```
# Load the dataset from CSV
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
%matplotlib inline

# Load the data
df = pd.read_csv('wine.CSV')
df.head()
```

```
   fixed acidity  volatile acidity  citric acid  residual sugar
chlorides  \
0            7.4              0.70         0.00             1.9
0.076
1            7.8              0.88         0.00             2.6
0.098
2            7.8              0.76         0.04             2.3
0.092
3           11.2              0.28         0.56             1.9
0.075
4            7.4              0.70         0.00             1.9
0.076

   free sulfur dioxide  total sulfur dioxide  density    pH  sulphates
\
0                 11.0                  34.0   0.9978  3.51       0.56

1                 25.0                  67.0   0.9968  3.20       0.68

2                 15.0                  54.0   0.9970  3.26       0.65

3                 17.0                  60.0   0.9980  3.16       0.58

4                 11.0                  34.0   0.9978  3.51       0.56


   alcohol  quality
0      9.4        5
1      9.8        5
2      9.8        5
3      9.8        6
4      9.4        5
```

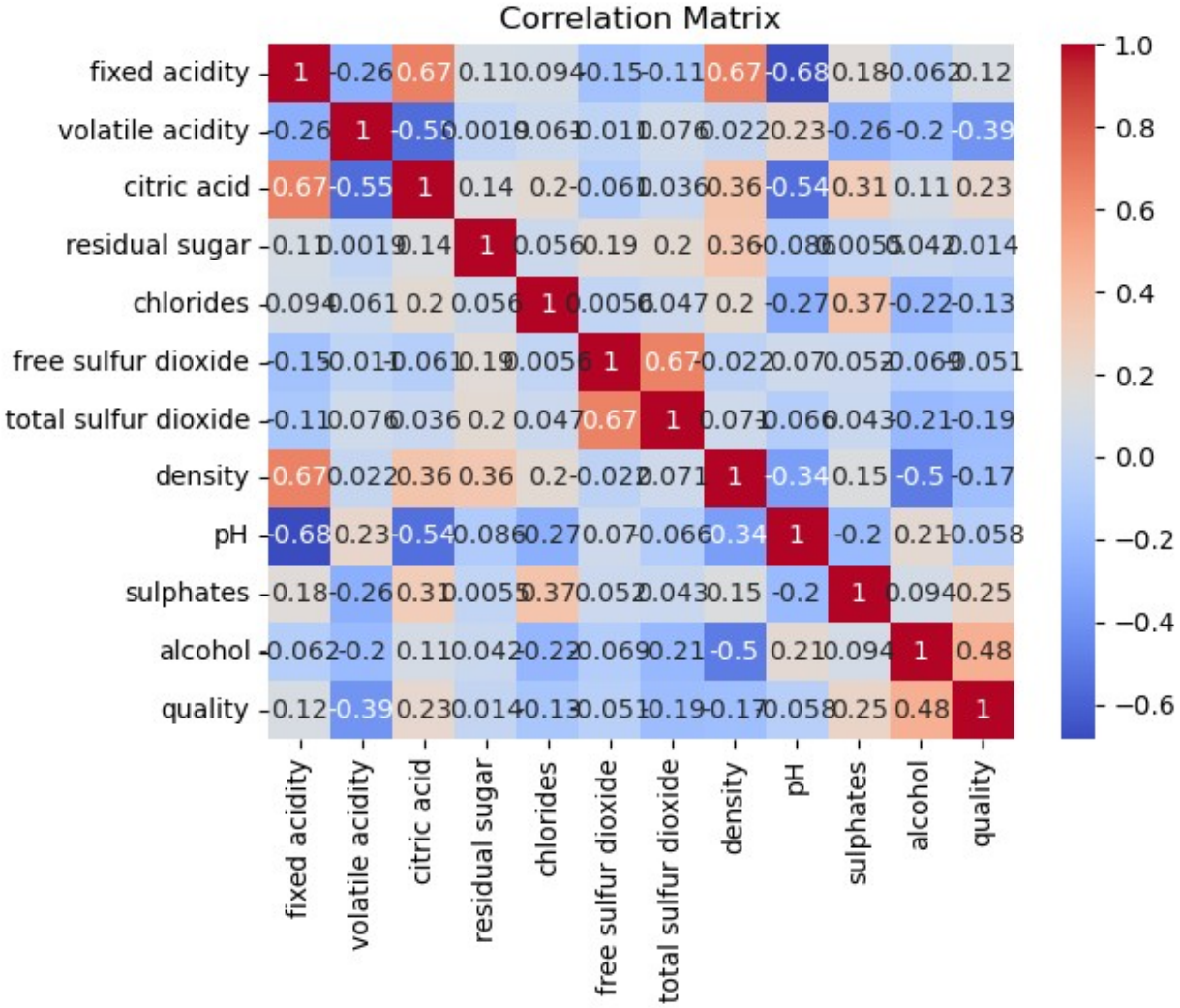
```
# Check for missing values
df.isnull().sum()
```

```
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

# Summary statistics of the dataset
print(df.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      quality
count  1599.000000  1599.000000  1599.000000  1599.000000
mean      3.311113     0.658149    10.422983     5.636023
