Analysis of Factors Affecting the Quality of Red Wine and Quality Prediction

Done By:

Tey Jia ying

Date:

6 Oct 2020

Introduction to research topic

A new wine brewery in Portugal, Jewel Wine, has engaged our service to analyze the different compositions found in red wine. They hope to obtain valuable insights to the factors that affect wine quality and will help them to improve their wine branding. As well as, some of their lower quality wines. There are several factors that go into producing a wine and these indicators can affect the wine quality which will be listed below.

Project Objective

This data analysis aims to evaluate how various factors affect the quality of red wine from a scale of 0 to 10, with 0 representing poor quality and 10 representing excellent quality. These factors include:

1. Fixed acidity
2. Volatile acidity
3. Citric acid
4. Residual sugar
5. Chlorides
6. Free sulfur dioxide
7. Total sulfur dioxide
8. Density
9. pH level
10. Sulphate
11. Alcohol
    We will also be doing a quality prediction model to predict the quality of red wine based on the factors inputted.

Initial Hypothesis

We hypothesize that the pH level is the most crucial factor in evaluating the grade of red wine. The pH level of wine is a measurement of its ripeness in relation to acidity. A wine that has a low pH level (acidic) tastes tart and crisp while wines with higher pH levels (basic) are considered more prone to bacterial growth. The average pH level for red wine is about 3.3 to 3.6 (Vinny, 2020).

Selected Dataset & Data Preparation

I. Problem Statement / Research Topic (2 Marks)

The research topic on quality evaluation of red wine will be addressed. We are researching this topic as we are interested in how to differentiate between poor quality and good quality red wine. By researching this topic, we hope to identify the determining factor of a good or bad wine. These findings can also be used by wine manufacturers to improve wines of lower quality.

II. Dataset and Data Preparation (2 Marks)

Data portal:UCI, Machine Learning Repository("https://archive.ics.uci.edu/ml/datasets/wine+quality")
Source of dataset(s):https://archive.ics.uci.edu/ml/datasets/wine+quality

Data Info: The two datasets are related to red and white variants of the Portuguese "Vinho Verde" wine. For more details, consult: [Web Link] or the reference [Cortez et al., 2009]. Due to privacy and logistic issues, only physicochemical (inputs) and sensory (the output) variables are available (e.g. there is no data about grape types, wine brand, wine selling price, etc.).

These datasets can be viewed as classification or regression tasks. The classes are ordered and not balanced (e.g. there are many more normal win

Data Structure: File type: CSV Columns: 11 Columns- Different Physicochemical headings, 1 column - overall quality
Row: 1599 Rows - Respective values Period of Data Collection: May 2004 - Feb 2007 Method: Recorded by a computerized system (iLab), which automatically manages the process of wine sample testing from producer requests to laboratory and sensory analysis. Each entry denotes a given test (analytical or sensory) and the final database was exported into a single sheet (.csv). Quality: Regarding the preferences, each sample was evaluated by a minimum of three sensory assessors (using blind tastes), which graded the wine in a scale that ranges from 0 (very bad) to 10 (excellent). The final sensory score is given by the median of these evaluations

---

Overview of Project & Explanation of Dataset:

Overview:

We have structured our project into 4 key sections. Below, we will share with you the steps we took together with insights for each section.

1. Data Cleaning
2. Exploratory Data Analysis
3. Analysis of Factors affecting Red Wine quality - Visualization
4. Wine Quality Prediction Model

Explanation of Dataset:

The dataset used for our project consist of quality and attributes for 1599 red wines. The dataset is related to red variants of the Portuguese "Vinho Verde" wine. In the dataset consists of 11 attributes for each red wine:

Attributes/Factors	Description
fixed acidity (tartaric acid - g / dm^3)	most acids involved with wine or fixed or nonvolatile (do not evaporate readily)
volatile acidity (acetic acid - g / dm^3)	the amount of acetic acid in wine, which at too high of levels can lead to an unpleasant, vinegar taste
citric acid (g / dm^3)	found in small quantities, citric acid can add 'freshness' and flavor to wines
residual sugar (g / dm^3)	the amount of sugar remaining after fermentation stops, it's rare to find wines with less than 1 gram/liter and wines with greater than 45 grams/liter are considered sweet
chlorides (sodium chloride - g / dm^3	the amount of salt in the wine
free sulfur dioxide (mg / dm^3)	the free form of SO2 exists in equilibrium between molecular SO2 (as a dissolved gas) and bisulfite ion; it prevents microbial growth and the oxidation of wine
total sulfur dioxide (mg / dm^3)	amount of free and bound forms of S02; in low concentrations, SO2 is mostly undetectable in wine, but at free SO2 concentrations over 50 ppm, SO2 becomes evident in the nose and taste of wine
density (g / cm^3)	the density of water is close to that of water depending on the percent alcohol and sugar content
pH	describes how acidic or basic a wine is on a scale from 0 (very acidic) to 14 (very basic); most wines are between 3-4 on the pH scale
sulphates (potassium sulphate - g / dm3)	a wine additive which can contribute to sulfur dioxide gas (S02) levels, wich acts as an antimicrobial and antioxidant
alcohol (% by volume)	the percent alcohol content of the wine

1. Data Cleaning

In [ ]:

```python
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
```

In [ ]:

```python
data_df = pd.read_csv ('winequality-red.csv', delimiter=',')
data_df.head()
```

Out[ ]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5

In [ ]:

```python
print("Rows, columns: " + str(data_df.shape))
```

```
Rows, columns: (1599, 12)
```

In [ ]:

```python
data_df.isnull().sum()
```

Out[ ]:

```
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
dtype: int64
```

Note: As shown above, since there is no missing data in our red wine dataset, we did not do any data cleaning.

2. Exploratory Data Analysis (EDA)

2A. Descriptive Statistical Analysis

After cleaning the data, we used the describe() function once again to look at the statistical summary of the numerical variables.

Insights:

From the describe() function, it computes a summary of statistics details like mean, percentile and standard deviation of the numerical values of the red wine quality dataset. Highlights of descriptive statistics are given below:

- Fixed acidity: The total of most acids involved with wine or fixed or nonvolatile has a mean value of 8.32 g/dm3 and median value of 7.90 g/dm3.
- Volatile acidity: The amount of acetic acid in wine has a mean value of 0.528 g/dm3 and median value of 0.520 g/dm3.
- Citric acid: Found in small quantities and a mean value of 0.271.
- Residual sugar: About 75% of the wines have a residual value below 2.6 g/dm3, while the remaining 25% of the wines have a residual value from range 2.6 to 15.5.
- Chlorides: Giving the wine a salty flavor, chlorides are mostly 0.079 g/dm3.
- Free sulfur dioxide: About 75% of the wines has a free sulfur dioxide value below 21 mg/dm3.
- Total sulfur dioxide: About 75% of the wines has a sulfur dioxide value below 62 mg/dm3.
- Density: Most wines have a mean value of 0.9967 and median value of 0.9968.
- pH: Most wines have a pH level of about 3.31.
- Sulphates: The wine additive ranges between 0.33 g/dm3 to 2 g/dm3, with most wines having a sulphte amount of 0.62 g/dm3.
- Alcohol: The percentage alcohol content of the wines are mostly 10.2%.
- Quality: Wine scores are in range between 3 to 8 and most of them have a score of 5.

