Student ID: DH20042

Name : 弗兰克

Data analysis

1. Wine quality analysis

In [341]:
```
#Load module and pkg
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV
import warnings
warnings.filterwarnings('ignore')
```

In [187]:
```
# Load dataframe
try:
    wine = pd.read_csv('winequality-red.csv', sep =';')
except:
    print("cannot find the file")
```

In [188]:
```
# Check the info of the dataframe
wine.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 [189]:
```
# let us visualize the dataframe
wine
```

Out[189]:

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

1599 rows × 12 columns

In [190]:
```
# Check duplicated wine information records

wine.duplicated().sum()
```

Out[190]:
```
240
```

In [191]:
```
#drop all duplicated records
wine.drop_duplicates()
```

Out[191]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56	9.4	5
1	7.8	0.880	0.00	2.6	0.098	25.0	67.0	0.99680	3.20	0.68	9.8	5
2	7.8	0.760	0.04	2.3	0.092	15.0	54.0	0.99700	3.26	0.65	9.8	5
3	11.2	0.280	0.56	1.9	0.075	17.0	60.0	0.99800	3.16	0.58	9.8	6
5	7.4	0.660	0.00	1.8	0.075	13.0	40.0	0.99780	3.51	0.56	9.4	5
...	...	...	...	...	...	...	...	...	...	...	...	...
1593	6.8	0.620	0.08	1.9	0.068	28.0	38.0	0.99651	3.42	0.82	9.5	6
1594	6.2	0.600	0.08	2.0	0.090	32.0	44.0	0.99490	3.45	0.58	10.5	5
1595	5.9	0.550	0.10	2.2	0.062	39.0	51.0	0.99512	3.52	0.76	11.2	6
1597	5.9	0.645	0.12	2.0	0.075	32.0	44.0	0.99547	3.57	0.71	10.2	5
1598	6.0	0.310	0.47	3.6	0.067	18.0	42.0	0.99549	3.39	0.66	11.0	6

1359 rows × 12 columns

In [192]:
```
# Check the basic statistics of the data
wine.describe()
```

Out[192]:

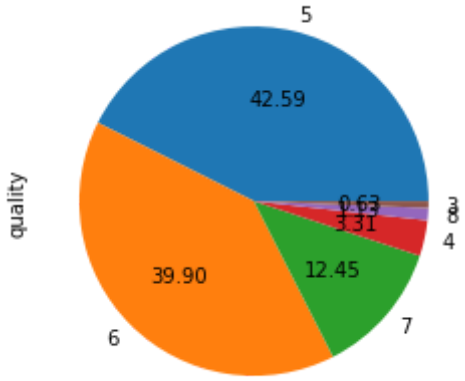
	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphate
count	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.00000
mean	8.319637	0.527821	0.270976	2.538806	0.087467	15.874922	46.467792	0.996747	3.311113	0.65814
std	1.741096	0.179060	0.194801	1.409928	0.047065	10.460157	32.895324	0.001887	0.154386	0.16950
min	4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	6.000000	0.990070	2.740000	0.33000
25%	7.100000	0.390000	0.090000	1.900000	0.070000	7.000000	22.000000	0.995600	3.210000	0.55000
50%	7.900000	0.520000	0.260000	2.200000	0.079000	14.000000	38.000000	0.996750	3.310000	0.62000
75%	9.200000	0.640000	0.420000	2.600000	0.090000	21.000000	62.000000	0.997835	3.400000	0.73000
max	15.900000	1.580000	1.000000	15.500000	0.611000	72.000000	289.000000	1.003690	4.010000	2.00000

In [193]:
```
#Check the values of each category of the quality attribute has
wine.quality.value_counts()
```

Out[193]:
```
5    681
6    638
7    199
4     53
8     18
3     10
Name: quality, dtype: int64
```

In [194]:
```
# Let plot the quality values
wine.quality.value_counts().plot(kind = 'pie', autopct = '%.2f')
```

Out[194]: <AxesSubplot:ylabel='quality'>

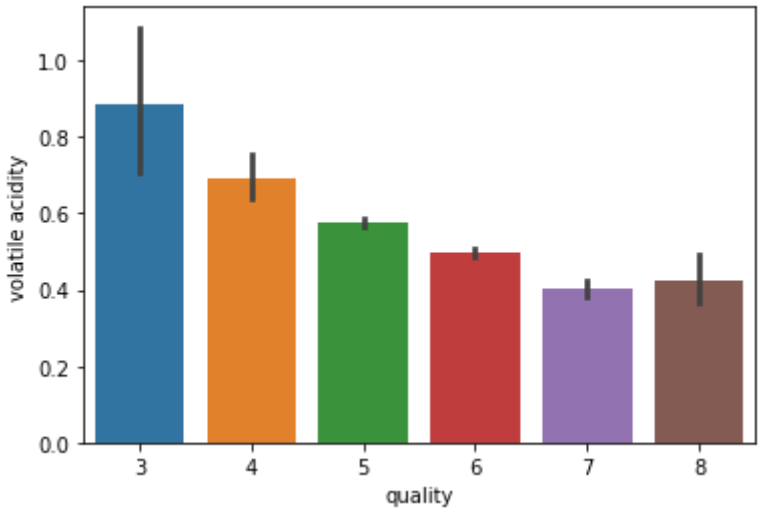


In [195]:
```
#Check the correlation
wine.corr().quality
```

Out[195]:
```
fixed acidity           0.124052
volatile acidity       -0.390558
citric acid             0.226373
residual sugar          0.013732
chlorides              -0.128907
free sulfur dioxide    -0.050656
total sulfur dioxide   -0.185100
density                -0.174919
pH                     -0.057731
sulphates               0.251397
alcohol                 0.476166
quality                 1.000000
Name: quality, dtype: float64
```

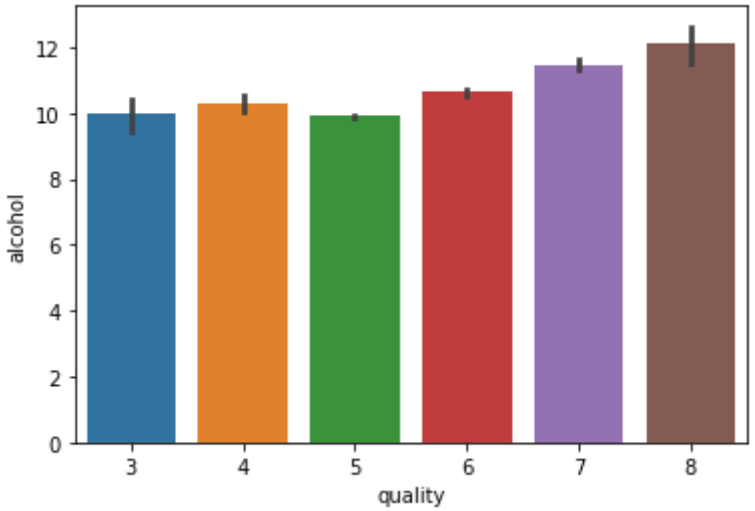
In [198]:
```
