In [4]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
path =r'C:\Users\aramaiah.ASUAD\Naresh_IT\MyDataScience\Data_Files\Visadataset.
visa_df=pd.read_csv(path)
visa_df.head(6)
```

Out[4]:

	case_id	continent	education_of_employee	has_job_experience	requires_job_training	no_of_e
0	EZYV01	Asia	High School	N	N	
1	EZYV02	Asia	Master's	Y	N	
2	EZYV03	Asia	Bachelor's	N	Y	
3	EZYV04	Asia	Bachelor's	N	N	
4	EZYV05	Africa	Master's	Y	N	
5	EZYV06	Asia	Master's	Y	N	

CATEGORICAL vs *CATEGORICAL*

In [ ]:
```
# CONTINENT a applicats
# Case_status
# as we know that there are 25480 observations are there
# in that 16k are from asia applicats
# out of 16k applicants how many visa approved
# out of 1k how mnay visa rejected
```

In [15]:
```
c1=visa_df['continent']=='Asia'
c2=visa_df['case_status']=='Certified'
c3=visa_df['case_status']=='Denied'
c4=visa_df['continent'].unique()
con1 = c1&c2
con2 = c1&c3
Certified_visa_count1=len(visa_df[con1])
Denied_visa_count2=len(visa_df[con2])
