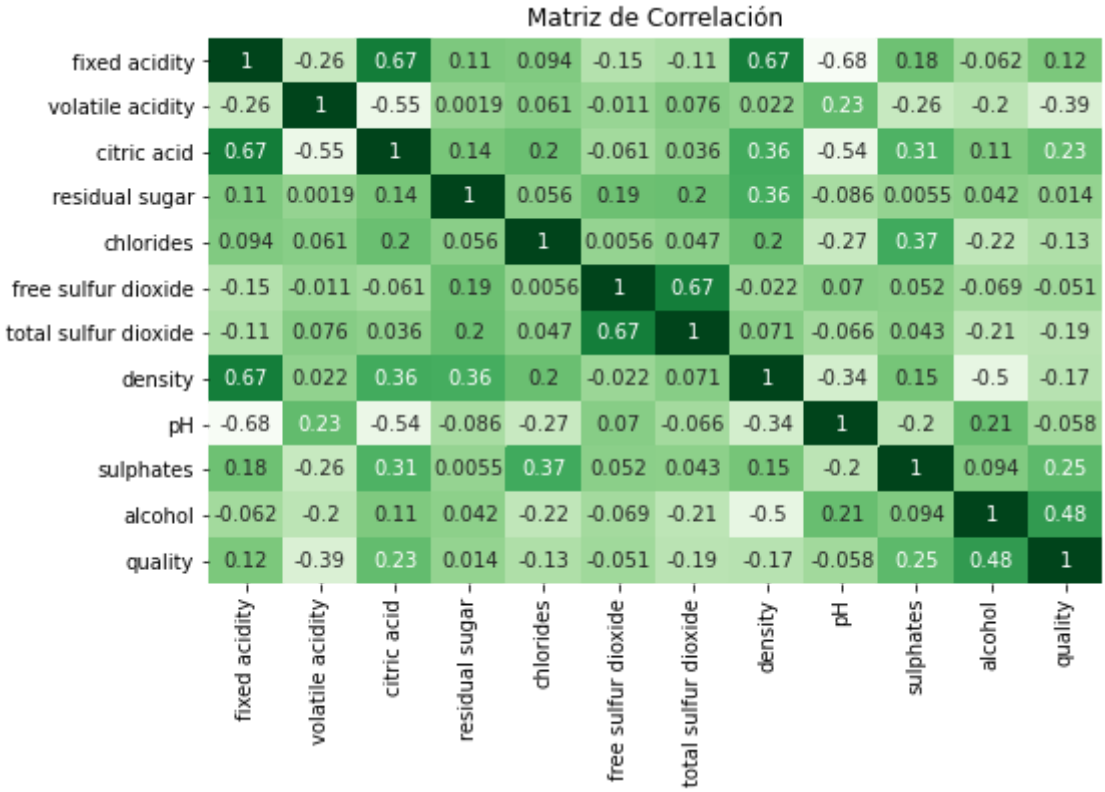
> **Respuesta:**
>
> **- En base al test realizado se observa que ninguna variable tiene medianas iguales entre los dos grupos de análisis por cuanto sus p-value son menores al nivel de significancia del 0.05, por lo cual pueden ser usadas para clasificar la calidad del vino.**

4.3.2 Correlación de Variables

Procedemos a realizar un análisis de correlación entre las distintas variables para determinar cuáles de ellas ejercen una mayor influencia sobre la calidad del vino. Para ello, se utilizará una matriz de correlación.

In [110]:

```
fig = plt.figure(figsize=(8,5))
sns.heatmap(red_wine_data.drop(columns=['y']).corr(),cmap='Greens',annot=True,cbar=Fals
plt.title('Matriz de Correlación')
plt.show()
```



> **Respuesta:**
>
> De la matriz de correlación presentada se puede concluir que la variable que tiene la mayor correlación con respecto a la variable quality es: alcohol=0.48, siendo es una correlación media.

4.3.3 Modelos de Clasificación

4.3.3.1 Conjuntos de datos tanto de entrenamiento (train) y test (test).

Procedemos a dividir nuestro conjunto de datos en conjuntos de entrenamiento (70%) y test (30%), además de escalar las variables.