# Check the distribution mean
sns.barplot(x = 'quality', y = 'volatile acidity', data = wine)
```

Out[198]:
```
<AxesSubplot:xlabel='quality', ylabel='volatile acidity'>
```



In [199]:
```
sns.barplot(x = 'quality', y = 'alcohol', data = wine)
```

Out[199]: <AxesSubplot:xlabel='quality', ylabel='alcohol'>



In [340]:
```
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import scale
from sklearn.metrics import confusion_matrix
from sklearn.preprocessing import StandardScaler
from pandas.plotting import scatter_matrix
from sklearn.pipeline import make_pipeline
```

In [242]:
```
# Now seperate the dataset as response variable and feature variabes
X = wine.drop('quality', axis=1)
y = wine['quality']
# Train and Test splitting of data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=50)
# Applying Standard scaling to get optimized result
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.fit_transform(X_test)
```

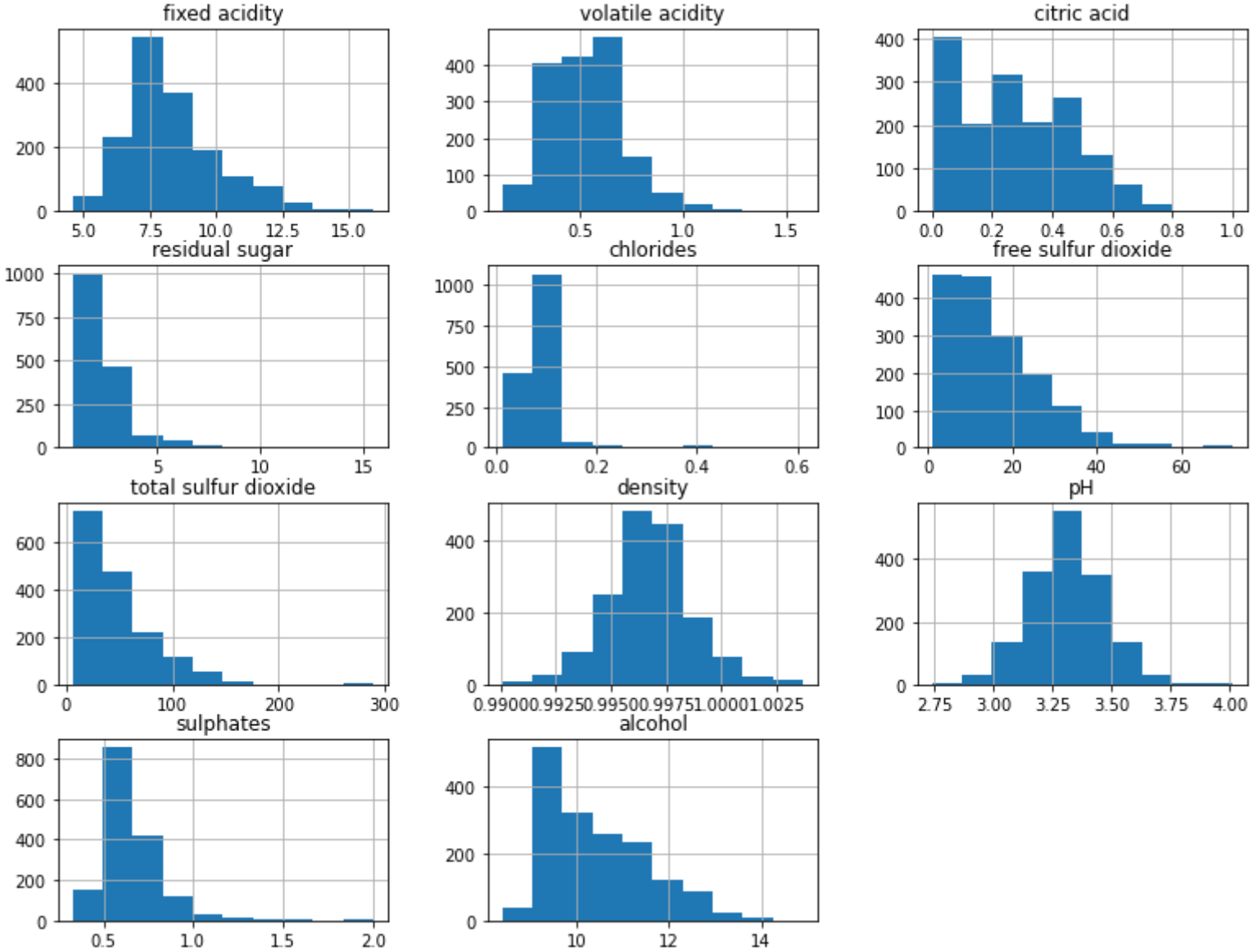
In [243]:
```
# Statistical characteristics of each numerical feature
print(wine.describe())
```

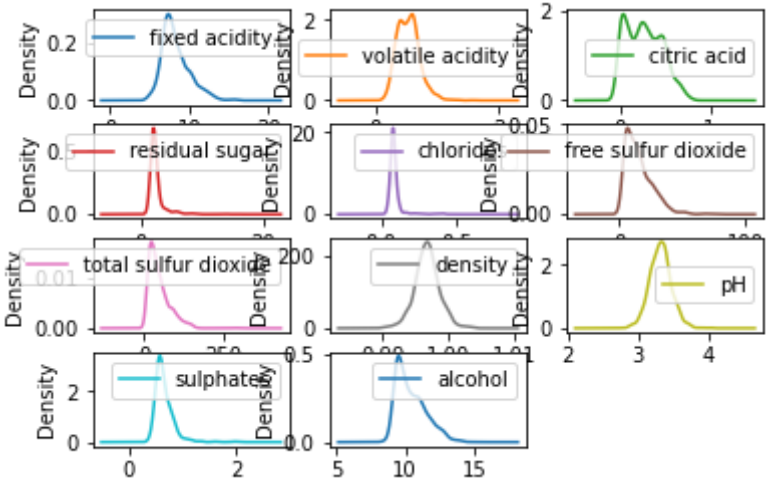
```
       fixed acidity  volatile acidity  citric acid  residual sugar  \
count    1599.000000       1599.000000  1599.000000     1599.000000
mean        8.319637          0.527821     0.270976        2.538806
std         1.741096          0.179060     0.194801        1.409928
min         4.600000          0.120000     0.000000        0.900000
25%         7.100000          0.390000     0.090000        1.900000
50%         7.900000          0.520000     0.260000        2.200000
75%         9.200000          0.640000     0.420000        2.600000
max        15.900000          1.580000     1.000000       15.500000