print(f'There are {Certified_visa_count1} got certified from Asia')
print(f'There are {Denied_visa_count2} got denied from Asia')
c4
```

```
There are 11012 got certified from Asia
There are 5849 got denied from Asia
```

Out[15]:
```
array(['Asia', 'Africa', 'North America', 'Europe', 'South America',
       'Oceania'], dtype=object)
```

In [ ]:
```
     Denied  Certified
Asia v1         v2
Europe v1       v2
```

In [23]:
```
# Step1 : make unique label
labels=visa_df['continent'].unique()
# Step-2 : create empty two lists
Certified_visa_count1=[]
Denied_visa_count2=[]
# Step-3 : iterate through the loop
for i in labels:
    c1=visa_df['continent']==i
    c2=visa_df['case_status']=='Certified'
    c3=visa_df['case_status']=='Denied'
    cert_con=c1&c2
    den_con =c1&c3

    Certified_visa_count1.append(len(visa_df[cert_con]))
    Denied_visa_count2.append(len(visa_df[den_con]))

Certified_visa_count1,Denied_visa_count2
cols=['Continent','Certified','Denied']
d1=pd.DataFrame(zip(labels,Certified_visa_count1,Denied_visa_count2),columns=cc

d1
```

Out[23]:

	Continent	Certified	Denied
0	Asia	11012	5849
1	Africa	397	154
2	North America	2037	1255
3	Europe	2957	775
4	South America	493	359
5	Oceania	122	70

In [24]:
```
d1.set_index('Continent')
```

Out[24]:

Continent	Certified	Denied
Asia	11012	5849
Africa	397	154
North America	2037	1255
Europe	2957	775
South America	493	359
Oceania	122	70

pd.crosstab

- will take two aguments
- one being index
- another being column

In [30]:
```
col1=visa_df['continent']
col2=visa_df['case_status']
res1=pd.crosstab(col1,col2)
res1
```

Out[30]:

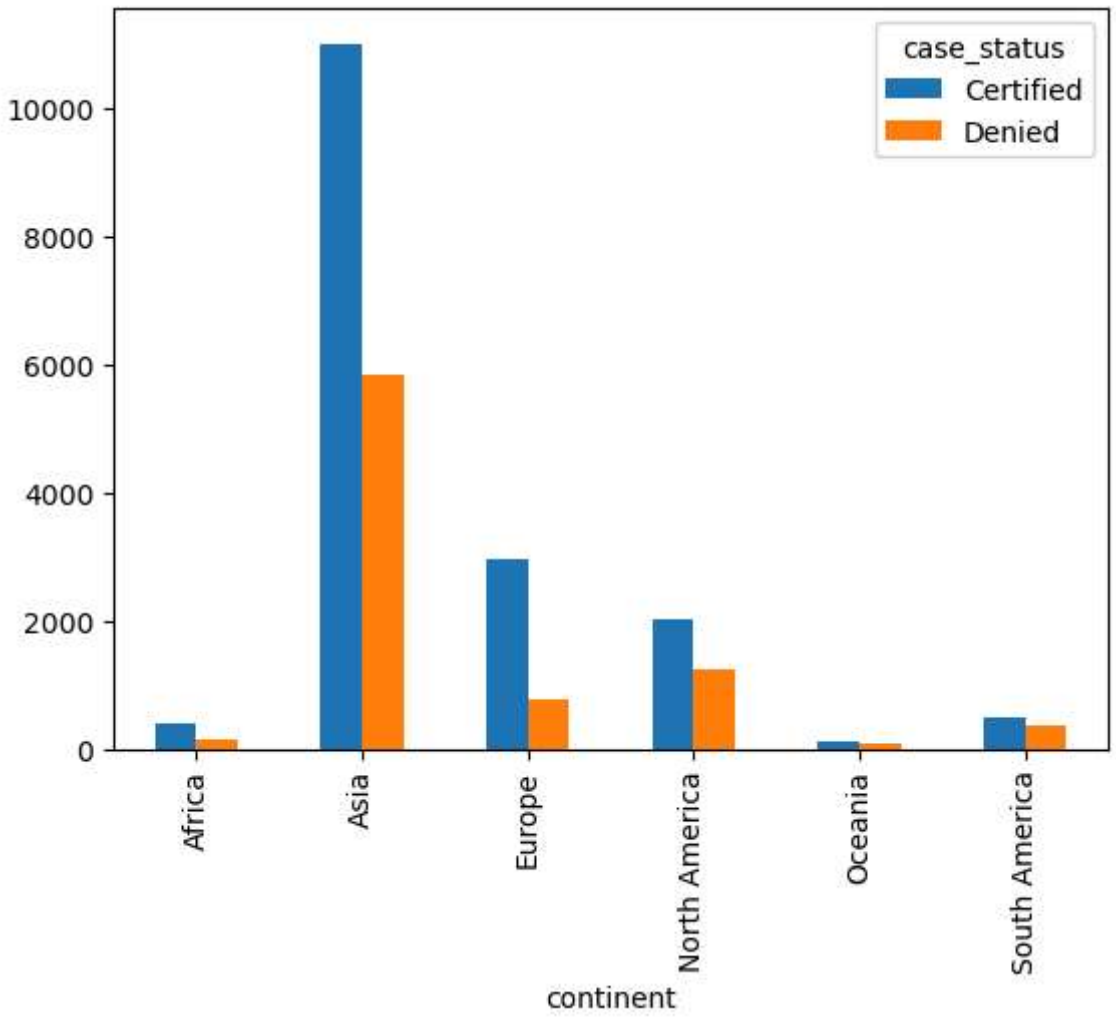
case_status	Certified	Denied
continent		
Africa	397	154
Asia	11012	5849
Europe	2957	775
North America	2037	1255
Oceania	122	70
South America	493	359

In [29]:
```
# in order to understand coulmn 1 and coumn 2
col1=[visa_df['continent'],visa_df['education_of_employee']]
col2=visa_df['case_status']
res2=pd.crosstab(col1,col2)
res2
```

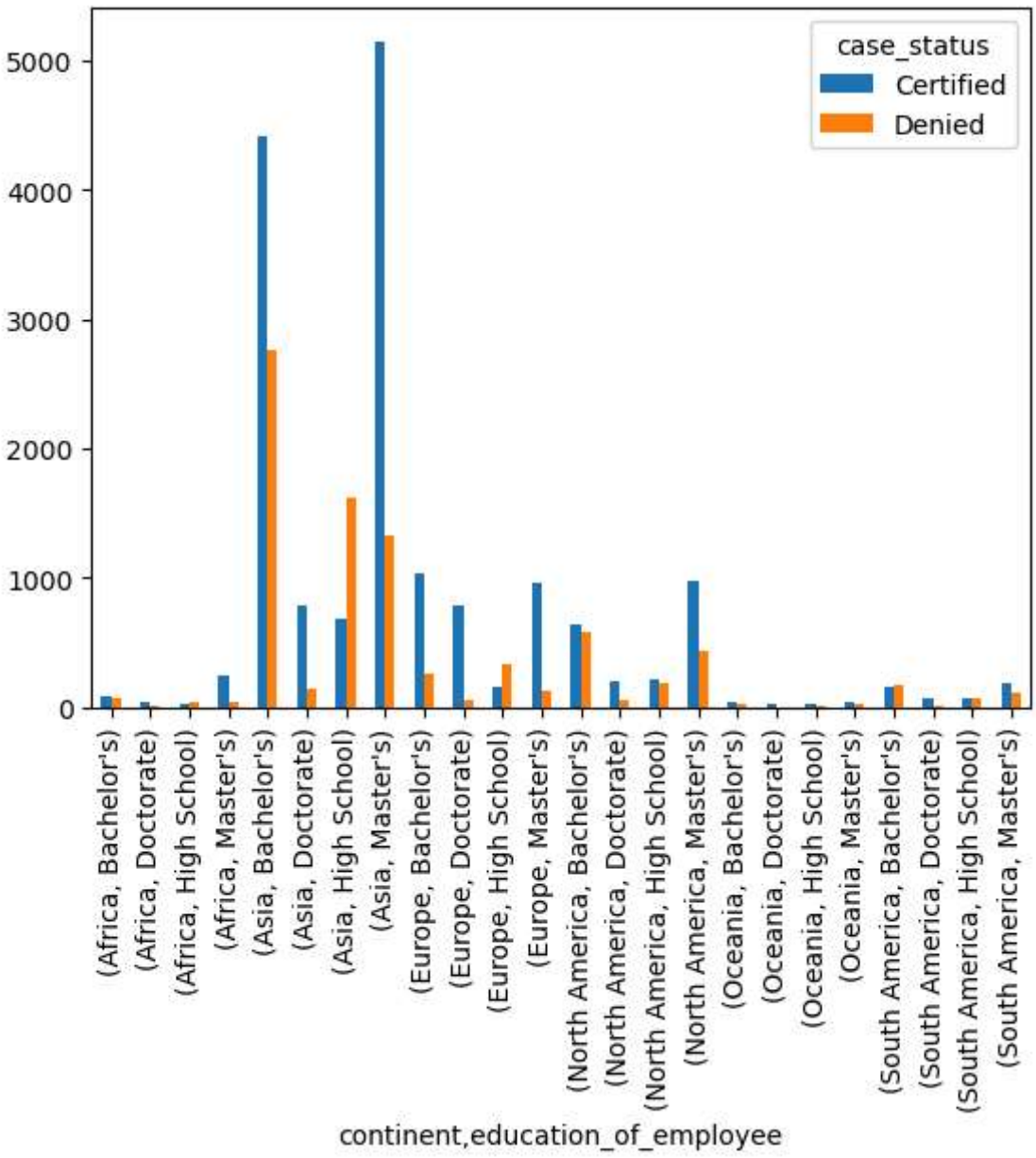
Out[29]:

	case_status	Certified	Denied
continent	education_of_employee		
Africa	Bachelor's	81	62
	Doctorate	43	11
	High School	23	43
	Master's	250	38
Asia	Bachelor's	4407	2761
	Doctorate	780	143
	High School	676	1614
	Master's	5149	1331
Europe	Bachelor's	1040	259
	Doctorate	788	58
	High School	162	328
	Master's	967	130
North America	Bachelor's	641	584
	Doctorate	207	51
	High School	210	191
	Master's	979	429
Oceania	Bachelor's	38	28
	Doctorate	19	3
	High School	19	17
	Master's	46	22
South America	Bachelor's	160	173
	Doctorate	75	14
	High School	74	63
	Master's	184	109

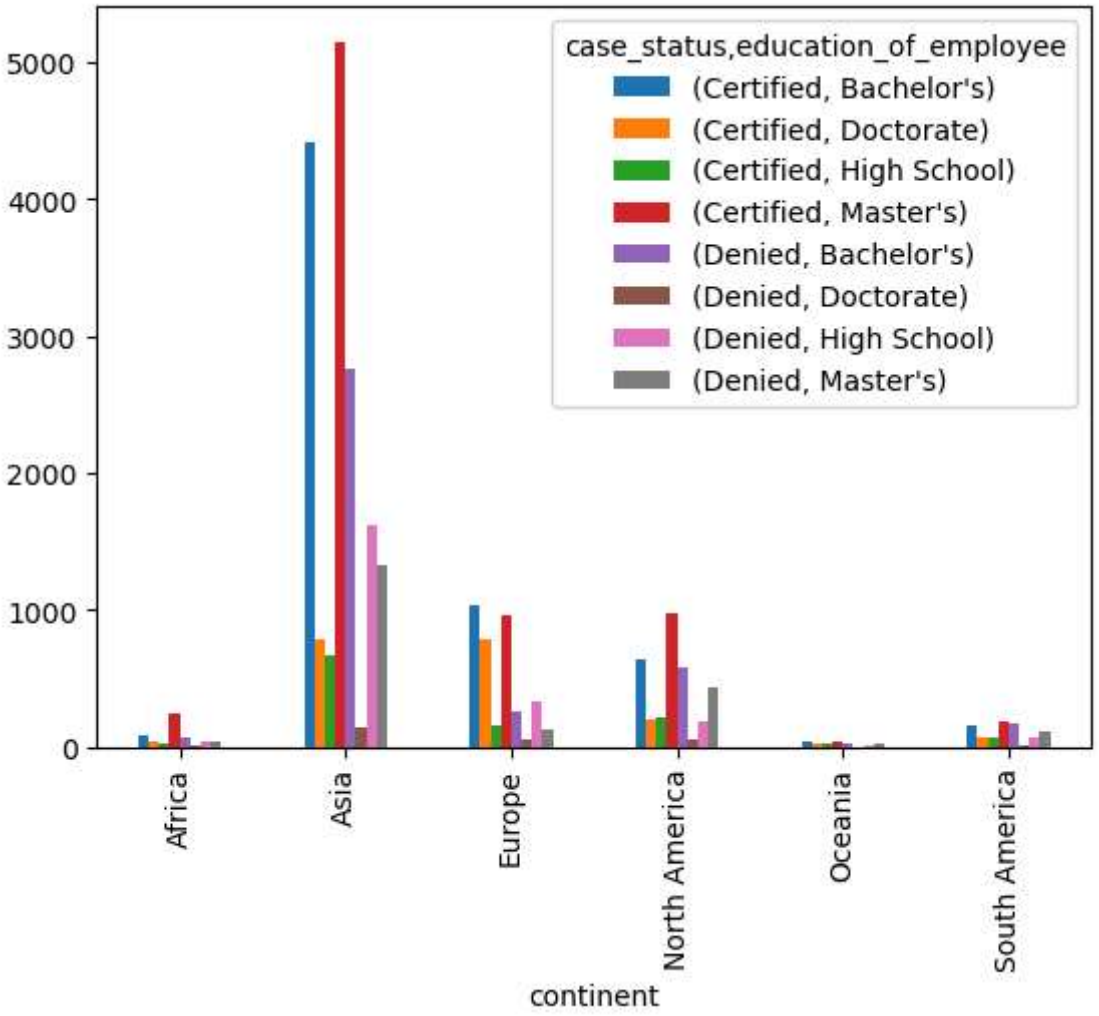
In [55]:
```
res1.plot(kind='bar')
plt.show()
```



In [56]:
```
res2.plot(kind='bar')
plt.show()
```



In [54]:
```
col1=visa_df['continent']
col2=visa_df['case_status']
col3=visa_df['education_of_employee']
r1=pd.crosstab(col1,[col2,col3])
r1.plot(kind='bar')
plt.show()
```

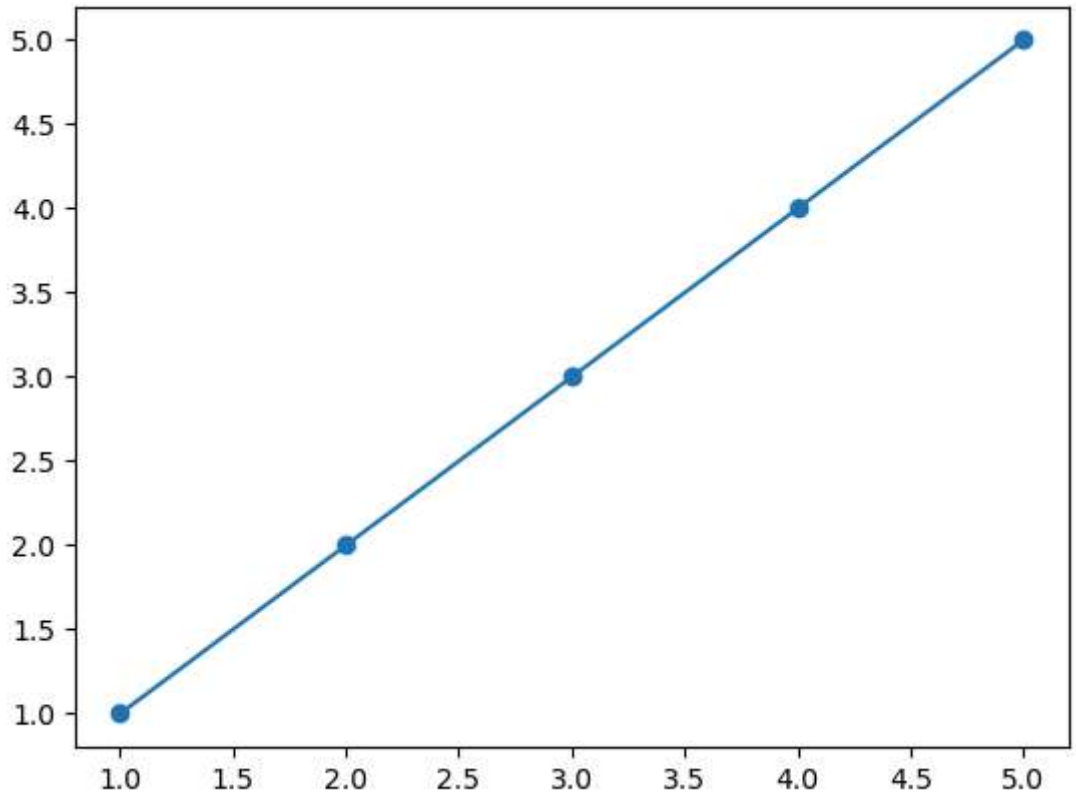


Numerical vs *Numerical*

In [41]:
```
x=[1,2,3,4,5]
y=[1,2,3,4,5]
#(1,1) (2,2) (3,3) (4,4) (5,5)
```

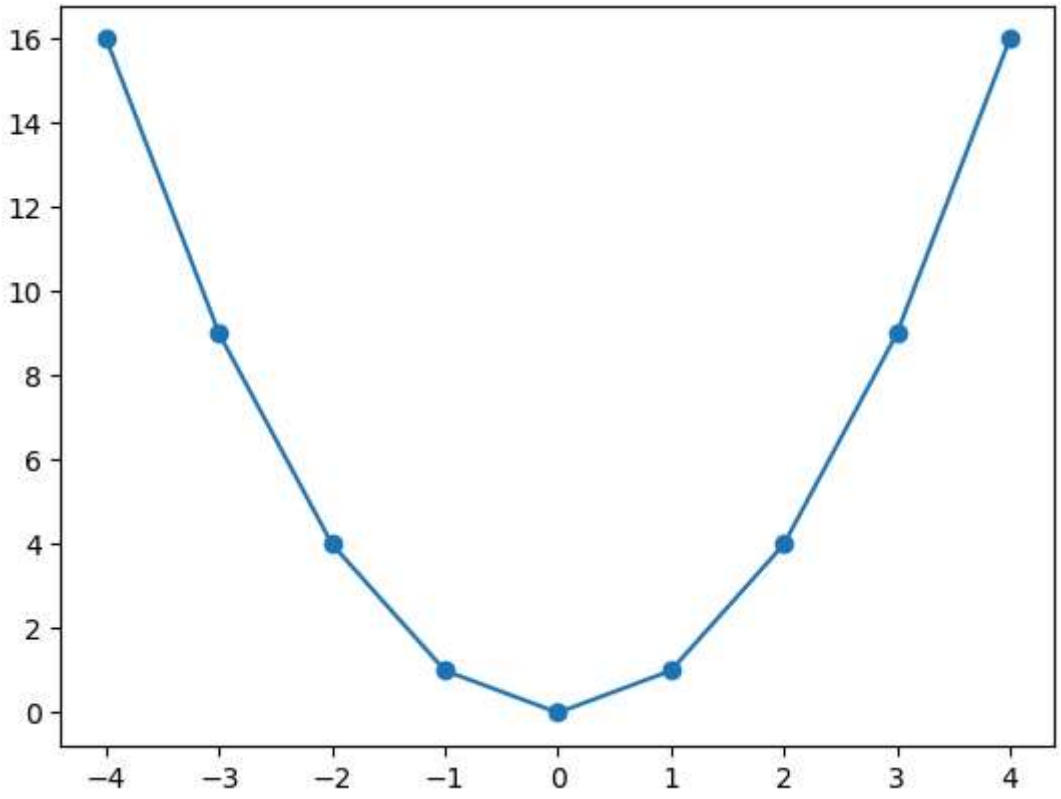
plt.scatter

In [42]:
```
plt.scatter(x,y)
plt.plot(x,y)
plt.show()
```



In [44]:
```
x=[-4,-3,-2,-1,0,1,2,3,4,5]
```

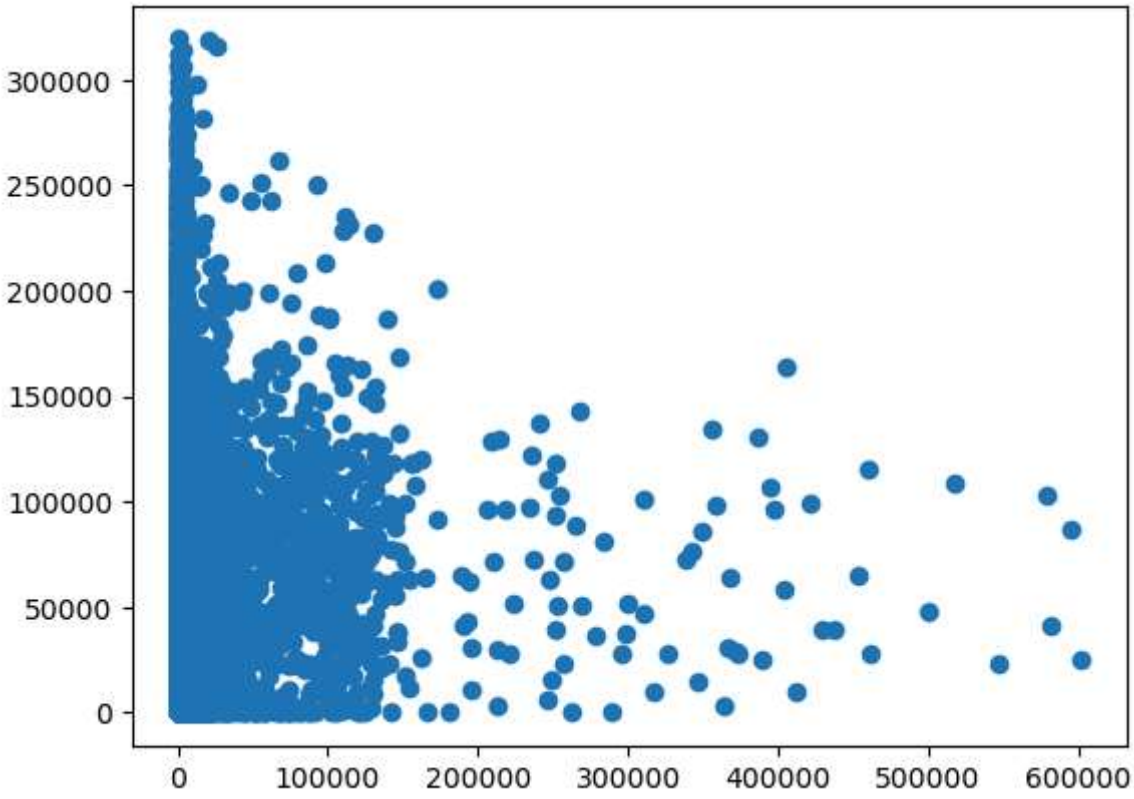
In [45]:
```
x=[i for i in range(-4,5)]
y=[i*i for i in x]
plt.scatter(x,y)
plt.plot(x,y)
plt.show() #relation
```



In [48]:
```
# extrct only numerical coulmns
num_cols=visa_df.select_dtypes(exclude= 'object')
num_cols.columns
```

Out[48]:
```
Index(['no_of_employees', 'yr_of_estab', 'prevailing_wage'], dtype='object')
```

In [49]:
```
col1=visa_df['no_of_employees']
col2=visa_df['prevailing_wage']
plt.scatter(col1,col2)
plt.show()
```



Pearson Correlation Coefficient

- r variefrom -1 to 1
- -1 to 0: Negative relation
- 0 to 1 : positive relation
- 0: No relation

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

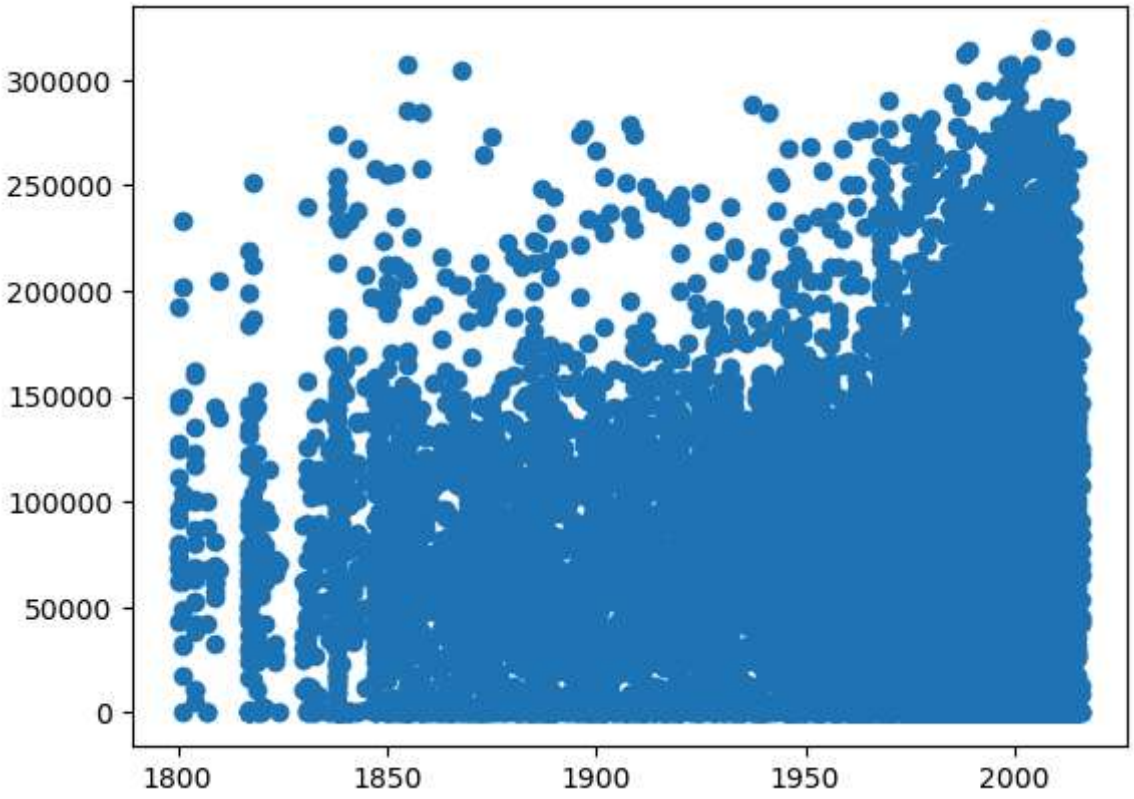
- when ypu do this pythn
- it gives the matrix
- in visa data we have 3 numerical columns arethere
- python will give a matrix wrt 3 numerical columns
- the values in eac field tells about the relation between the variables

In [50]:
```
visa_df.corr(numeric_only=True)
```

Out[50]:

	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]:
```
# check the scatter plot between yr_of_estab
# with prevailing_WAGE
# we are seeing the relation is 0.012342
col1=visa_df['yr_of_estab']
col2=visa_df['prevailing_wage']
plt.scatter(col1,col2)
plt.show()
```



In [59]:
```
wine=pd.read_csv("C:\\Users\\aramaiah.ASUAD\\Naresh_IT\\MyDataScience\\Data_Fil
wine.head()
```

Out[59]:

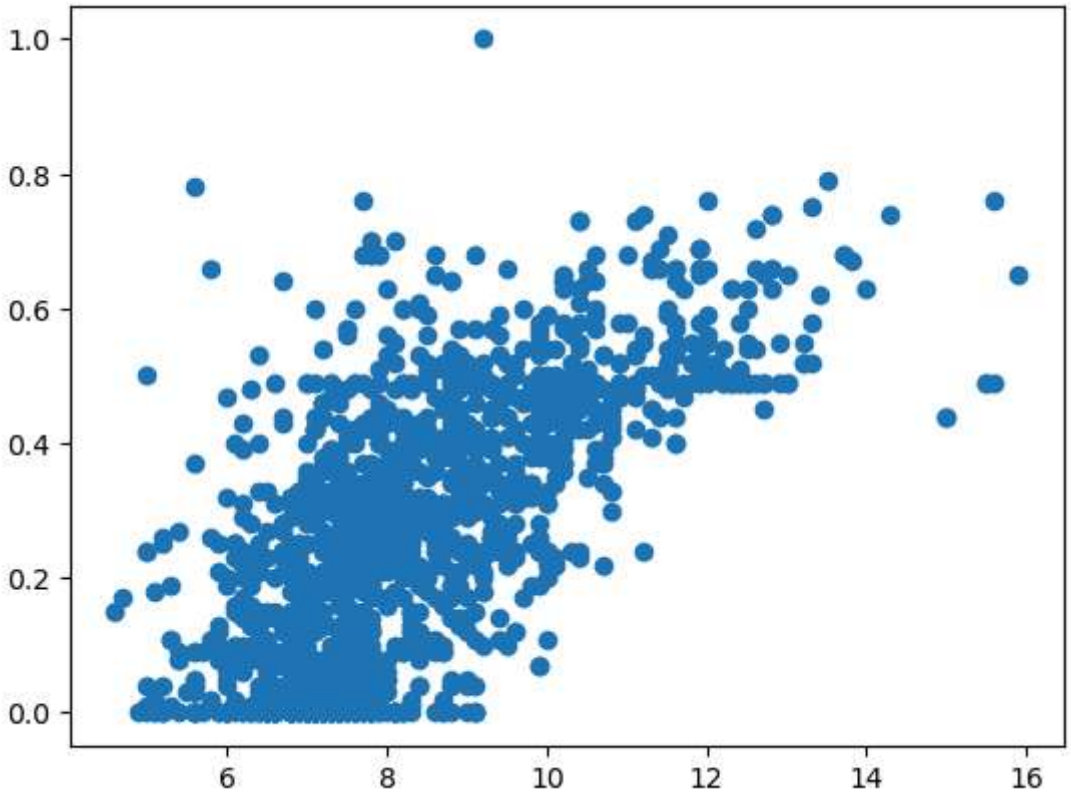
	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	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 [60]:
```
wine.corr()
```

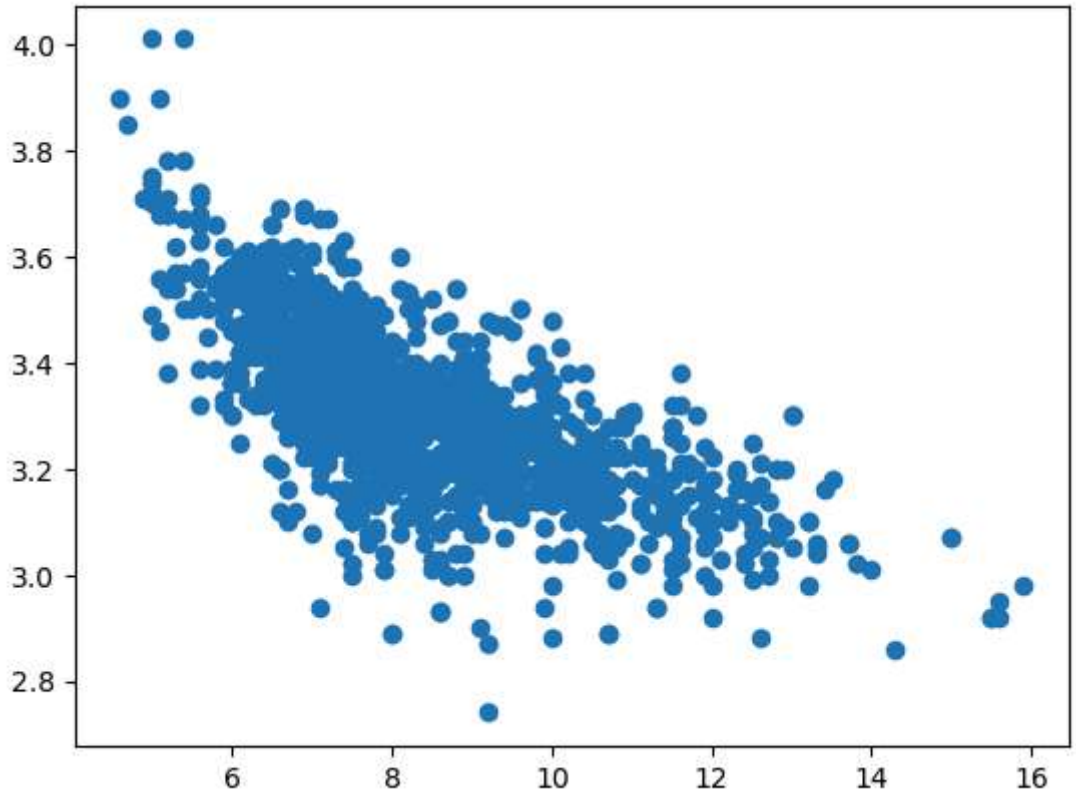
Out[60]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	
