In [1]:

```python
# Import the required packages
# And read the data
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

file_location="C:\\Users\\omkar\\OneDrive\\Documents\\Data science\\Naresh I
visa_df=pd.read_csv(file_location)
visa_df.head()
```

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 | |
| **3** | EZYV04 | Asia | Bachelor's | N | N | |
| **4** | EZYV05 | Africa | Master's | Y | N | |

In [2]:

```python
visa_df.dtypes
```

Out[2]:

```
case_id                   object
continent                 object
education_of_employee     object
has_job_experience        object
requires_job_training     object
no_of_employees            int64
yr_of_estab                int64
region_of_employment      object
prevailing_wage          float64
unit_of_wage              object
full_time_position        object
case_status               object
dtype: object
```

In [3]:

```python
visa_df.columns
```

Out[3]:

```
Index(['case_id', 'continent', 'education_of_employee', 'has_job_experienc
e',
       '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]:
```
visa_df['continent']
```

Out[4]:
```
0          Asia
1          Asia
2          Asia
3          Asia
4        Africa
          ...  
25475      Asia
25476      Asia
25477      Asia
25478      Asia
25479      Asia
Name: continent, Length: 25480, dtype: object
```

In [5]:
```
type(visa_df['continent'])
```

Out[5]:
```
pandas.core.series.Series
```

In [6]:
```
col=['continent']
visa_df[col]
```

Out[6]:

| | continent |
|---|---|
| **0** | Asia |
| **1** | Asia |
| **2** | Asia |
| **3** | Asia |
| **4** | Africa |
| **...** | ... |
| **25475** | Asia |
| **25476** | Asia |
| **25477** | Asia |
| **25478** | Asia |
| **25479** | Asia |

25480 rows × 1 columns

In [7]:

```python
visa_df['continent'] # series
visa_df[['continent']] # dataframe
```

Out[7]:

| | continent |
|---|---|
| **0** | Asia |
| **1** | Asia |
| **2** | Asia |
| **3** | Asia |
| **4** | Africa |
| **...** | ... |
| **25475** | Asia |
| **25476** | Asia |
| **25477** | Asia |
| **25478** | Asia |
| **25479** | Asia |

25480 rows × 1 columns

In [8]:

```python
list(visa_df['continent'].values)
```

Out[8]:

```
['Asia',
 'Asia',
 'Asia',
 'Asia',
 'Africa',
 'Asia',
 'Asia',
 'North America',
 'Asia',
 'Europe',
 'Asia',
 'Asia',
 'Asia',
 'Asia',
 'Asia',
 'Asia',
 'Europe',
 'Asia',
 'Africa',
 'Asia'
```

In [9]:

```python
visa_df[['continent']].values # Dont use this
```

Out[9]:

```
array([['Asia'],
       ['Asia'],
       ['Asia'],
       ...,
       ['Asia'],
       ['Asia'],
       ['Asia']], dtype=object)
```

```
In [10]: l1=[1,2,3,4]
         l2=[10,20,30,40]
         l1+l2
```

```
Out[10]: [1, 2, 3, 4, 10, 20, 30, 40]
```

```
In [11]: a1=np.array(l1)
         a2=np.array(l2)
         a1+a2
```

```
Out[11]: array([11, 22, 33, 44])
```

```
In [12]: # Read multiplle columns
         # continent
         # education_of_employee
         cols=['continent','education_of_employee']
         visa_df[cols].head()
```

Out[12]:

| | continent | education_of_employee |
|---|---|---|
| **0** | Asia | High School |
| **1** | Asia | Master's |
| **2** | Asia | Bachelor's |
| **3** | Asia | Bachelor's |
| **4** | Africa | Master's |

*unique*

```
In [13]: visa_df['continent'].unique()
         # There are almost 25k entries are there under contienent column
         # but some values reapeted as 25k
```

```
Out[13]: array(['Asia', 'Africa', 'North America', 'Europe', 'South America',
                'Oceania'], dtype=object)
```

*nunique*