         chlorides  free sulfur dioxide  total sulfur dioxide      density  \
count  1599.000000          1599.000000           1599.000000  1599.000000
mean      0.087467            15.874922             46.467792     0.996747
std       0.047065            10.460157             32.895324     0.001887
min       0.012000             1.000000              6.000000     0.990070
25%       0.070000             7.000000             22.000000     0.995600
50%       0.079000            14.000000             38.000000     0.996750
75%       0.090000            21.000000             62.000000     0.997835
max       0.611000            72.000000            289.000000     1.003690

                pH    sulphates      alcohol
count  1599.000000  1599.000000  1599.000000
mean      3.311113     0.658149    10.422983
std       0.154386     0.169507     1.065668
min       2.740000     0.330000     8.400000
25%       3.210000     0.550000     9.500000
50%       3.310000     0.620000    10.200000
75%       3.400000     0.730000    11.100000
max       4.010000     2.000000    14.900000
```

In [254]:
```
# Histograms
wine.hist(bins=10,figsize=(13, 10))
plt.show()
# Density
wine.plot(kind='density', subplots=True, layout=(4,3), sharex=False)
plt.show()
```




fixed acidity
volatile acidity
citric acid
residual sugar
chlorides
free sulfur dioxide
total sulfur dioxide
density
pH
sulphates
alcohol
Density

In [255]:
```
# Create pivot_table
group_names = ['fixed acidity', 'volatile acidity', 'citric acid', 'residual sugar', 'chlorides', 'free
wine_pivot_table = wine.pivot_table(group_names,
               ['quality'], aggfunc='median')
print(wine_pivot_table)
```

