Bi variate Analysis

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as pt
import seaborn as sns
```

In [13]:
```
visadf=pd.read_csv('C:/Users/Anuja_PC/OneDrive/Documents/dataFiles/Visadataset.c
visadf
```

Out[13]:

	case_id	continent	education_of_employee	has_job_experience	requires_job_t
0	EZYV01	Asia	High School	N	
1	EZYV02	Asia	Master's	Y	
2	EZYV03	Asia	Bachelor's	N	
3	EZYV04	Asia	Bachelor's	N	
4	EZYV05	Africa	Master's	Y	
...	...	...	...	...	
25475	EZYV25476	Asia	Bachelor's	Y	
25476	EZYV25477	Asia	High School	Y	
25477	EZYV25478	Asia	Master's	Y	
25478	EZYV25479	Asia	Master's	Y	
25479	EZYV25480	Asia	Bachelor's	Y	

25480 rows × 12 columns

- Analyse the two variables,
- that may be 2 categorical columns,
- may be 2 numerical columns or
- 1 categorical and 1 numerical

Categorical v/s Categorical

In [ ]:
```
visadf.select_dtypes(include='object').columns
```

In [ ]:
```
visadf.select_dtypes(exclude='object').columns  #numerical columns
```

Continent and case status

In [ ]:
```
visadf['case_status'].value_counts()
```

- **How many asia ppl got the visa certified?**

- **How many asia ppl got the visa denied?**

In [ ]:
```
cond1=visadf['continent']== 'Asia'
cond2=visadf['case_status']== 'Certified'
con = cond1 & cond2
visadf[con]
```

In [ ]:
```
len(visadf[con])

print(f"Number of employee visa certified are: {len(visadf[con])}")
```

In [ ]:
```
cond1=visadf['continent']== 'Asia'
cond2=visadf['case_status']== 'Denied'
con=cond1 & cond2
visadf[con]
print(f"Number of employee visa denied are: {len(visadf[con])}")
