In [10]:
```
#1) Create a data frame with null values and fill
  #the null values by python.
```

In [12]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

names=['avinash','suresh',np.nan,'ramesh']
age=[20,23,24,null]
city=[np.nan,'banglore','delhi']
marks=[85,90,95,np.nan]
fathers_name=['mr.k',np.nan,'mr.l','mr.p']

cols=['names','age','city','marks','fathers_name']
index=['A','B','C','D','E']

pd.DataFrame=(zip(names,age,city,marks,Fathers_name),
              columns=cols,
              index=index)
```

```
  Cell In[12], line 17
    index=index)
    ^
SyntaxError: invalid syntax. Maybe you meant '==' or ':=' instead of '='?
```

In [23]:
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

names=['avinash','suresh',np.nan,'ramesh']
age=[20,23,24,np.nan]
city=[np.nan,'banglore','delhi','calcutta']
marks=[85,90,95,np.nan]
fathers_name=['mr.k',np.nan,'mr.l','mr.p']

cols=['names','age','city','marks','fathers_name']
index=['A','B','C','D']

data1=pd.DataFrame(zip(names,age,city,marks,fathers_name),
              columns=cols,
              index=index)
data1
```

Out[23]:

	names	age	city	marks	fathers_name
A	avinash	20.0	NaN	85.0	mr.k
B	suresh	23.0	banglore	90.0	NaN
C	NaN	24.0	delhi	95.0	mr.l
D	ramesh	NaN	calcutta	NaN	mr.p

In [24]:
```python
data1.fillna('python')
```

Out[24]:

	names	age	city	marks	fathers_name
A	avinash	20.0	python	85.0	mr.k
B	suresh	23.0	banglore	90.0	python
C	python	24.0	delhi	95.0	mr.l
D	ramesh	python	calcutta	python	mr.p

In [ ]:
```python
#2) Explain the types of sampling in statistics-theory.
```

In [ ]:
```python
#3) Create a one data frame with outliers and handling the
#outliers by different
   #methods-python.
```

In [48]:
```python
import pandas as pd
import numpy as np


Q1=np.percentile(visa_df['value'],25)
Q2=np.percentile(visa_df['value'],50)
Q3=np.percentile(visa_df['value'],75)

IQR=Q3-Q1
c1=visa_df['continent']<lb
c2=visa_df['continent']>ub
con=c1|c2

c1=visa_df['continent']>lb
c2=visa_df['continent']<ub
con=c1&c2
non_outliers_df=visa_df[c1&c2]
non_outliers_df
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
File C:\anaconda\Lib\site-packages\pandas\core\indexes\base.py:3653, in Index.get_loc(self, key)
   3652 try:
-> 3653     return self._engine.get_loc(casted_key)
   3654 except KeyError as err:

File C:\anaconda\Lib\site-packages\pandas\_libs\index.pyx:147, in pandas._libs.index.IndexEngine.get_loc()

File C:\anaconda\Lib\site-packages\pandas\_libs\index.pyx:176, in pandas._libs.index.IndexEngine.get_loc()

File pandas\_libs\hashtable_class_helper.pxi:7080, in pandas._libs.hashtable.PyObjectHashTable.get_item()

File pandas\_libs\hashtable_class_helper.pxi:7088, in pandas._libs.hashtable.PyObjectHashTable.get_item()

KeyError: 'value'

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call last)
Cell In[48], line 9
      2 import numpy as np
      4 #data={'no':[1,2,3,4],
      5  #     'value':[10,20,30,40]}
      6 #df=pd.DataFrame(data)
      7 #df
----> 9 Q1=np.percentile(visa_df['value'],25)
     10 Q2=np.percentile(visa_df['value'],50)
     11 Q3=np.percentile(visa_df['value'],75)

File C:\anaconda\Lib\site-packages\pandas\core\frame.py:3761, in DataFrame.__getitem__(self, key)
   3759 if self.columns.nlevels > 1:
   3760     return self._getitem_multilevel(key)
-> 3761 indexer = self.columns.get_loc(key)
   3762 if is_integer(indexer):
   3763     indexer = [indexer]

File C:\anaconda\Lib\site-packages\pandas\core\indexes\base.py:3655, in Index.get_loc(self, key)
   3653     return self._engine.get_loc(casted_key)
   3654 except KeyError as err:
-> 3655     raise KeyError(key) from err
   3656 except TypeError:
   3657     # If we have a listlike key, _check_indexing_error will raise
   3658     #  InvalidIndexError. Otherwise we fall through and re-raise
   3659     #  the TypeError.
   3660     self._check_indexing_error(key)

KeyError: 'value'
```