std       0.154386     0.169507     1.065668     0.807569
min       2.740000     0.330000     8.400000     3.000000
25%       3.210000     0.550000     9.500000     5.000000
50%       3.310000     0.620000    10.200000     6.000000
```

```
75%         3.400000      0.730000     11.100000      6.000000
max         4.010000      2.000000     14.900000      8.000000
```

```
# Correlation matrix to understand feature relationships
correlation_matrix = df.corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title("Correlation Matrix")
plt.show()
```




```
# Preprocessing: Selecting features and target variable
X = df[['fixed acidity', 'volatile acidity', 'citric acid', 'residual
sugar', 'chlorides', 'free sulfur dioxide', 'total sulfur dioxide',
'density','pH','sulphates','alcohol']]
y = df['quality']
```

```
# Splitting the dataset into training and testing sets
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Building the Linear Regression Model
from sklearn.linear_model import LinearRegression
model = LinearRegression()

model.fit(X_train, y_train)

LinearRegression()

y_pred = model.predict(X_test)
y_pred

array([5.34666441, 5.05631345, 5.66446972, 5.46451484, 5.72518476,
       5.27928659, 5.03421667, 5.12623347, 5.74534288, 5.68665032,
       6.13959677, 5.23386892, 5.54991474, 5.25825299, 5.44810502,
       6.46828999, 5.15018088, 5.59105157, 6.5560658 , 5.32255751,
       5.3918385 , 5.19610791, 5.94475739, 6.36197631, 5.35484893,
       5.41907575, 6.36483321, 5.35121573, 5.172392  , 6.16987311,