In [ ]:

```python
data_df.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
```

In [ ]:

```python
data_df.describe()
```

Out[ ]:

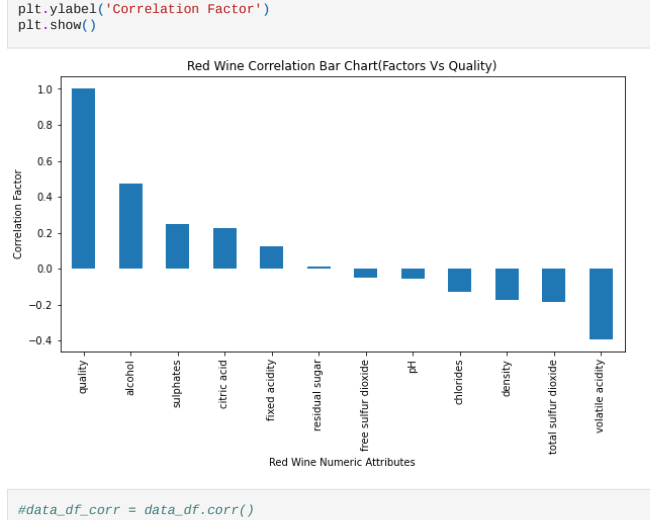
	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	qua
count	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000
mean	8.319637	0.527821	0.270976	2.538806	0.087467	15.874922	46.467792	0.996747	3.311113	0.658149	10.422983	5.636
std	1.741096	0.179060	0.194801	1.409928	0.047065	10.460157	32.895324	0.001887	0.154386	0.169507	1.065668	0.807
min	4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	6.000000	0.990070	2.740000	0.330000	8.400000	3.000
25%	7.100000	0.390000	0.090000	1.900000	0.070000	7.000000	22.000000	0.995600	3.210000	0.550000	9.500000	5.000
50%	7.900000	0.520000	0.260000	2.200000	0.079000	14.000000	38.000000	0.996750	3.310000	0.620000	10.200000	6.000
75%	9.200000	0.640000	0.420000	2.600000	0.090000	21.000000	62.000000	0.997835	3.400000	0.730000	11.100000	6.000
max	15.900000	1.580000	1.000000	15.500000	0.611000	72.000000	289.000000	1.003690	4.010000	2.000000	14.900000	8.000

2B. Correlation Analysis

Next, we conducted a Correlation Analysis to better understand the relationship between the 11 key factors affecting quality of red wine.

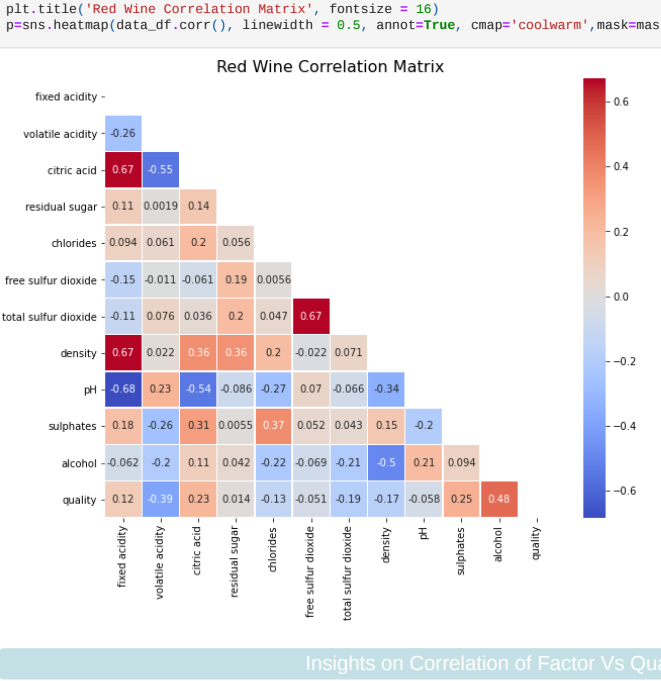
In [ ]:

```python
corr_mat = data_df.corr()
plt.figure(figsize = (10,5))
corr_mat['quality'].sort_values(ascending = False).plot(kind = 'bar');
plt.title('Red Wine Correlation Bar Chart(Factors Vs Quality)')
plt.xlabel('Red Wine Numeric Attributes')
plt.ylabel('Correlation Factor')
plt.show()
```



In [ ]:

```python
#data_df_corr = data_df.corr()
#data_df_corr.style.background_gradient(cmap = 'coolwarm').set_precision(2)
import numpy as np
rs = np.random.RandomState(33)


# Generate a mask for the upper triangle
mask = np.triu(np.ones_like(data_df.corr(), dtype=bool))

plt.figure(figsize=(10,8))
plt.title('Red Wine Correlation Matrix', fontsize = 16)
p=sns.heatmap(data_df.corr(), linewidth = 0.5, annot=True, cmap='coolwarm',mask=mask)
```



Insights on Correlation of Factor Vs Quality

We identified four variables which had significant correlation with the red wine quality

1. Alcohol Vs Quality(0.48)
2. Sulphates Vs Quality(0.25)
3. Critic Acid Vs Quality(0.23)
4. Volatile acidity Vs Quality(-0.39)

3. Analysis of Factors affecting Red Wine quality -Data Visualization

For the analysis of Factors affecting Red Wine quality, we are going to focus on the 4 most significant factors in the above section and have the analyzed their impact on Red Wine quality(score between 0 and 10). We will analyze 4 the attributes with the help of visualizations analysis.The analysis will comprise of the following sections:

Part A: Analysis of Red Wine Quality

- Red Wine Quality distribution
- Classification of quality

Part B:Analysis of each factor VS Quality

3A. Volatile acidity(acetic acid - g / dm^3) Vs Quality
3B. Citric acid(g / dm^3) Vs Quality
3C. Sulphates(potassium sulphate - g / dm3) Vs Quality
3D. Alcohol(% by volume) Vs Quality

Part A: Analysis and Classification of Red Wine Quality

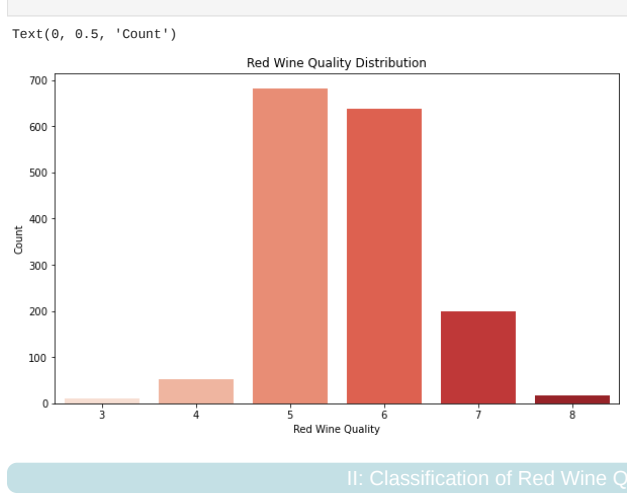
I: Distribution of Red Wine Quality

In [ ]:

```python
fig = plt.figure(figsize=(10,6))
ax1 = fig.add_subplot(1,1,1)

sns.countplot(x = "quality", data = data_df, ax = ax1, palette="Reds")
ax1.set_title('Red Wine Quality Distribution', fontsize = 12)
ax1.set_xlabel('Red Wine Quality')
ax1.set_ylabel('Count')
```

Out[ ]:

```
Text(0, 0.5, 'Count')
```



II: Classification of Red Wine Quality

In [ ]:

```python
data_df["quality"].describe()
```

Out[ ]:

```
count    1599.000000
mean        5.636023
std         0.807569
min         3.000000
25%         5.000000
50%         6.000000
75%         6.000000
max         8.000000
Name: quality, dtype: float64
```

Insights

1. Quality ranges from 3 to 8
2. Majority of wine which have a middle quality of 5 and 6

Defining a good quality wine

Also, since the median quality is 6 and mean quality is ~5.6 , **we defined a bottle of wine as 'good quality' if it had a quality score of 7 or higher, and if it had a score of less than 7, it was deemed 'poor quality'.**

Part B:Analysis of each factor VS Quality

In this section, we will only focus on the 4 most significant variables(Volatile acidity,Citric acid(g / dm^3), Sulphates and Alcohol) as mentioned in the correlation matrix above. We will dive deeper to analyse the relationship between each of the 4 factors and the red wine quality and produce some insights from it. Using the insights, we will be suggesting some useful recommendations that our client, Jewel wine can take into consideration in order to produce a good quality wine.