```
         alcohol  chlorides  citric acid  density  fixed acidity  \
quality
low         10.0      0.080         0.08  0.99660            7.5
medium      10.0      0.080         0.24  0.99680            7.8
high        11.6      0.073         0.40  0.99572            8.7

         free sulfur dioxide    pH  residual sugar  sulphates  \
quality
low                      9.0  3.38             2.1       0.56
medium                  14.0  3.31             2.2       0.61
high                    11.0  3.27             2.3       0.74

         total sulfur dioxide  volatile acidity
quality
low                      26.0              0.68
medium                   40.0              0.54
high                     27.0              0.37
```

In [262]:
```
corr_matrix = wine.corr()
corr_matrix
```

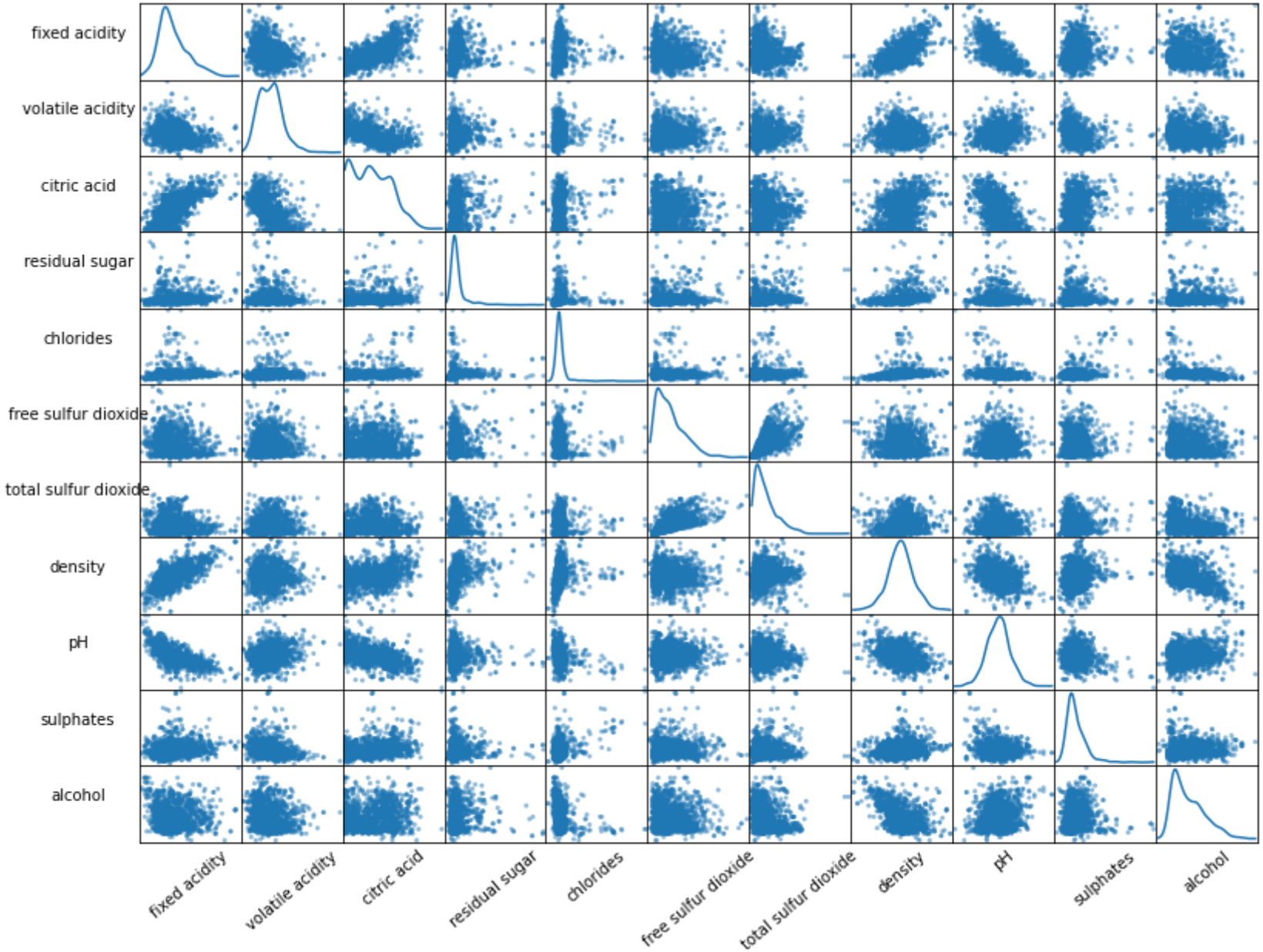
Out[262]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol
fixed acidity	1.000000	-0.256131	0.671703	0.114777	0.093705	-0.153794	-0.113181	0.668047	-0.682978	0.183006	-0.061668
volatile acidity	-0.256131	1.000000	-0.552496	0.001918	0.061298	-0.010504	0.076470	0.022026	0.234937	-0.260987	-0.202288
citric acid	0.671703	-0.552496	1.000000	0.143577	0.203823	-0.060978	0.035533	0.364947	-0.541904	0.312770	0.109903
residual sugar	0.114777	0.001918	0.143577	1.000000	0.055610	0.187049	0.203028	0.355283	-0.085652	0.005527	0.042075
chlorides	0.093705	0.061298	0.203823	0.055610	1.000000	0.005562	0.047400	0.200632	-0.265026	0.371260	-0.221141
free sulfur dioxide	-0.153794	-0.010504	-0.060978	0.187049	0.005562	1.000000	0.667666	-0.021946	0.070377	0.051658	-0.069408
total sulfur dioxide	-0.113181	0.076470	0.035533	0.203028	0.047400	0.667666	1.000000	0.071269	-0.066495	0.042947	-0.205654
density	0.668047	0.022026	0.364947	0.355283	0.200632	-0.021946	0.071269	1.000000	-0.341699	0.148506	-0.496180