       5.25263058, 5.50657406, 5.75422105, 5.39101712, 5.45331031,
       5.02757499, 6.16173243, 5.68661555, 5.6486077 , 6.165471  ,
       5.52872593, 5.24414488, 6.17724727, 5.16500868, 5.87598332,
       5.81317121, 6.41982782, 5.6059474 , 5.15232137, 5.55634632,
       5.16044852, 5.10449459, 5.58371721, 6.33425313, 4.95134985,
       4.98364804, 6.01041999, 5.40809804, 5.83802638, 5.2486897 ,
       5.60717482, 5.96630957, 5.27619063, 5.30380113, 6.4949309 ,
       5.42033967, 6.34273471, 5.24618531, 6.41317317, 5.31237924,
       6.41746963, 4.74315748, 5.79362039, 5.8283184 , 6.17598768,
       5.29723707, 6.76198733, 5.89745261, 6.07833712, 6.43522754,
       5.29499011, 6.4546625 , 5.45007864, 5.69644693, 5.72368681,
       6.41233601, 5.31025119, 5.84548953, 6.31433877, 5.20585049,
       6.10141578, 5.70349712, 5.78679322, 5.93173502, 5.1852885 ,
       5.74819506, 5.17351769, 5.69336056, 4.99158806, 5.52004223,
       5.06867029, 5.13831807, 5.84991801, 5.72612872, 5.47766711,
       6.12476389, 5.73551897, 5.44180611, 6.08785125, 5.24667513,
       6.68434941, 5.26499691, 6.15359147, 4.74493131, 5.82508834,
       5.9872331 , 6.17033538, 5.50859099, 5.02156367, 5.83326942,
       6.21086737, 5.26363047, 5.75354145, 5.38942262, 5.39641713,
       5.25966957, 6.21024761, 5.69536196, 5.58586923, 5.82155344,
       5.79362039, 5.14962195, 5.01142496, 6.34824026, 5.55634632,
       5.08213438, 5.05668453, 5.3517036 , 5.11920475, 5.66948552,
       6.01614582, 6.03912287, 6.2439487 , 5.48155178, 5.86335248,
       5.26302973, 6.06162683, 5.4041289 , 5.99869245, 5.06897434,