3A. Volatile acidity Vs Quality

I: Removing Outliers for Volatile Acidity

In [ ]:

```python
#REMOVING OUTLIERS OF FIXED ACIDITY
va_quartile_1=data_df["volatile acidity"].quantile(q=0.25)
va_quartile_3=data_df["volatile acidity"].quantile(q=0.75)
print(va_quartile_1)
print(va_quartile_3)

va_IQR=va_quartile_3-va_quartile_1

va_lower_bound=float(va_quartile_1-1.5*va_IQR)
va_upper_bound=float(va_quartile_3+1.5*va_IQR)
print(va_lower_bound) #max value--> any thing outside max or min value--> outliers
print(va_upper_bound)#min value

va_cond1=data_df["volatile acidity"]>=va_lower_bound
va_cond2=data_df["volatile acidity"]<=va_upper_bound
data_df_no_va_outlier=data_df[va_cond1 & va_cond2]
print(len(data_df_no_va_outlier))
```

```
0.39
0.64
0.015000000000000013
1.0150000000000001
1580
```

II: Range of volatile acidity that gives better quality wine (supporting evidence for boxplot)

In [ ]:

```python
better_quality_va=data_df_no_va_outlier[data_df_no_va_outlier["quality"].isin([7,8])]
better_quality_va[["volatile acidity","quality"]].describe()
```

Out[ ]:

	volatile acidity	quality
count	217.000000	217.000000
mean	0.405530	7.082949
std	0.144963	0.276443
min	0.120000	7.000000
25%	0.300000	7.000000
50%	0.370000	7.000000
75%	0.490000	7.000000
max	0.915000	8.000000

II: Range of volatile acidity that gives better quality wine (supporting evidence for boxplot)

In [ ]:

```python
poorer_quality_va=data_df_no_va_outlier[data_df_no_va_outlier["quality"].isin([3,4])]
poorer_quality_va[["volatile acidity","quality"]].describe()
```

Out[ ]:

	volatile acidity	quality
count	54.000000	54.000000
mean	0.656019	3.87037
std	0.184448	0.33905
min	0.230000	3.00000
25%	0.522500	4.00000
50%	0.632500	4.00000
75%	0.807500	4.00000
max	0.980000	4.00000

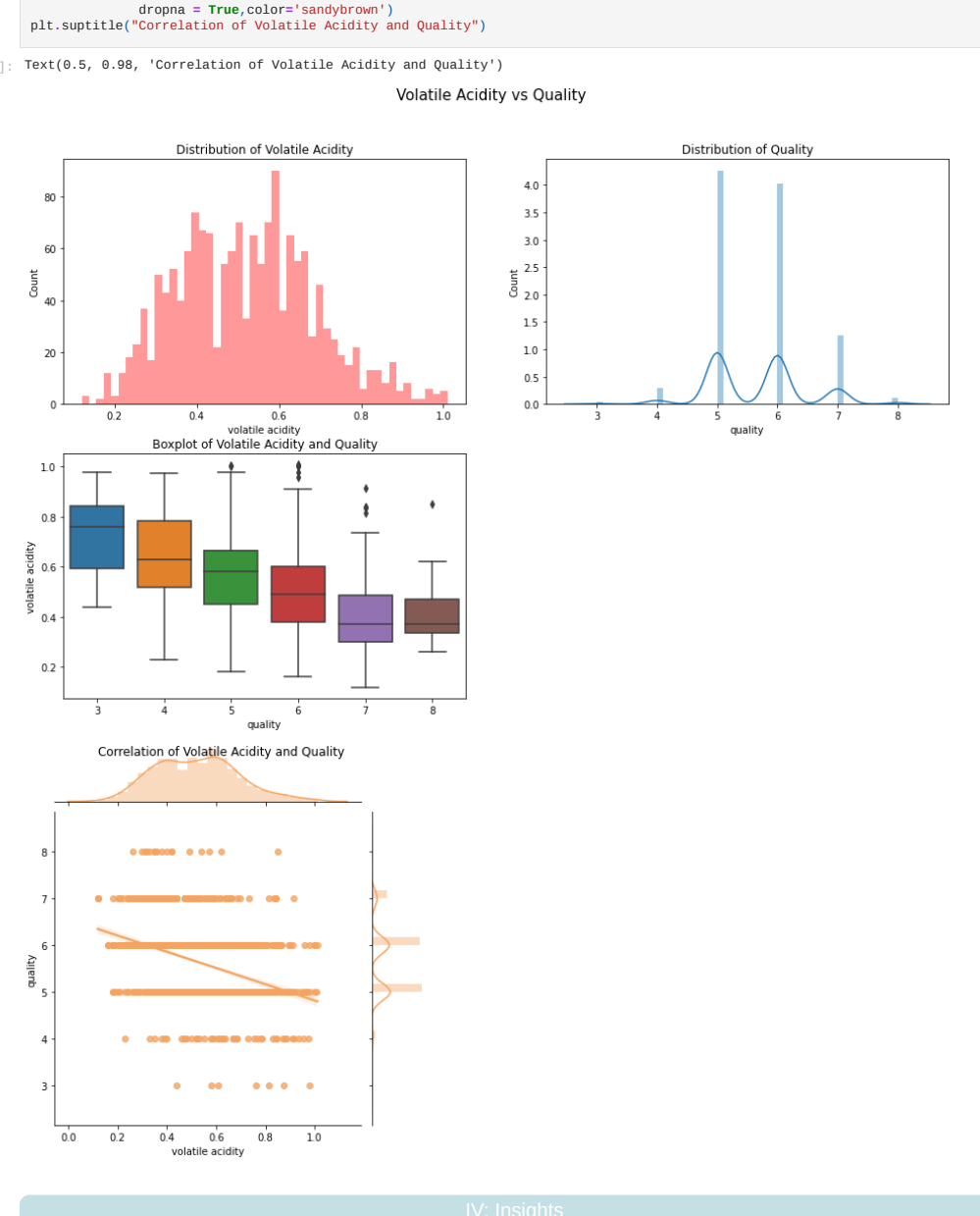
III: Plotting of graphs--Volatile Acidity VS Quality

In [ ]:

```python
figure = plt.figure(figsize=(16,10))

# Distribution of Volatile acidity
ax1 = figure.add_subplot(2, 2, 1)
sns.distplot(data_df_no_va_outlier['volatile acidity'],
             kde=False,
             norm_hist=False,
             bins=50,
             color='red',
             ax=ax1).set_title('Distribution of Volatile Acidity')
ax1.set_ylabel("Count")

# Distribution of Quality
ax2 = figure.add_subplot(2, 2, 2)
sns.distplot(data_df_no_va_outlier['quality'],
             bins=50,
             ax=ax2).set_title('Distribution of Quality')
ax2.set_ylabel("Count")

# Boxplot
ax3 = figure.add_subplot(2, 2, 3)
sns.boxplot(data = data_df_no_va_outlier, x = 'quality', y = 'volatile acidity',ax=ax3).set_title("Boxplot of Volatile Acidity an

plt.suptitle("Volatile Acidity vs Quality", fontsize=15)

# Correlation of Volatile Acidity and Quality
sns.jointplot(x = "volatile acidity", y = "quality",
              kind = "reg", data = data_df_no_va_outlier,
              dropna = True,color='sandybrown')
plt.suptitle("Correlation of Volatile Acidity and Quality")
```

Out[ ]:

```
Text(0.5, 0.98, 'Correlation of Volatile Acidity and Quality')
```



IV: Insights

a) **Distribution Chart of Volatile Acidity:**
The quantity of volatile acidity ranges between 0.12 to 0.98 g/dm^3 where the distribution mainly clusters around 0.37 to 0.63 g/dm^3.

b) **Distribution Chart of Quality:**
We can see that the chart is middle-skewed where most of the wine quality ranges from 5 to 6 which implies that most of the red wine possess middle quality.

c) **Boxplot for Volatile Acidity:**
For a better wine quality of 7 to 8, the quantity of volatile acidity ranges from 0.30 to 0.49 g/dm^3. Whereas for a lower wine quality of 3 to 4, the quantity of volatile acidity will range from 0.50 to 0.80 g.dm^3. This might implies that as the quality of the wine decreases, the quantity of volatile acidity required also increases. However, this is exceptional for the good quality wine of 7 and 8 as both have the similar quantity of volatile acidity.

d) **Joint Plot for Volatile Acidity and Quality:**
We have combined the two distribution charts together to see their relationship. We learned that there is an negative relationship between volatile acidity and quality as the orange line shows a negative linear graph. This evidences the results from the boxplot where the increase in volatile acidity shows a decrease in quality.

Recommendations
Hence, we suggest our client, Jewel wine to lessen the composition of volatile acidity to produce a good quality wine. A good amount of volatile acid composition would be around 0.30 to 0.49 g/dm^3. This is because volatile acid is the amount of acetic acid in wine, which at too high of levels can lead to an unpleasant, vinegar taste.

3B. Citric acid Vs Quality

I: Removing Outliers for Citric Acid

In [ ]:

```python
#REMOVING OUTLIERS OF CITRIC ACIDITY
ca_quartile_1=data_df["citric acid"].quantile(q=0.25)
ca_quartile_3=data_df["citric acid"].quantile(q=0.75)
print(ca_quartile_1)
print(ca_quartile_3)

ca_IQR=ca_quartile_3-ca_quartile_1

ca_lower_bound=float(ca_quartile_1-1.5*ca_IQR)
ca_upper_bound=float(ca_quartile_3+1.5*ca_IQR)
print(ca_lower_bound) #max value--> any thing outside max or min value--> outliers
print(ca_upper_bound) #min value

ca_cond1=data_df["citric acid"]>=ca_lower_bound
ca_cond2=data_df["citric acid"]<=ca_upper_bound
data_df_no_ca_outlier=data_df[ca_cond1 & ca_cond2]
print(len(data_df_no_ca_outlier))
```

```
0.09
0.42
-0.40499999999999997
0.9149999999999999
1598
```

II: Range of Citric Acid that gives better quality wine (supporting evidence for boxplot)

In [ ]:

```python
better_quality_ca=data_df_no_ca_outlier[data_df_no_ca_outlier["quality"].isin([7,8])]
better_quality_ca[["citric acid","quality"]].describe()
```

Out[ ]:

	citric acid	quality
count	217.000000	217.000000
mean	0.376498	7.082949
std	0.194439	0.276443
min	0.000000	7.000000
25%	0.300000	7.000000
50%	0.400000	7.000000
75%	0.490000	7.000000
max	0.760000	8.000000

III: Range of citric acid that gives better quality wine (supporting evidence for boxplot)

In [ ]:

```python
poorer_quality_ca=data_df_no_ca_outlier[data_df_no_ca_outlier["quality"].isin([3,4])]
poorer_quality_ca[["citric acid","quality"]].describe()
```

Out[ ]:

	citric acid	quality
count	62.000000	62.000000
mean	0.160323	3.838710
std	0.179854	0.370801
min	0.000000	3.000000
25%	0.020000	4.000000
50%	0.075000	4.000000
75%	0.267500	4.000000
max	0.660000	4.000000

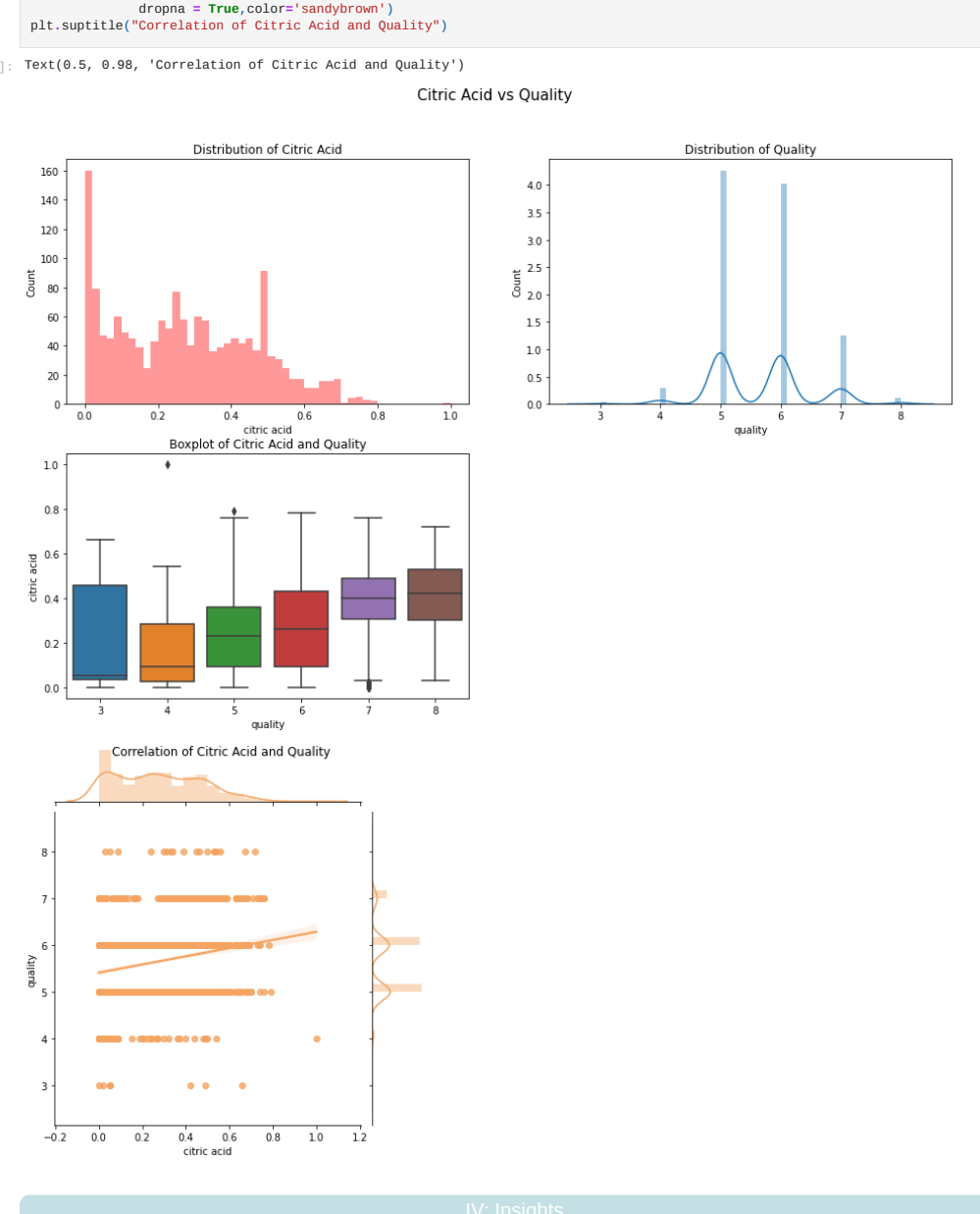
III: Plotting of graphs--Critic Acid VS Quality

In [ ]:

```python
figure = plt.figure(figsize=(16,10))

# Distribution of Citric Acid
ax1 = figure.add_subplot(2, 2, 1)
sns.distplot(data_df_no_ca_outlier['citric acid'],
             kde=False,
             norm_hist=False,
             bins=50,
             color='red',
             ax=ax1).set_title('Distribution of Citric Acid')
ax1.set_ylabel("Count")

# Distribution of Quality
ax2 = figure.add_subplot(2, 2, 2)
sns.distplot(data_df_no_va_outlier['quality'],
             bins=50,
             ax=ax2).set_title('Distribution of Quality')
ax2.set_ylabel("Count")

# Boxplot
ax3 = figure.add_subplot(2, 2, 3)
sns.boxplot(data = data_df_no_va_outlier, x = 'quality', y = 'citric acid',ax=ax3).set_title("Boxplot of Citric Acid and Quality"

plt.suptitle("Citric Acid vs Quality", fontsize=15)

# Correlation of Citric Acid and Quality
sns.jointplot(x = "citric acid", y = "quality",
              kind = "reg", data = data_df_no_va_outlier,
              dropna = True,color='sandybrown')
plt.suptitle("Correlation of Citric Acid and Quality")
```

Out[ ]:

```
Text(0.5, 0.98, 'Correlation of Citric Acid and Quality')
```



IV: Insights

a) **Distribution Chart of Citric Acid:**
The quantity of citric acid ranges between 0.00 to 0.76 g/dm^3 where the distribution mainly clusters around 0.075 to 0.76 g/dm^3. The quantity of citric acid is quite small in the overall quantity.

b) **Distribution Chart of Quality:**
We can see that the chart is middle-skewed where most of the wine quality ranges from 5 to 6 which implies that most of the red wine possess middle quality.

c) **Boxplot for Citric Acid:**
For a better wine quality of 7 to 8, the quantity of citric acid ranges from 0.30 to 0.49 g/dm^3. Whereas for a lower wine quality of 3 to 4, the quantity of citric acid will range from 0.02 to 0.27 g.dm^3. This might implies that as the quality of the wine increases, the quantity of citric acid required also increases. Note that, the quantity of volatile acidity of the good quality wine of 7 and 8 does not differ much.

d) **Joint Plot for Citric Acid and Quality:**
We have combined the two distribution charts together to see their relationship. We learned that there is an positive relationship between citric acid and quality as the orange line shows a positive linear graph. This evidences the results from the boxplot where the increase in citric acid shows a increase in quality.

Recommendations
Hence, we suggest our client, Jewel wine to increase the composition of citric acid acidity to produce a good quality wine. A good amount of citric acid composition would be around 0.30 to 0.49 g/dm^3 to produce a good quality wine. This is because citric acid is the essence to add 'freshness' and flavor to wines.