In [ ]:

```

```

In [ ]:

```
#ans4)Explain central limit theorem by random data-python.
```

```
Central Limit Theorem:
The Central Limit Theorem (CLT) is a fundamental theorem
in statistics. If the data does not follow normality, then we have to
divide the data into
samples and calculate the sample means, then the distribution of
these follow normality.
• Example:
• It is ideally shown that dividing into 30samples and calculate 30
sample means
• If we take n=10 it is nearer to sample distribution
```

In [ ]:

```
#5) Plot the Bar plot, histogram plot, heat map, scatter plot,
#pie chart for the
#  given data set. -python.
```

In [25]:

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

path=r"C:\Users\tanma\DATASCIENCE\data\Visadataset.xlsx"
visa_df=pd.read_excel(path)
visa_df
```

Out[25]:

has_job_experience	requires_job_training	no_of_employees	yr_of_estab	region_of_employme
N	N	14513	2007	W
Y	N	2412	2002	Northea
N	Y	44444	2008	W
N	N	98	1897	W
Y	N	1082	2005	Sou
...	...	...	...	
Y	Y	2601	2008	Sou
Y	N	3274	2006	Northea
Y	N	1121	1910	Sou
Y	Y	1918	1887	W
Y	N	3195	1960	Midw

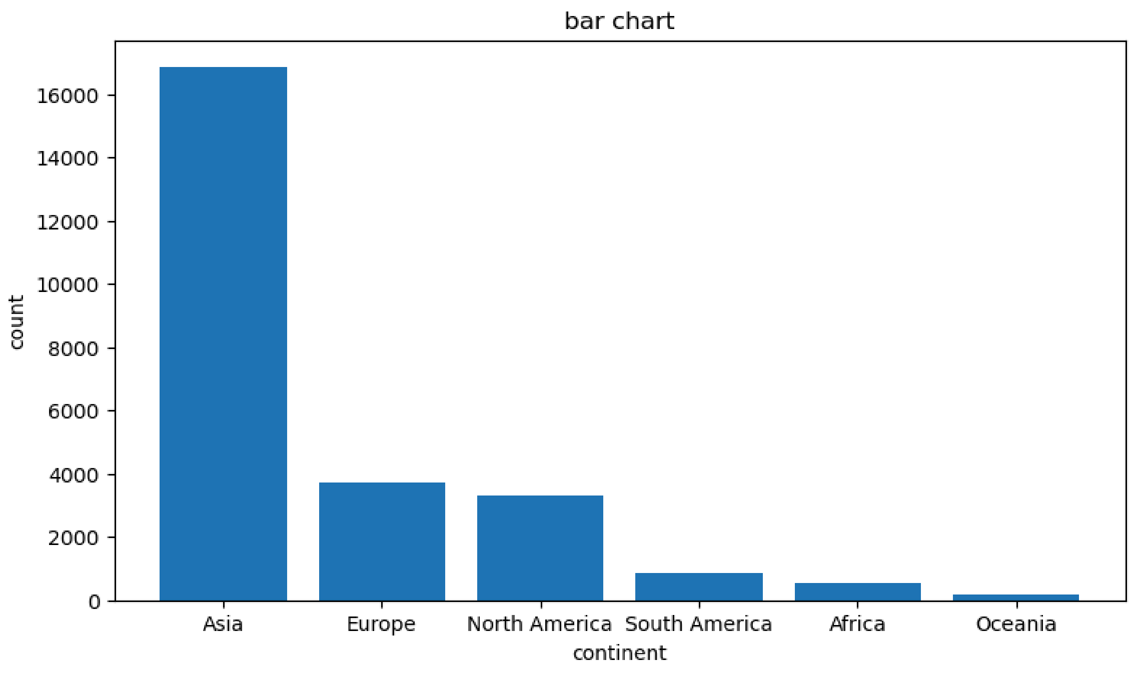
In [26]:
```
continent_vc=visa_df['continent'].value_counts()
l1=continent_vc.keys()