       5.70161041, 6.14167652, 5.11821365, 5.67658854, 5.79362039,
       6.0891404 , 5.22103588, 5.90134727, 5.48941228, 5.93412645,
       6.3118134 , 5.71785286, 6.13152024, 4.9898825 , 5.39143155,
       5.63146602, 4.70626967, 5.232132  , 5.04110749, 4.99137335,
```

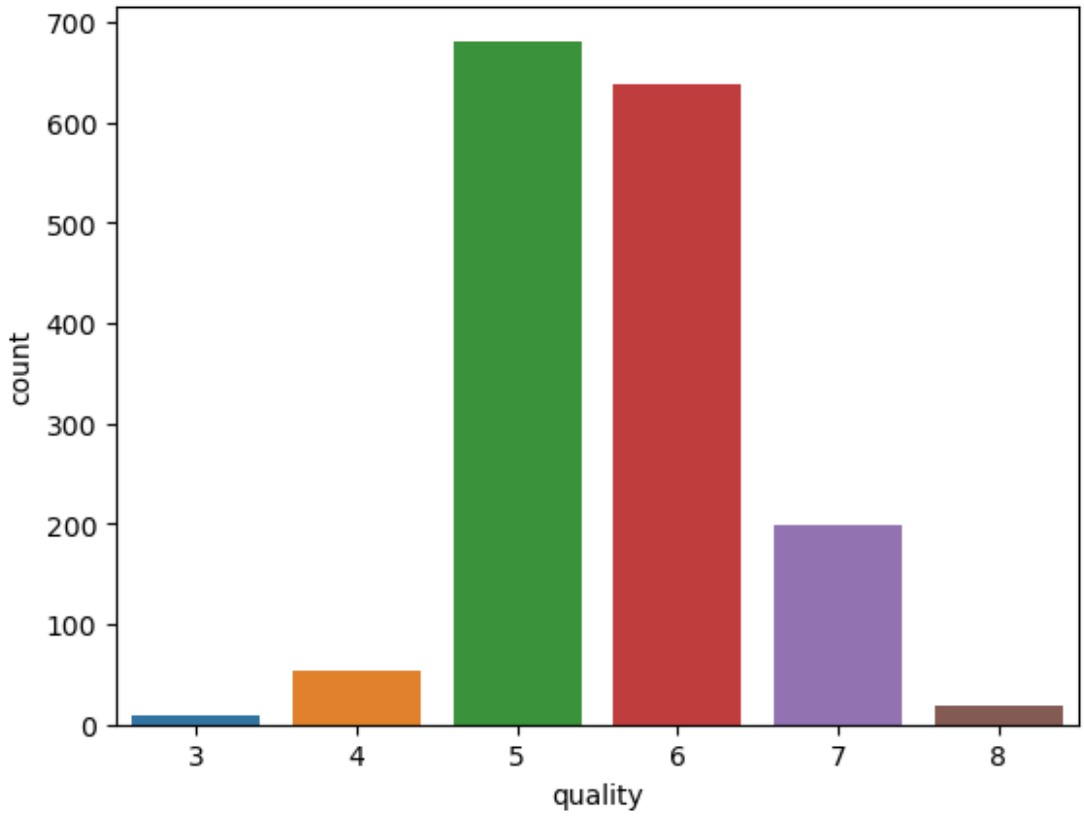
```
       5.20669998, 5.11005631, 6.29652093, 5.48263655, 5.73380671,
       5.86096397, 6.11131909, 5.38204246, 5.39418516, 5.11161705,
       4.74487438, 6.34043215, 5.57642863, 6.52465957, 5.18100269,
       6.37846442, 5.39147732, 5.7435927 , 6.71436012, 5.48263655,
       5.42573746, 6.08035849, 5.6017508 , 6.52660959, 5.79174569,
       5.32807323, 4.92850887, 5.40669848, 5.49983794, 6.12476389,
       5.36974106, 5.78401123, 5.48534309, 5.02135392, 6.65592712,
       5.62370825, 4.83368748, 5.73347951, 5.68074781, 6.09738854,
       5.99258428, 5.16969289, 5.7770828 , 6.59697123, 6.37009025,
       5.77981876, 5.46465189, 5.19009343, 5.80517998, 5.30830978,
       5.09158113, 6.24863165, 6.33674607, 5.99341483, 5.16829696,
       4.81289689, 5.22265229, 6.44901207, 5.48931502, 5.31886287,
       5.5589884 , 5.04938167, 6.32905554, 5.98208683, 6.04415923,
       6.12476389, 5.37906696, 5.72368681, 4.795237  , 5.03676054,
       5.68938109, 5.01079638, 5.83995808, 6.13732216, 5.24782156,
       5.56627333, 6.00210169, 5.3626292 , 6.68219105, 5.11532126,
       5.78120835, 5.62454656, 5.31952796, 5.51514228, 5.20719665,
       5.13154551, 5.48620652, 5.85075029, 5.71919777, 6.80397753,
