

```
In [2]: 1 import sys
        2 import pandas as pd
        3 import numpy as np
        4 import matplotlib.pyplot as plt
```

```
In [3]: 1 df=pd.read_csv('sonarData.csv')
```

```
In [4]: 1 df
      0  0.0200  0.0371  0.0428  0.0207  0.0954  0.0986  0.1539  0.1601  0.3109  0.2111
      1  0.0453  0.0523  0.0843  0.0689  0.1183  0.2583  0.2156  0.3481  0.3337  0.2872
      2  0.0262  0.0582  0.1099  0.1083  0.0974  0.2280  0.2431  0.3771  0.5598  0.6194
      3  0.0100  0.0171  0.0623  0.0205  0.0205  0.0368  0.1098  0.1276  0.0598  0.1264
      4  0.0762  0.0666  0.0481  0.0394  0.0590  0.0649  0.1209  0.2467  0.3564  0.4459
      ...    ...    ...    ...    ...    ...    ...    ...    ...    ...
    203  0.0187  0.0346  0.0168  0.0177  0.0393  0.1630  0.2028  0.1694  0.2328  0.2684
    204  0.0323  0.0101  0.0298  0.0564  0.0760  0.0958  0.0990  0.1018  0.1030  0.2154
    205  0.0522  0.0437  0.0180  0.0292  0.0351  0.1171  0.1257  0.1178  0.1258  0.2529
    206  0.0303  0.0353  0.0490  0.0608  0.0167  0.1354  0.1465  0.1123  0.1945  0.2354
    207  0.0260  0.0363  0.0136  0.0272  0.0214  0.0338  0.0655  0.1400  0.1843  0.2354
```

208 rows × 61 columns

```
In [5]: 1 print("No. of rows =",df.shape[0])
```

No. of rows = 208

```
In [6]: 1 print("No. of columns =",df.shape[1])
```

No. of columns = 61

```
In [7]: 1 df.dtypes.value_counts()
```

```
Out[7]: float64    60
        object     1
        dtype: int64
```

In [8]: 1 df.describe()

Out[8]:

	Freq_1	Freq_2	Freq_3	Freq_4	Freq_5	Freq_6	Freq_7
count	208.000000	208.000000	208.000000	208.000000	208.000000	208.000000	208.000000
mean	0.029164	0.038437	0.043832	0.053892	0.075202	0.104570	0.121747
std	0.022991	0.032960	0.038428	0.046528	0.055552	0.059105	0.061788
min	0.001500	0.000600	0.001500	0.005800	0.006700	0.010200	0.003300
25%	0.013350	0.016450	0.018950	0.024375	0.038050	0.067025	0.080900
50%	0.022800	0.030800	0.034300	0.044050	0.062500	0.092150	0.106950
75%	0.035550	0.047950	0.057950	0.064500	0.100275	0.134125	0.154000
max	0.137100	0.233900	0.305900	0.426400	0.401000	0.382300	0.372900

8 rows × 60 columns



In [9]: 1 df.isnull().sum()

Out[9]: Freq\_1 0  
Freq\_2 0  
Freq\_3 0  
Freq\_4 0  
Freq\_5 0  
..  
Freq\_57 0  
Freq\_58 0  
Freq\_59 0  
Freq\_60 0  
Label 0  
Length: 61, dtype: int64

In [10]: 1 pip install plotly pandas

Requirement already satisfied: plotly in d:\anaconda\lib\site-packages (5.3.1)  
Requirement already satisfied: pandas in d:\anaconda\lib\site-packages (1.5.3)  
Requirement already satisfied: tenacity>=6.2.0 in d:\anaconda\lib\site-packages (from plotly) (8.2.2)  
Requirement already satisfied: six in d:\anaconda\lib\site-packages (from plotly) (1.16.0)  
Requirement already satisfied: python-dateutil>=2.8.1 in d:\anaconda\lib\site-packages (from pandas) (2.8.2)  
Requirement already satisfied: pytz>=2020.1 in d:\anaconda\lib\site-packages (from pandas) (2022.7)  
Requirement already satisfied: numpy>=1.21.0 in d:\anaconda\lib\site-packages (from pandas) (1.24.3)  
Note: you may need to restart the kernel to use updated packages.