```

In [ ]:
```
visadf['continent'].unique()
```

In [ ]:
```
visadf['case_status'].unique()
```

In [ ]:
```
contkey = visadf['continent'].keys

contkey
```

In [ ]:
```
contkey = visadf['continent'].unique()
certList,deniedlist=[],[]
for i in contkey:
    cond1=(visadf['continent']== i)
    cond2=visadf['case_status']== 'Certified'
    cond3=visadf['case_status']== 'Denied'

    cert_cond = cond1&cond2
    den_cond = cond1&cond3

    certList.append(len(visadf[cert_cond]))
    deniedlist.append(len(visadf[den_cond]))

certList,deniedlist
```

In [ ]:
```
pd.DataFrame(zip(certList,deniedlist),columns=['Certified','Denied'],index=contk
```

In [ ]:
```
visadf['continent'].unique()
```

In [ ]:
```
visadf['continent'].value_counts()
```

In [ ]:
```
col1=visadf['continent']
col2=visadf['case_status']
result1=pd.crosstab(col1,col2)   # 1st argument is index, second is column
result1
```

In [ ]:
```
result1.plot(kind='bar')
```

In [ ]:
```
col1=visadf['continent']
col2=visadf['case_status']
```

```
result= pd.crosstab(col2,col1)
```

In [ ]:
```
result
```

In [ ]:
```
result.plot(kind='bar')
```

In [ ]:
```
col1=visadf['continent']
col2=visadf['education_of_employee']
col3=visadf['case_status']
col=[col1,col2]
result2= pd.crosstab(col,col3)
result2
```

In [ ]:
```
result2.plot(kind='bar')
```

In [ ]:
```

```

In [ ]:
```
col1=visadf['continent']
col2=visadf['education_of_employee']
col3=visadf['case_status']
col=[col3,col2]
result3= pd.crosstab(col,col1)
result3
```

In [ ]:
```