       6.20404528, 6.04410296, 5.38204246, 6.50598024, 5.85449947,
       6.30306847, 5.05268393, 4.92613186, 5.94872379, 6.32176541,
       5.18546252, 5.8361393 , 5.40120414, 5.17199122, 5.3095161 ,
       5.49911144, 5.66556707, 6.21315993, 6.22227229, 5.26433184,
       6.48967503, 4.95165562, 5.37197617, 5.49931461, 5.3577211 ,
       5.82641444, 4.97385804, 6.03912287, 5.03990278, 5.76144224,
       5.67870975, 6.57726748, 5.67261468, 5.5851728 , 4.92156862,
       6.38162382, 5.10784567, 6.30108784, 6.21224582, 6.50221084,
       5.51985221, 5.16412612, 6.23283235, 5.32903476, 5.25839032,
       5.32882382, 5.89753508, 5.92128255, 6.26545355, 6.57918909,
       5.55219907, 5.56483453, 5.51937934, 5.61558301, 5.39101712,
       5.68815279, 5.23225544, 5.2805354 , 6.2724663 , 5.19707213])
```

```
from sklearn.metrics import r2_score
from sklearn.metrics import mean_squared_error
r2 = r2_score(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
print("Mean Squared Error:", mse)
print("R-squared:", r2)
```

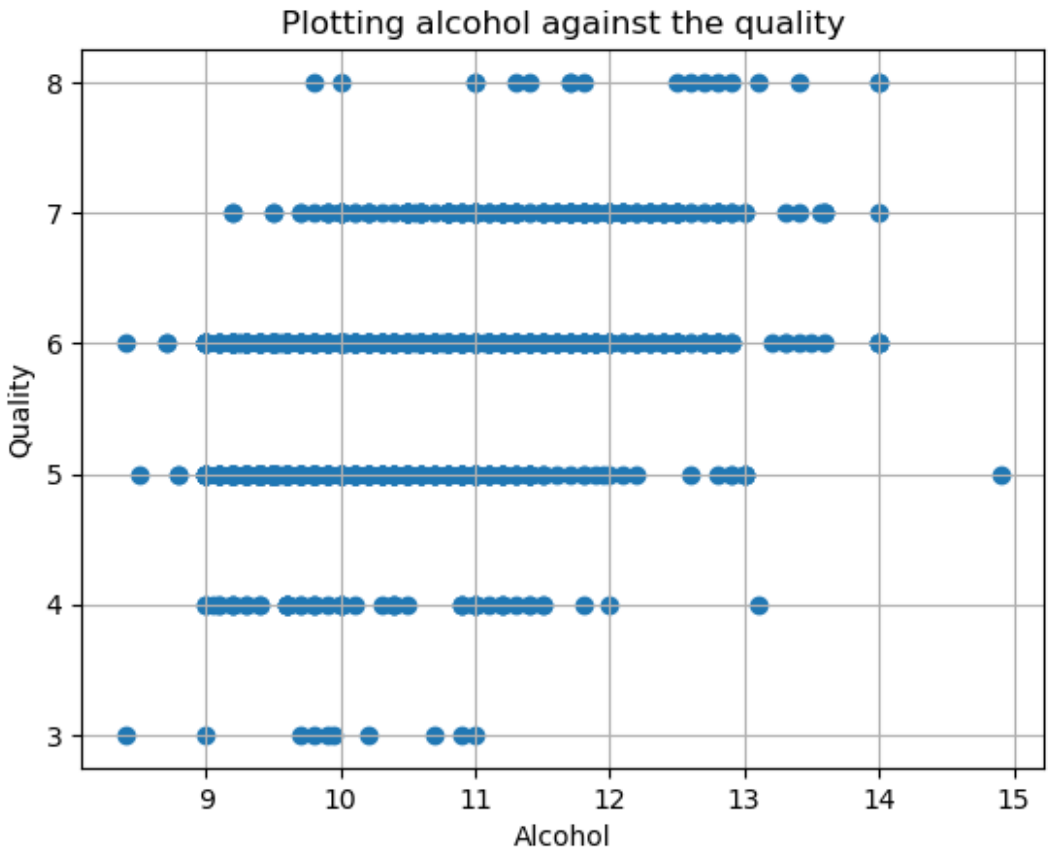
```
Mean Squared Error: 0.39002514396395427
R-squared: 0.403180341279623
```

```
sns.countplot(df['quality'])
plt.show()
```

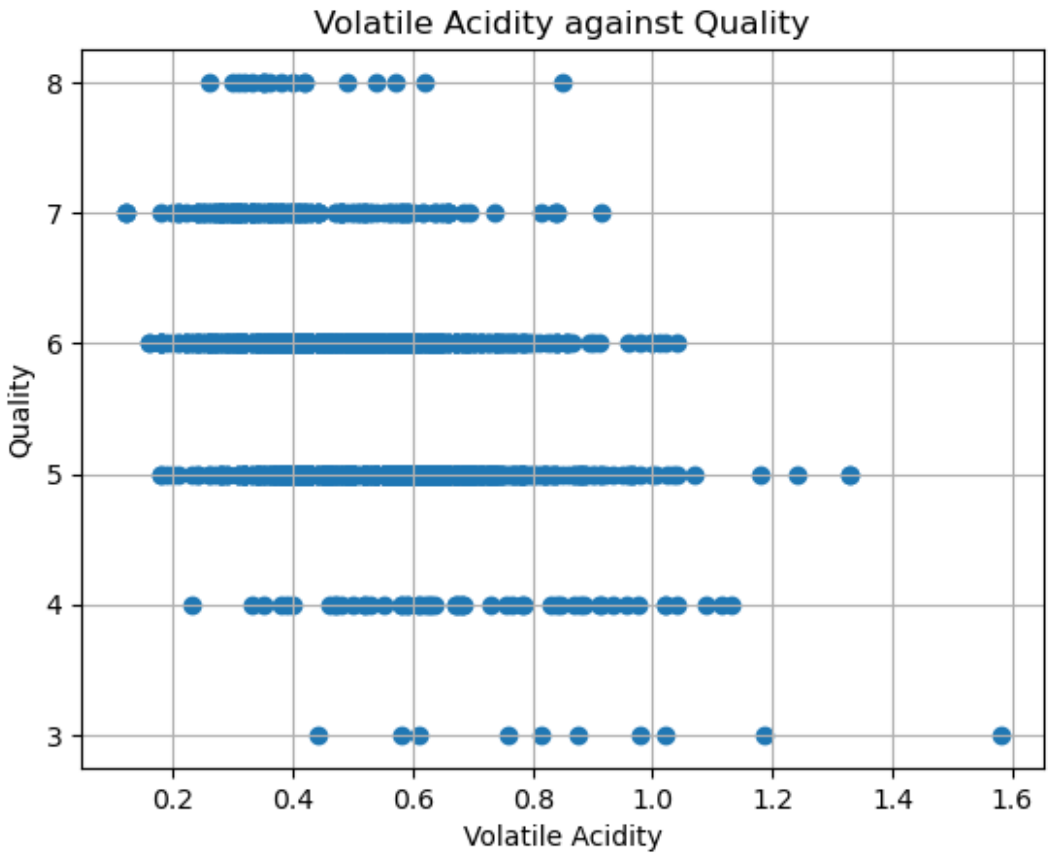
```
C:\Users\Logeshwaran\anaconda3\lib\site-packages\seaborn\
_decorators.py:36: FutureWarning: Pass the following variable as a
keyword arg: x. From version 0.12, the only valid positional argument
will be `data`, and passing other arguments without an explicit
keyword will result in an error or misinterpretation.