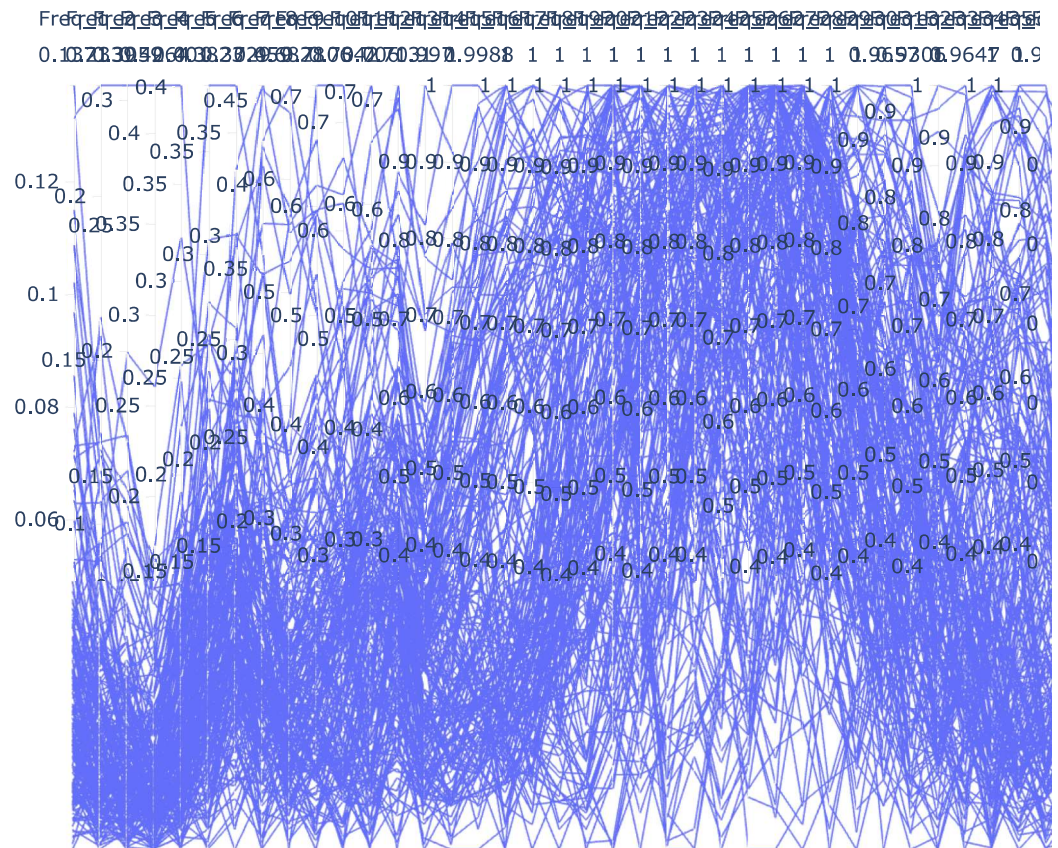
```
In [11]: 1 import plotly.express as px
```

```
In [12]: 1 fig = px.parallel_coordinates(df)
```

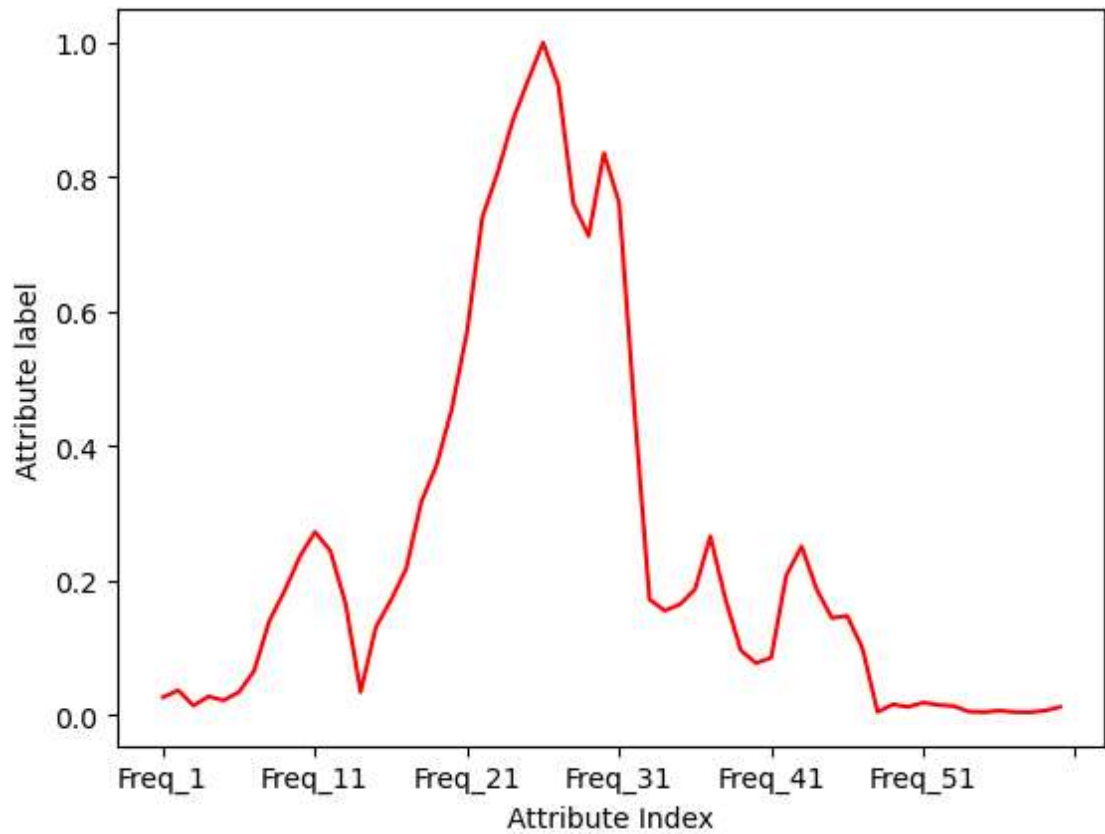
D:\anaconda\Lib\site-packages\plotly\express\\_core.py:279: FutureWarning: iteritems is deprecated and will be removed in a future version. Use .items instead.

```
    dims = [
```

```
In [13]: 1 fig.show()
```

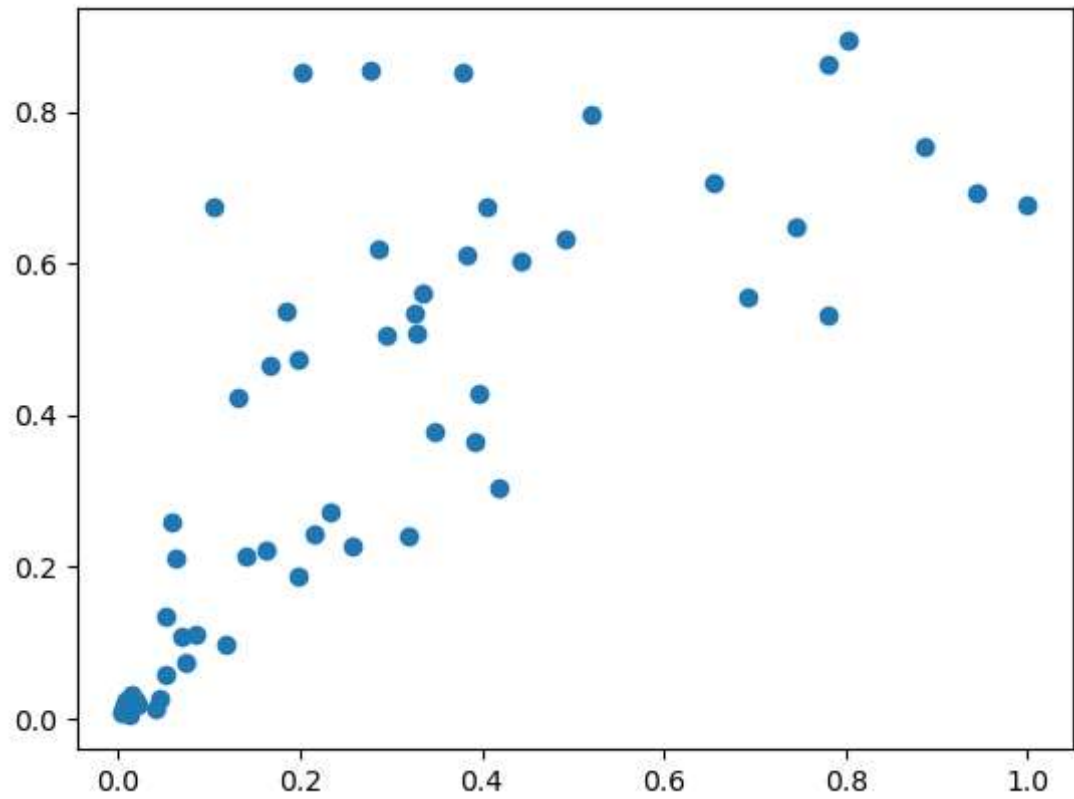


```
In [14]: ▶ 1 for i in range(208):
2           if df.iat[i,60] == "M":
3               pcolor="red"
4           else:
5               pcolor="blue"
6 datarow=df.iloc[i,0:60]
7 datarow.plot(color=pcolor)
8 plt.xlabel("Attribute Index")
9 plt.ylabel(("Attribute label"))
10 plt.show()
```



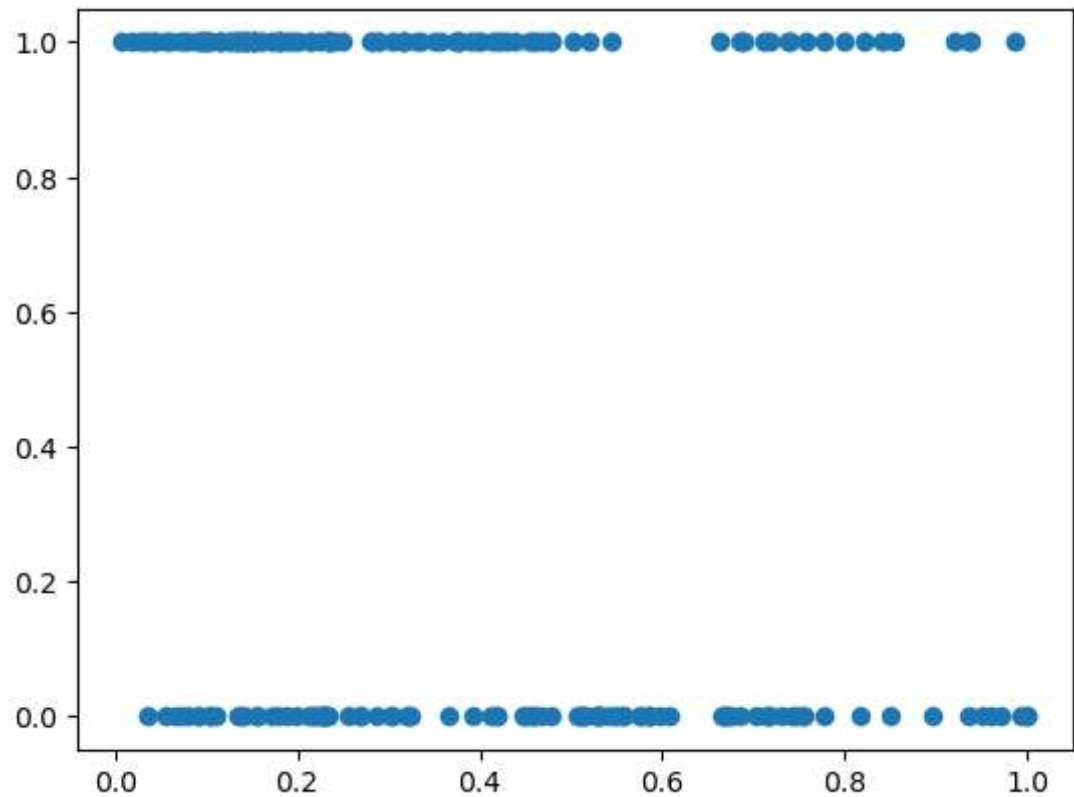
```
In [15]: ▶ 1 #correlation plot between attribute 2 and attribute 3(0-based indexing)
          2 dt1=df.iloc[1,0:60]
          3 dt2=df.iloc[2,0:60]
          4 plt.scatter(dt1, dt2)
```

Out[15]: <matplotlib.collections.PathCollection at 0x2a9102dfb90>



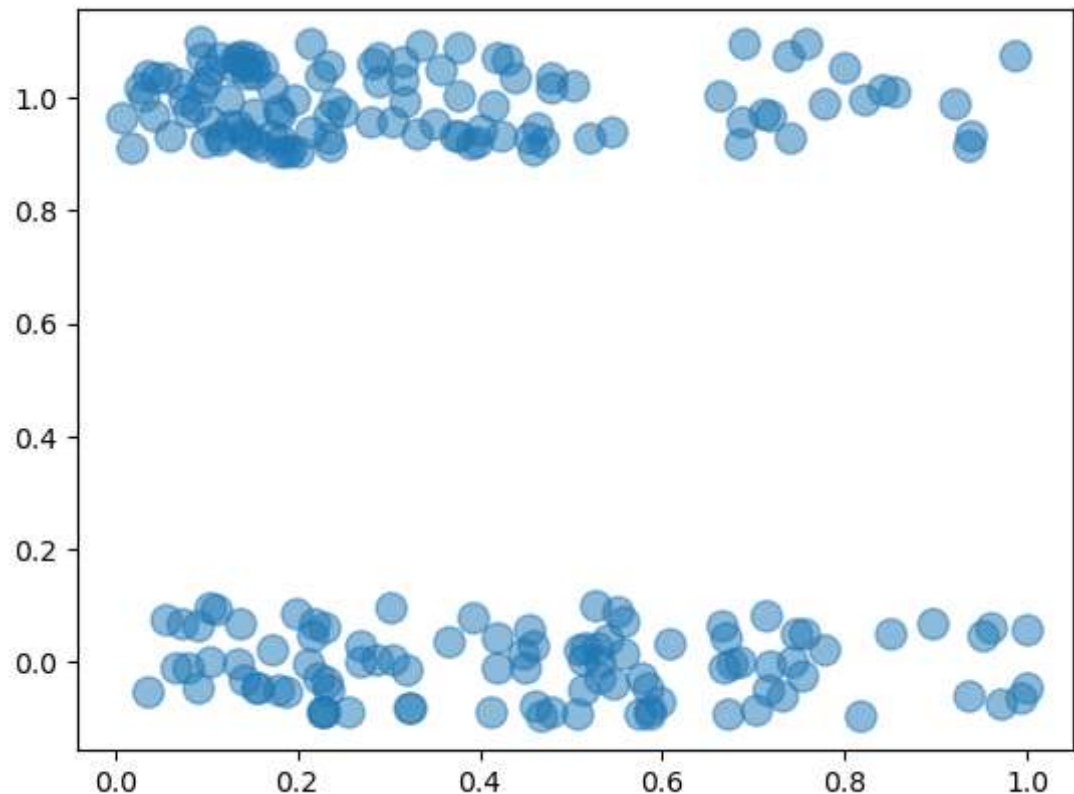
```
In [16]: 1 target=[]
2         for i in range(208):
3             if df.iat[i,60]=="M":
4                 target.append(1.0)
5             else:
6                 target.append(0.0)
7         #plotting 35th attribute
8         datarow=df.iloc[0:208,35]
9         plt.scatter(datarow,target)
```

Out[16]: <matplotlib.collections.PathCollection at 0x2a9108fce50>



```
In [17]: 1 # to improve visualisation caused due to points getting on top on one
2 # and make somewhat transparent, here alpha is for transparency and s
3 from random import uniform
4 target=[]
5 for i in range(208):
6     if df.iat[i,60]=="M":
7         target.append(1.0 + uniform(-0.1,0.1))
8     else:
9         target.append(0.0 + uniform(-0.1,0.1))
10 datarow=df.iloc[0:208,35]
11 plt.scatter(datarow,target,alpha=0.5,s=120)
```

Out[17]: <matplotlib.collections.PathCollection at 0x2a910cadf10>



```
In [18]: 1 # Below corr function uses pearson's correlation coefficient to calculate
2 # two attributes freq_2 and freq_3
3 correlation = df['Freq_2'].corr(df['Freq_3'])
4 correlation
```

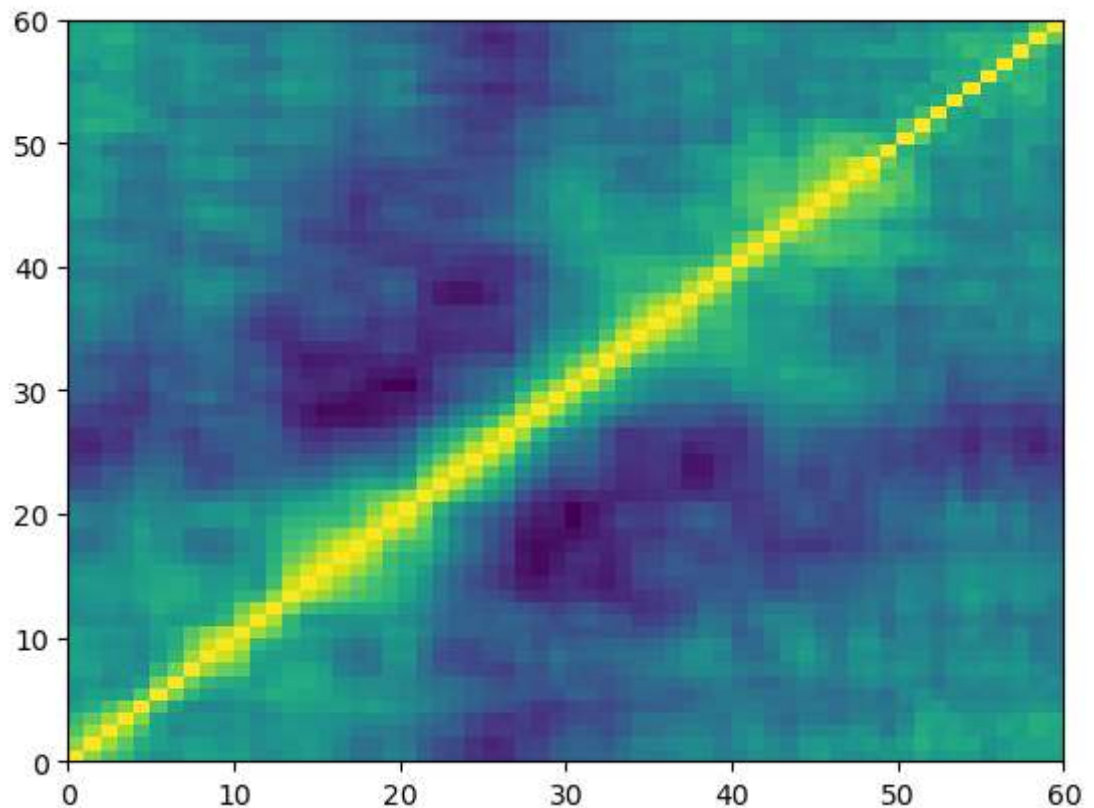
Out[18]: 0.7799158719104263



```
In [20]: 1 # heatmap of correlation between attributes of the dataframe
2 # for pairwise collinearity > 0.7, it is multi-collinearity and can lead to
3 # if it is 1 it is a mistake and have included same thing twice
4 from pandas import DataFrame
5 cormatrix = DataFrame(df.corr())
6 plt.pcolor(cormatrix)
7 plt.show()
```

C:\Users\Hp\AppData\Local\Temp\ipykernel\_27448\2863054692.py:5: FutureWarning:

The default value of numeric\_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric\_only to silence this warning.



```
In [ ]: 1 # End of this project
```

```
In [ ]: 1
```