Foundations of Data Science Using Python

Session 5: Data Handling and Data Visualization using Python

Analyzing Data using Python

First, load the dataset using pandas, we will learn more about the dataset.

1. Describing the dataset

The method **describe()** gives the parameters like count, mean, standard deviation, minimum, maximum.

```
import pandas
df=pandas.read_csv('F:\PYTHON\MyPythonPrograms\winequality-red.csv',sep=';')
print(df.describe())
```

```
F:\PYTHON\MyPythonPrograms>py DataAnalysis.py
       fixed acidity  volatile acidity  citric acid  residual sugar    chlorides  ...      density           pH    sulphates      alcohol      qua
count    1599.000000       1599.000000  1599.000000     1599.000000  1599.000000  ...  1599.000000  1599.000000  1599.000000  1599.000000  1599.00
mean        8.319637          0.527821     0.270976        2.538806     0.087467  ...     0.996747     3.311113     0.658149    10.422983     5.63
std         1.741096          0.179060     0.194801        1.409928     0.047065  ...     0.001887     0.154386     0.169507     1.065668     0.80
min         4.600000          0.120000     0.000000        0.900000     0.012000  ...     0.990070     2.740000     0.330000     8.400000     3.00
25%         7.100000          0.390000     0.090000        1.900000     0.070000  ...     0.995600     3.210000     0.550000     9.500000     5.00
50%         7.900000          0.520000     0.260000        2.200000     0.079000  ...     0.996750     3.310000     0.620000    10.200000     6.00
75%         9.200000          0.640000     0.420000        2.600000     0.090000  ...     0.997835     3.400000     0.730000    11.100000     6.00
max        15.900000          1.580000     1.000000       15.500000     0.611000  ...     1.003690     4.010000     2.000000    14.900000     8.00

[8 rows x 12 columns]

F:\PYTHON\MyPythonPrograms>
```

2. Shape of the dataset

The dimensions of the dataset can be obtained from **shape** tuple.

```
import pandas
df=pandas.read_csv('F:\PYTHON\MyPythonPrograms\winequality-red.csv',sep=';')
print(df.shape)
```

```
F:\PYTHON\MyPythonPrograms>py DataAnalysis.py
(1599, 12)

F:\PYTHON\MyPythonPrograms>
```

From the output we can observe that the dataset contains 1599 instances and 12 attributes.

3. Extracting data from the dataset

In case we need to extract the first ten rows from the dataset, we can invoke **head()** method on the data frame with 10 as an argument to the method **head()**

```
import pandas
df=pandas.read_csv('F:\PYTHON\MyPythonPrograms\winequality-red.csv',sep=';')
print(df.head(10)
```

```
F:\PYTHON\MyPythonPrograms>py DataAnalysis.py
   fixed acidity  volatile acidity  citric acid  residual sugar  chlorides  ...  density    pH  sulphates  alcohol  quality
0            7.4              0.70         0.00             1.9      0.076  ...   0.9978  3.51       0.56      9.4        5
1            7.8              0.88         0.00             2.6      0.098  ...   0.9968  3.20       0.68      9.8        5
2            7.8              0.76         0.04             2.3      0.092  ...   0.9970  3.26       0.65      9.8        5
3           11.2              0.28         0.56             1.9      0.075  ...   0.9980  3.16       0.58      9.8        6
4            7.4              0.70         0.00             1.9      0.076  ...   0.9978  3.51       0.56      9.4        5
5            7.4              0.66         0.00             1.8      0.075  ...   0.9978  3.51       0.56      9.4        5
6            7.9              0.60         0.06             1.6      0.069  ...   0.9964  3.30       0.46      9.4        5
7            7.3              0.65         0.00             1.2      0.065  ...   0.9946  3.39       0.47     10.0        7
8            7.8              0.58         0.02             2.0      0.073  ...   0.9968  3.36       0.57      9.5        7
9            7.5              0.50         0.36             6.1      0.071  ...   0.9978  3.35       0.80     10.5        5

[10 rows x 12 columns]

F:\PYTHON\MyPythonPrograms>
```