fixed acidity	1.000000	-0.256131	0.671703	0.114777	0.093705	-0.153794	-0.113181	0.668047	-
volatile acidity	-0.256131	1.000000	-0.552496	0.001918	0.061298	-0.010504	0.076470	0.022026	
citric acid	0.671703	-0.552496	1.000000	0.143577	0.203823	-0.060978	0.035533	0.364947	-
residual sugar	0.114777	0.001918	0.143577	1.000000	0.055610	0.187049	0.203028	0.355283	-
chlorides	0.093705	0.061298	0.203823	0.055610	1.000000	0.005562	0.047400	0.200632	-
free sulfur dioxide	-0.153794	-0.010504	-0.060978	0.187049	0.005562	1.000000	0.667666	-0.021946	
total sulfur dioxide	-0.113181	0.076470	0.035533	0.203028	0.047400	0.667666	1.000000	0.071269	-
density	0.668047	0.022026	0.364947	0.355283	0.200632	-0.021946	0.071269	1.000000	-
pH	-0.682978	0.234937	-0.541904	-0.085652	-0.265026	0.070377	-0.066495	-0.341699	
sulphates	0.183006	-0.260987	0.312770	0.005527	0.371260	0.051658	0.042947	0.148506	-
alcohol	-0.061668	-0.202288	0.109903	0.042075	-0.221141	-0.069408	-0.205654	-0.496180	
quality	0.124052	-0.390558	0.226373	0.013732	-0.128907	-0.050656	-0.185100	-0.174919	-