3C. Sulphates Vs Quality

I: Removing Outliers for Sulphates

In [ ]:

```python
#REMOVING OUTLIERS OF FIXED ACIDITY
sul_quantile = data_df.groupby("quality")["sulphates"].quantile([0.25, 0.75]).unstack(level=1)

quartile_1=data_df["sulphates"].quantile(q=0.25) # 25% of hourly rate data are below $29.00, isit normal
quartile_3=data_df["sulphates"].quantile(q=0.75)
print(quartile_1)
print(quartile_3)

IQR=quartile_3-quartile_1

lower_bound=float(quartile_1-1.5*IQR)
upper_bound=float(quartile_3+1.5*IQR)
print(lower_bound) #max value--> any thing outside max or min value--> outliers
print(upper_bound)#min value

cond1=data_df["sulphates"]>=lower_bound
cond2=data_df["sulphates"]<=upper_bound
data_df_no_outlier=data_df[cond1 & cond2]
```

```
0.55
0.73
0.28000000000000014
0.9999999999999999
```

II: Range of Sulphates that gives better quality wine (supporting evidence for boxplot)

In [ ]:

```python
better_quality_sul=data_df_no_outlier[data_df_no_outlier["quality"].isin([7,8])]
better_quality_sul[["sulphates","quality"]].describe()
```

Out[ ]:

	sulphates	quality
count	207.000000	207.000000
mean	0.726522	7.082126
std	0.110077	0.275422
min	0.390000	7.000000
25%	0.650000	7.000000
50%	0.740000	7.000000
75%	0.815000	7.000000
max	0.990000	8.000000

II: Range of Sulphates that gives poorer quality wine (supporting evidence for boxplot)

In [ ]:

```python
poorer_quality_sul=data_df_no_outlier[data_df_no_outlier["quality"].isin([3,4])]
poorer_quality_sul[["sulphates","quality"]].describe()
```

Out[ ]:

	sulphates	quality
count	60.000000	60.000000
mean	0.551833	3.833333
std	0.094214	0.375823
min	0.330000	3.000000
25%	0.490000	4.000000
50%	0.555000	4.000000
75%	0.592500	4.000000
max	0.860000	4.000000

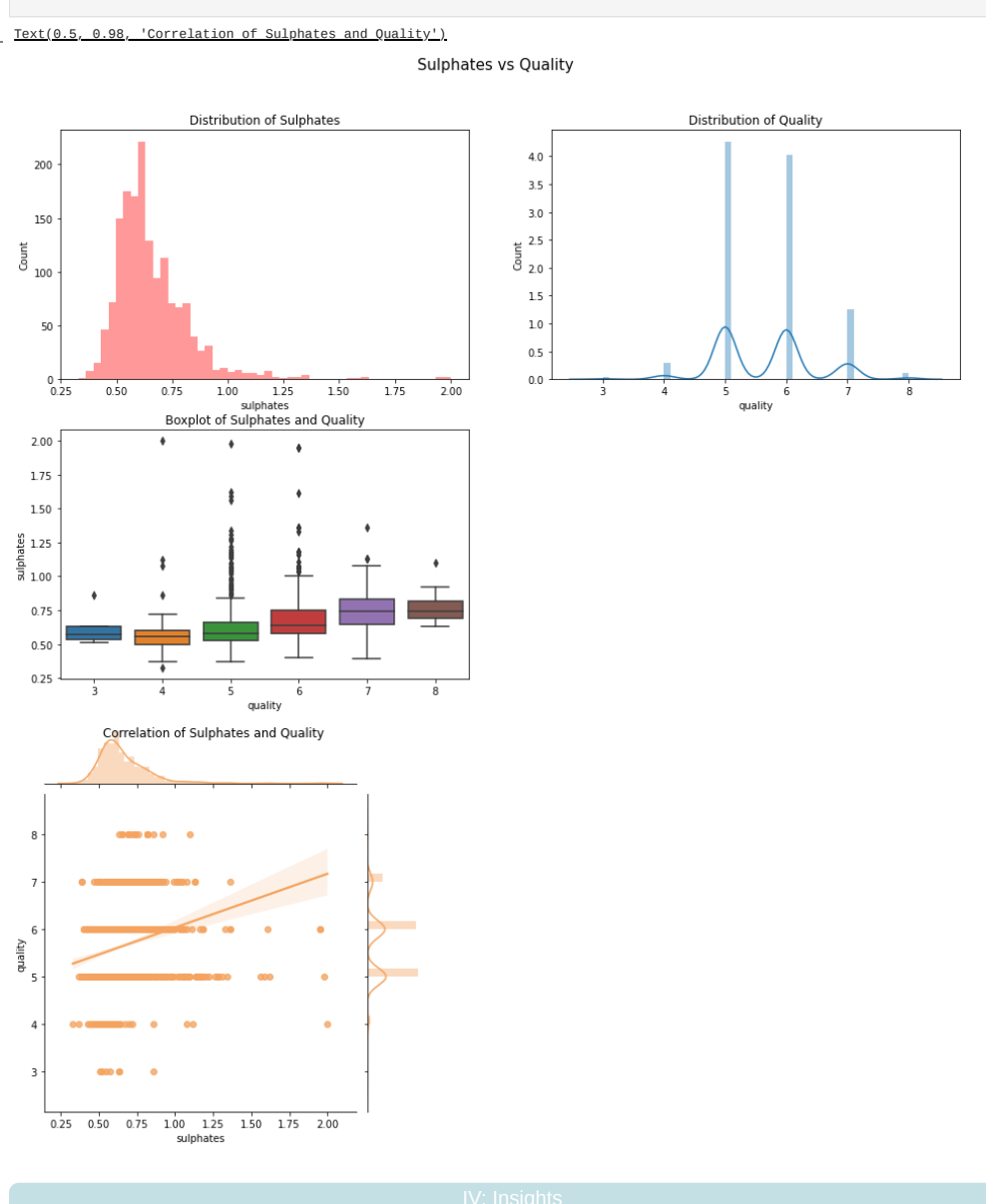
III: Plotting of graphs--Sulphates VS Quality

In [ ]:

```python
figure = plt.figure(figsize=(16,10))

# Distribution of Sulphates
ax1 = figure.add_subplot(2, 2, 1)
sns.distplot(data_df_no_va_outlier['sulphates'],
             kde=False,
             norm_hist=False,
             bins=50,
             color='red',
             ax=ax1).set_title('Distribution of Sulphates')
ax1.set_ylabel("Count")

# Distribution of Quality
ax2 = figure.add_subplot(2, 2, 2)
sns.distplot(data_df_no_va_outlier['quality'],
             bins=50,
             ax=ax2).set_title('Distribution of Quality')
ax2.set_ylabel("Count")

# Boxplot
ax3 = figure.add_subplot(2, 2, 3)
sns.boxplot(data = data_df_no_va_outlier, x = 'quality', y = 'sulphates',ax=ax3).set_title("Boxplot of Sulphates and Quality")

plt.suptitle("Sulphates vs Quality", fontsize=15)

# Correlation of Sulphates and Quality
sns.jointplot(x = "sulphates", y = "quality",
              kind = "reg", data = data_df_no_va_outlier,
              dropna = True,color='sandybrown')
plt.suptitle("Correlation of Sulphates and Quality")
```

Out[ ]:

```
Text(0.5, 0.98, 'Correlation of Sulphates and Quality')
```



IV: Insights

a) **Distribution Chart of Sulphates:**
The distribution chart of sulphate is left-skewed which means that the quantity of sulphate is quite less in this dataset. The quantity of Sulphates ranges between 0.33 to 0.99 g/dm^3 where the distribution mainly clusters around 0.55 to 0.74 g/dm^3.

b) **Distribution Chart of Quality:**
We can see that the chart is middle-skewed where most of the wine quality ranges from 5 to 6 which implies that most of the red wine possess middle quality.

c) **Boxplot for Sulphates:**
For a better wine quality of 7 to 8, the quantity of sulphate ranges from 0.65 to 0.81 g/dm^3. Whereas for a lower wine quality of 3 to 4, the quantity of sulphate will range from 0.49 to 0.59g/dm^3. This might implies that as the quality of the wine increases, the quantity of sulphate required also increases. However, there might be a possibility that the unique figure was depended on the small number of high and low quality data ,hence resulting in an inaccurate measurement. In fact, the low quality wine of 3 and 4 shows contrasting average where the distribution of datasets within the quality lies towards one side of the quartiles.

d) **Joint Plot for Sulphates and Quality:**
We have combined the two distribution charts together to see their relationship. We learned that there is an positive relationship between Sulphate and quality as the orange line shows a positive linear graph. This evidences the results from the boxplot where the increase in sulphate results in an increase in quality as well. The graph is quite steeped up which shows that the relationship between sulphate and wine is quite strong.

Recommendations
Hence, we suggest our client, Jewel wine to increase the composition of sulphates to produce a good quality wine. A good amount of sulphate composition would be around 0.65 to 0.81 g/dm^3 to produce a good quality wine. This is because sulphate is a wine additive which can contribute to sulfur dioxide gas (S02) levels, wich acts as an antimicrobial and antioxidant to kill bacteria in wine. Hence, having a higher composition of sulphate can produce a good quality wine.

3D. Alcohol Vs Quality

I: Removing Outliers for Alcohol

In [ ]:

```python
#REMOVING OUTLIERS OF Alcohol
quartile_1=data_df["alcohol"].quantile(q=0.25) # 25% of hourly rate data are below $29.00, isit normal
quartile_3=data_df["alcohol"].quantile(q=0.75)
print(quartile_3)

IQR=quartile_3-quartile_1

lower_bound=float(quartile_1-1.5*IQR)
upper_bound=float(quartile_3+1.5*IQR)
print(lower_bound) #max value--> any thing outside max or min value--> outliers
print(upper_bound)#min value

cond1=data_df["alcohol"]>=lower_bound
cond2=data_df["alcohol"]<=upper_bound
data_df_no_outlier=data_df[cond1 & cond2]
```

```
9.5
11.1
7.1000000000000005
13.5
```

II: Range of Alcohol that gives better quality wine (supporting evidence for boxplot)

In [ ]:

```python
better_quality_al=data_df_no_outlier[data_df_no_outlier["quality"].isin([7,8])]
better_quality_al[["alcohol","quality"]].describe()
```

Out[ ]:

	alcohol	quality
count	210.000000	210.000000
mean	11.443095	7.076190
std	0.923750	0.265937
min	9.200000	7.000000
25%	10.800000	7.000000
50%	11.500000	7.000000
75%	12.100000	7.000000
max	13.400000	8.000000

II: Range of Alcohol that gives poorer quality wine (supporting evidence for boxplot)

In [ ]:

```python
poorer_quality_al=data_df_no_outlier[data_df_no_outlier["quality"].isin([3,4])]
poorer_quality_al[["alcohol","quality"]].describe()
```

Out[ ]:

	alcohol	quality
count	63.000000	63.000000
mean	10.215873	3.841270
std	0.918178	0.368359
min	8.400000	3.000000
25%	9.600000	4.000000
50%	10.000000	4.000000
75%	11.000000	4.000000
max	13.100000	4.000000

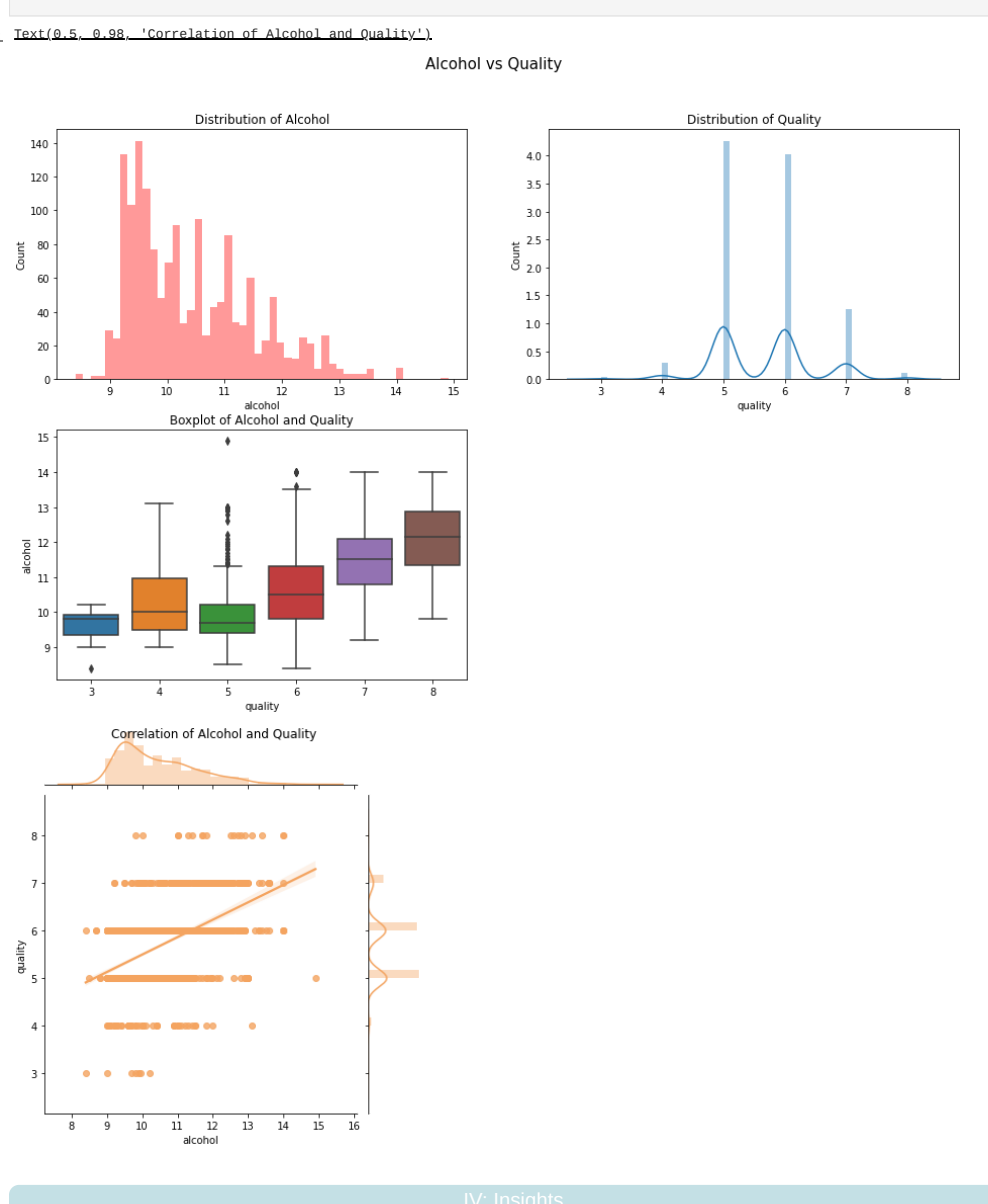
III: Plotting of graphs--Alcohol VS Quality

In [ ]:

```python
figure = plt.figure(figsize=(16,10))

# Distribution of Alcohol
ax1 = figure.add_subplot(2, 2, 1)
sns.distplot(data_df_no_va_outlier['alcohol'],
             kde=False,
             norm_hist=False,
             bins=50,
             color='red',
             ax=ax1).set_title('Distribution of Alcohol')
ax1.set_ylabel("Count")

# Distribution of Quality
ax2 = figure.add_subplot(2, 2, 2)
sns.distplot(data_df_no_va_outlier['quality'],
             bins=50,
             ax=ax2).set_title('Distribution of Quality')
ax2.set_ylabel("Count")

# Boxplot
ax3 = figure.add_subplot(2, 2, 3)
sns.boxplot(data = data_df_no_va_outlier, x = 'quality', y = 'alcohol',ax=ax3).set_title("Boxplot of Alcohol and Quality")

plt.suptitle("Alcohol vs Quality", fontsize=15)

# Correlation of Alcohol and Quality
sns.jointplot(x = "alcohol", y = "quality",
              kind = "reg", data = data_df_no_va_outlier,
              dropna = True,color='sandybrown')
plt.suptitle("Correlation of Alcohol and Quality")
```

Out[ ]:

```
Text(0.5, 0.98, 'Correlation of Alcohol and Quality')
```



IV: Insights

a) **Distribution Chart of Alcohol:**
We have removed the outliers for fixed acidity to allow a clearer distribution to be formed. The distribution of alcohol ranges from 8.4 to 13.4 g / dm^3 and most of the wine is found to have acidity of 10.0 to 11.5 vol%.

b)**Distribution Chart of Quality:**
We can see that the chart is middle-skewed where most of the wine quality ranges from 5 to 6 which implies that most of the red wine possess middle quality.

c) **Boxplot for Alcohol:**
For a better wine quality of 7 to 8, the volume of alcohol ranges from 10.8 to 12.1 g/dm^3. Whereas for poorer wine quality of 3 to 4, the volume of alcohol ranges from 9.6 to 11.0 g/dm^3. By looking at these values, a clear relationship could not be defined since they intersects each other and there is only a small difference in the values.

Nevertheless, by looking at the average alcohol content across each quality, it shows a positive relationship between alcohol volume and quality where the quality increases with alcohol. However, both good quality wines (rated 7&8) as well as low quality wine (3 & 4) has a similar average alcohol content when being compared within their categories. This shows that a good quality of wine ,be it rating of 7 or 8 has the same alcohol content. Likewise for a bad quality of wine rated 3 and 4.

d)**Joint Plot for Alcohol and Quality:**
We have combined the two distribution charts together to see their relationship. We learned that there is a strong positive relationship between alcohol and quality as the orange line shows a very steep positive linear graph. Thus, indicating that there is indeed a positive correlation relationship between the two variables.

Recommendations
Hence, we suggest our client, Jewel wine to increase the composition of alcohol to produce a good quality wine. A good amount of alcohol composition would be around 10.8 to 12.1 g/dm^3 to produce a good quality wine.

4. Wine Quality Prediction Model

From our analysis above, we now have a better understanding on the relationship between each wine's attribute and quality.

As our objective is to help Jewel Wine better understand what affects the quality of wine and improve their wine manufacturing process, in this section we will look at how we can better predict the Red Wine quality given the factors we had analyzed.

Approach: Develop a Decision Tree

I: Encoding quality in "Good" OR "Bad"

In this section, we categorised the wine quality into 2 sections, good and bad quality wine for clearer wine prediction later on.

In [ ]:

```python
new_df=data_df[["fixed acidity","volatile acidity","citric acid","residual sugar","chlorides","free sulfur dioxide","total sulfu

def quality_separator(quality):
    if quality>=7:
        return "Good"
    else:
        return "Bad"

new_df["quality_cat"]=new_df["quality"].apply(quality_separator)
new_df
```

Out[ ]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality	quality_cat
0	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56	9.4	5	Bad
1	7.8	0.880	0.00	2.6	0.098	25.0	67.0	0.99680	3.20	0.68	9.8	5	Bad
2	7.8	0.760	0.04	2.3	0.092	15.0	54.0	0.99700	3.26	0.65	9.8	5	Bad
3	11.2	0.280	0.56	1.9	0.075	17.0	60.0	0.99800	3.16	0.58	9.8	6	Bad
4	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56	9.4	5	Bad
...	...	...	...	...	...	...	...	...	...	...	...	...	...
1594	6.2	0.600	0.08	2.0	0.090	32.0	44.0	0.99490	3.45	0.58	10.5	5	Bad
1595	5.9	0.550	0.10	2.2	0.062	39.0	51.0	0.99512	3.52	0.76	11.2	6	Bad
1596	6.3	0.510	0.13	2.3	0.076	29.0	40.0	0.99574	3.42	0.75	11.0	6	Bad
1597	5.9	0.645	0.12	2.0	0.075	32.0	44.0	0.99547	3.57	0.71	10.2	5	Bad
1598	6.0	0.310	0.47	3.6	0.067	18.0	42.0	0.99549	3.39	0.66	11.0	6	Bad

1599 rows × 13 columns

II: SciKit Learn to train and test our datasets

In [ ]:

```python
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn import datasets, linear_model
```

In [ ]:

```python
features = ["fixed acidity","volatile acidity","citric acid","residual sugar","chlorides","free sulfur dioxide","total sulfur dio

df_x = new_df[features]
df_y = new_df['quality_cat']

df_x.head()
df_y.head()

print( set(df_x) )
print( set(df_y) )

from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test = train_test_split(df_x,
                                                    df_y,
                                                    test_size=0.20, # 20% Testing Data
                                                    random_state=10)

display(x_train.head())
display(y_train.head())
```

```
{'sulphates', 'chlorides', 'pH', 'citric acid', 'volatile acidity', 'fixed acidity', 'free sulfur dioxide', 'total sulfur dioxid
e', 'density', 'alcohol', 'residual sugar'}
{'Bad', 'Good'}
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol
1407	6.0	0.51	0.00	2.10	0.064	40.0	54.0	0.99500	3.54	0.93	10.7
1202	8.6	0.42	0.39	1.80	0.068	6.0	12.0	0.99516	3.35	0.69	11.7
245	7.3	0.66	0.00	2.00	0.084	6.0	23.0	0.99830	3.61	0.96	9.9
1002	9.1	0.29	0.33	2.05	0.063	13.0	27.0	0.99516	3.26	0.84	11.7
939	6.2	0.46	0.17	1.60	0.073	7.0	11.0	0.99425	3.61	0.54	11.4

```
1407     Bad
1202    Good
245      Bad
1002    Good
939      Bad
Name: quality_cat, dtype: object
```

III: Growing Decision Tree

In [ ]:

```python
#FULL TREE
```

In [ ]:

```python
from sklearn.tree import DecisionTreeClassifier
factor_dtree=DecisionTreeClassifier(criterion="entropy") #entropy--> KPI-->measures how chaotic the data is
factor_dtree.fit(x_train,y_train)

print("Depth of Tree",factor_dtree.tree_.max_depth)
```

```
Depth of Tree 13
```

In [ ]:

```python
#PRUNED TREE
```

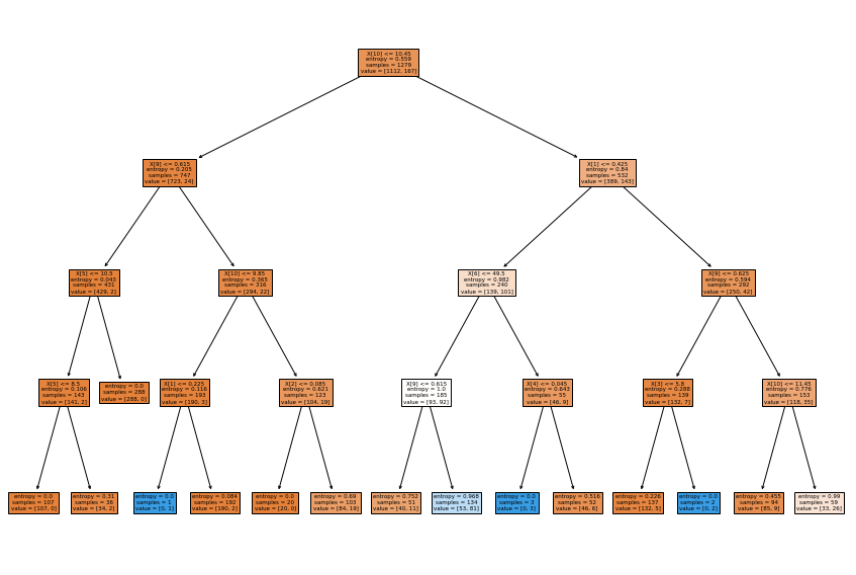
In [ ]:

```python
pruned_dtree=DecisionTreeClassifier(criterion="entropy",max_depth=4) #Change depth of tree

pruned_dtree.fit(x_train,y_train)
print("Depth of Tree",pruned_dtree.tree_.max_depth)

from sklearn import tree
from matplotlib.pyplot import figure

figure(figsize=(15,10))
tree.plot_tree(pruned_dtree, filled=True, impurity=True)
```

```
Depth of Tree 4
```

Out[ ]:

```
[Text(381.13392857142856, 489.24, 'X[10] <= 10.45\nentropy = 0.559\nsamples = 1279\nvalue = [1112, 167]'),
 Text(164.41071428571428, 380.52000000000004, 'X[9] <= 0.615\nentropy = 0.205\nsamples = 747\nvalue = [723, 24]'),
 Text(89.67857142857143, 271.8, 'X[5] <= 10.5\nentropy = 0.043\nsamples = 431\nvalue = [429, 2]'),
 Text(59.785714285714285, 163.08000000000004, 'X[5] <= 8.5\nentropy = 0.106\nsamples = 143\nvalue = [141, 2]'),
 Text(29.892857142857142, 54.360000000000014, 'entropy = 0.0\nsamples = 107\nvalue = [107, 0]'),
 Text(89.67857142857143, 54.360000000000014, 'entropy = 0.31\nsamples = 36\nvalue = [34, 2]'),
 Text(119.57142857142857, 163.08000000000004, 'entropy = 0.0\nsamples = 288\nvalue = [288, 0]'),
 Text(239.14285714285714, 271.8, 'X[10] <= 9.85\nentropy = 0.365\nsamples = 316\nvalue = [294, 22]'),
 Text(179.35714285714286, 163.08000000000004, 'X[1] <= 0.225\nentropy = 0.116\nsamples = 193\nvalue = [190, 3]'),
 Text(149.46428571428572, 54.360000000000014, 'entropy = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(209.25, 54.360000000000014, 'entropy = 0.084\nsamples = 192\nvalue = [190, 2]'),
 Text(298.92857142857144, 163.08000000000004, 'X[2] <= 0.085\nentropy = 0.621\nsamples = 123\nvalue = [104, 19]'),
 Text(269.0357142857143, 54.360000000000014, 'entropy = 0.0\nsamples = 20\nvalue = [20, 0]'),
 Text(328.82142857142856, 54.360000000000014, 'entropy = 0.69\nsamples = 103\nvalue = [84, 19]'),
 Text(597.8571428571429, 380.52000000000004, 'X[1] <= 0.425\nentropy = 0.84\nsamples = 532\nvalue = [389, 143]'),
 Text(478.2857142857143, 271.8, 'X[6] <= 49.5\nentropy = 0.982\nsamples = 240\nvalue = [139, 101]'),
 Text(418.5, 163.08000000000004, 'X[9] <= 0.615\nentropy = 1.0\nsamples = 185\nvalue = [93, 92]'),
 Text(388.60714285714283, 54.360000000000014, 'entropy = 0.752\nsamples = 51\nvalue = [40, 11]'),
 Text(448.3928571428571, 54.360000000000014, 'entropy = 0.968\nsamples = 134\nvalue = [53, 81]'),
 Text(538.0714285714286, 163.08000000000004, 'X[4] <= 0.045\nentropy = 0.643\nsamples = 55\nvalue = [46, 9]'),
 Text(508.17857142857144, 54.360000000000014, 'entropy = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(567.9642857142857, 54.360000000000014, 'entropy = 0.516\nsamples = 52\nvalue = [46, 6]'),
 Text(717.4285714285714, 271.8, 'X[9] <= 0.625\nentropy = 0.594\nsamples = 292\nvalue = [250, 42]'),
 Text(657.6428571428571, 163.08000000000004, 'X[3] <= 5.8\nentropy = 0.288\nsamples = 139\nvalue = [132, 7]'),
 Text(627.75, 54.360000000000014, 'entropy = 0.226\nsamples = 137\nvalue = [132, 5]'),
 Text(687.5357142857142, 54.360000000000014, 'entropy = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(777.2142857142857, 163.08000000000004, 'X[10] <= 11.45\nentropy = 0.776\nsamples = 153\nvalue = [118, 35]'),
 Text(747.3214285714286, 54.360000000000014, 'entropy = 0.455\nsamples = 94\nvalue = [85, 9]'),
 Text(807.1071428571429, 54.360000000000014, 'entropy = 0.99\nsamples = 59\nvalue = [33, 26]')]
```



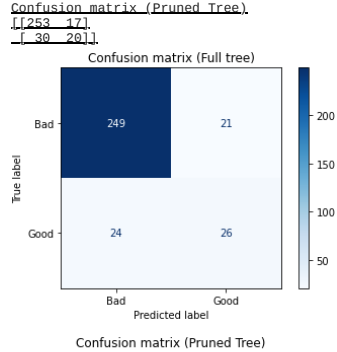
In [ ]:

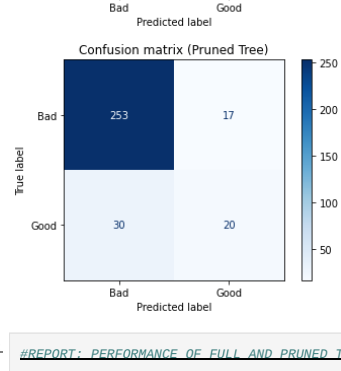
```python
#COMPARING FULL TREE & PRUNED TREE  using Confusion Matrix
```

In [ ]:

```python
from sklearn.metrics import plot_confusion_matrix
full_results = factor_dtree.predict(x_test)
pruned_results = pruned_dtree.predict(x_test)
cm_f = confusion_matrix(y_test, full_results) #(actual,prediction)
cm_p = confusion_matrix(y_test, pruned_results)

titles_options = [("Confusion matrix (Full tree)", factor_dtree),
                  ("Confusion matrix (Pruned Tree)", pruned_dtree)]

for title, classifier in titles_options:
    disp = plot_confusion_matrix(classifier, x_test, y_test,
                                 cmap=plt.cm.Blues,
                                 normalize=None)
    disp.ax_.set_title(title)
    print(title)
    print(disp.confusion_matrix)

plt.show()
```

```
Confusion matrix (Full tree)
[[249  21]
 [ 24  26]]
Confusion matrix (Pruned Tree)
[[253  17]
 [ 30  20]]
```





In [ ]:

```python
#REPORT: PERFORMANCE OF FULL AND PRUNED TREE
```

In [ ]:

```python
from sklearn.metrics import classification_report
print()
print("----Full Tree----")
print(classification_report(y_test,full_results))
print()
print("----Pruned Tree----")
print(classification_report(y_test,pruned_results))
```

```
----Full Tree----
              precision    recall  f1-score   support

         Bad       0.91      0.92      0.92       270
        Good       0.55      0.52      0.54        50

    accuracy                           0.86       320
   macro avg       0.73      0.72      0.73       320
weighted avg       0.86      0.86      0.86       320


----Pruned Tree----
              precision    recall  f1-score   support

         Bad       0.89      0.94      0.92       270
        Good       0.54      0.40      0.46        50

    accuracy                           0.85       320
   macro avg       0.72      0.67      0.69       320
weighted avg       0.84      0.85      0.84       320
```

In [ ]:

```python
print("Full tree accuracy", factor_dtree.score(x_test, y_test))
print("Pruned tree accuracy", pruned_dtree.score(x_test, y_test))
```

```
Full tree accuracy 0.859375
Pruned tree accuracy 0.853125
```

Insights

Based on the classification report as shown above:

1. Pruned tree has a lower precision of 0.89 than the full tree which has a precision of 0.91
2. Pruned tree has a higher recall of 0.94 than full tree which has a recall of 0.92.
3. Pruned tree has a lower accuracy of 0.853 than full tree which has a accuracy of 0.859.

Although full tree has a higher precision and accuracy, **we would be using the pruned decision tree to make wine quality prediction as we are dealing with beverages for human consumption in this case Hence, recall is more important than precision n accuracy in this case. If there is bad quality wine that we did not managed to spot ,it could possibly lead to some health issues in the people who drank Jewel wine"s red wine.

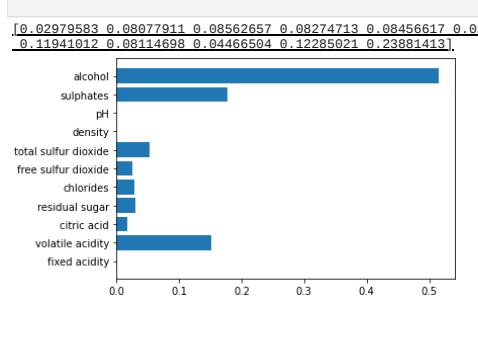
In [ ]:

```python
#FEATURE IMPORTANCE
```

In [ ]:

```python
import matplotlib.pyplot as plt

fig, ax = plt.subplots()
ax.barh(features, pruned_dtree.feature_importances_)
print(factor_dtree.feature_importances_)
```

```
[0.02979583 0.08077911 0.06562657 0.08274713 0.08456617 0.02959871
 0.11941012 0.08114698 0.04466504 0.12285021 0.23881413]
```



Insights

- Alcohol has highest feature importance of 0.24
- Sulphate and Volatile Acidity have around the same feature importance ranging aound 0.11.
- pH & Density has absolutely no feature importance.

Recommendations</u>
As shown in the above bar chart, alcohol is the most important factor in manufacturing a good quality, followed by sulphates volatile acidity and citric acid. Hence, we would recommend our client ,Jewel wine to focus most on getting the right alcohol content (recommended alcohol content is stated in previous section) to produce a good quality wine. Also, they can put minimal focus on the compositions of pH & density as it does not really affect the quality of red wine.