

```
In [1]: # Import the packages
# Read the data

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

path=r"C:\Users\omkar\OneDrive\Documents\Data science\Naresh IT\Datafiles\V:
visa_df=pd.read_csv(path)
visa_df.head(3)
```

```
Out[1]:
```

	case_id	continent	education_of_employee	has_job_experience	requires_job_training	no_
0	EZYV01	Asia	High School	N	N	
1	EZYV02	Asia	Master's	Y	N	
2	EZYV03	Asia	Bachelor's	N	Y	

### *Categorical vs Categorical*

```
In [ ]: # continent
# case_status
# as we know that there are 25480 observations are there
# in that 16k are from asia applicants
# out of 16k applicants how many visa certified
# out of 16k applicants how many visa denied
```

```
In [11]: c1=visa_df['continent']=='Asia'
c2=visa_df['case_status']=='Certified'
c3=visa_df['case_status']=='Denied'

cert_con=c1&c2
den_con=c1&c3

certified_count=len(visa_df[cert_con])
denied_count=len(visa_df[den_con])
print(f"there are {certified_count} got certified visa from Asia")
print(f"there are {denied_count} got denied visa from Asia")
```

there are 11012 got certified visa from Asia  
there are 5849 got denied visa from Asia

```
In [ ]:
```

	Denied	Certified
Asia	v1	v2
Europe	v1	v2

```
In [21]: # step-1: make unique lables
labels=visa_df['continent'].unique()
# step-2: create empty two lists
certified_count=[]
denied_count=[]
# step-3: iterate through 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_count.append(len(visa_df[cert_con]))
    denied_count.append(len(visa_df[den_con]))

cols=['Continent','Certified','Denied']
d1=pd.DataFrame(zip(labels,
                    certified_count,
                    denied_count), columns=cols)
d1.set_index('Continent')
```

Out[21]:

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

```
In [20]: d1.set_index('Continent')
```

Out[20]:

	Certified	Denied
Continent		
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 arguments
- index
- column

```
In [25]: col1=[visa_df['continent']]
col2=visa_df['case_status']
result1=pd.crosstab(col1,col2)
result1
```

```
Out[25]:
```

	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

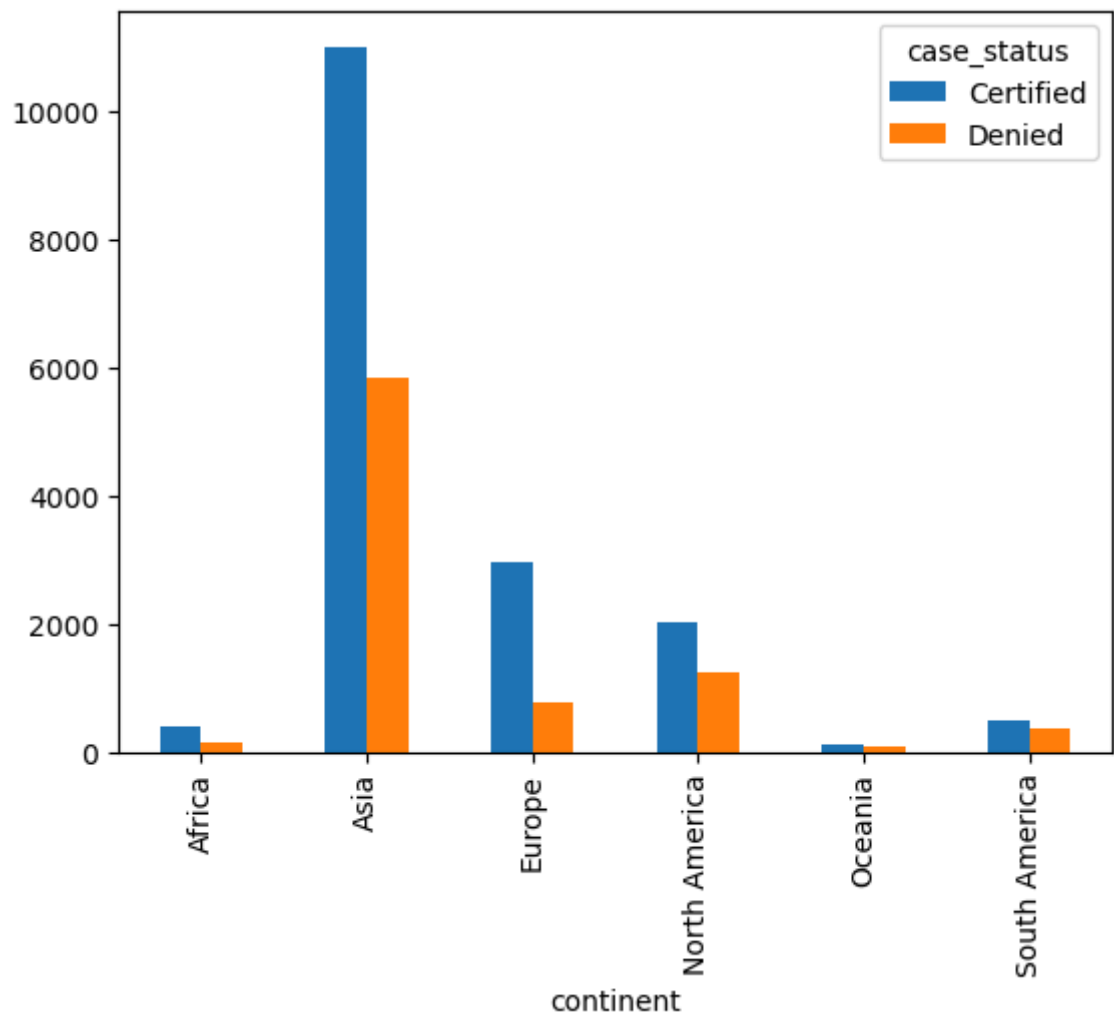
```
In [26]: col1=[visa_df['continent'],
            visa_df['education_of_employee']]
col2=visa_df['case_status']
result2=pd.crosstab(col1,col2)
result2
```

Out[26]:

		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

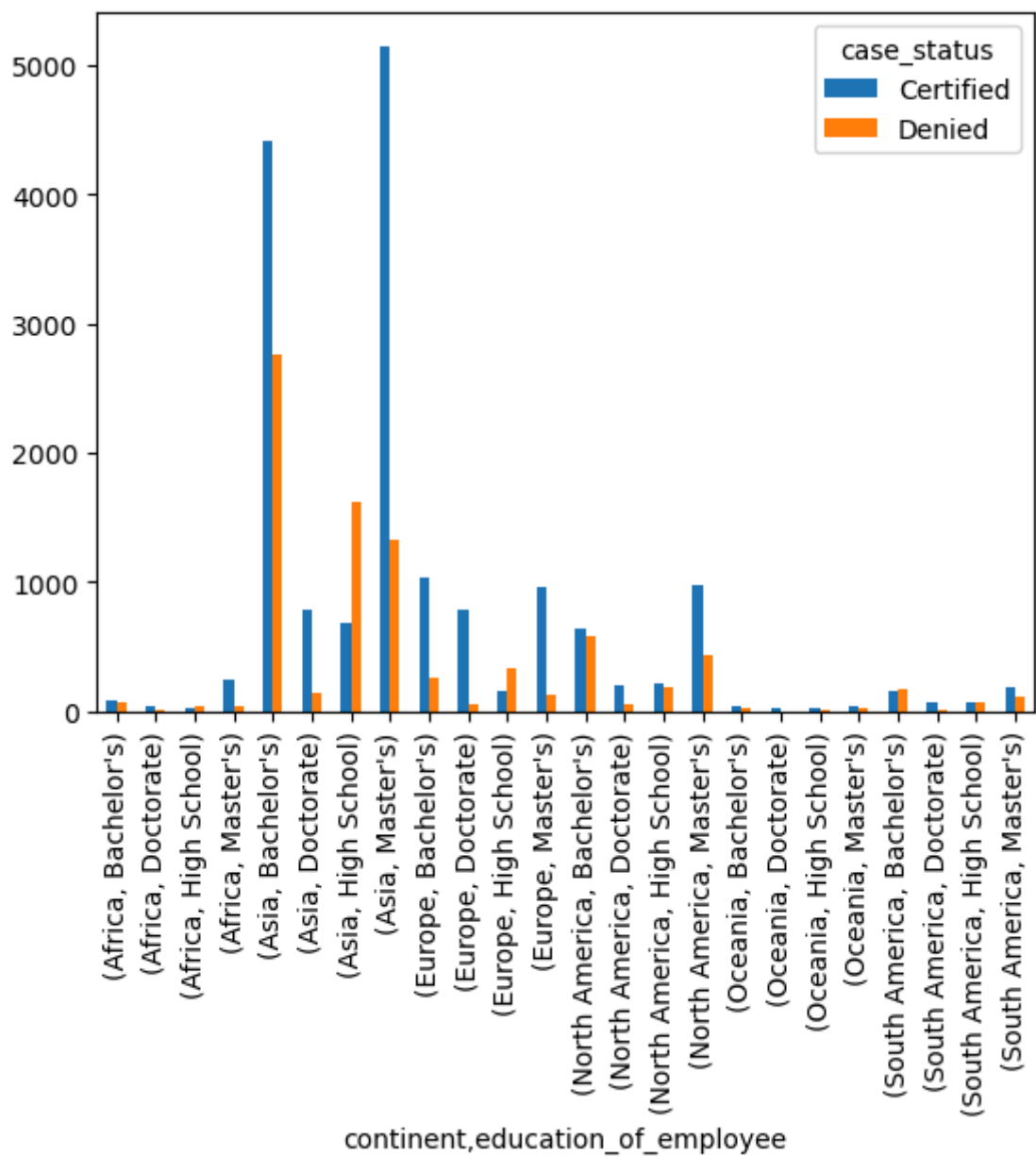
```
In [27]: result1.plot(kind='bar')
```

```
Out[27]: <Axes: xlabel='continent'>
```



```
In [28]: result2.plot(kind='bar')
```

```
Out[28]: <Axes: xlabel='continent,education_of_employee'>
```

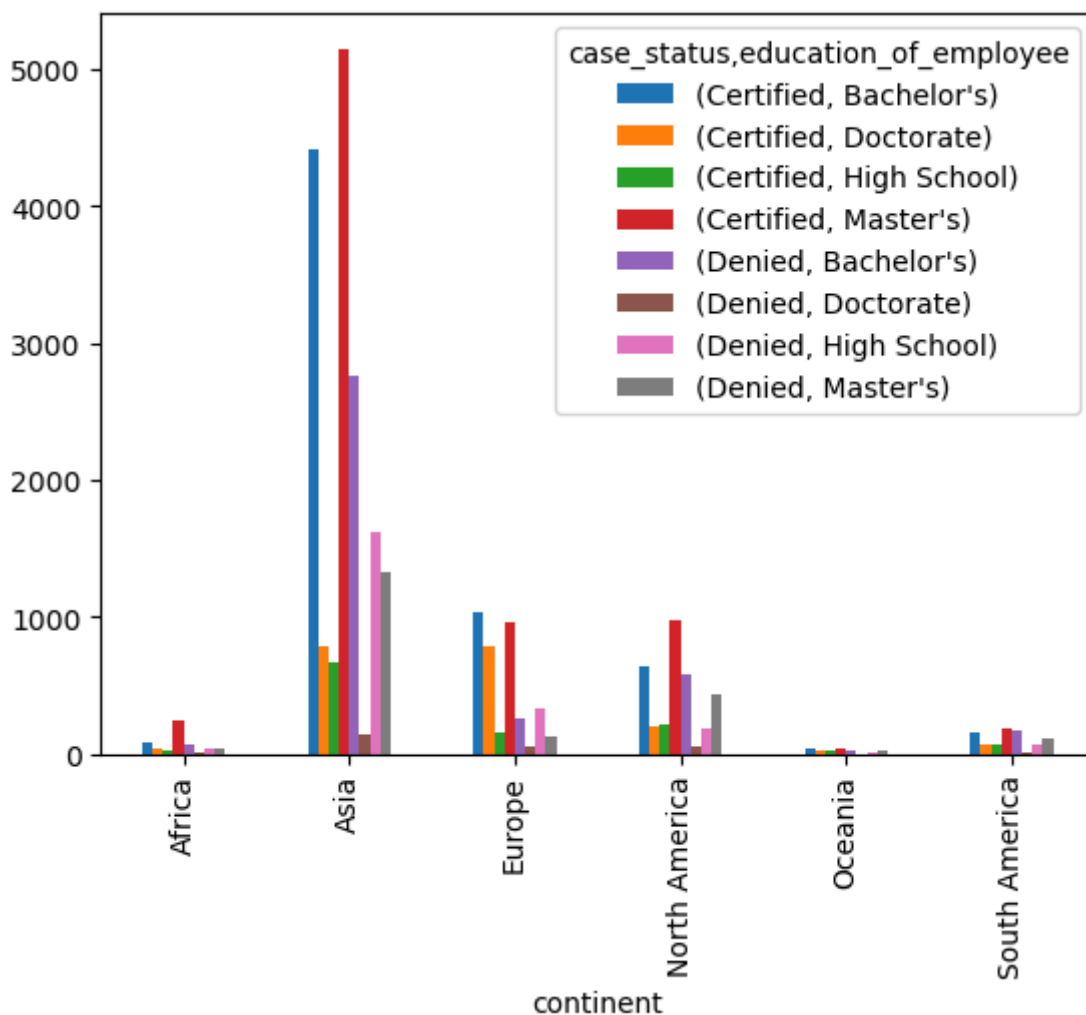


```
Out[30]: <Axes: xlabel='case_status'>
```



```
In [32]: 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')
```

Out[32]: <Axes: xlabel='continent'>



In [1]: *# Read the packages and data*

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

path=r"C:\Users\omkar\OneDrive\Documents\Data science\Naresh IT\Datafiles\V:
visa_df=pd.read_csv(path)
visa_df.head(3)
```

Out[1]:

	case_id	continent	education_of_employee	has_job_experience	requires_job_training	no_
0	EZYV01	Asia	High School	N	N	
1	EZYV02	Asia	Master's	Y	N	
2	EZYV03	Asia	Bachelor's	N	Y	

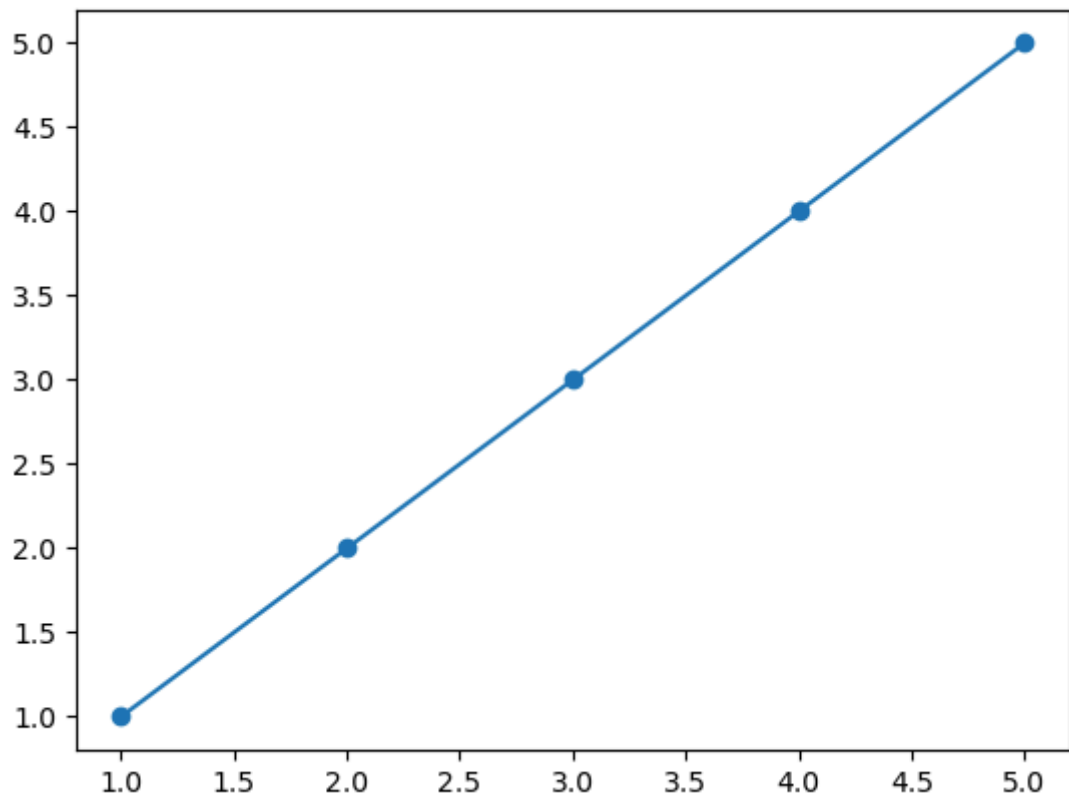


## *Numerical vs Numerical*

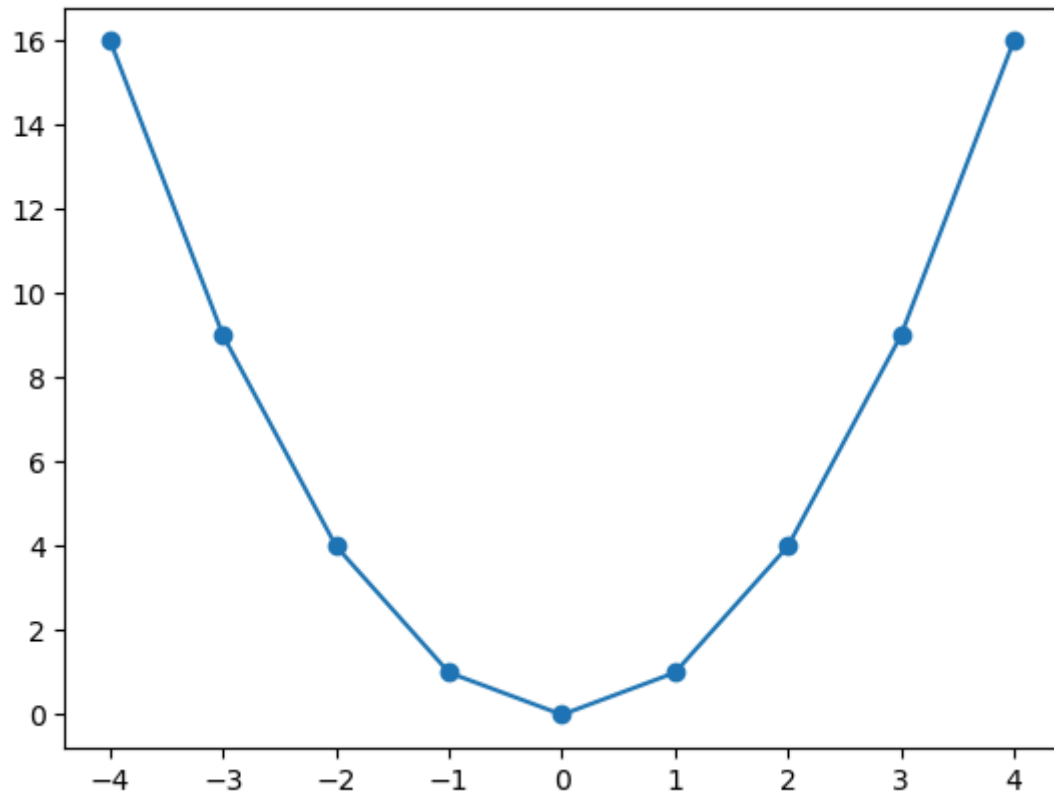
```
In [2]: x=[1,2,3,4,5]  
y=[1,2,3,4,5]  
# (1,1) (2,2) (3,3) (4,4) (5,5)
```

**plt.scatter**

```
In [5]: plt.scatter(x,y)  
plt.plot(x,y)  
plt.show()
```



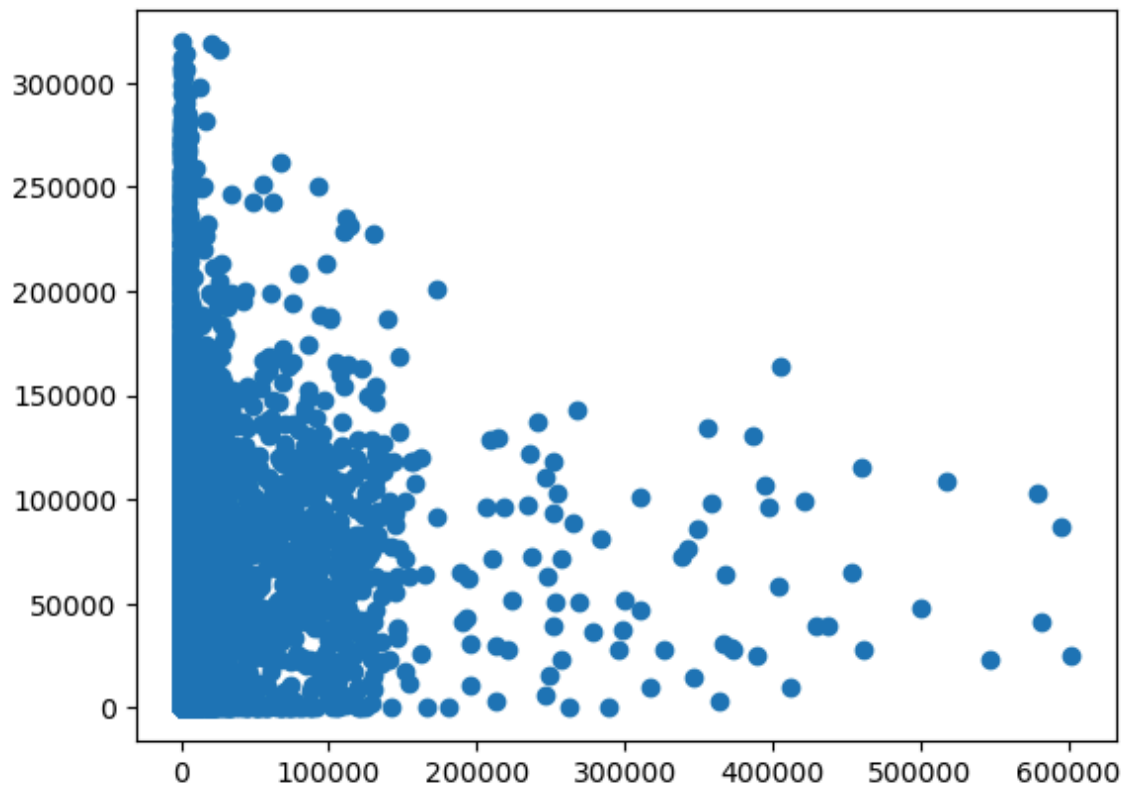
```
In [10]: x=[i for i in range(-4,5)]  
y=[i*i for i in x]  
plt.scatter(x,y)  
plt.plot(x,y)  
plt.show()
```



```
In [12]: # extract only numerical columns  
num_cols=visa_df.select_dtypes(exclude='object')  
num_cols.columns
```

```
Out[12]: Index(['no_of_employees', 'yr_of_estab', 'prevailing_wage'], dtype='object')
```

```
In [14]: col1=visa_df['no_of_employees']
col2=visa_df['prevailing_wage']
plt.scatter(col1,col2)
plt.show() # No relation
```



### Pearson Correlation Coefficient

- r varies from -1 to 1
- -1 to 0 : Negative relation
- 0 to 1: Postive relation
- 0: No relation

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

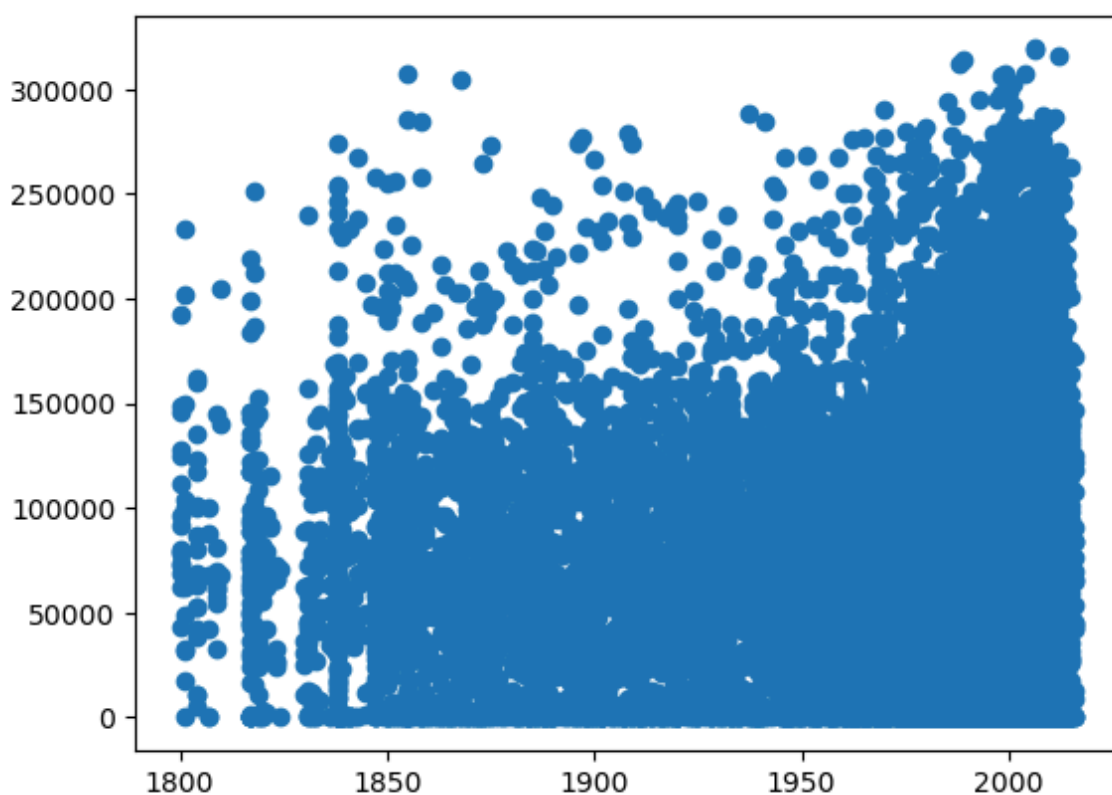
- when you do this python
- It gives the matrix
- in Visa data we have 3 numerical columns are there
- python will give a matrix w.r.t 3 numerical columns
- The values in each field tells about the relation between the variables

```
In [16]: visa_df.corr(numeric_only=True)
```

```
Out[16]:
```

	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 [18]: # 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 [21]: wine=pd.read_csv("C:\\Users\\omkar\\OneDrive\\Documents\\Data science\\Nares\\wine.csv")
wine.head()
```


```
Out[21]:
```

	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	
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	

In [22]: wine.corr()

Out[22]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	densi
<b>fixed acidity</b>	1.000000	-0.256131	0.671703	0.114777	0.093705	-0.153794	-0.113181	0.66804
<b>volatile acidity</b>	-0.256131	1.000000	-0.552496	0.001918	0.061298	-0.010504	0.076470	0.02202
<b>citric acid</b>	0.671703	-0.552496	1.000000	0.143577	0.203823	-0.060978	0.035533	0.36494
<b>residual sugar</b>	0.114777	0.001918	0.143577	1.000000	0.055610	0.187049	0.203028	0.35528
<b>chlorides</b>	0.093705	0.061298	0.203823	0.055610	1.000000	0.005562	0.047400	0.20063
<b>free sulfur dioxide</b>	-0.153794	-0.010504	-0.060978	0.187049	0.005562	1.000000	0.667666	-0.02194
<b>total sulfur dioxide</b>	-0.113181	0.076470	0.035533	0.203028	0.047400	0.667666	1.000000	0.07126
<b>density</b>	0.668047	0.022026	0.364947	0.355283	0.200632	-0.021946	0.071269	1.00000
<b>pH</b>	-0.682978	0.234937	-0.541904	-0.085652	-0.265026	0.070377	-0.066495	-0.34169
<b>sulphates</b>	0.183006	-0.260987	0.312770	0.005527	0.371260	0.051658	0.042947	0.14850
<b>alcohol</b>	-0.061668	-0.202288	0.109903	0.042075	-0.221141	-0.069408	-0.205654	-0.49618
<b>quality</b>	0.124052	-0.390558	0.226373	0.013732	-0.128907	-0.050656	-0.185100	-0.17497

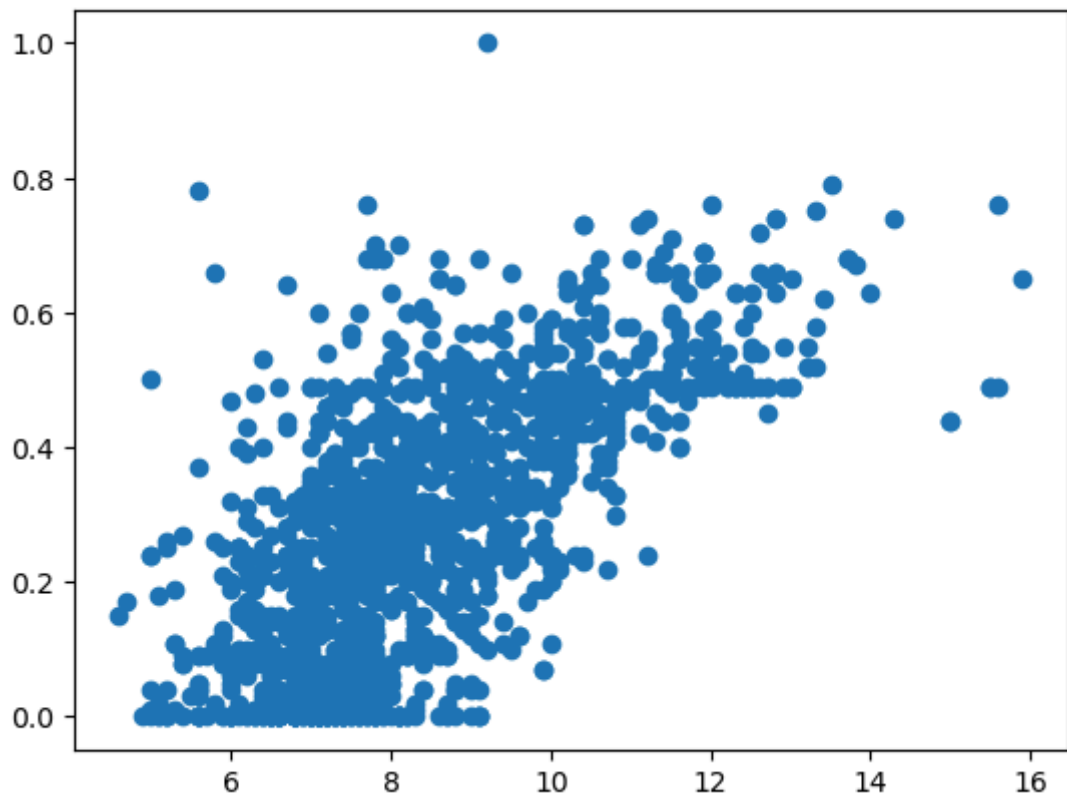


In [23]: wine.columns

Out[23]: Index(['fixed acidity', 'volatile acidity', 'citric acid', 'residual sugar',  
          'chlorides', 'free sulfur dioxide', 'total sulfur dioxide', 'density',  
          'pH', 'sulphates', 'alcohol', 'quality'],  
          dtype='object')

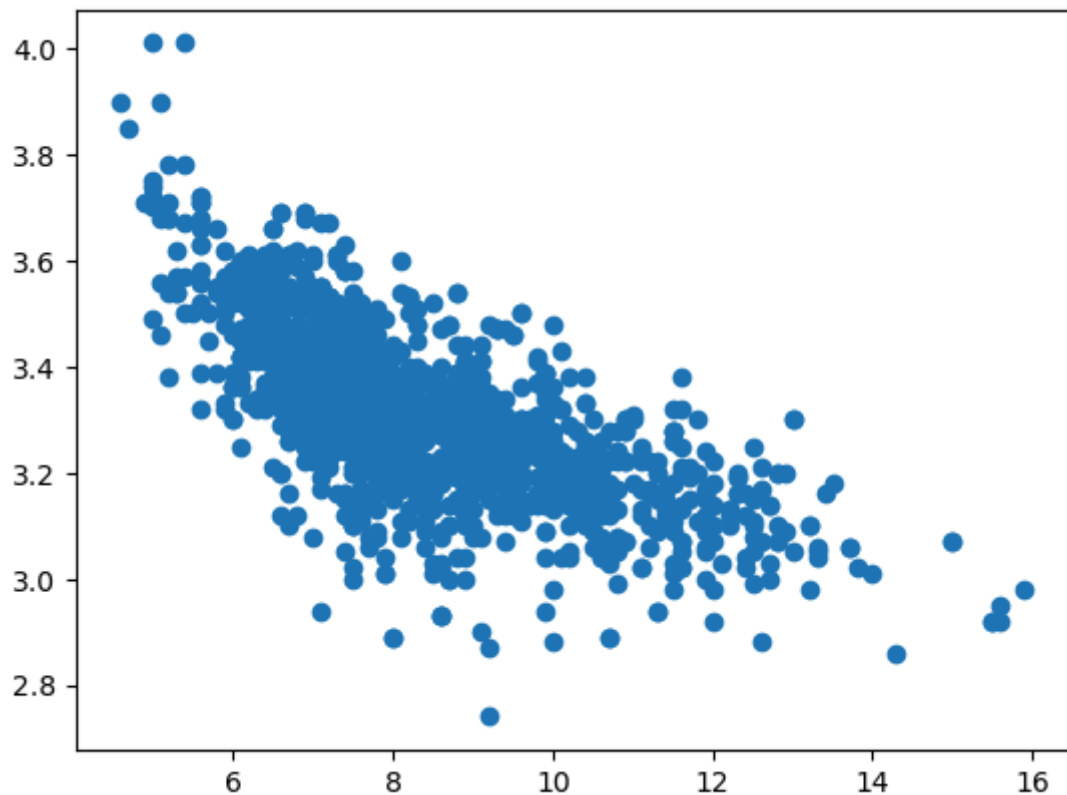
```
In [24]: # 'fixed acidity' and 'citric acid' : 0.67 +ve  
col1=wine['fixed acidity']  
col2=wine['citric acid']  
plt.scatter(col1,col2)
```

Out[24]: <matplotlib.collections.PathCollection at 0x1993b9d7250>



```
In [25]: # 'fixed acidity' and 'pH' : 0.68 -ve
col1=wine['fixed acidity']
col2=wine['pH']
plt.scatter(col1,col2)
```

Out[25]: <matplotlib.collections.PathCollection at 0x1993ba22a90>



### *heat-map*

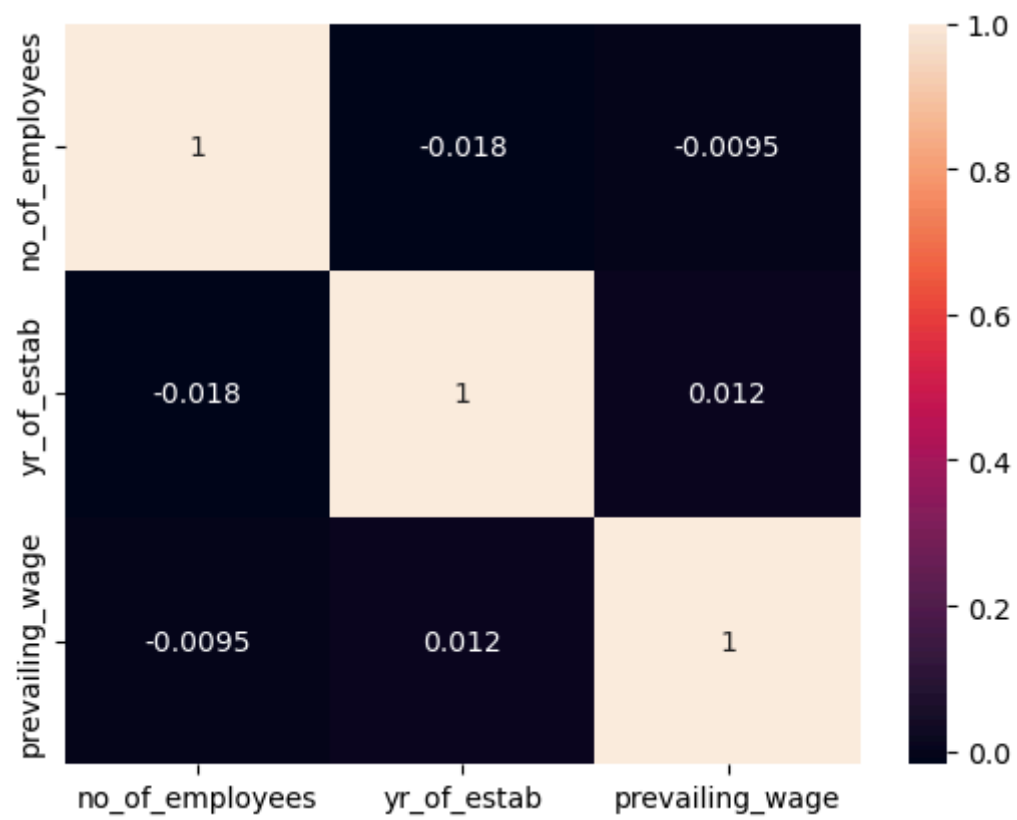
- heat map is useful to visulization of matrix
- it is under seaborn pacakges
- heat map will varies the values and gives the color about the values

```
In [28]: corr_visa=visa_df.corr(numeric_only=True)
corr_visa
# this is a matrix we want apply a heat map
```

Out[28]:

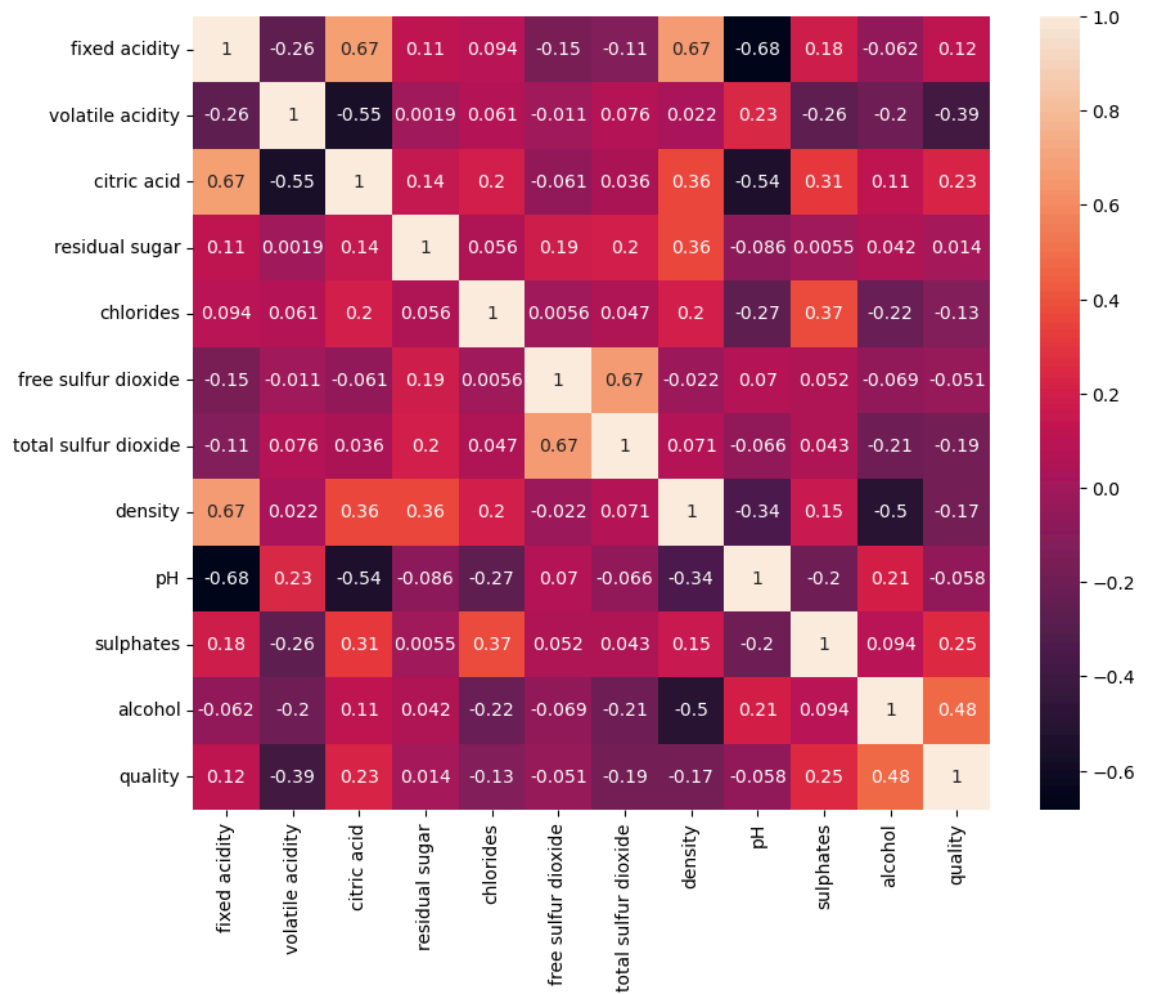
	no_of_employees	yr_of_estab	prevailing_wage
no_of_employees	1.000000	-0.017770	-0.009523
yr_of_estab	-0.017770	1.000000	0.012342
prevailing_wage	-0.009523	0.012342	1.000000

```
In [31]: sns.heatmap(corr_visa,  
                    annot=True)  
plt.show()
```





```
In [33]: corr_wine=wine.corr(numeric_only=True)
plt.figure(figsize=(10,8))
sns.heatmap(corr_wine,annot=True)
plt.show()
```



```
In [ ]:
```