4. Performing operations around a variable

Certain operations can be performed on a variable. For instance, let us understand how to group data on a variable. The **groupby()** method is helpful in grouping the data based on a dataset attribute.

```
import pandas
df=pandas.read_csv('F:\PYTHON\MyPythonPrograms\winequality-red.csv',sep=';')
print(df.groupby('quality').size())
```

```
F:\PYTHON\MyPythonPrograms>py DataAnalysis.py
quality
3     10
4     53
5    681
6    638
7    199
8     18
dtype: int64

F:\PYTHON\MyPythonPrograms>
```

Total 1599

Data Visualization using Python

Data can be visualized as plots and charts in python. The python libraries viz., **pandas, Matplotlib** and **seaborn** can be used for Data Analysis and Data Visualization. Let us discuss two kinds of plots, univariate and multivariate in visualizing the data. Univariate plot denotes that only one variable is being examined unlike multiple variables in multivariate plot. Histograms, Density plots, Box and Whisker plots helps us in visualizing the data univariate plots whereas Correlation Matrix and Scatterplot Matrix for data multivariate plots.

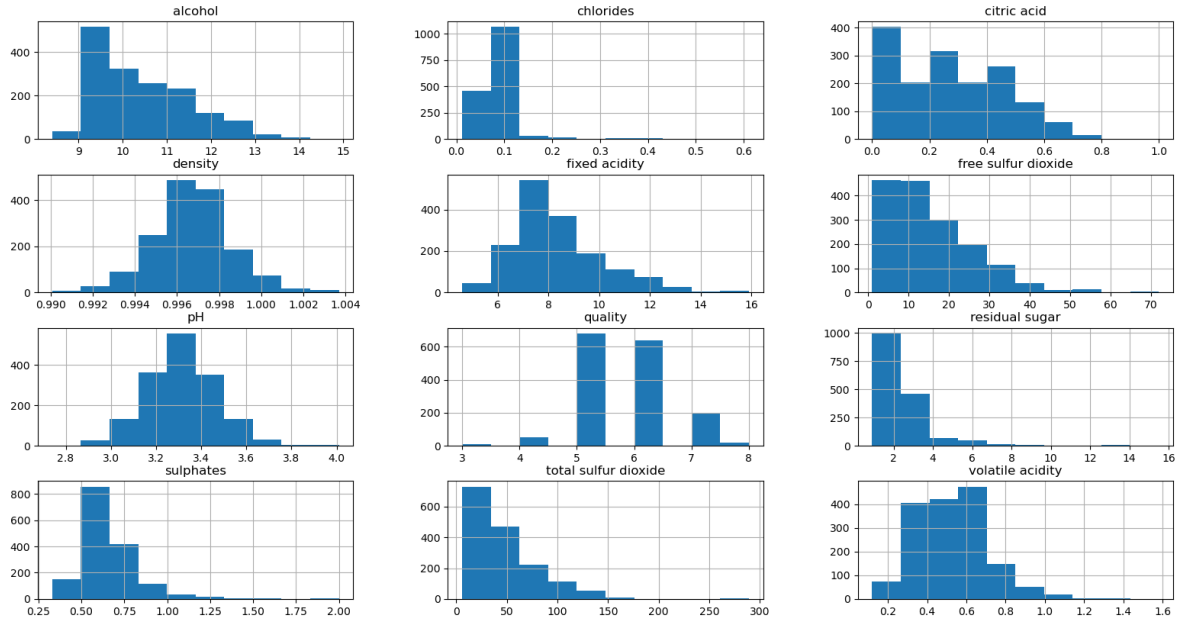
- Univariate plots to understand each attribute
- Multivariate plots to understand the relationships between attributes

Visualizing Data-Univariate Plots

1. Histograms

One way to visualize the data in Machine Learning is histograms. The histograms group data into bins and give us an idea of how many observations each bin holds. The shape of the bins denotes whether an attribute is Gaussian, skewed, or has an exponential distribution. The information about the outliers can also be known through Histograms. In python, the method **hist()** is used to represent the data in terms of a histogram.

```
import pandas
import matplotlib.pyplot as plt
#create the dataframe from CSV file using pandas
df=pandas.read_csv('F:\PYTHON\MyPythonPrograms\winequality-red.csv',sep=';')
#Representation of data as a histogram
df.hist()
#For displaying the plot
plt.show()
```



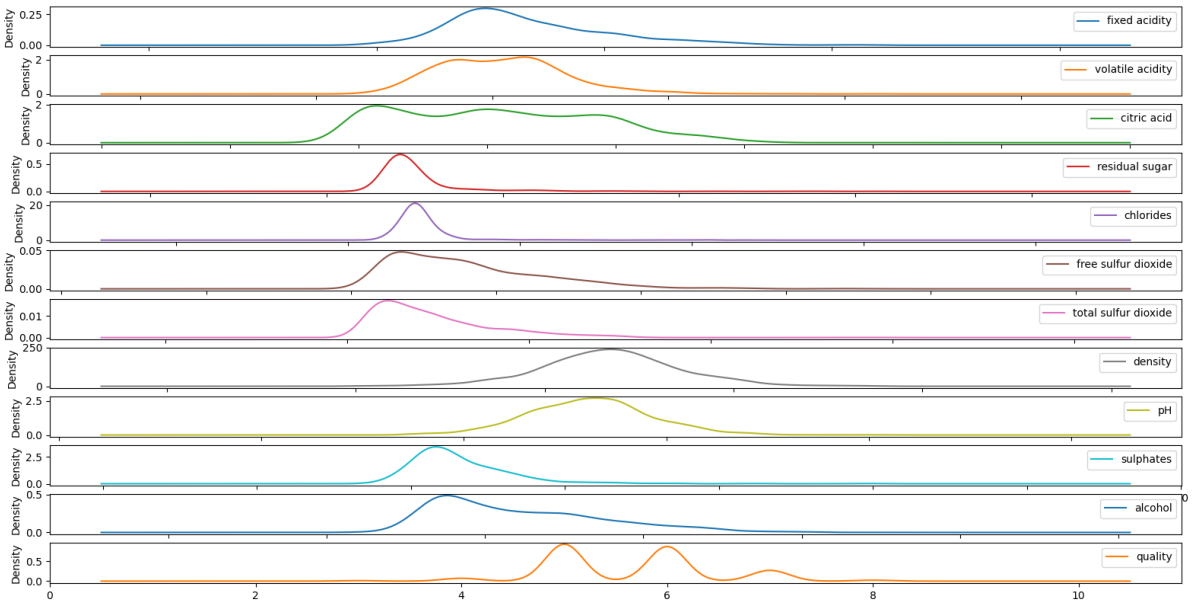
From the histograms, we can notice that the attributes 'total sulphur oxide', 'free sulphur oxide' and 'residual sugar' have an exponential distribution. The attributes 'density', 'pH', 'fixed acidity', and 'Volatile acidity' have Gaussian or nearly Gaussian distributions.

2. Density Plots

The distribution of an attribute in a data set can be visualized through density plots. It plots the graph on a continuous interval or time period. Density plots are variation of Histograms. The density plot charts the values from a selected column as equally binned distributions. A density plot appears to be an abstracted histogram, where each bin has a smooth curve drawn through its top.

Density plot is a smoothed, continuous version of a histogram estimated from the data set. The most common form of estimation is known as kernel density estimation. In this method, a continuous curve (the kernel) is drawn at every individual data point and all of these curves are then added together to make a single smooth density estimation. The kernel most often used is a Gaussian which produces a Gaussian bell curve at each data point.