pH	-0.682978	0.234937	-0.541904	-0.085652	-0.265026	0.070377	-0.066495	-0.341699	1.000000	-0.196648	0.205633
sulphates	0.183006	-0.260987	0.312770	0.005527	0.371260	0.051658	0.042947	0.148506	-0.196648	1.000000	0.093595
alcohol	-0.061668	-0.202288	0.109903	0.042075	-0.221141	-0.069408	-0.205654	-0.496180	0.205633	0.093595	1.000000

In [344]:
```
group_names = ['fixed acidity', 'volatile acidity', 'citric acid', 'residual sugar', 'chlorides', 'free
# Correlation matrix
correlations = wine.corr()
# Plot figsize
fig, ax = plt.subplots(figsize=(10, 10))
# Generate Color Map
colormap = sns.diverging_palette(220, 10, as_cmap=True)
# Generate Heat Map, allow annotations and place floats in map
sns.heatmap(correlations, cmap=colormap, annot=True, fmt=".2f")
plt.show()
```


fixed acidity 1.00 -0.26 0.67 0.11 0.09 -0.15 -0.11 0.67 -0.68 0.18 -0.06
volatile acidity -0.26 1.00 -0.55 0.00 0.06 -0.01 0.08 0.02 0.23 -0.26 -0.20
citric acid 0.67 -0.55 1.00 0.14 0.20 -0.06 0.04 0.36 -0.54 0.31 0.11
residual sugar 0.11 0.00 0.14 1.00 0.06 0.19 0.20 0.36 -0.09 0.01 0.04
chlorides 0.09 0.06 0.20 0.06 1.00 0.01 0.05 0.20 -0.27 0.37 -0.22
free sulfur dioxide -0.15 -0.01 -0.06 0.19 0.01 1.00 0.67 -0.02 0.07 0.05 -0.07
total sulfur dioxide -0.11 0.08 0.04 0.20 0.05 0.67 1.00 0.07 -0.07 0.04 -0.21
density 0.67 0.02 0.36 0.36 0.20 -0.02 0.07 1.00 -0.34 0.15 -0.50
pH -0.68 0.23 -0.54 -0.09 -0.27 0.07 -0.07 -0.34 1.00 -0.20 0.21
sulphates 0.18 -0.26 0.31 0.01 0.37 0.05 0.04 0.15 -0.20 1.00 0.09
alcohol -0.06 -0.20 0.11 0.04 -0.22 -0.07 -0.21 -0.50 0.21 0.09 1.00
fixed acidity
volatile acidity
citric acid
residual sugar
chlorides
free sulfur dioxide
total sulfur dioxide
density
pH
sulphates
alcohol
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6

In [345]:
```
# Scatterplot Matrix
sm = scatter_matrix(wine, figsize=(13, 10), diagonal='kde')