result3.plot(kind='bar')
```

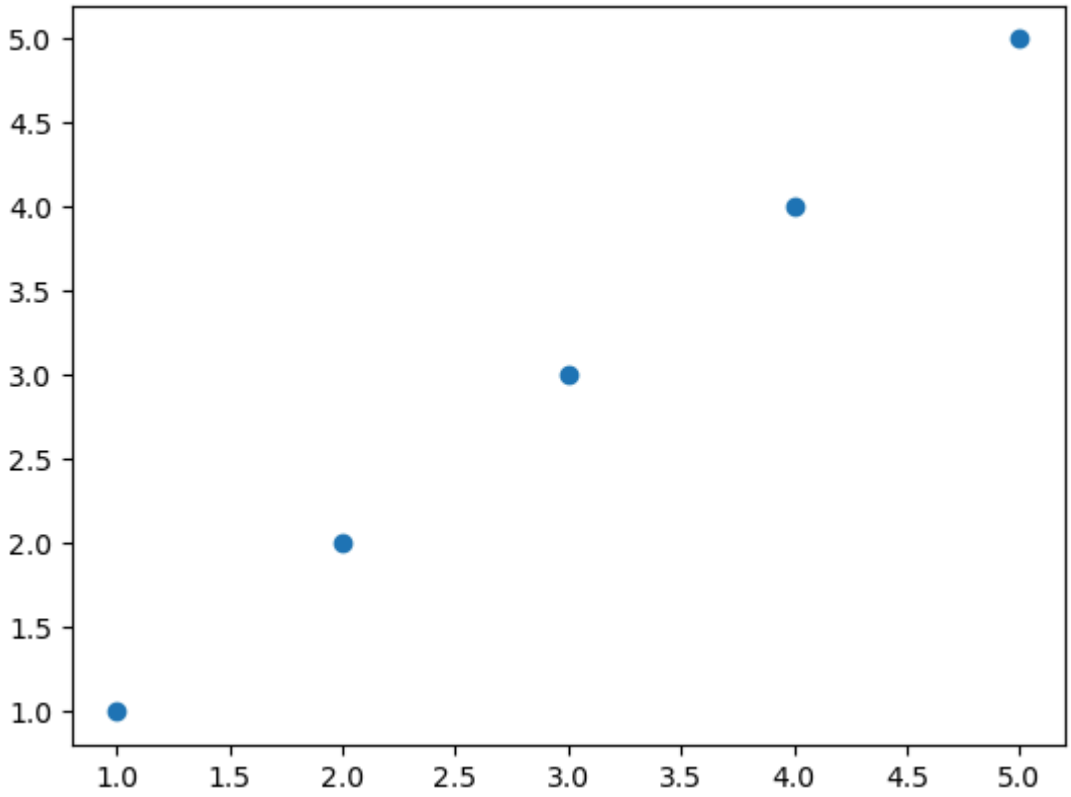
In [ ]:
```

```

Numerical-Numerical

In [ ]:
```
- In order to plot numerical v/s numerical we need to use scatter plots
- Scatter plots give the relation between 2 columns
- it is undet matplotlib
```

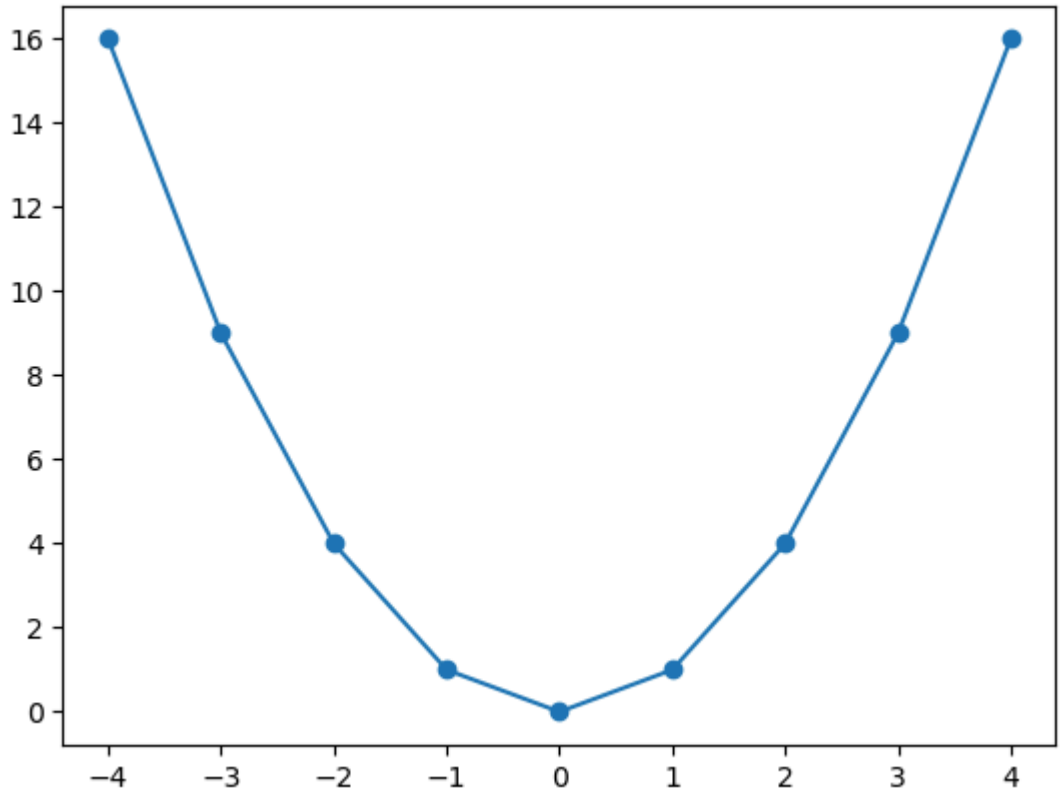
In [21]:
```
x=[1,2,3,4,5]
y=[1,2,3,4,5]

# y=x plot

pt.scatter(x,y)
```

Out[21]:
```
<matplotlib.collections.PathCollection at 0x214ff2d2190>
```



In [ ]:
```
pt.plot(x,y)
```

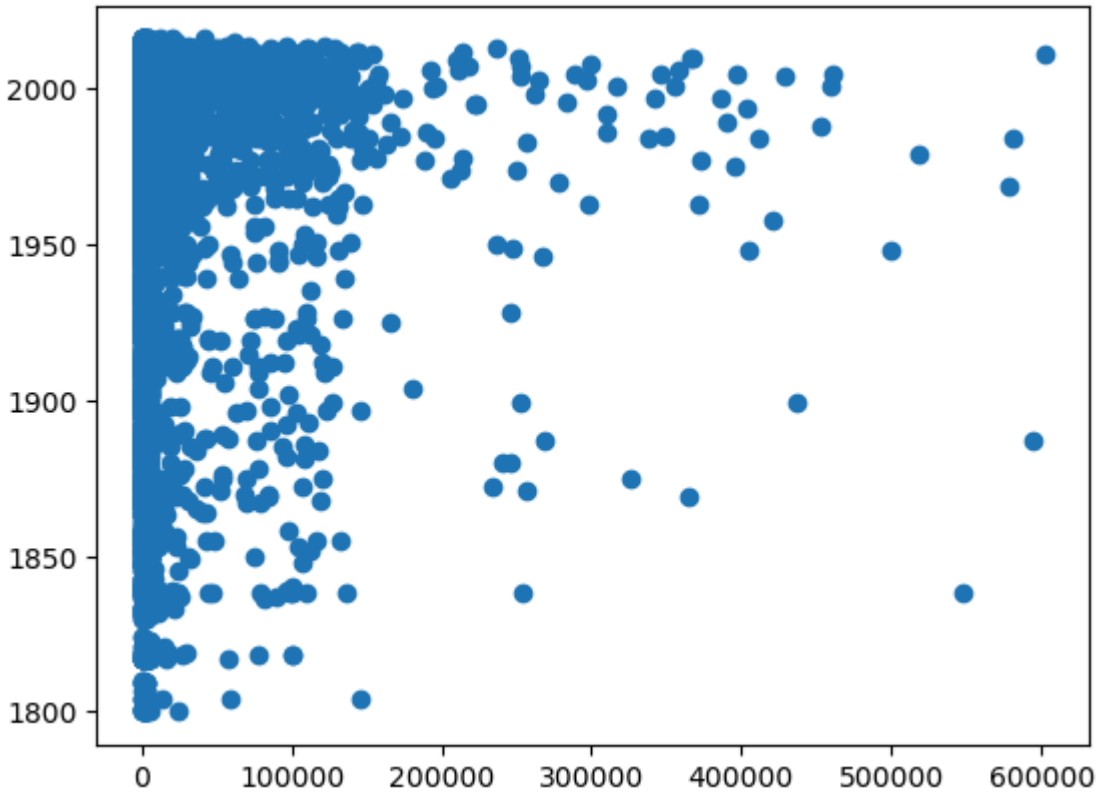
In [22]:
```
x=[i for i in range(-4,5)]
y=[i*i for i in x]

pt.scatter(x,y)
pt.plot(x,y)
pt.show()
```



Scatterplot1

In [24]:
```
col1=visadf['no_of_employees']
col2=visadf['yr_of_estab']
pt.scatter(col1,col2)
#pt.plot(col1,col2)
```

Out[24]: <matplotlib.collections.PathCollection at 0x214ffa85ad0>



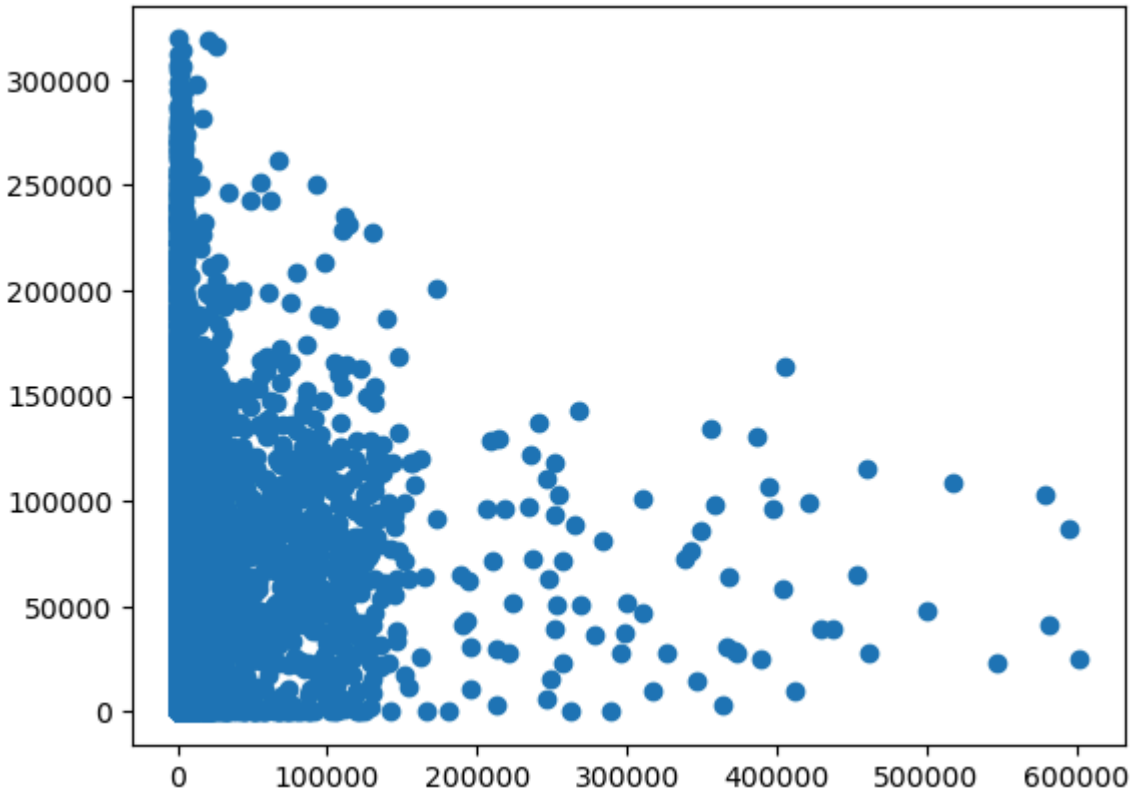
- Observation - no relation

Scatterplot2

In [ ]:
```
visadf.select_dtypes(exclude='object').columns  #numerical columns
```

In [25]:
```
col1=visadf['no_of_employees']
col2=visadf['prevailing_wage']
pt.scatter(col1,col2)
```

Out[25]: <matplotlib.collections.PathCollection at 0x214ffa85e50>



In [ ]:
```
col1=visadf['yr_of_estab']
col2=visadf['prevailing_wage']
pt.scatter(col1,col2)
```

- all 3 no relation

In [ ]:
```
pt.figure(figsize=(14,3))
col1=visadf['no_of_employees']
col2=visadf['yr_of_estab']
pt.subplot(1,3,1).scatter(col1,col2)

col1=visadf['no_of_employees']
col2=visadf['yr_of_estab']
pt.subplot(1,3,2).scatter(col1,col2)