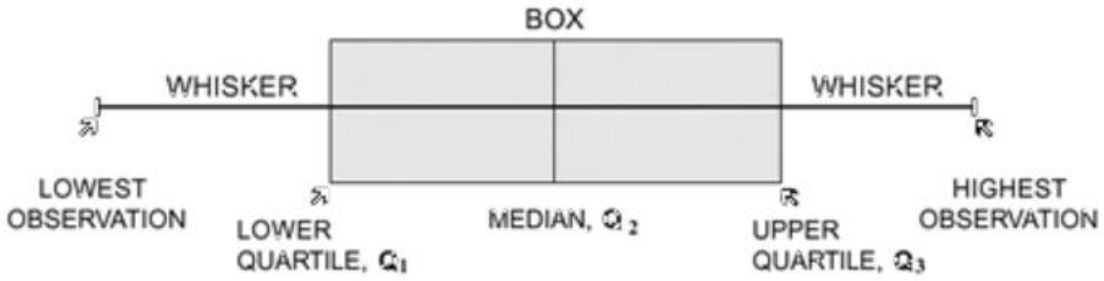
```
import pandas
import matplotlib.pyplot as plt
#create the dataframe from CSV file using pandas
df=pandas.read_csv('F:\PYTHON\MyPythonPrograms\winequality-red.csv',sep=';')
#Representation of data as a density plots
df.plot(kind='density',subplots=True,sharex=False)
#For displaying the plot
plt.show()
```



3. Box and Whisker Plots

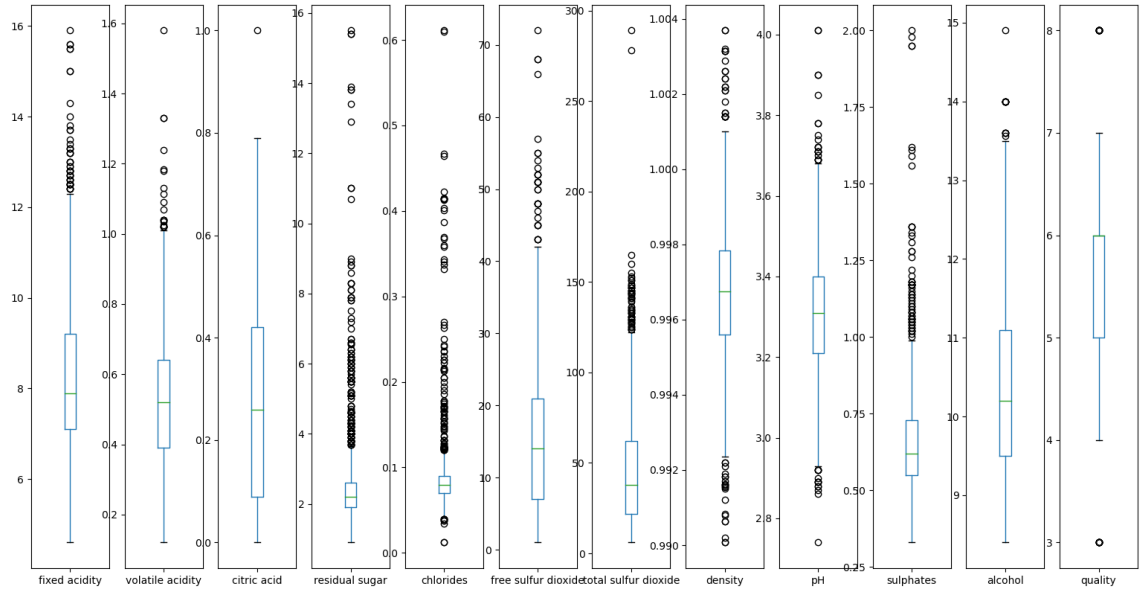
A box and whisker plot (sometimes called a boxplot) is a graph that presents information from a five-number summary. It does not show a distribution in as much detail as a histogram does, but is especially useful for indicating whether a distribution is skewed and whether there are potential unusual observations (outliers) in the data set. Box and whisker plots are also very useful when large numbers of observations are involved and when two or more data sets are being compared. A box and whisker plot is a way of summarizing a set of data measured on an interval scale. It is often used in explanatory data analysis. This type of graph is used to show the shape of the distribution, its central value, and its variability. In a box and whisker plot:

- the ends of the box are the upper and lower quartiles, so the box spans the interquartile range
- the median is marked by a vertical line inside the box
- the whiskers are the two lines outside the box that extend to the highest and lowest observations.



Box and Whisker plots are ideal for comparing distributions because the centre, spread and overall range are immediately apparent. A box plot summarizes how each attribute is distributed. It also draws a line for the median and a box around the 25th and 75th percentiles. Whiskers tells us how the data is spread, and the dots outside the Whiskers give candidate outlier values.

```
import pandas
import matplotlib.pyplot as plt
#create the dataframe from CSV file using pandas
df=pandas.read_csv('F:\PYTHON\MyPythonPrograms\winequality-red.csv',sep=';')
#Representation of data as a Box plots
df.plot(kind='box',subplots=True,sharex=False,sharey=False)
#For displaying the plot
plt.show()
```



Here, the attributes like ‘total sulphur dioxide’, ‘sulphates’, and ‘residual sugar’ appear skewed towards smaller values.

Visualizing Data-Multivariate Plots

A multivariate analysis examines more than two variables. For two variables, we call it bivariate.

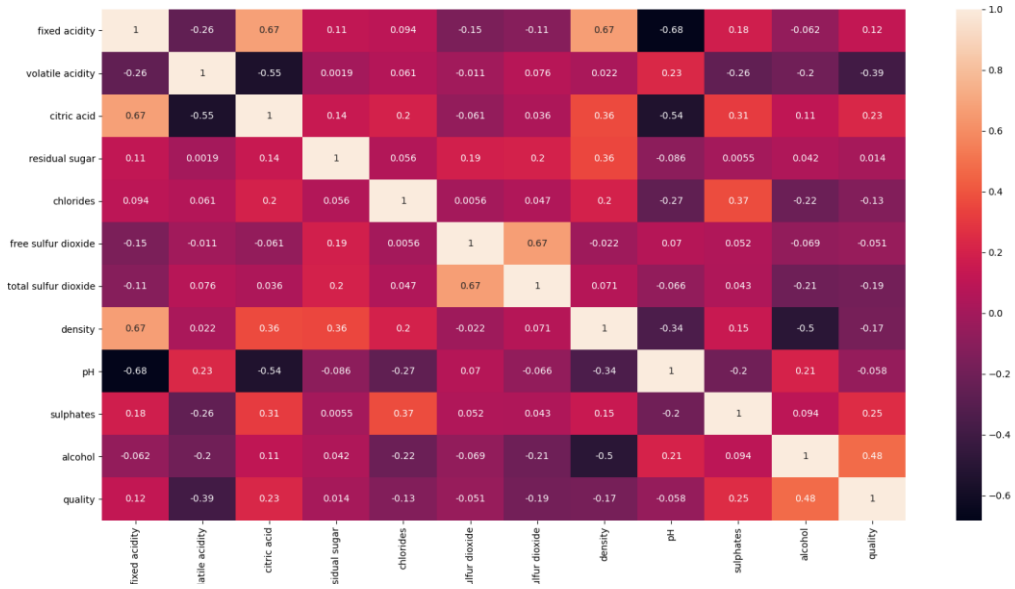
1. Correlation Matrix Plot

Correlation Matrix plot denotes how changes between two variables relate. Two variables that change in the same direction are positively correlated. A change in opposite directions implies negative correlation.

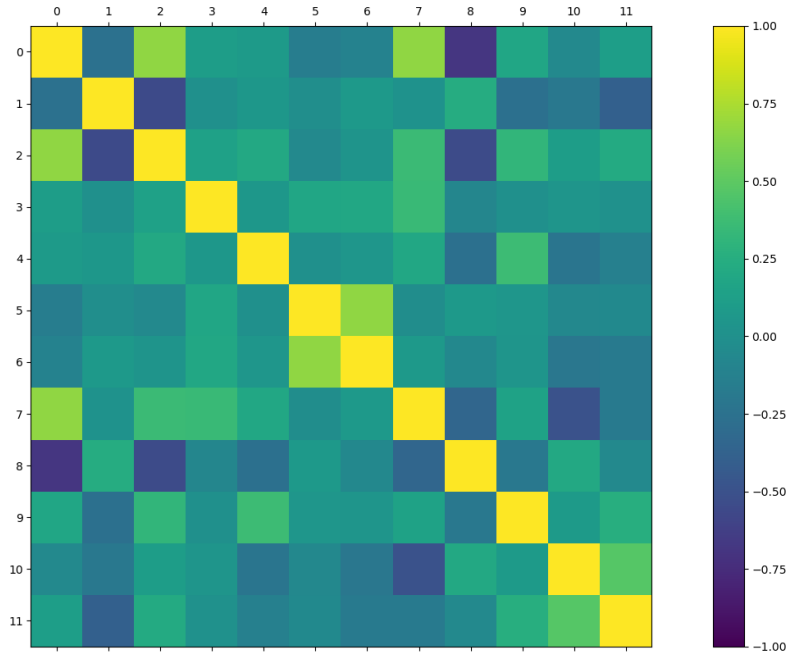
```
import pandas
import seaborn
import matplotlib.pyplot as plt
#create the dataframe from CSV file using pandas
df=pandas.read_csv('F:\PYTHON\MyPythonPrograms\winequality-red.csv',sep=';')
#Representation of data as a Correlation Matrix plot
correlationsMatrix=df.corr()
print(correlationsMatrix)
seaborn.heatmap(correlationsMatrix, annot=True)
plt.show()
```

```
F:\PYTHON\MyPythonPrograms>py DataVisualization.py
                      fixed acidity  volatile acidity  citric acid  residual sugar  chlorides  ...   density        pH  sulphates   alcohol   qual
fixed acidity              1.000000         -0.256131     0.671703        0.114777   0.093705  ...  0.668047 -0.682978   0.183006 -0.061668  0.124
volatile acidity          -0.256131          1.000000    -0.552496        0.001918   0.061298  ...  0.022026  0.234937  -0.260987 -0.202288 -0.390
citric acid                0.671703         -0.552496     1.000000        0.143577   0.203823  ...  0.364947 -0.541904   0.312770  0.109903  0.226
residual sugar             0.114777          0.001918     0.143577        1.000000   0.055610  ...  0.355283 -0.085652   0.005527  0.042075  0.013
chlorides                  0.093705          0.061298     0.203823        0.055610   1.000000  ...  0.200632 -0.265026   0.371260 -0.221141 -0.128
free sulfur dioxide       -0.153794         -0.010504    -0.060978        0.187049   0.005562  ... -0.021946  0.070377   0.051658 -0.069408 -0.050
total sulfur dioxide      -0.113181          0.076470     0.035533        0.203028   0.047400  ...  0.071269 -0.066495   0.042947 -0.205654 -0.185
density                    0.668047          0.022026     0.364947        0.355283   0.200632  ...  1.000000 -0.341699   0.148506 -0.496180 -0.174
pH                        -0.682978          0.234937    -0.541904       -0.085652  -0.265026  ... -0.341699  1.000000  -0.196648  0.205633 -0.057
sulphates                  0.183006         -0.260987     0.312770        0.005527   0.371260  ...  0.148506 -0.196648   1.000000  0.093595  0.251
alcohol                   -0.061668         -0.202288     0.109903        0.042075  -0.221141  ... -0.496180  0.205633   0.093595  1.000000  0.476
quality                    0.124052         -0.390558     0.226373        0.013732  -0.128907  ... -0.174919 -0.057731   0.251397  0.476166  1.000

[12 rows x 12 columns] Correlation Matrix
```



