### Import the packages

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

In [2]:
```
visa_df=pd.read_csv(r"C:\Users\omkar\OneDrive\Documents\Data science\Naresh IT\N
visa_df.head(2)
```

Out[2]:

|   | case_id | continent | education_of_employee | has_job_experience | requires_job_training |
|---|---|---|---|---|---|
| **0** | EZYV01 | Asia | High School | N | N |
| **1** | EZYV02 | Asia | Master's | Y | N |

### Bi variate analysis

- Analyse the two variables
- Two categorical columns
- Two numerical columns
- One categorical and one numerical

### Categorical vs Categorical

In [3]:
```
visa_df.columns
```

Out[3]:
```
Index(['case_id', 'continent', 'education_of_employee', 'has_job_experience',
       'requires_job_training', 'no_of_employees', 'yr_of_estab',
       'region_of_employment', 'prevailing_wage', 'unit_of_wage',
       'full_time_position', 'case_status'],
      dtype='object')
```

In [4]:
```
cat=visa_df.select_dtypes(include='object').columns
cat
```

Out[4]:
```
Index(['case_id', 'continent', 'education_of_employee', 'has_job_experience',
       'requires_job_training', 'region_of_employment', 'unit_of_wage',
       'full_time_position', 'case_status'],
      dtype='object')
```

In [5]:
```
num_cols=visa_df.select_dtypes(exclude='object').columns
num_cols
```

Out[5]:
```
Index(['no_of_employees', 'yr_of_estab', 'prevailing_wage'], dtype='object')
```

### continent-case_status

In [6]:
```
visa_df['continent'].value_counts()
```

```
Out[6]:  continent
         Asia             16861
         Europe            3732
         North America     3292
         South America      852
         Africa             551
         Oceania            192
         Name: count, dtype: int64
```

```
In [7]:  visa_df['case_status'].value_counts()
```

```
Out[7]:  case_status
         Certified    17018
         Denied        8462
         Name: count, dtype: int64
```

### How many asia people got the visa Certified

### How many asia people got the visa Denied

```
In [11]: con1=(visa_df['continent']=='Asia')
         con2=visa_df['case_status']=='Certified'
         con=con1&con2
         len(visa_df[con])
         print(f"the number of employees visa certified are: {len(visa_df[con])}")
```

```
the number of employees visa certified are: 11012
```

```
In [12]: con1=(visa_df['continent']=='Asia')
         con2=visa_df['case_status']=='Denied'
         con=con1&con2
         len(visa_df[con])
         print(f"the number of employees visa certified are: {len(visa_df[con])}")
```

```
the number of employees visa certified are: 5849
```

```
In [19]: keys=visa_df['continent'].unique()
         certified_list,denied_list=[],[]
         for i in keys:
             con1=(visa_df['continent']==i)
             con2=visa_df['case_status']=='Certified'
             con3=visa_df['case_status']=='Denied'
             certi_con=con1&con2
             denied_con=con1&con3
             certified_list.append(len(visa_df[certi_con]))
             denied_list.append(len(visa_df[denied_con]))

         pd.DataFrame(zip(certified_list,denied_list),
                     columns=['Certified','Denied'],
                     index=keys)
```

Out[19]:

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

### Cross tab

- cross tab will take two arguments
    - first argument: index
    - second argument: column

In [21]:

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

Out[21]:

| 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 [24]:

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

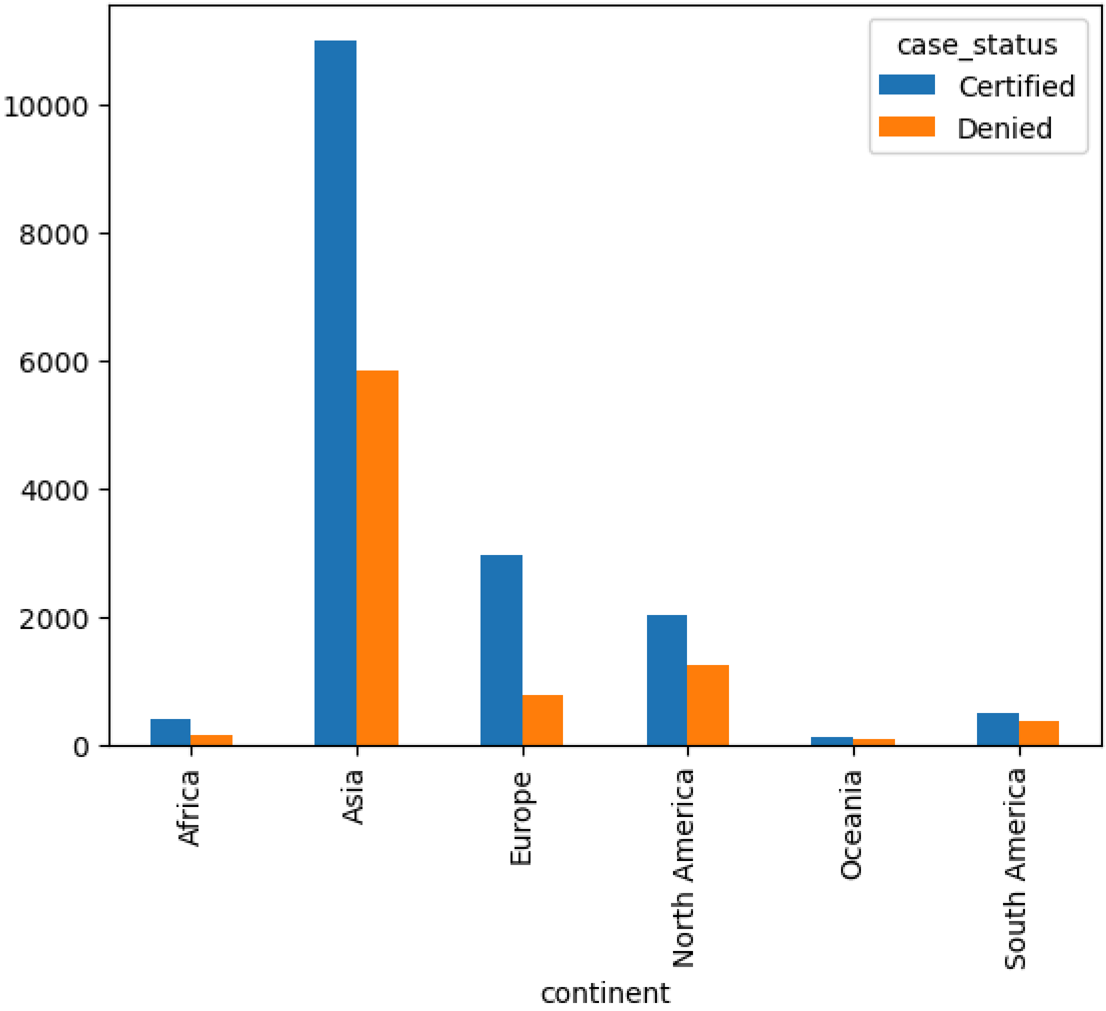
Out[24]:

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

In [25]:

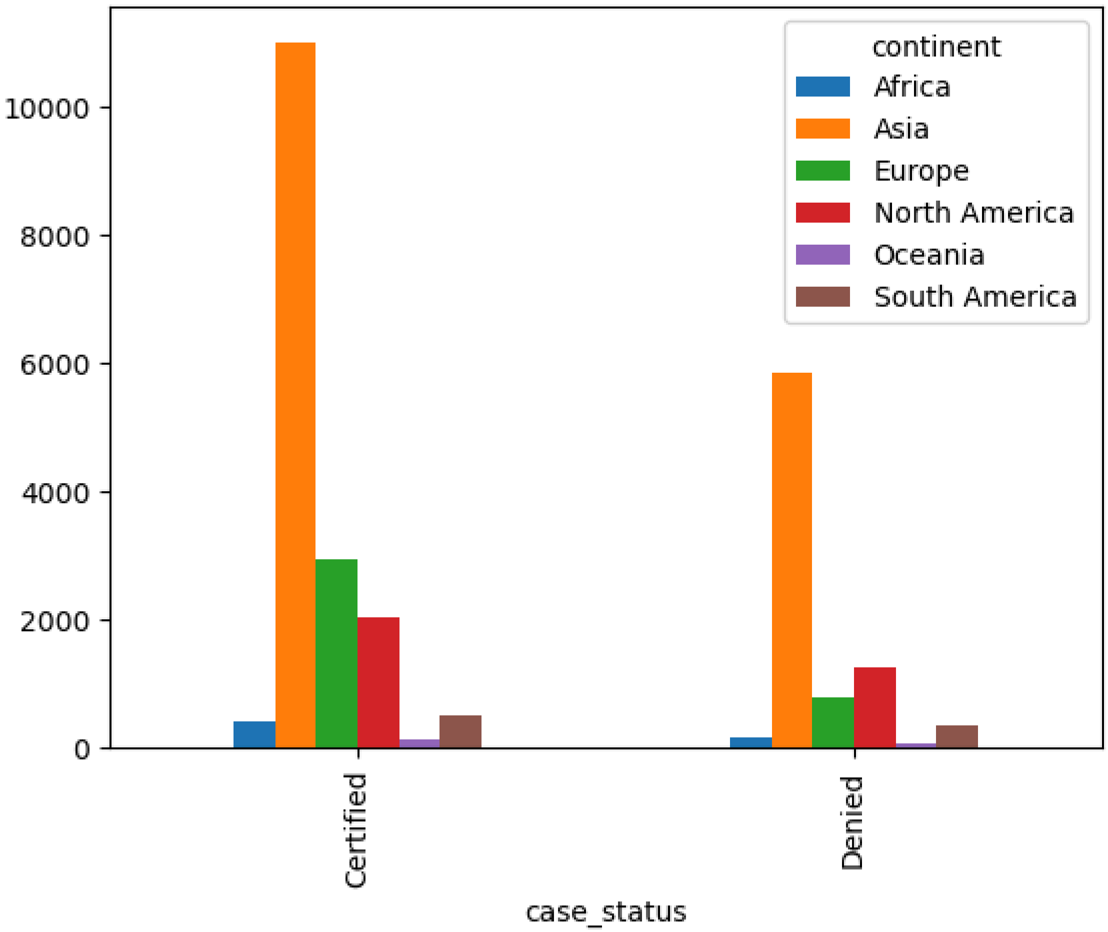
```
result1.plot(kind='bar')
```

```
Out[25]:  <Axes: xlabel='continent'>
```



```
In [26]:  result2.plot(kind='bar')
```

```
Out[26]:  <Axes: xlabel='case_status'>
```



In [28]:
```
col1=visa_df['continent']
col2=visa_df['education_of_employee']
col3=visa_df['case_status']
col=[col1,col2]
result3=pd.crosstab(col,col3)
result3
```

Out[28]:

| | 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 |

#### Numerical-Numerical

- In order to plot numerical vs numerical we need to use scatter plots
- Scatter plots will give the relation between two numerical columns
- It is under matplotlib