l2=continent_vc.values
continent_vc_df=pd.DataFrame(zip(l1,l2),
                             columns=['continent','count'])
continent_vc_df
```

Out[26]:

	continent	count
0	Asia	16861
1	Europe	3732
2	North America	3292
3	South America	852
4	Africa	551
5	Oceania	192

bar chart

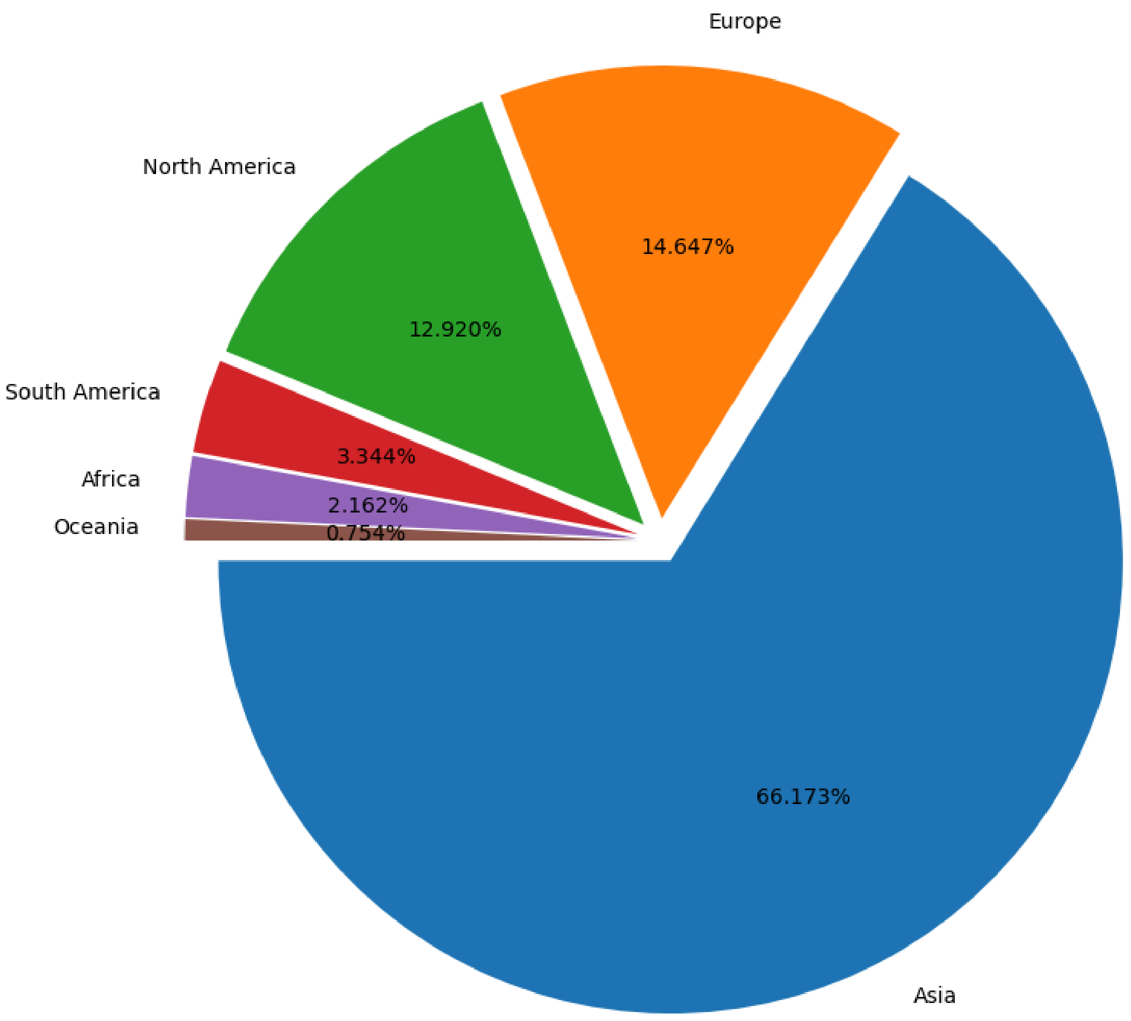
In [28]:
```
plt.figure(figsize=(9,5))
plt.bar('continent','count',data=continent_vc_df)
plt.xlabel('continent')
plt.ylabel('count')
plt.title('bar chart')
plt.show()
```



In [34]:
```
keys=visa_df['continent'].value_counts().keys()
values=visa_df['continent'].value_counts().values