col1=visadf['no_of_employees']
col2=visadf['yr_of_estab']
pt.subplot(1,3,3).scatter(col1,col2)
```

Pearsons correlation coefficient

-- inspect image - edit as html - copy - paste in jupitor - esc M - shift enter

- It give the amount of relation between variables
- It is denoted with r
- r varies from -1 to 1
- r value for positive relation from 0 to 1
- for negative relation r varies from -1 to 0
- for no relation r approx. equals to 0

- in pythn code we have corr function under pandas
- this gives covariance matrix
- in this data we have 3 num calumns , so will get 9
- all upper and lower triangles represents co-variance
- all trace matrix are variance

In [ ]:

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}$$

In [ ]:
```
visadf.corr(numeric_only=True)
```

- Observations - the correlationvalue between no_of_employees,yr_of_estab,prevailing_wage is 0, which indicates no relation

In [39]:
```
wineQualityDf=pd.read_excel(r"C:\Users\Anuja_PC\OneDrive\Documents\DataScience_N
wineQualityDf
```

Out[39]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulph
0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
1	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
3	7.8	0.880	0.00	2.6	0.098	25.0	67.0	0.99680	3.20	
4	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
...	...	...	...	...	...	...	...	...	...	
3193	6.3	0.510	0.13	2.3	0.076	29.0	40.0	0.99574	3.42	
3194	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
3195	5.9	0.645	0.12	2.0	0.075	32.0	44.0	0.99547	3.57	
3196	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