In [111]:

```
X_train, X_test, y_train, y_test = train_test_split(X,y,stratify=y,test_size=0.3, rando

scaler = StandardScaler()
scaler.fit(X_train)

X_train_scaler = pd.DataFrame(scaler.transform(X_train), columns = X.columns)
X_test_scaler = pd.DataFrame(scaler.transform(X_test), columns = X.columns)
```

4.3.3.2 Reducción de dimensionalidad

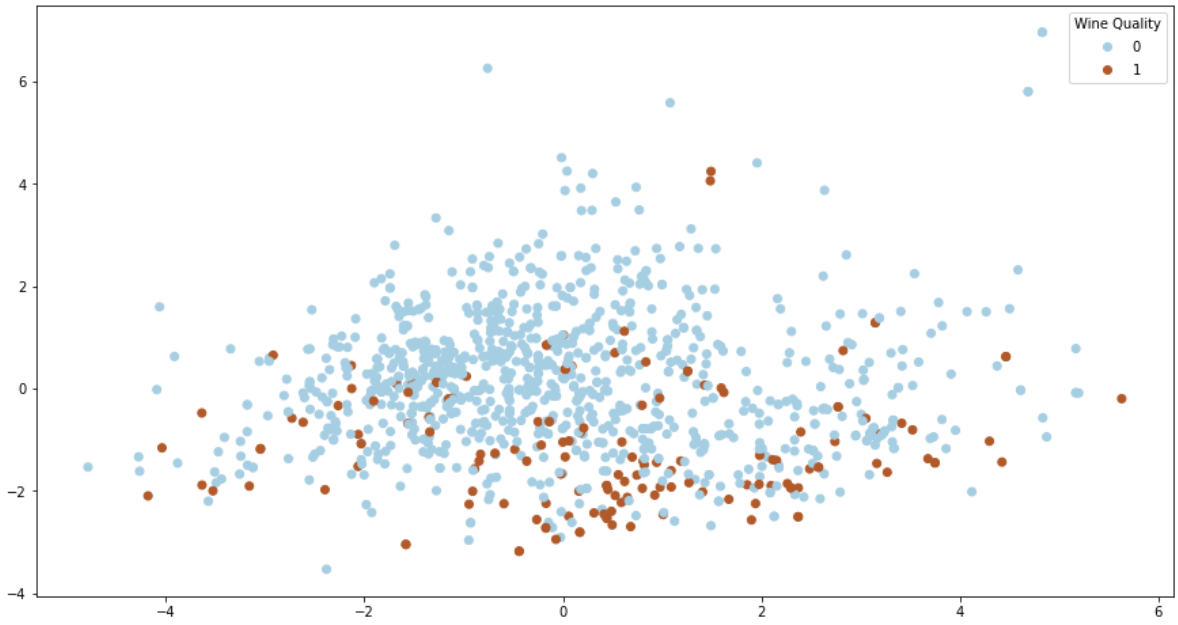
Utilizando el metodo del PCA, se procederá a reducir en dos dimensiones, para poder visualizar el conjunto de datos.

In [112]:

```
f,ax=plt.subplots(figsize=(15,8))
pca=decomposition.PCA(n_components=2)
pca.fit(X_train_scaler) # entrenas con el fit
X_train_pca=pca.transform(X_train_scaler) #transformar a X_train en dos dim
X_test_pca=pca.transform(X_test_scaler) #transformar a X_test en dos dim

scatter=ax.scatter(x=X_train_pca[:,0],y=X_train_pca[:,1],c=y_train, cmap='Paired')
ax.legend(*scatter.legend_elements(num=1),loc="upper right", title="Wine Quality")
plt.show()

n_pcs= pca.components_.shape[0]
# get the index of the most important feature on EACH component
# LIST COMPREHENSION HERE
most_important = [np.abs(pca.components_[i]).argmax() for i in range(n_pcs)]
initial_feature_names = X_train_scaler.columns # get the names
most_important_names = [initial_feature_names[most_important[i]] for i in range(n_pcs)]
print("Importancia de Variables")
dic = {'PC{}'.format(i+1): most_important_names[i] for i in range(n_pcs)}
# build the dataframe
df = pd.DataFrame(sorted(dic.items()))
df.head()
```




```
Importancia de Variables
```

Out[112]:

	0	1
0	PC1	fixed acidity
1	PC2	total sulfur dioxide

4.3.3.3 Implementación de modelos

4.3.3.3.1 SVC

Calcularemos el valor óptimo de los hiperparámetros C y gamma, utilizando una búsqueda de rejilla con validación cruzada con folds=4, para encontrar dichos valores.

In [113]:

```
param_grid = {'C':[  0.01, 0.1, 1, 10, 50, 100 ,200],
              'gamma':[0.001, 0.01, 0.1, 1 , 10]}
svm = SVC()
svm_cv = GridSearchCV(svm, param_grid, cv=4)
svm_cv.fit(X_train_scaler, y_train)

print("El valor óptimo de K es:",svm_cv.best_params_)
print("Accurancy del valor óptimo  es  {0:.2f} %".format(svm_cv.best_score_*100))
```

```
El valor óptimo de K es: {'C': 10, 'gamma': 1}
Accurancy del valor óptimo  es  89.90 %
```

> **Respuesta:** Mediante el uso de una búsqueda de rejilla con validación cruzada para encontrar los valores óptimos de C y gamma, se generó que los valores óptimos para este caso son de: C=10 y gamma=1, y un accurancy de 89.90%.

Con la mejor combinación de hiperparámetros encontradados, se entrena un clasificador `SVC` (con *train*), también calcularemos el *accuracy* del modelo obtenido sobre test y la matriz de confusión.

In [114]:

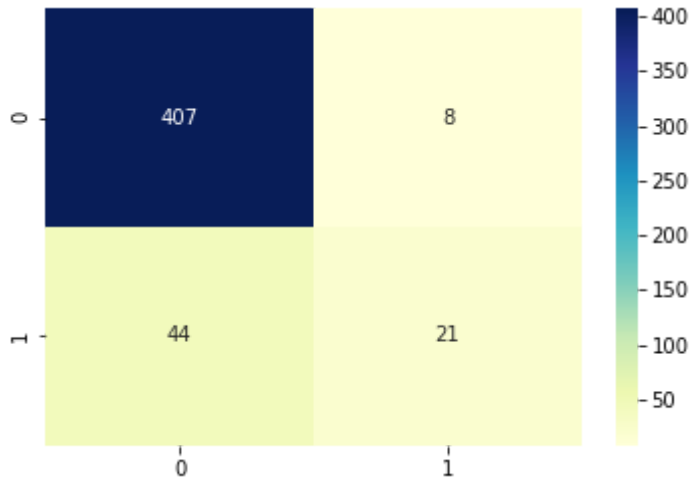
```
svm = SVC(C=10,gamma=1)
svm.fit(X_train_scaler,y_train)
y_test_prediction = svm.predict(X_test_scaler)
y_train_prediction = svm.predict(X_train_scaler)

print("Accurancy sobre test es  {0:.2f} %".format(accuracy_score(y_test_prediction,y_te

matriz=confusion_matrix(y_test,y_test_prediction)
print("Matriz de Confusión")
sns.heatmap(matriz, annot=True,cmap="YlGnBu",fmt='d')
plt.show()

```

```
Accurancy sobre test es  89.17 %
Matriz de Confusión
```



Respuesta: Las predicciones obtenidas con los datos de test son: Accurancy de **89.17%**.

4.3.3.3.2 Random Forest

Calcularemos el valor óptimo de los hiperparámetros n_estimators y max_depth, utilizando una búsqueda de rejilla con validación cruzada (GridSearchCV) para encontrar dichos valores.

X_train_scaler , y_train /*variables escalares*/

In [115]:

```
param_grid_fr = {'n_estimators':[50, 100 , 200],
              'max_depth':[8,13]}
rfm = RandomForestClassifier(random_state=123)
rfm_cv = GridSearchCV(rfm, param_grid_fr, cv=4)
rfm_cv.fit(X_train_scaler, y_train)


print("El valor óptimo de K es:",rfm_cv.best_params_)
print("Accurancy del valor óptimo  es  {0:.2f} %".format(rfm_cv.best_score_*100))

```

```
El valor óptimo de K es: {'max_depth': 13, 'n_estimators': 200}
Accurancy del valor óptimo  es  90.97 %
```

> **Respuesta:** Mediante el uso de una búsqueda de rejilla con validación cruzada para encontrar los valores óptimos de max_depth y n_estimators,cuyos valores para este caso son de:max_depth= 13, n_estimators=200, con un valor de accurancy de 90.97%, relativamente mejor al modelo anterior y con el cual se procederá a estimar la precisión en el conjunto de datos de Test.

Representación de los resultados a partir de tablas y gráficas.

Con la mejor combinación de hiperparámetros encontradados, se entrena un clasificador RandomForestClassifier (con *train*), también calcularemos el *accuracy* del modelo obtenido sobre test y la matriz de confusión.

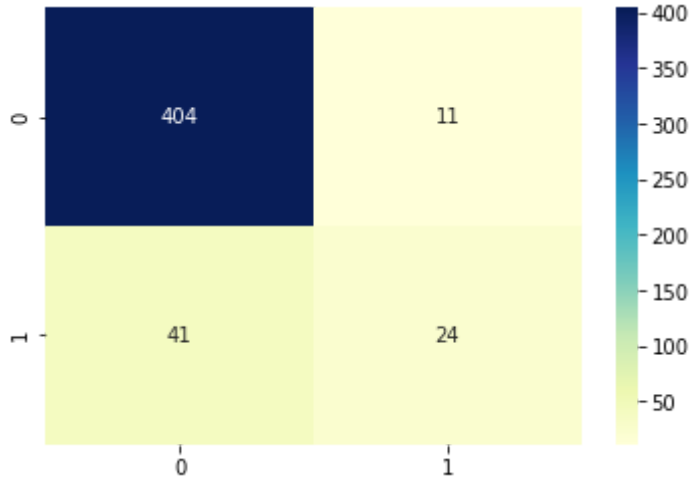
In [116]:

```
rf= RandomForestClassifier(n_estimators=200,max_depth= 13, random_state=123)
rf.fit(X_train_scaler,y_train)
y_pred_test=rf.predict(X_test_scaler)
y_pred_train=rf.predict(X_train_scaler)

print("Accurancy sobre test es  {0:.2f} %".format(accuracy_score(y_pred_test,y_test)*10

matriz=confusion_matrix(y_test,y_pred_test)
print("Matriz de Confusión")
sns.heatmap(matriz, annot=True,cmap="YlGnBu",fmt='d')
plt.show()
```

```
Accurancy sobre test es  89.17 %
Matriz de Confusión
```

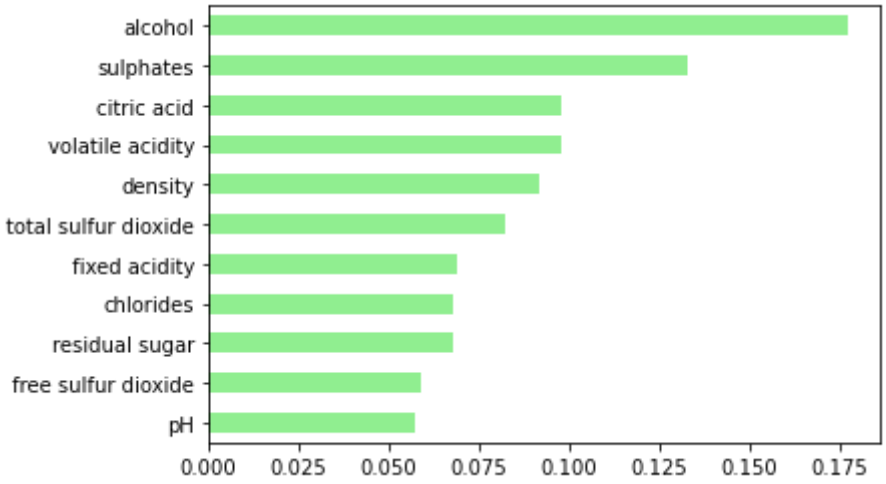


> **Respuesta:** Las predicciones obtenidas son los datos de test para este modelo son: Accurancy de **89.17 %**,

Para conocer cuáles han sido las variables más decisivas a la hora de generar el modelo, utilizamos la característica del modelo **feature importance**.

In [117]:

```
importances_rf = pd.Series(rf.feature_importances_, index = X_train_scaler.columns)
# Sort importances_rf
sorted_importances_rf = importances_rf.sort_values()
# Make a horizontal bar plot
sorted_importances_rf.plot(kind='barh', color='lightgreen'); plt.show()
#cuales son las variables mas relevantes para la clasificacion
```



> **Respuesta:** De la gráfica anterior con feature importance se puede concluir que la variable mas decisiva para la clasificación de la calidad del vino, es el alcohol seguida de la variable sulphates.

Conclusiones.

De todo el proceso realizado se puede concluir que los datos no vienen de una de una distribución normal, asi tambien de la matriz de correlación presentada se puede concluir que la variable que tiene una correlación media con respecto a la variable objetivo es: alcohol=0.41, apalancada tambien al momento de implementar el feature importance en el cual gráficamente esta variable es muy decisiva para la clasificación del la calidad del vino.

Con respecto a valor nulos o atípicos se observa un gran número de observaciones que se podrian considerar como valores extremos especialmente en las variables residual sugar y chlorides, esto gracias a los gráficos de cajas, sin embargo al revisar sus estadísticos básicos se evidencia que existe un total del 25% de sus registros dentro de esta clasificación, por tal motivo se concluye que esos datos no son atípicos y es mas se debe tener en cuenta que estos datos corresponden a los registros de las pruebas fisicoquímicas, lo cual no es tan problable que existan errores.

Al implementar un modelo el que nos permita clasificar de la mejor manera la calidad de los vinos se ha considerado el modelo **RandomForestClassifier**, el mismo que es tiene una precisión global de 89.17%, el cual es relativamente bueno, siendo mucho mas eficiente al momento de clasificar a los de baja calidad.

In [118]:

```
contribuciones=['Investigación previa','Redacción de las respuestas','Desarrollo código
firmas=['RM','RM','RM']
pd.DataFrame([contribuciones,firmas],index=['Contirbuciones','Firmas']).T
```

Out[118]:

	Contirbuciones	Firmas
0	Investigación previa	RM
1	Redacción de las respuestas	RM
2	Desarrollo código	RM

In [ ]:

```

```