#Change label rotation
[s.xaxis.label.set_rotation(40) for s in sm.reshape(-1)]
[s.yaxis.label.set_rotation(0) for s in sm.reshape(-1)]
#May need to offset label when rotating to prevent overlap of figure
[s.get_yaxis().set_label_coords(-0.6,0.5) for s in sm.reshape(-1)]
#Hide all ticks
[s.set_xticks(()) for s in sm.reshape(-1)]
[s.set_yticks(()) for s in sm.reshape(-1)]
plt.show()
```


fixed acidity
volatile acidity
citric acid
residual sugar
chlorides
free sulfur dioxide
total sulfur dioxide
density
pH
sulphates
alcohol

In [347]:
```
# Dividing wine as low, medium and high by giving the limit for the quality
bins = (2,4,6,8)
group_names = ['low','medium', 'high']
wine['quality'] = pd.cut(wine['quality'], bins = bins, labels = group_names)
# Now lets assign a labels to our quality variable
label_quality = LabelEncoder()
wine['quality'] = label_quality.fit_transform(wine['quality'])
'''
```

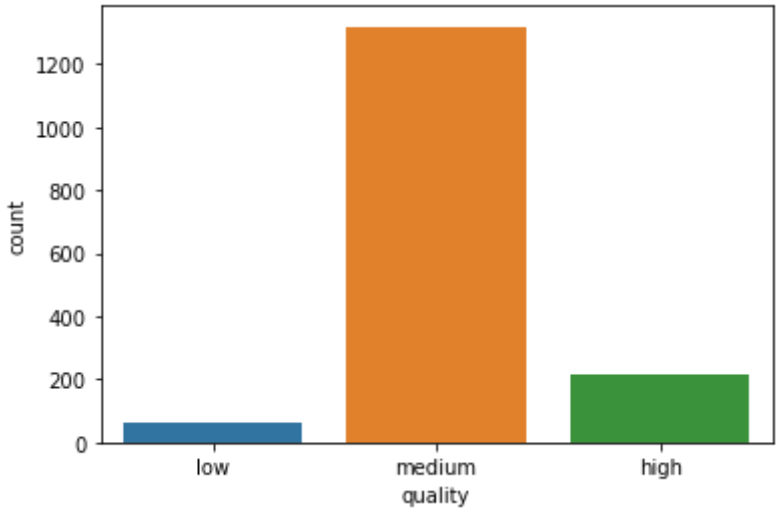
```
  File "/var/folders/57/yhl6qkcj0wsd5j12rr54v5400000gn/T/ipykernel_40670/1689526706.py", line 9
    '''
       ^
SyntaxError: EOF while scanning triple-quoted string literal
```

In [343]:
```
wine.quality.value_counts()
```

Out[343]:
```
medium    1319
high       217
low         63
Name: quality, dtype: int64
```

In [289]:
```
sns.countplot(wine['quality'])
```

Out[289]: <AxesSubplot:xlabel='quality', ylabel='count'>



In [321]:
```
X = wine.iloc[:,:-1]
y = wine.quality
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.3)
```

In [322]:
```
X_train = scale(X_train)
X_test = scale (X_test)
```

In [323]:
```
rfc = RandomForestClassifier(n_estimators = 200)
rfc.fit(X_train, y_train)
y_pred = rfc.predict(X_test)
print(confusion_matrix(y_test,y_pred))
```

```
[[ 27   0  30]
 [  0   0  17]
 [ 19   0 387]]
```

In [ ]: