# Assignment 1

**Name: Guo Xinfu**

**ID: n01611988**

## 1. Import numpy, pandas, vasualization libraries and set %matplotlib inline

In [2]:
```
import numpy as np
import pandas as pd
%matplotlib inline
import matplotlib .pyplot as plt
import seaborn as sns
```

## 2. Check the info() of the df

In [3]:
```
df = pd.read_csv("WineQT.csv")
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1143 entries, 0 to 1142
Data columns (total 13 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   fixed acidity         1142 non-null   float64
 1   volatile acidity      1139 non-null   float64
 2   citric acid           1140 non-null   float64
 3   residual sugar        1143 non-null   float64
 4   chlorides             1143 non-null   float64
 5   free sulfur dioxide   1143 non-null   float64
 6   total sulfur dioxide  1143 non-null   float64
 7   density               1141 non-null   float64
 8   pH                    1143 non-null   float64
 9   sulphates             1143 non-null   float64
 10  alcohol               1143 non-null   float64
 11  quality               1141 non-null   float64
 12  Id                    1143 non-null   int64  
dtypes: float64(12), int64(1)
memory usage: 116.2 KB
```

## 3. Check the head of df

In [4]:
```
df.head()
```

Out[4]:

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

## 4. What are the top 5 alcohol for wine dataset?

In [8]:
```
df_top_alcohol = df.sort_values(by='alcohol', ascending=False).head(5)
df_top_alcohol
```

Out[8]:

| | fixed acidity | volatile acidity | citric acid | residual sugar | chlorides | free sulfur dioxide | total sulfur dioxide | density | pH | sulphates | al |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **462** | 15.9 | 0.36 | 0.65 | 7.5 | 0.096 | 22.0 | 71.0 | 0.99760 | 2.98 | 0.84 | |
| **329** | 8.8 | 0.46 | 0.45 | 2.6 | 0.065 | 7.0 | 18.0 | 0.99470 | 3.32 | 0.79 | |
| **98** | 5.2 | 0.34 | 0.00 | 1.8 | 0.050 | 27.0 | 63.0 | 0.99160 | 3.68 | 0.79 | |
| **898** | 5.0 | 0.38 | 0.01 | 1.6 | 0.048 | 26.0 | 60.0 | 0.99084 | 3.70 | 0.75 | |
| **419** | 5.0 | 0.42 | 0.24 | 2.0 | 0.060 | 19.0 | 50.0 | 0.99170 | 3.72 | 0.74 | |

## 5. What is the data type of the density column?

In [9]:
```
df["density"].dtypes
```

Out[9]:
```
dtype('float64')
```

## 6. Check how many missing values are in the dataset?

In [11]:
```
df.isnull().sum().sum()
#df.isnull().sum() which is about how many missing values in every column.
```

Out[11]:
```
12
```

## 7. Fill missing parts with the appropriate technique based on each

In [13]:
```
df.fillna(0, inplace=True)
df
```

Out[13]:

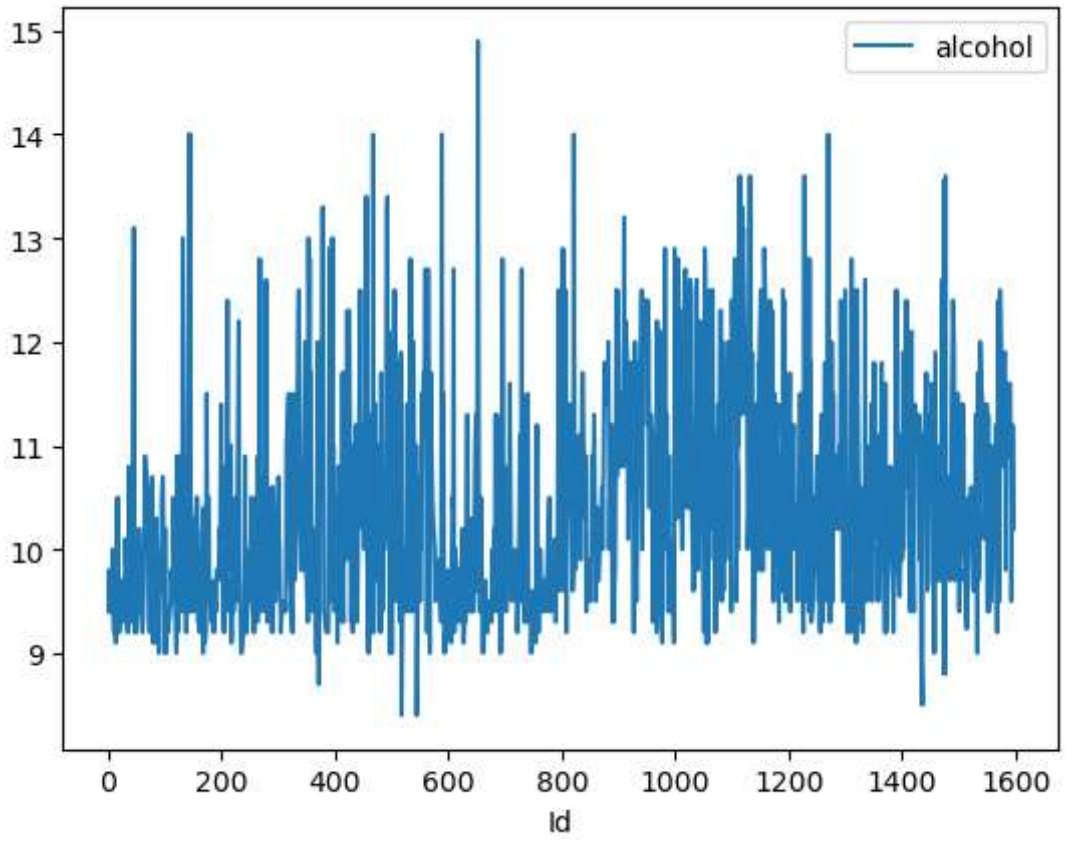
| | fixed acidity | volatile acidity | citric acid | residual sugar | chlorides | free sulfur dioxide | total sulfur dioxide | density | pH | sulphates | a |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 7.4 | 0.700 | 0.00 | 1.9 | 0.076 | 11.0 | 34.0 | 0.99780 | 3.51 | 0.56 | |
| **1** | 7.8 | 0.880 | 0.00 | 2.6 | 0.098 | 25.0 | 67.0 | 0.99680 | 3.20 | 0.68 | |
| **2** | 7.8 | 0.760 | 0.04 | 2.3 | 0.092 | 15.0 | 54.0 | 0.99700 | 3.26 | 0.65 | |
| **3** | 11.2 | 0.280 | 0.56 | 1.9 | 0.075 | 17.0 | 60.0 | 0.99800 | 3.16 | 0.58 | |
| **4** | 7.4 | 0.700 | 0.00 | 1.9 | 0.076 | 11.0 | 34.0 | 0.99780 | 3.51 | 0.56 | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| **1138** | 6.3 | 0.510 | 0.13 | 2.3 | 0.076 | 29.0 | 40.0 | 0.99574 | 3.42 | 0.75 | |
| **1139** | 6.8 | 0.620 | 0.08 | 1.9 | 0.068 | 28.0 | 38.0 | 0.99651 | 3.42 | 0.82 | |
| **1140** | 6.2 | 0.600 | 0.08 | 2.0 | 0.090 | 32.0 | 44.0 | 0.99490 | 3.45 | 0.58 | |
| **1141** | 5.9 | 0.550 | 0.10 | 2.2 | 0.062 | 39.0 | 51.0 | 0.99512 | 3.52 | 0.76 | |
| **1142** | 5.9 | 0.645 | 0.00 | 2.0 | 0.075 | 32.0 | 44.0 | 0.99547 | 3.57 | 0.71 | |

1143 rows × 13 columns

## 8. Create a simple plot of the dataframe indicating the alcohol per wine.

In [18]:
```
df.plot("Id", "alcohol")
```

Out[18]: <Axes: xlabel='Id'>




## 9. Change the name of the column "residual sugar" to "sugar"

In [21]:
```
df.rename(columns={"residual sugar":"sugar"}, inplace=True)
df.head()
```

Out[21]:

| | fixed acidity | volatile acidity | citric acid | sugar | chlorides | free sulfur dioxide | total sulfur dioxide | density | pH | sulphates | alcoho |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 7.4 | 0.70 | 0.00 | 1.9 | 0.076 | 11.0 | 34.0 | 0.9978 | 3.51 | 0.56 | 9.4 |
| **1** | 7.8 | 0.88 | 0.00 | 2.6 | 0.098 | 25.0 | 67.0 | 0.9968 | 3.20 | 0.68 | 9.8 |
| **2** | 7.8 | 0.76 | 0.04 | 2.3 | 0.092 | 15.0 | 54.0 | 0.9970 | 3.26 | 0.65 | 9.8 |
| **3** | 11.2 | 0.28 | 0.56 | 1.9 | 0.075 | 17.0 | 60.0 | 0.9980 | 3.16 | 0.58 | 9.8 |
| **4** | 7.4 | 0.70 | 0.00 | 1.9 | 0.076 | 11.0 | 34.0 | 0.9978 | 3.51 | 0.56 | 9.4 |

## 10. Sort the dataset based on "quality" and "alcohol" feature.

In [25]:
```
df.sort_values(by=["quality", "alcohol"], ascending=False, inplace=True)
df
```

Out[25]:

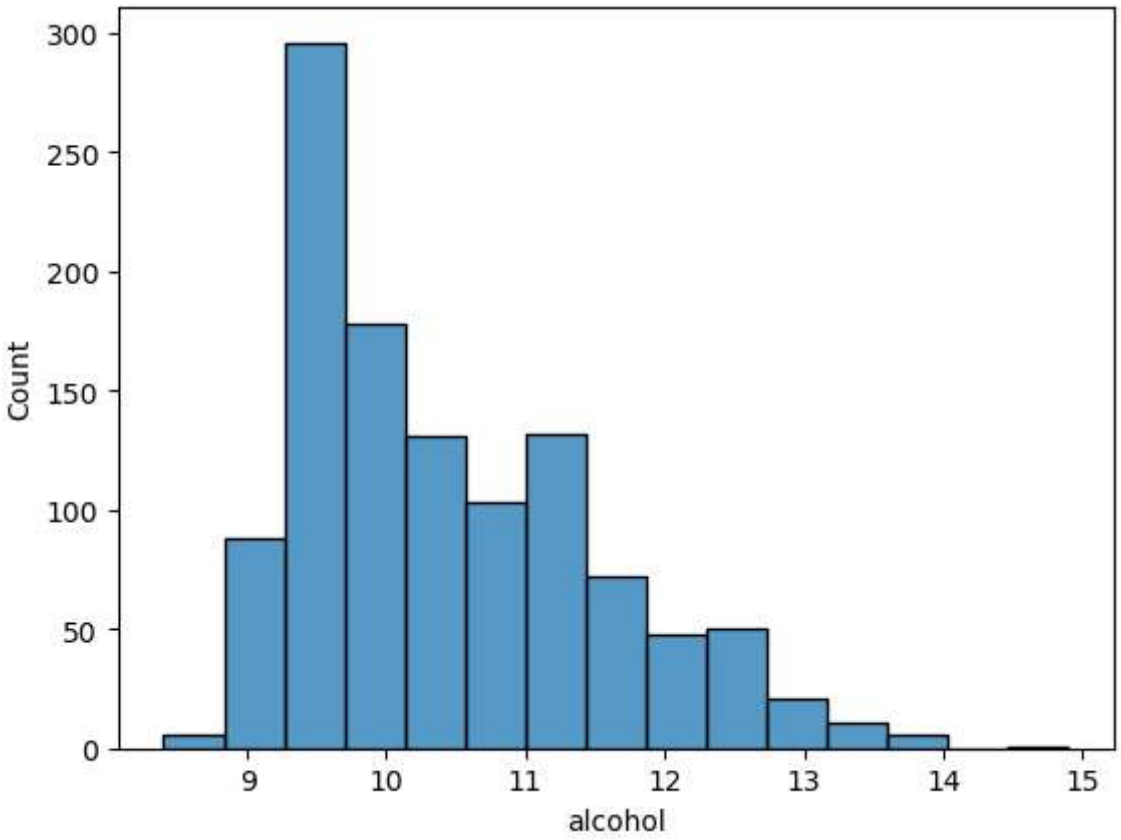
| | fixed acidity | volatile acidity | citric acid | sugar | chlorides | free sulfur dioxide | total sulfur dioxide | density | pH | sulphates | alc |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **419** | 5.0 | 0.42 | 0.24 | 2.0 | 0.060 | 19.0 | 50.0 | 0.99170 | 3.72 | 0.74 | |
| **321** | 11.3 | 0.62 | 0.67 | 5.2 | 0.086 | 6.0 | 19.0 | 0.99880 | 3.22 | 0.69 | |
| **793** | 7.9 | 0.54 | 0.34 | 2.5 | 0.076 | 8.0 | 17.0 | 0.99235 | 3.20 | 0.72 | |
| **271** | 5.6 | 0.85 | 0.05 | 1.4 | 0.045 | 12.0 | 88.0 | 0.99240 | 3.56 | 0.82 | |
| **190** | 7.9 | 0.35 | 0.46 | 3.6 | 0.078 | 15.0 | 37.0 | 0.99730 | 3.35 | 0.86 | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **1046** | 7.3 | 0.98 | 0.05 | 2.1 | 0.061 | 20.0 | 49.0 | 0.99705 | 3.31 | 0.55 | |
| **324** | 11.6 | 0.58 | 0.66 | 2.2 | 0.074 | 10.0 | 47.0 | 1.00080 | 3.25 | 0.57 | |
| **368** | 10.4 | 0.61 | 0.49 | 2.1 | 0.200 | 5.0 | 16.0 | 0.99940 | 3.16 | 0.63 | |
| **1075** | 6.8 | 0.00 | 0.05 | 2.0 | 0.070 | 6.0 | 14.0 | 0.99562 | 3.51 | 0.66 | |
| **1112** | 7.8 | 0.60 | 0.26 | 2.0 | 0.080 | 31.0 | 131.0 | 0.99622 | 3.21 | 0.52 | |

1143 rows × 13 columns

## 11. Plot the histogram of "density".

In [28]:
```
sns.histplot(data=df['alcohol'], bins=15)
```

Out[28]: <Axes: xlabel='alcohol', ylabel='Count'>

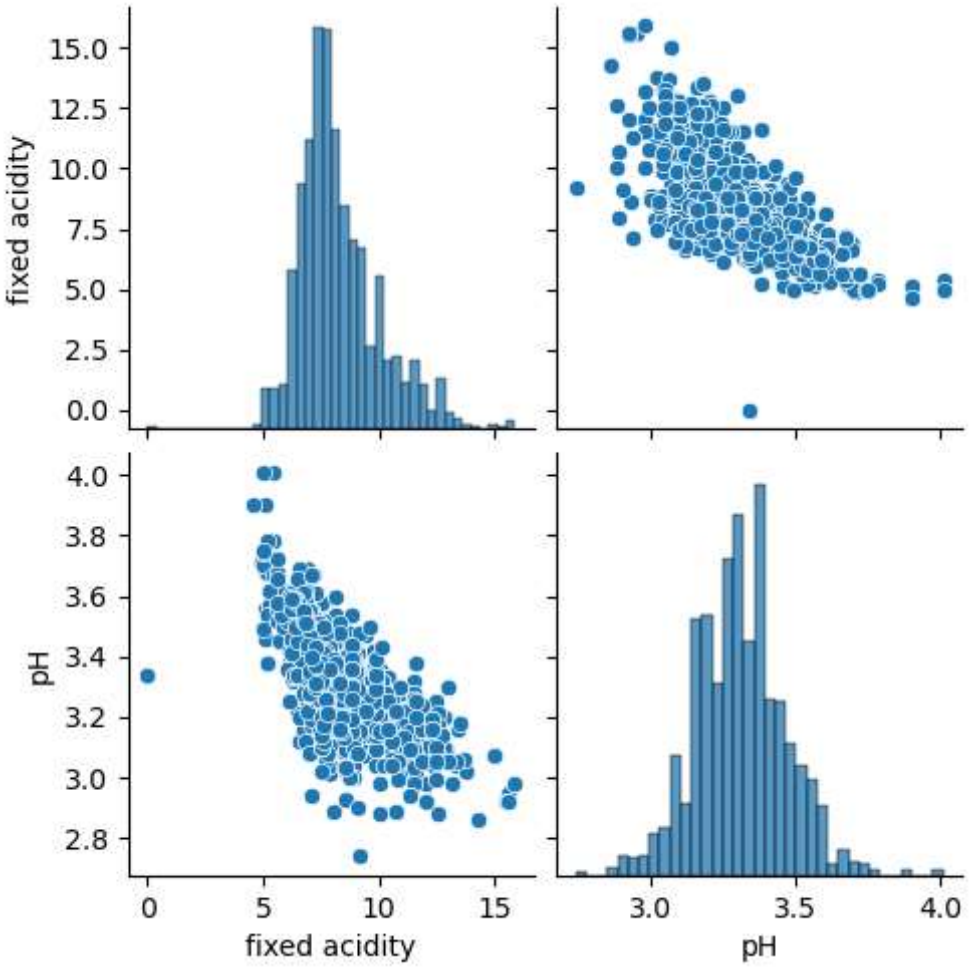


## 12. Plot distribution and relationship in a dataset and analyze the relationship between "fixed acidity" and "pH" columns

In [29]:
```
sns.pairplot(df[["fixed acidity", "pH"]])
```

```
E:\anaconda\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning: The figure lay
out has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

Out[29]: <seaborn.axisgrid.PairGrid at 0x2b6a4315c50>




In [30]:
```
#the relationship between "fixed acidity" and "pH" is linear, strong and negative.
```