values
```

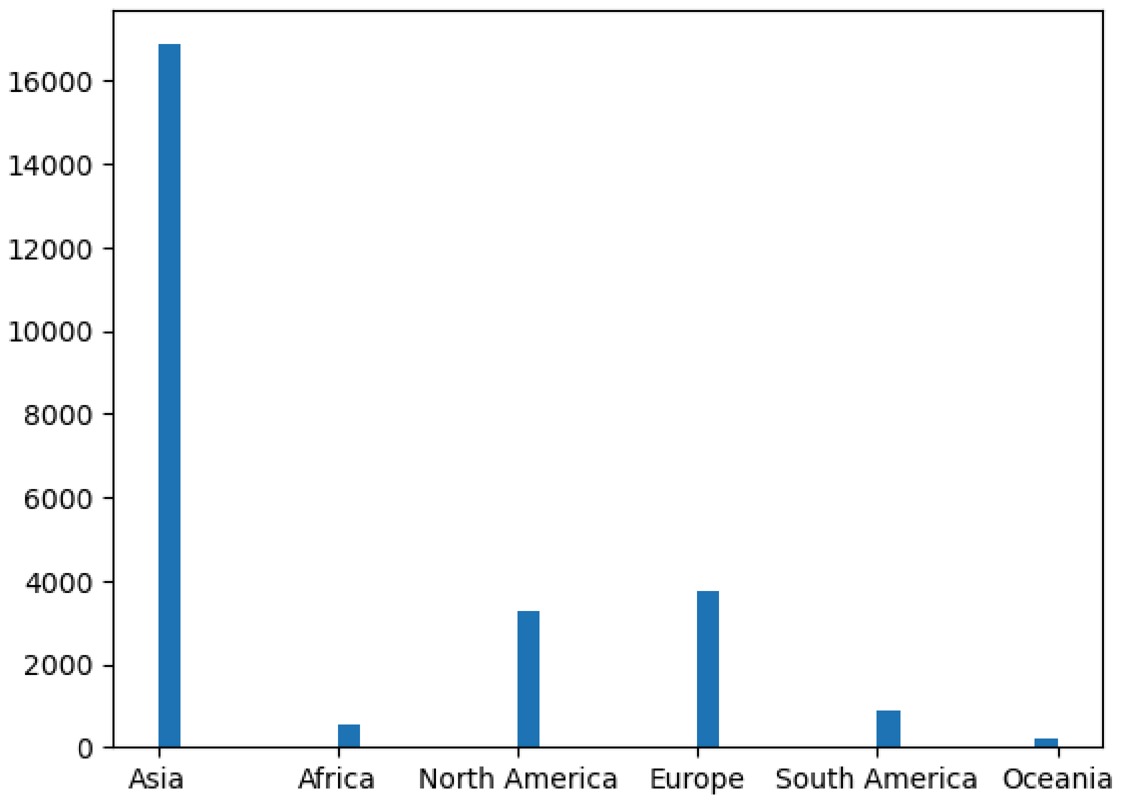
Out[34]:
```
array([16861,  3732,  3292,   852,   551,   192], dtype=int64)
```

In [38]:
```
plt.pie(values,labels=keys,
        autopct="%0.3f%%",
        explode=[0.1,0.1,0.1,0.1,0.1,0.1],
        startangle=180,
        radius=2)
plt.show()
```



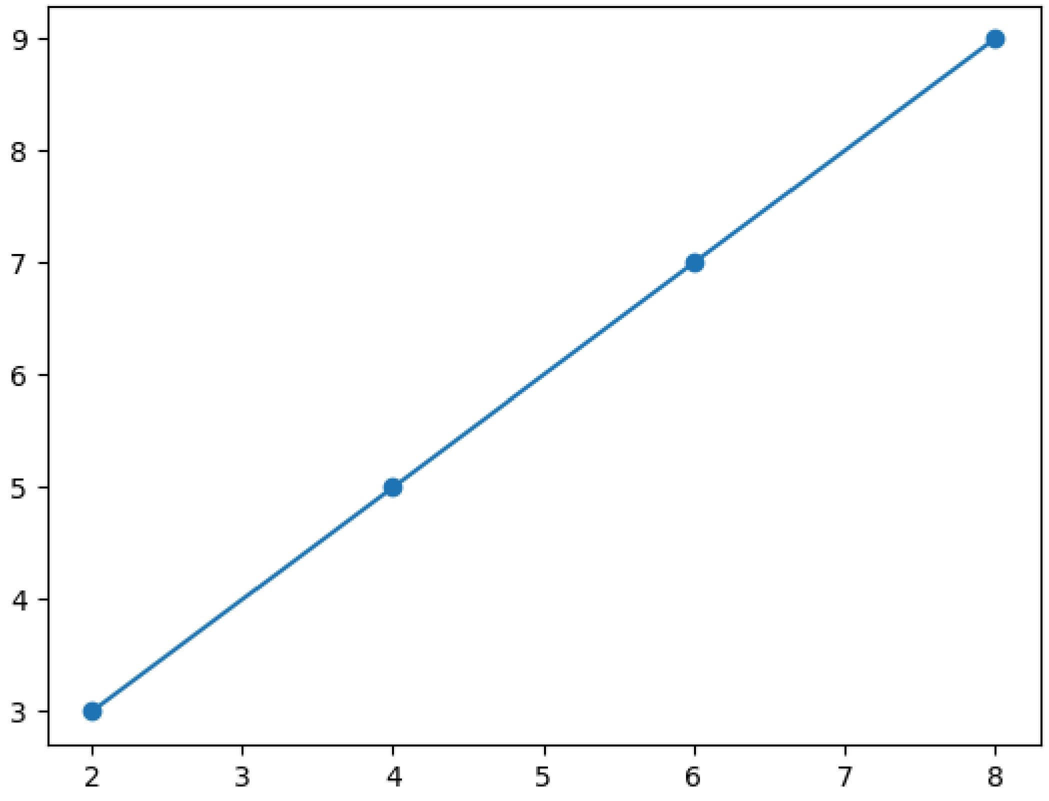
histogram

```
In [39]: f,i,n=plt.hist(visa_df['continent'],bins=40)
```



scatter plot

```
In [41]: x=[2,4,6,8]
         y=[3,5,7,9]
         plt.scatter(x,y)
         plt.plot(x,y)
         plt.show()
```



```
In [ ]:

In [ ]:

In [ ]:

In [ ]:
```