  warnings.warn(
```




```
#Let's plot a scatter plot of alcohol against its quality
import matplotlib.pyplot as plt
plt.scatter(df['alcohol'],df['quality'])
plt.xlabel('Alcohol')
plt.ylabel('Quality')
plt.title('Plotting alcohol against the quality')
plt.grid(True)
plt.show()
```




```
#Plotting volatile acidity against quality
import matplotlib.pyplot as plt
plt.scatter(df['volatile acidity'],df['quality'])
plt.xlabel('Volatile Acidity')
plt.ylabel('Quality')
plt.title('Volatile Acidity against Quality')
plt.grid(True)
plt.show()
```




```
df.corr()
```

| | fixed acidity | volatile acidity | citric acid | residual sugar | chlorides | free sulfur dioxide | total sulfur dioxide | density | pH | sulphates | alcohol | quality |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| fixed acidity | 1.000000 | -0.256131 | 0.671703 | 0.114777 | 0.093705 | -0.153794 | -0.113181 | 0.668047 | -0.682978 | 0.183006 | -0.061668 | 0.124052 |
| volatile acidity | -0.256131 | 1.000000 | -0.552496 | 0.001918 | 0.061298 | -0.010504 | 0.076470 | 0.022026 | 0.234937 | -0.260987 | -0.202288 | -0.390558 |
| citric acid | 0.671703 | -0.552496 | 1.000000 | 0.143577 | 0.203823 | -0.060978 | 0.035533 | 0.364947 | -0.541904 | 0.312770 | 0.109903 | 0.226373 |
| residual sugar | 0.114777 | 0.001918 | 0.143577 | 1.000000 | 0.055610 | 0.187049 | 0.203028 | 0.355283 | -0.085652 | 0.005527 | 0.042075 | 0.013732 |
| chlorides | 0.093705 | 0.061298 | 0.203823 | 0.055610 | 1.000000 | 0.005562 | 0.047400 | 0.200632 | -0.265026 | 0.371260 | -0.221141 | -0.128907 |
| 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 | -0.050656 |
| 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 | -0.185100 |
| density | 0.668047 | 0.022026 | 0.364947 | 0.355283 | 0.200632 | -0.021946 | 0.071269 | 1.000000 | -0.341699 | 0.148506 | -0.496180 | -0.174919 |
| pH | -0.682978 | 0.234937 | -0.541904 | -0.085652 | -0.265026 | 0.070377 | -0.066495 | -0.341699 | 1.000000 | -0.196648 | 0.205633 | -0.057731 |
| sulphates | 0.183006 | -0.260987 | 0.312770 | 0.005527 | 0.371260 | 0.051658 | 0.042947 | 0.148506 | -0.196648 | 1.000000 | 0.093595 | 0.251397 |
| alcohol | -0.061668 | -0.202288 | 0.109903 | 0.042075 | -0.221141 | -0.069408 | -0.205654 | -0.496180 | 0.205633 | 0.093595 | 1.000000 | 0.476166 |
| quality | 0.124052 | -0.390558 | 0.226373 | 0.013732 | -0.128907 | -0.050656 | -0.185100 | -0.174919 | -0.057731 | 0.251397 | 0.476166 | 1.000000 |

```
residual sugar                1.000000   0.055610              0.187049

chlorides                     0.055610   1.000000              0.005562

free sulfur dioxide           0.187049   0.005562              1.000000

total sulfur dioxide          0.203028   0.047400              0.667666

density                       0.355283   0.200632             -0.021946

pH                           -0.085652  -0.265026              0.070377

sulphates                     0.005527   0.371260              0.051658

alcohol                       0.042075  -0.221141             -0.069408

quality                       0.013732  -0.128907             -0.050656


                      total sulfur dioxide   density         pH
sulphates  \
fixed acidity                    -0.113181  0.668047 -0.682978
0.183006
volatile acidity                  0.076470  0.022026  0.234937  -
0.260987
citric acid                       0.035533  0.364947 -0.541904
0.312770
residual sugar                    0.203028  0.355283 -0.085652
0.005527
chlorides                         0.047400  0.200632 -0.265026
0.371260
free sulfur dioxide               0.667666 -0.021946  0.070377
0.051658
total sulfur dioxide              1.000000  0.071269 -0.066495
0.042947
density                           0.071269  1.000000 -0.341699
0.148506
pH                               -0.066495 -0.341699  1.000000  -
0.196648
sulphates                         0.042947  0.148506 -0.196648
1.000000
alcohol                          -0.205654 -0.496180  0.205633
0.093595
quality                          -0.185100 -0.174919 -0.057731
0.251397

                       alcohol   quality
fixed acidity        -0.061668  0.124052
volatile acidity     -0.202288 -0.390558
```

```
citric acid           0.109903  0.226373
residual sugar        0.042075  0.013732
chlorides            -0.221141 -0.128907
free sulfur dioxide  -0.069408 -0.050656
total sulfur dioxide -0.205654 -0.185100
density              -0.496180 -0.174919
pH                    0.205633 -0.057731
sulphates             0.093595  0.251397
alcohol               1.000000  0.476166
quality               0.476166  1.000000
```