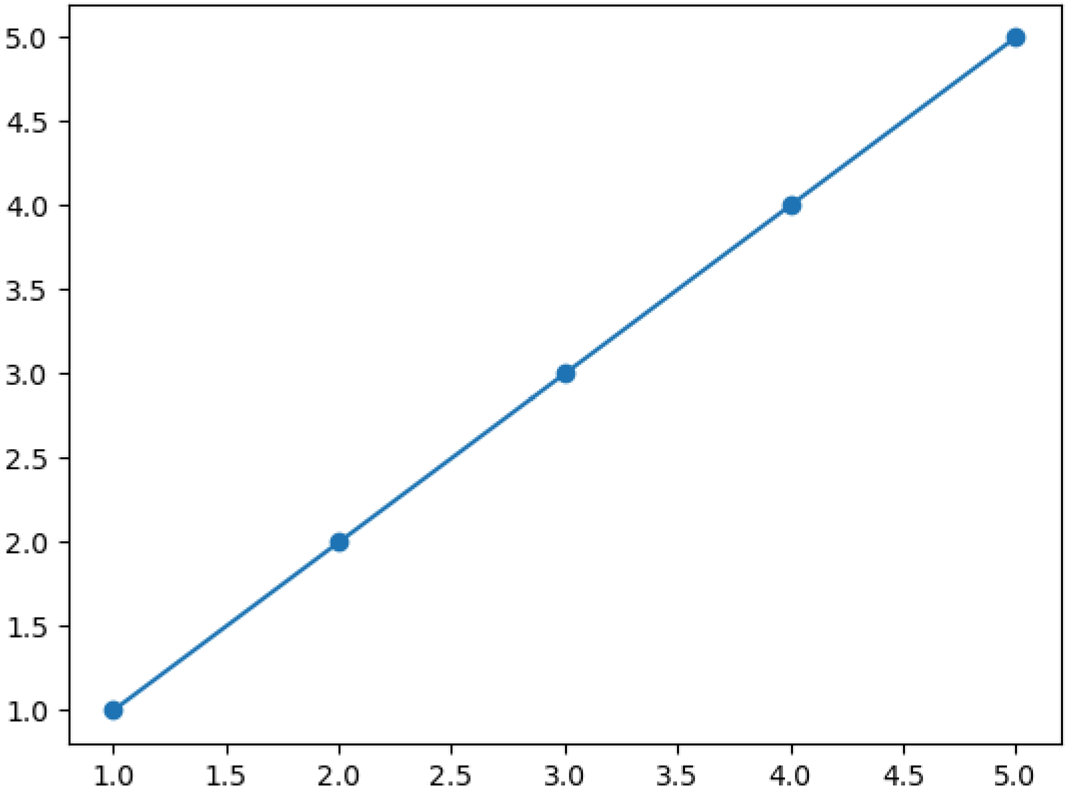
$plt.scatter$

In [ ]:

```
#y=x graph
```

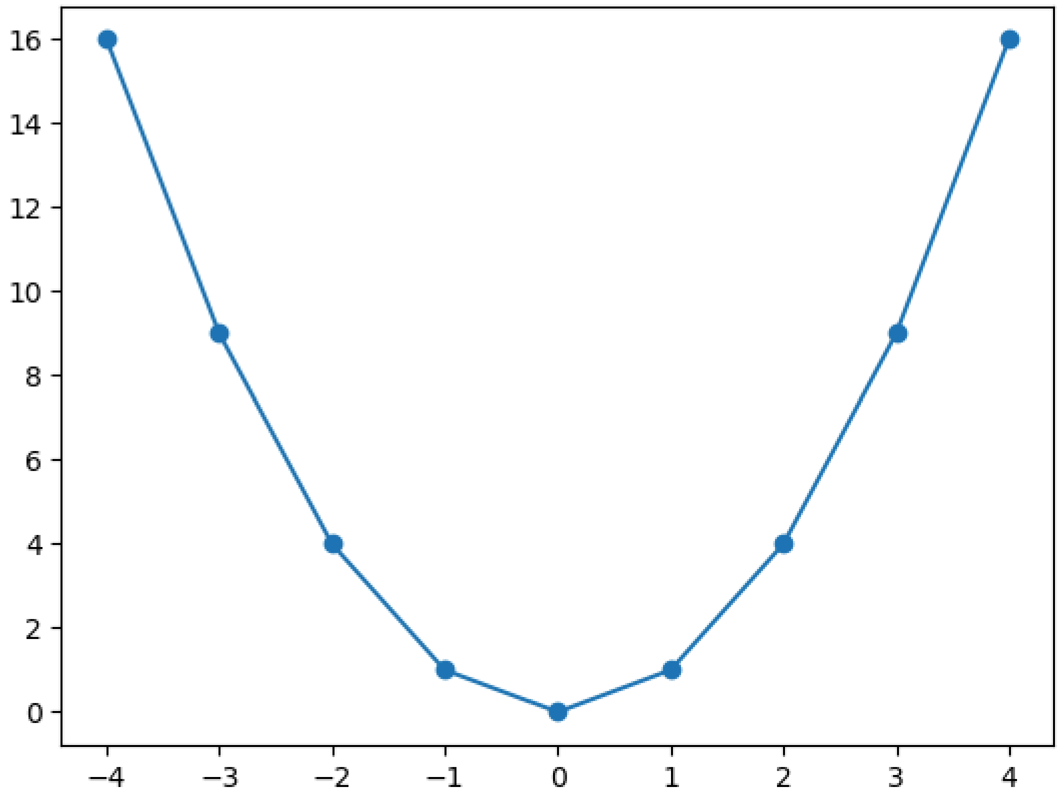
In [31]:

```
x=[1,2,3,4,5]
y= [1,2,3,4,5]
# it is y=x plot
plt.scatter(x,y)
plt.plot(x,y)
plt.show()
```