In [62]:
```
# fixed acidity columns and citric acid column :0.67 +ve
col11=wine['fixed acidity']
col22=wine['citric acid']
plt.scatter(col11,col22)
plt.show()
```



In [63]:
```
# fixed acidity columns and pH column :0.67 +ve
col11=wine['fixed acidity']
col33=wine['pH']
plt.scatter(col11,col33)
plt.show()
```



heat map

- heat map is useful to visulation of matrix
- it is under seaborn pacakges
- heat map will varies the varies and gives the color about the value

In [65]:
```
corr_visa=visa_df.corr(numeric_only=True)
corr_visa
# this is a matrix we want apply to heat map
```

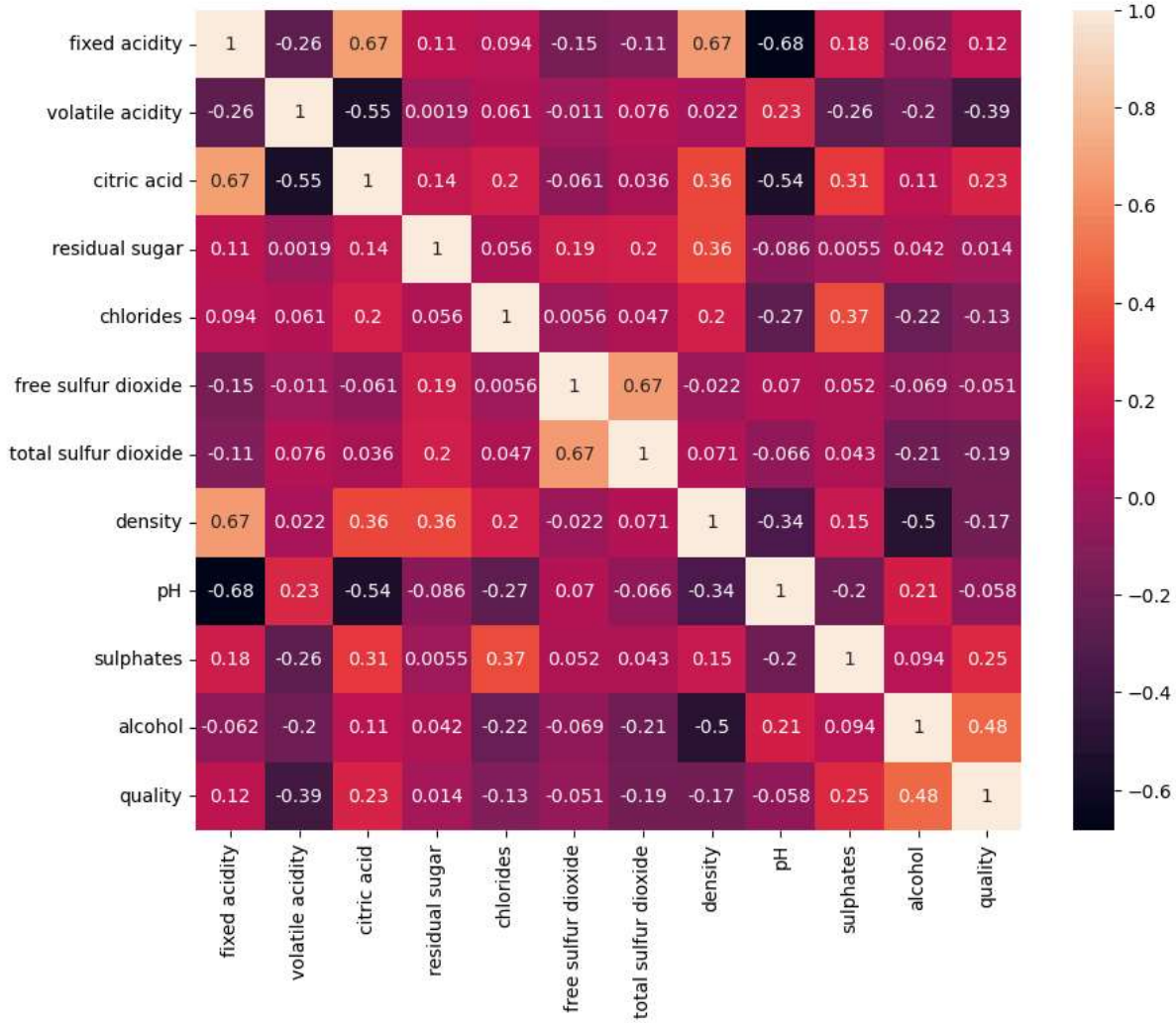
Out[65]:

	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 [67]:
```
sns.heatmap(corr_visa,annot=True)
```

Out[67]: <Axes: >

In [70]:
```
corr_wine=wine.corr(numeric_only=True)
plt.figure(figsize=(10,8))
sns.heatmap(corr_wine,annot=True)
plt.show()
```



In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: