

Deena 20104016

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as pp
```

Problem Statement

LINEAR REGRESSION

```
In [2]: a = pd.read_csv("11_winequality-red.csv")
```

Out[2]:

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

1599 rows × 12 columns

HEAD

In [3]:

Out[3]:

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

Data Cleaning and Preprocessing

In [4]:

Out[4]:

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

In [5]:

Out[5]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide
count	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000
mean	8.319637	0.527821	0.270976	2.538806	0.087467	15.874922	46.467791
std	1.741096	0.179060	0.194801	1.409928	0.047065	10.460157	32.895301
min	4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	6.000000
25%	7.100000	0.390000	0.090000	1.900000	0.070000	7.000000	22.000000
50%	7.900000	0.520000	0.260000	2.200000	0.079000	14.000000	38.000000
75%	9.200000	0.640000	0.420000	2.600000	0.090000	21.000000	62.000000
max	15.900000	1.580000	1.000000	15.500000	0.611000	72.000000	289.000000

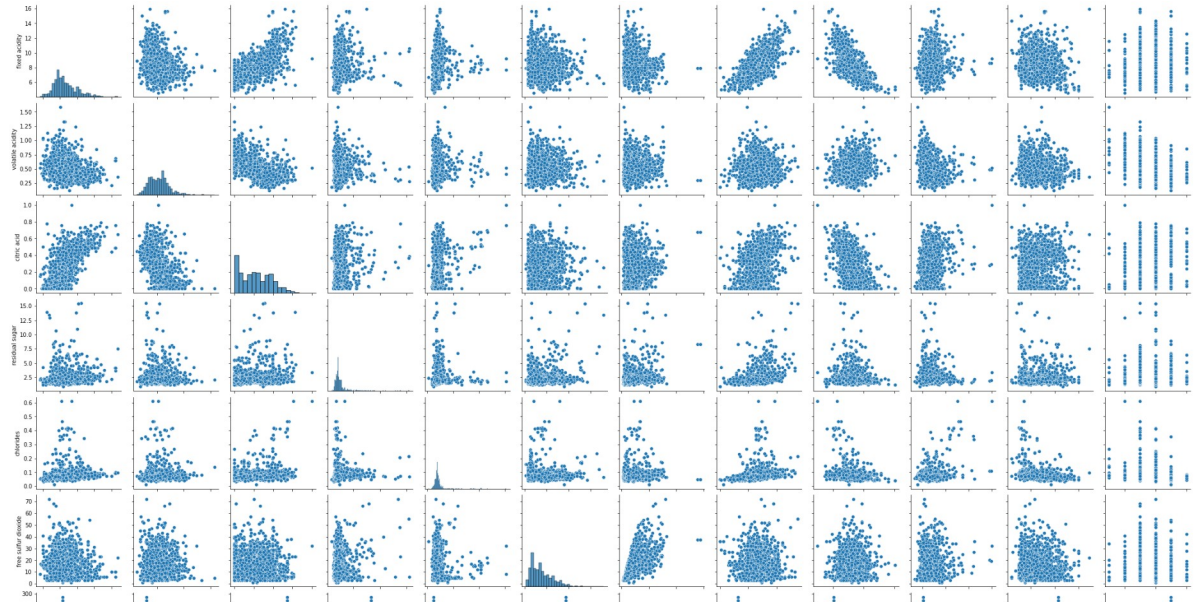
To display heading

In [6]:

```
Out[6]: Index(['fixed acidity', 'volatile acidity', 'citric acid', 'residual sugar',  
             'chlorides', 'free sulfur dioxide', 'total sulfur dioxide', 'density',  
             'pH', 'sulphates', 'alcohol', 'quality'],  
          dtype='object')
```

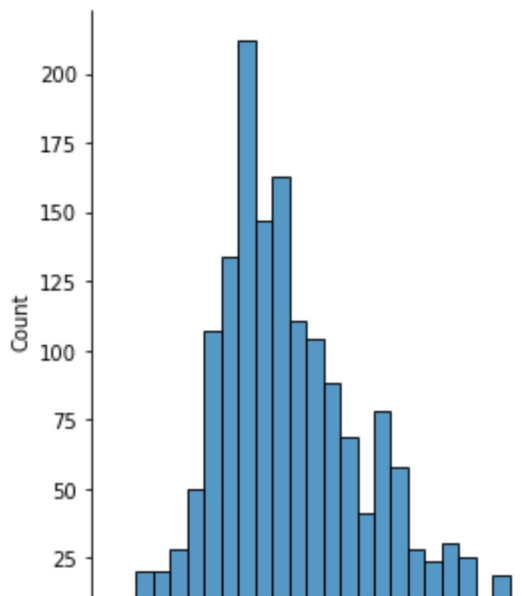
In [7]:

```
Out[7]: <seaborn.axisgrid.PairGrid at 0x25f2c221130>
```



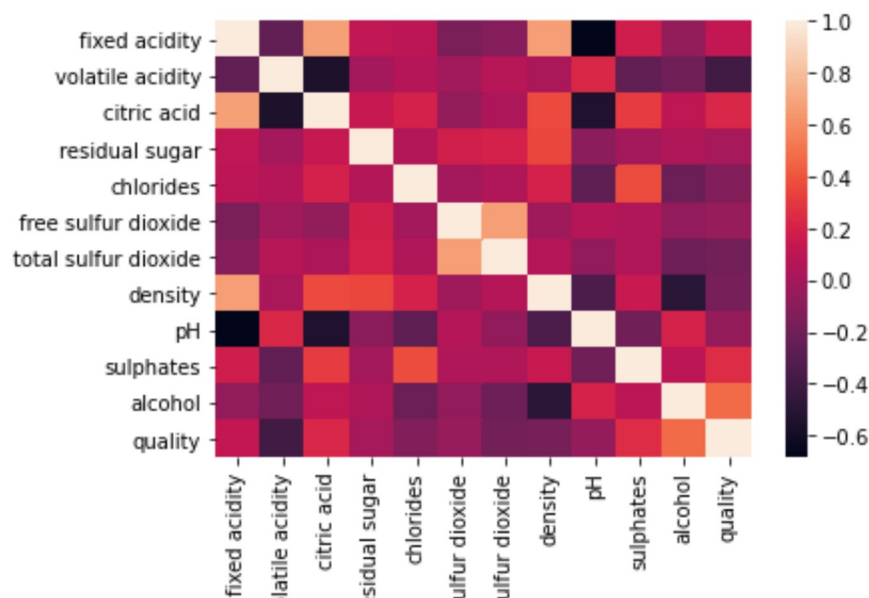
In [8]:

```
Out[8]: <seaborn.axisgrid.FacetGrid at 0x25f2b1298b0>
```



In [9]:

Out[9]: <AxesSubplot:>



TO TRAIN THE MODEL - MODEL BUILDING

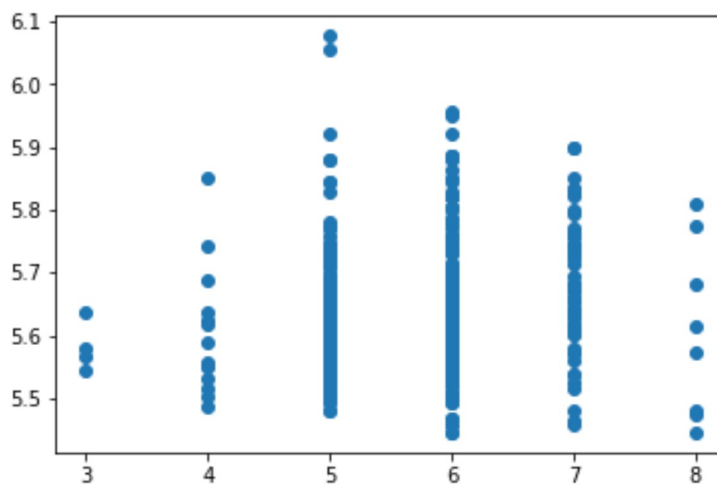
In [10]: `x = a[['fixed acidity']]`In [11]: `# to split my dataset into training and test data
from sklearn.model_selection import train_test_split`In [12]: `from sklearn.linear_model import LinearRegression
lr = LinearRegression()`Out[12]: `LinearRegression()`In [13]: `coeff = pd.DataFrame(lr.coef_, x.columns, columns=['Co-efficient'])`

Out[13]:

	Co-efficient
fixed acidity	0.057914

```
In [14]: prediction= lr.predict(x_test)
```

```
Out[14]: <matplotlib.collections.PathCollection at 0x25f34ad1a60>
```



```
In [15]:
```

```
Out[15]: 0.014037973770738121
```

RIDGE & LASSO

```
In [16]: from sklearn.linear_model import Ridge,Lasso  
rr=Ridge(alpha=10)
```

```
Out[16]: Ridge(alpha=10)
```

```
In [17]:
```

```
Out[17]: 0.014039689432659208
```

```
In [18]: la=Lasso(alpha=10)
```

```
Out[18]: Lasso(alpha=10)
```

```
In [19]:
```

```
Out[19]: -0.0001421891431563349
```