In [32]:

```
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 [33]:  num_cols
```

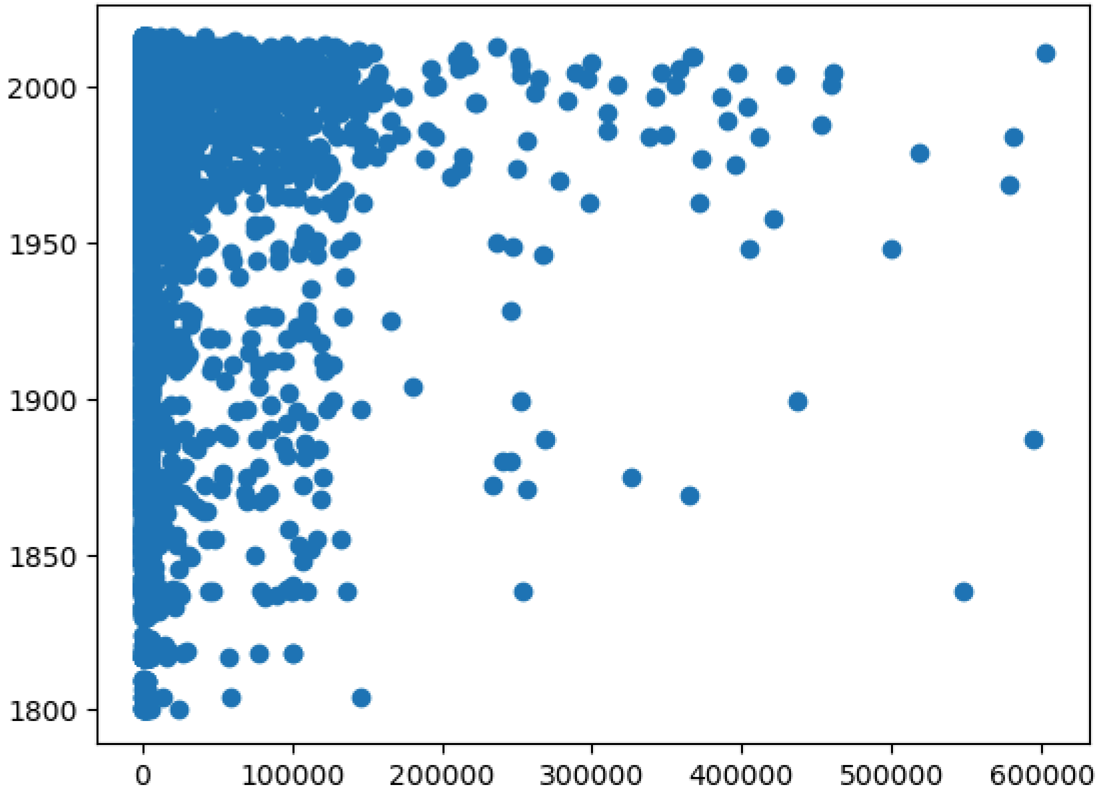
```
Out[33]:  Index(['no_of_employees', 'yr_of_estab', 'prevailing_wage'], dtype='object')
```

**Scatter plot-1**

- no_of_employees and yr_of_estab

```
In [34]:  col1=visa_df['no_of_employees']
          col2=visa_df['yr_of_estab']
          plt.scatter(col1,col2)
```

```
Out[34]:  <matplotlib.collections.PathCollection at 0x1bef3f24d50>
```



**Observation**: No relation

```
pearson correlation coeff=0
```

**Scatter plot-2**

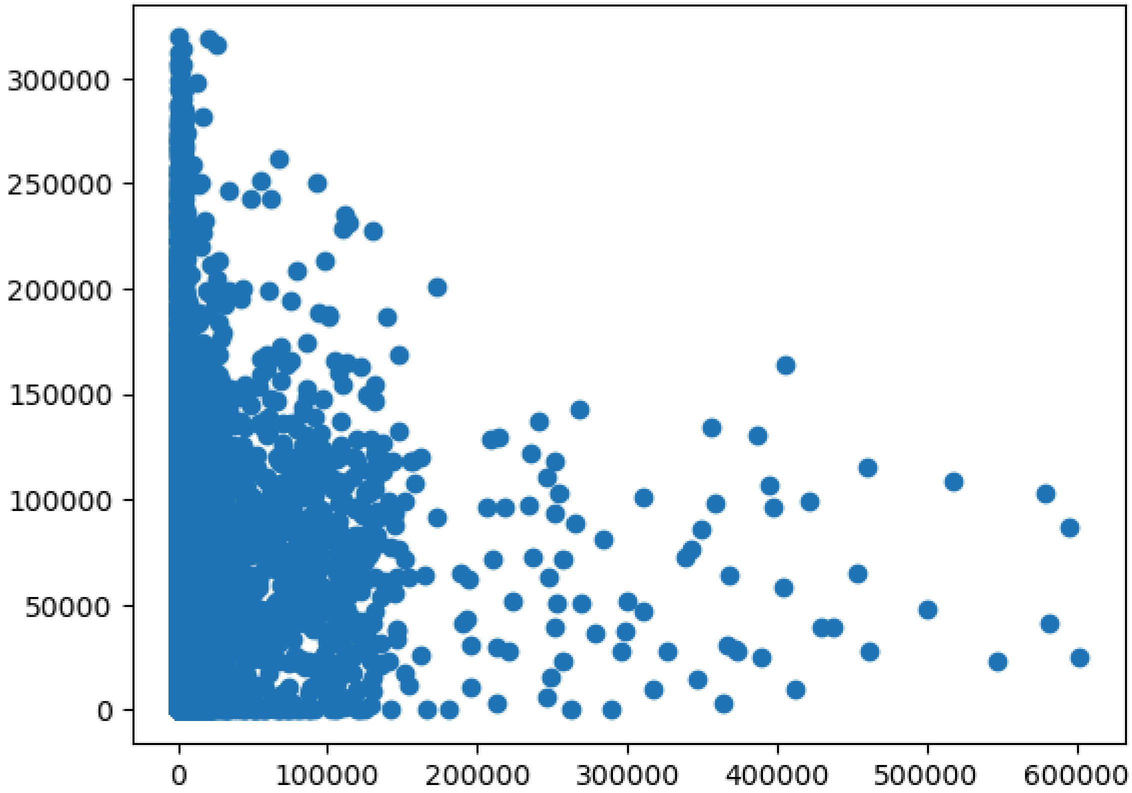
- no_of_employees and prevailing_wage

In [36]:

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

Out[36]:

```
<matplotlib.collections.PathCollection at 0x1bef3e8cd50>
```



**Observation**: No relation

```
pearson correlation coeff=0
```

**Scatter plot-2**

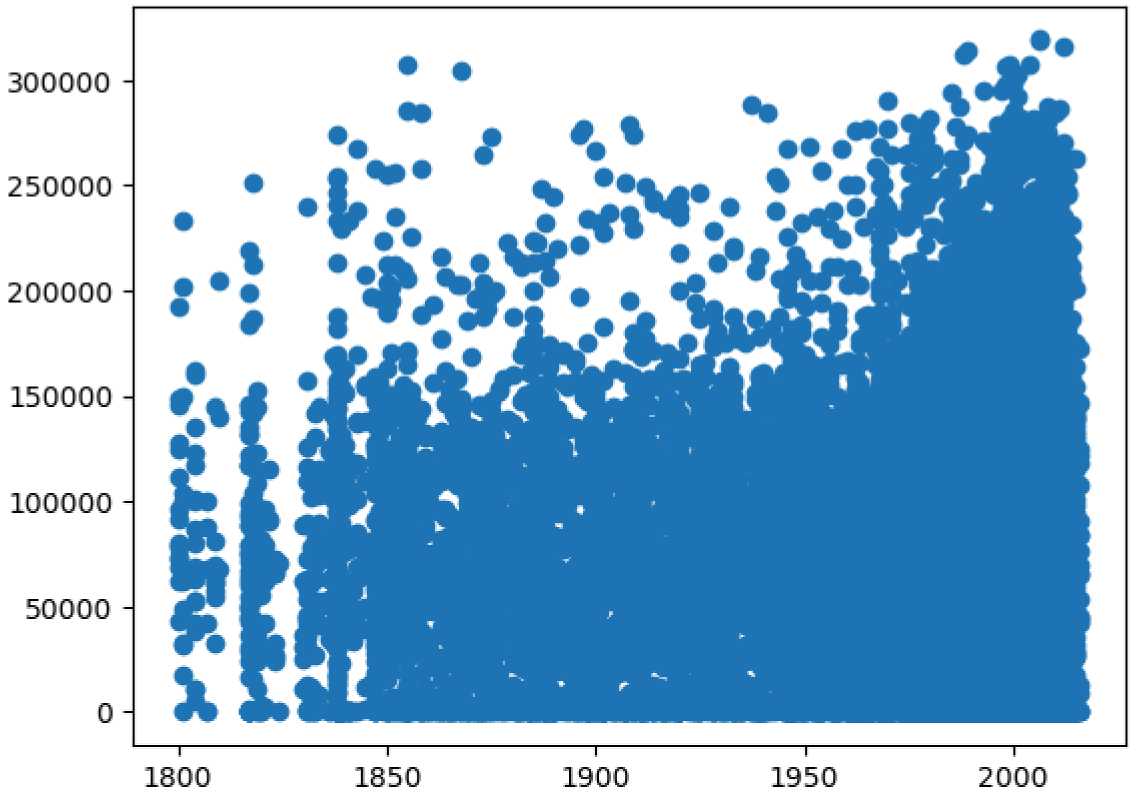
- yr_of_estab and prevailing_wage

In [37]:

```
col1=visa_df['yr_of_estab']
col2=visa_df['prevailing_wage']
plt.scatter(col1,col2)
```

Out[37]:

```
<matplotlib.collections.PathCollection at 0x1bef37c8a50>
```



**Observation**: No relation

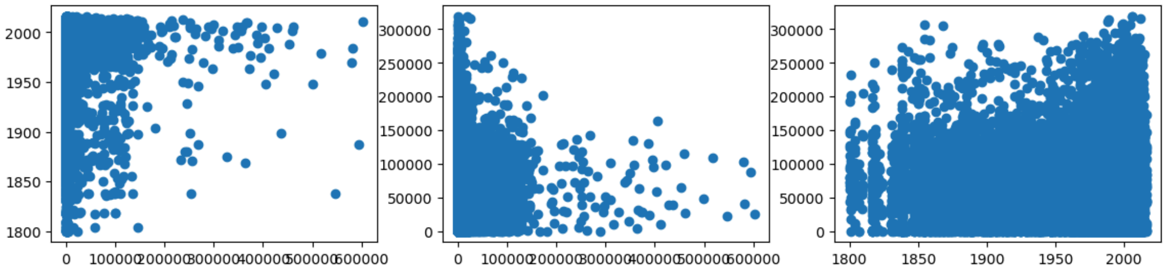
```
pearson correlation coeff=0
```

In [40]:

```
plt.figure(figsize=(14,3))

col1=visa_df['no_of_employees']
col2=visa_df['yr_of_estab']
plt.subplot(1,3,1).scatter(col1,col2)


col1=visa_df['no_of_employees']
col2=visa_df['prevailing_wage']
plt.subplot(1,3,2).scatter(col1,col2)

col1=visa_df['yr_of_estab']
col2=visa_df['prevailing_wage']
plt.subplot(1,3,3).scatter(col1,col2)
```

Out[40]:

```
<matplotlib.collections.PathCollection at 0x1bef51f6090>
```



**Correlation Coeffiecient**

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

- Perason correlation coeffiecient will give the amount of relation between variables
- It is denoted with r
- r varies from -1 to 1
- For postive relation r varies from 0 to 1
- For negative relation r varies from -1 to 0
- For no relation r approximate 0
- In python code we have **corr** function under pandas
  - This will give covariance matrix
  - we already discussed covariance matrix is rows and columns type
  - In this data we have 3 numerical columns
  - so we will get 3*3 9 values
  - All trace of the matrix represnts variance
  - Upper trainagle and lower traingle represents co variance

In [42]:
```
visa_df.corr(numeric_only=True)
```

Out[42]:

| | 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 |

**Observations**

- The correlation value between no_of_employees and yr_of_estb approximately zero
  - which indicates no relation
- The correlation value between no_of_employees and prevailing_wage approximately zero
  - which indicates no relation
- The correlation value between yr_of_estb and prevailing_wage approximately zero

        - which indicates no relation

In [ ]:

```
```