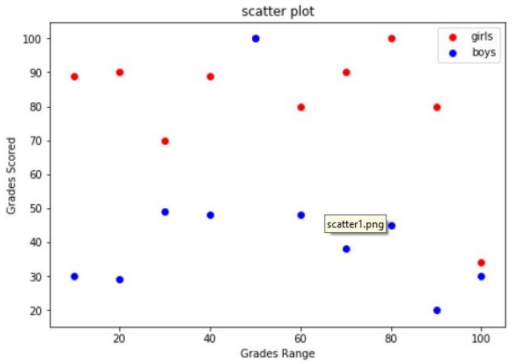

```
import pandas, numpy
import matplotlib.pyplot as plt
#create the dataframe from CSV file using pandas
df=pandas.read_csv('F:\PYTHON\MyPythonPrograms\winequality-red.csv',sep=';')
#Representation of data as a Correlation Matrix plot
correlations=df.corr()
fig=plt.figure()
ax=fig.add_subplot(111)
cax=ax.matshow(correlations,vmin=-1,vmax=1)
fig.colorbar(cax)
ticks=numpy.arange(0,12,1)
ax.set_xticks(ticks)
ax.set_yticks(ticks)
#For displaying the plot
plt.show()
```



This matrix is symmetrical around a left diagonal.

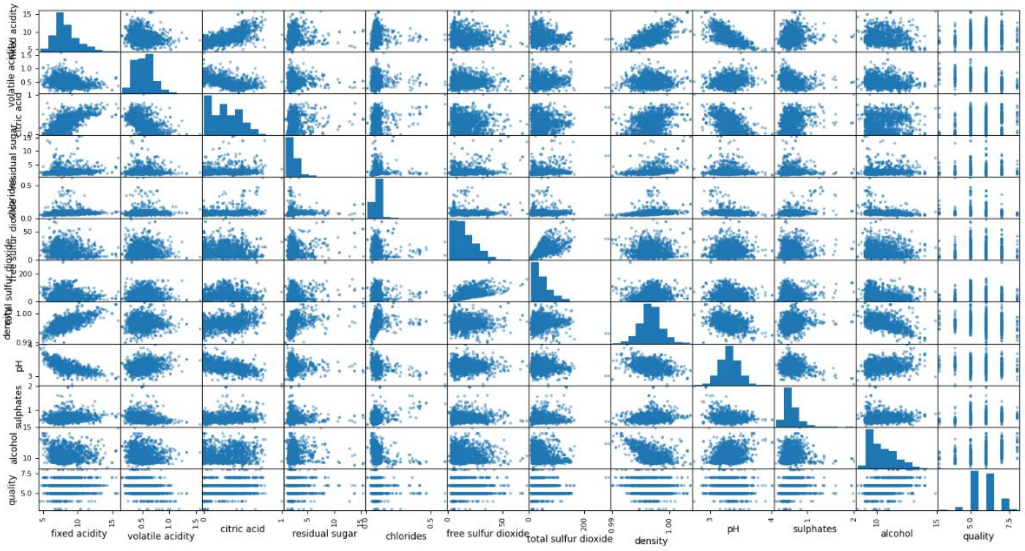
2. Scatterplot Matrix

Scatter plots are used to plot the data points on horizontal and vertical axes in the attempt to show how much one variable is affected by another. Each row in the data table is represented by a marker, the position depends on its values in the columns set on the X and Y axes. A third variable can be set to correspond to the colour or size of the markers, thus adding yet another dimension to the plot.



Scatterplot matrices depict how two variables relate as dots in two dimensions. Plotting all scatterplots for a data together in one place results in a scatterplot matrix. These plots can spot structured relationships between variables.

```
import pandas
import matplotlib.pyplot as plt
#create the dataframe from CSV file using pandas
df=pandas.read_csv('F:\PYTHON\MyPythonPrograms\winequality-red.csv',sep=';')
#Representation of data as a Scatteredplot
pandas.plotting.scatter_matrix(df)
plt.show()
```



This is symmetrical too. The left diagonal has histograms of attributes because it does not makes sense to plot an attribute's scatterplot with itself