3197	6.0	0.310	0.47	3.6	0.067	18.0	42.0	0.99549	3.39	

3198 rows × 12 columns

In [42]:
```
wineDataDf=wineQualityDf.drop_duplicates()
```

In [45]:
```
wineDataDf.iloc[1:,:]
```

Out[45]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulph
1	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	
3	7.8	0.880	0.00	2.6	0.098	25.0	67.0	0.99680	3.20	
5	7.8	0.760	0.04	2.3	0.092	15.0	54.0	0.99700	3.26	
7	11.2	0.280	0.56	1.9	0.075	17.0	60.0	0.99800	3.16	
11	7.4	0.660	0.00	1.8	0.075	13.0	40.0	0.99780	3.51	
...	...	...	...	...	...	...	...	...	...	
3187	6.8	0.620	0.08	1.9	0.068	28.0	38.0	0.99651	3.42	
3189	6.2	0.600	0.08	2.0	0.090	32.0	44.0	0.99490	3.45	
3191	5.9	0.550	0.10	2.2	0.062	39.0	51.0	0.99512	3.52	
3195	5.9	0.645	0.12	2.0	0.075	32.0	44.0	0.99547	3.57	
3197	6.0	0.310	0.47	3.6	0.067	18.0	42.0	0.99549	3.39	

1359 rows × 12 columns

In [46]:
```
wineDataDf.columns
```

Out[46]:
```
Index(['fixed acidity', 'volatile acidity', 'citric acid', 'residual sugar',
       'chlorides', 'free sulfur dioxide', 'total sulfur dioxide', 'density',
       'pH', 'sulphates', 'alcohol', 'quality'],
      dtype='object')
```

In [ ]:
```
# if corellation on 12 columns
# 12 * 12
```

In [47]:
```
len(wineDataDf.columns)
```

Out[47]:
```
12
```

In [48]:
```
winecorr=wineDataDf.corr()
winecorr
```

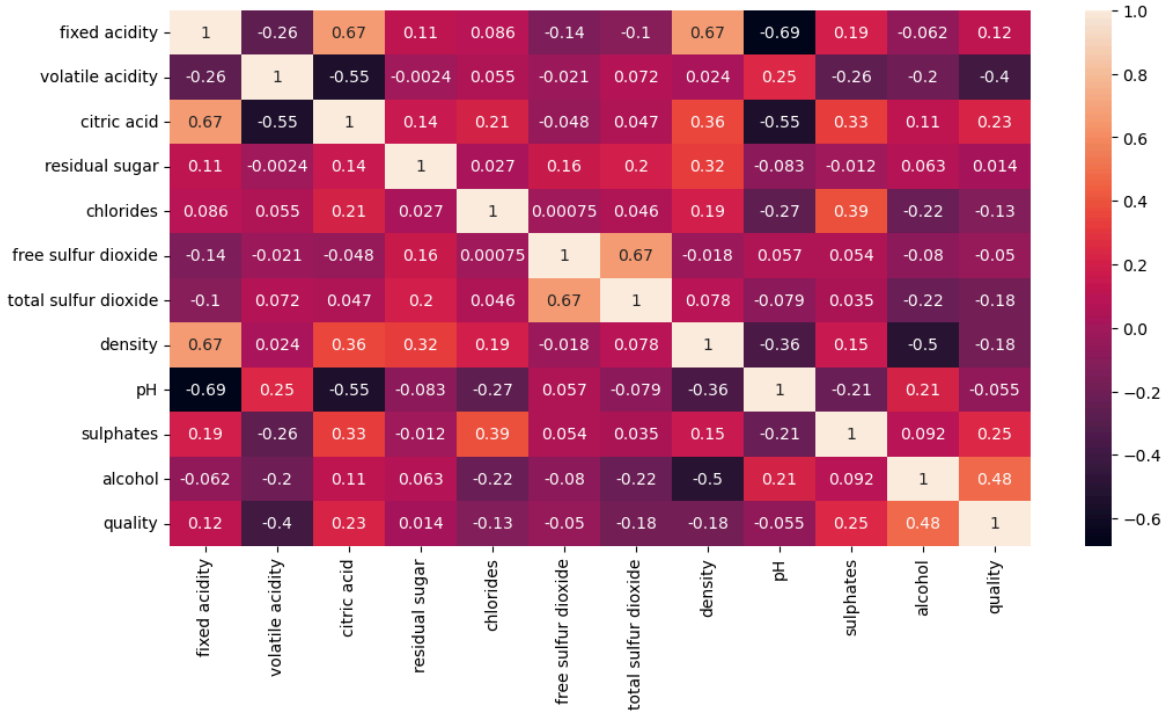
Out[48]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	
fixed acidity	1.000000	-0.255124	0.667437	0.111025	0.085886	-0.140580	-0.103777	(
volatile acidity	-0.255124	1.000000	-0.551248	-0.002449	0.055154	-0.020945	0.071701	(
citric acid	0.667437	-0.551248	1.000000	0.143892	0.210195	-0.048004	0.047358	(
residual sugar	0.111025	-0.002449	0.143892	1.000000	0.026656	0.160527	0.201038	(
chlorides	0.085886	0.055154	0.210195	0.026656	1.000000	0.000749	0.045773	(
free sulfur dioxide	-0.140580	-0.020945	-0.048004	0.160527	0.000749	1.000000	0.667246	-(
total sulfur dioxide	-0.103777	0.071701	0.047358	0.201038	0.045773	0.667246	1.000000	(
density	0.670195	0.023943	0.357962	0.324522	0.193592	-0.018071	0.078141	1
pH	-0.686685	0.247111	-0.550310	-0.083143	-0.270893	0.056631	-0.079257	-(
sulphates	0.190269	-0.256948	0.326062	-0.011837	0.394557	0.054126	0.035291	(
alcohol	-0.061596	-0.197812	0.105108	0.063281	-0.223824	-0.080125	-0.217829	-(
quality	0.119024	-0.395214	0.228057	0.013640	-0.130988	-0.050463	-0.177855	-(

$Heat - Map$

- Heat map will provide the matrix representation of correlation value
- it represents values in colour format
- beside matrix it will display color bar
- color bar means like a scae of values with colour
- It is under seaborn package

In [49]:
```
pt.figure(figsize=(12,6))
sns.heatmap(winecorr,annot=True)
pt.show()
```



In [51]:
```
import seaborn as sns
sns.__version__
```

Out[51]: '0.13.2'

In [50]:
```
numVisaDf=visadf.select_dtypes(exclude=object)
numVisaDf
```

Out[50]:

	no_of_employees	yr_of_estab	prevailing_wage
0	14513	2007	592.2029
1	2412	2002	83425.6500
2	44444	2008	122996.8600
3	98	1897	83434.0300
4	1082	2005	149907.3900
...	...	...	...
25475	2601	2008	77092.5700
25476	3274	2006	279174.7900
25477	1121	1910	146298.8500
25478	1918	1887	86154.7700
25479	3195	1960	70876.9100

25480 rows × 3 columns

In [52]:
```
visaCorr= numVisaDf.corr()
visaCorr
```

Out[52]:

	no_of_employees	yr_of_estab	prevailing_wage
no_of_employees	1.000000	-0.017770	-0.009523
yr_of_estab	-0.017770	1.000000	0.012342
prevailing_wage	-0.009523	0.012342	1.000000

In [53]:
```
sns.heatmap(visaCorr,annot=True)
```

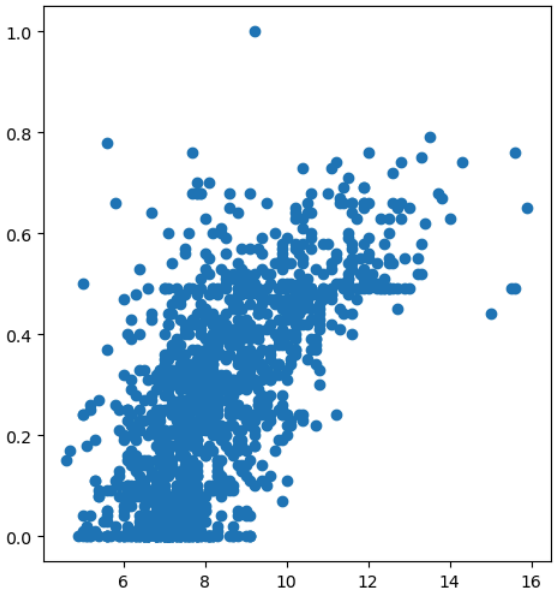
Out[53]: <Axes: >

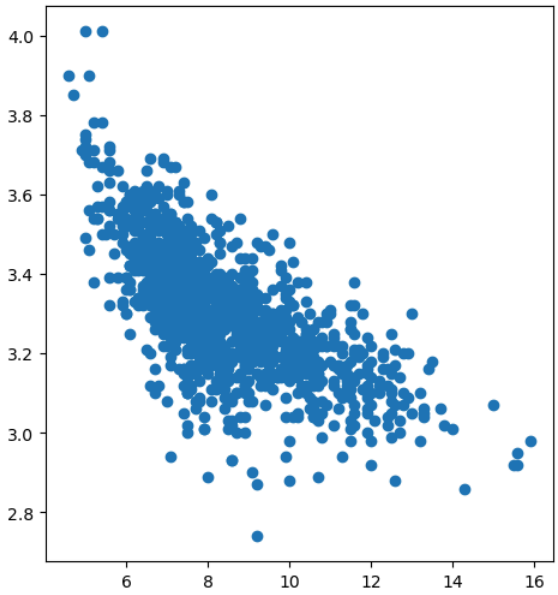
In [ ]:
```
# draw the scatter plots for wine data columns , which is having highest positiv
# and which is having highest negative correlation
```

In [26]:
```
wineQualityDf.columns
```

Out[26]:
```
Index(['fixed acidity', 'volatile acidity', 'citric acid', 'residual sugar',
       'chlorides', 'free sulfur dioxide', 'total sulfur dioxide', 'density',
       'pH', 'sulphates', 'alcohol', 'quality'],
      dtype='object')
```

In [54]:
```
pt.figure(figsize=(12,6))
col1=wineDataDf['fixed acidity']
col2=wineDataDf['citric acid']
pt.subplot(1,2,1).scatter(col1,col2)
col3=wineDataDf['pH']
pt.subplot(1,2,2).scatter(col1,col3)
pt.show()
```


1.0
0.8
0.6
0.4
0.2
0.0
6
8
10
12
14
16


4.0
3.8
3.6
3.4
3.2
3.0
2.8
6
8
10
12
14
16