```
In [14]: visa_df['continent'].nunique()
```

```
Out[14]: 6
```

```
In [15]: len(visa_df['continent'].unique())
```

```
Out[15]: 6
```

```
In [16]: visa_df['continent'].unique()       # array of entries
         visa_df['continent'].nunique()      # 6
         len(visa_df['continent'].unique())  # 6
```

```
Out[16]: 6
```

In [17]:

```
# How many Asia count is avaialable out of 25k entries
# How many Africa count is avaialable out of 25k entries
```

$Value - Counts$

In [18]:

```
visa_df['continent'].value_counts()
```

Out[18]:

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

In [19]:

```
type(visa_df['continent'].value_counts())
```

Out[19]:

```
pandas.core.series.Series
```

In [20]:

```
visa_df['continent'].value_counts(normalize=True)
```

Out[20]:

```
continent
Asia             0.661735
Europe           0.146468
North America    0.129199
South America    0.033438
Africa           0.021625
Oceania          0.007535
Name: proportion, dtype: float64
```

In [21]:

```
# Save in one variable
# convert into dictionary
# Sepaerate keys in one list and values in another list
# create a data frame

continent_values=dict(visa_df['continent'].value_counts())
keys=list(continent_values.keys())
values=list(continent_values.values())
continent_df=pd.DataFrame(zip(keys,values),
                          columns=['Continent','Count'])
continent_df.to_csv('continent_df.csv',index=False)
```

In [22]:

```
continent_values

d1={'Names':['Ramesh','Suresh','Sathish'],
   'Age':[20,25,30]}
pd.DataFrame(d1)
```

Out[22]:

| | Names | Age |
|---|---|---|
| **0** | Ramesh | 20 |
| **1** | Suresh | 25 |
| **2** | Sathish | 30 |

In [23]:
```
# ValueError: If using all scalar values, you must pass an index

continent_values=dict(visa_df['continent'].value_counts())
pd.DataFrame(continent_values,index=['Count'])

# In the dictionay if values in the form of list: then no need to provide i
# In the dict  if values are as single (scalar)  then provide index
```

Out[23]:

| | Asia | Europe | North America | South America | Africa | Oceania |
|---|---|---|---|---|---|---|
| **Count** | 16861 | 3732 | 3292 | 852 | 551 | 192 |

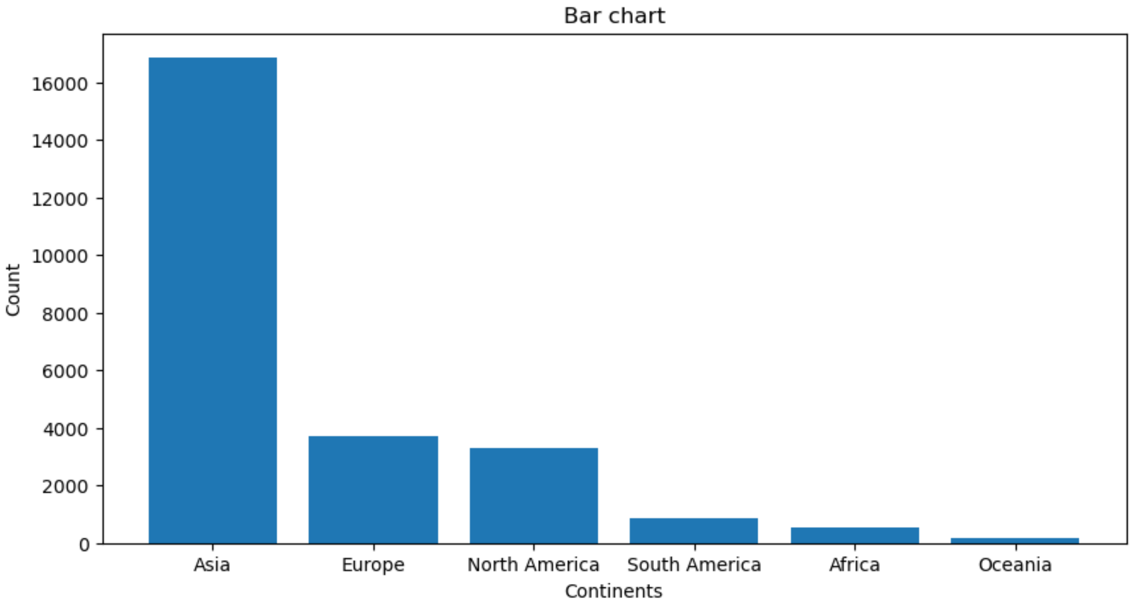
In [24]:
```
continent_df   # Frequency table
```

Out[24]:

| | 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 [25]:
```
# Bar chart requires  one categorical column , one numerical column
# Continent df output consider as a frequency table
# In that we have one cat column: Contienent (X-axis)
# Count is a numerical column: (Y-axis)
plt.figure(figsize=(10,5))
plt.bar('Continent','Count',data=continent_df)
plt.xlabel("Continents")
plt.ylabel("Count")
plt.title("Bar chart")
plt.savefig("Continent_bar_chart.jpg")
plt.show()
```



- We read Categorical Column
- Unique
- nunique
- Value counts
- From the value counts we created a dataframe (Frequency table)
- Bar chart

#### Bar chart require two things

- under Matplotlib
- X-axis (categorical column)
- y-axis (numerical column)

#### Count plot

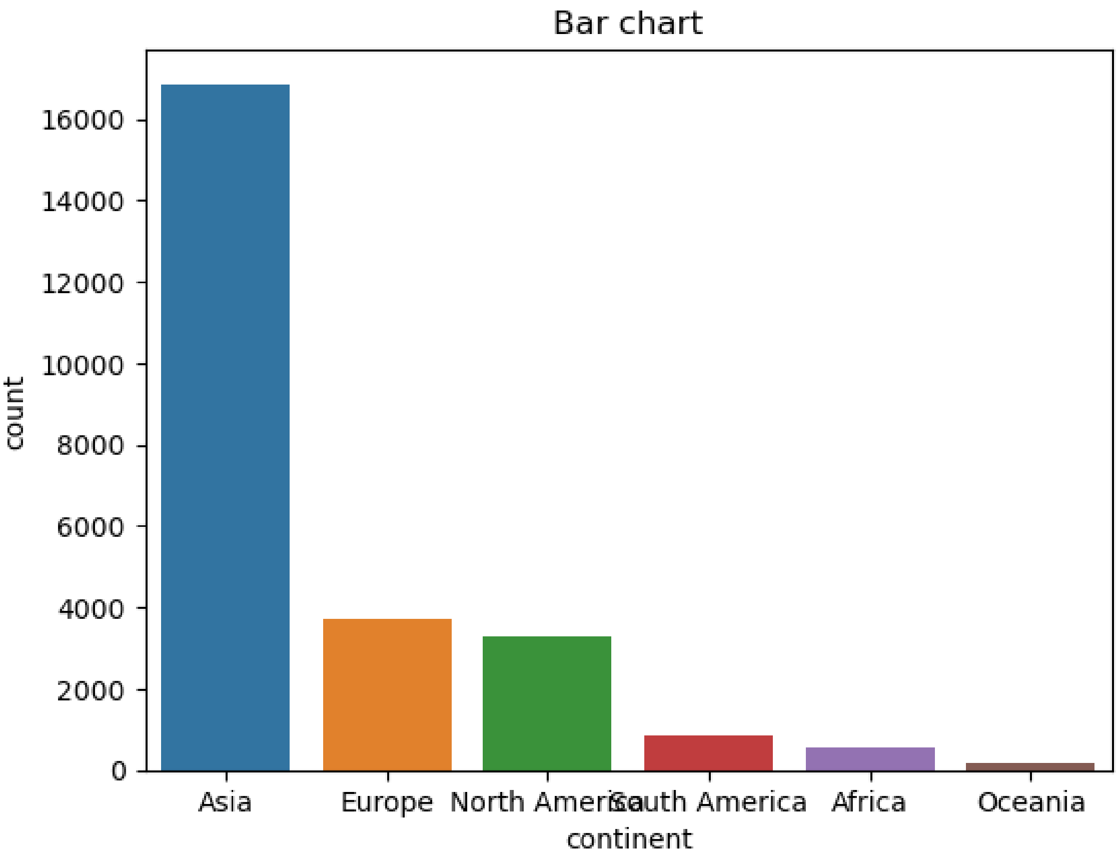
- Under seaborn package
- It requires directly column name from original dataframe

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

```
Out[26]: Index(['case_id', 'continent', 'education_of_employee', 'has_job_experienc
         e',
                '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 [33]:

```
order_lables=['Asia','Europe','North America','South America','Africa','Oce
sns.countplot(data=visa_df,
              x='continent',
             order=order_lables)
plt.title("Bar chart")
plt.show()
```



In [42]:

```
visa_df['continent'].value_counts().keys()
#visa_df['continent'].value_counts().to_list()
```

Out[42]:

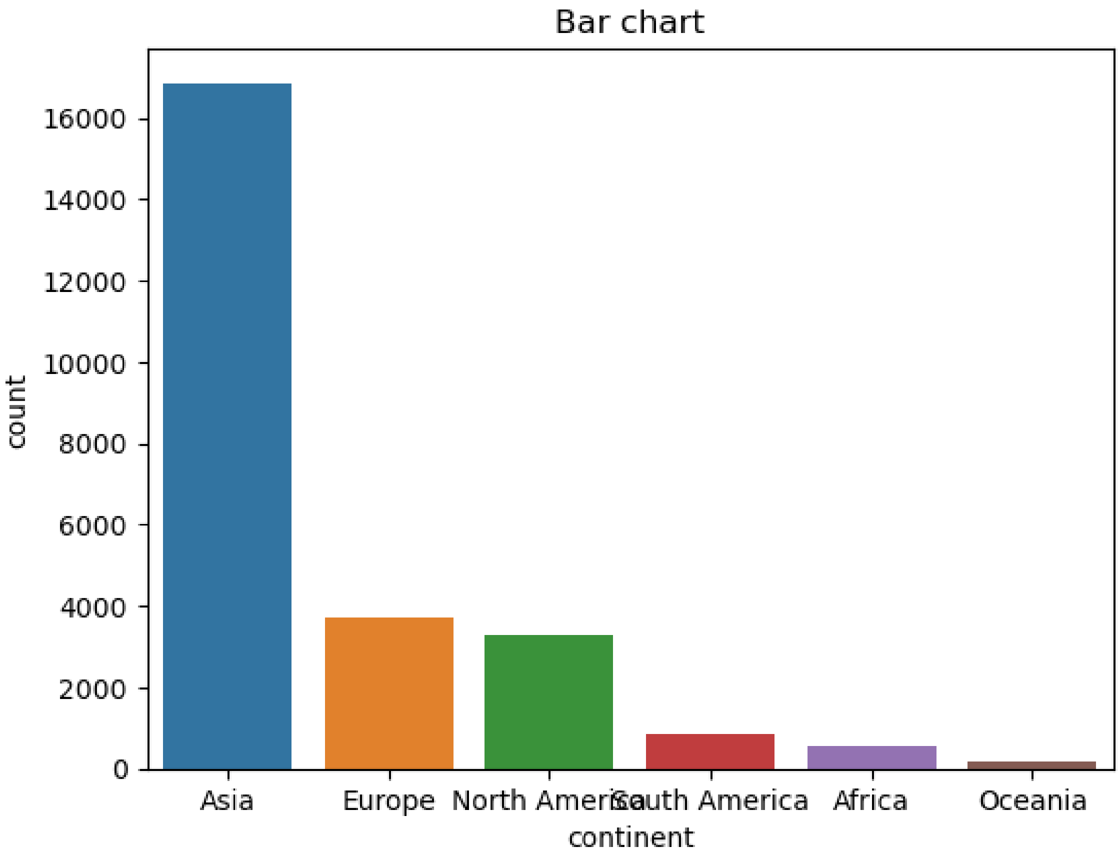
```
Index(['Asia', 'Europe', 'North America', 'South America', 'Africa',
       'Oceania'],
      dtype='object', name='continent')
```

In [ ]:

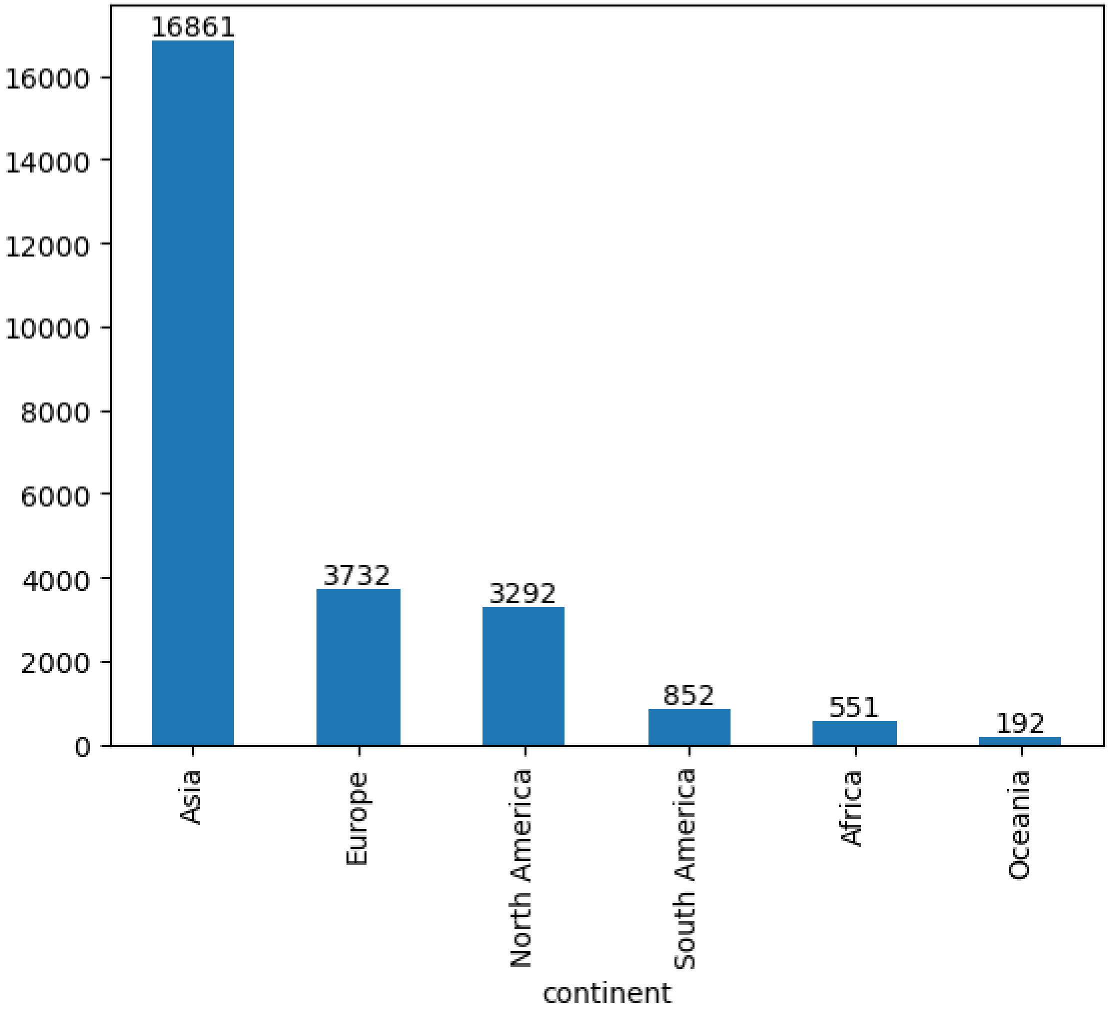
```
index
array
tuple
list
```

In [41]:

```
visa_df['continent'].value_counts()
# In the series
# we have left side :  keys
# we have right side: values
keys=visa_df['continent'].value_counts().keys()  # keys
values=visa_df['continent'].value_counts().to_list() # values
sns.countplot(data=visa_df,
              x='continent',
             order=keys)
plt.title("Bar chart")
plt.show()
```



```
In [53]: values=visa_df['continent'].value_counts()
         ax=values.plot(kind='bar')
         ax.bar_label(ax.containers[0])
         plt.show()
```

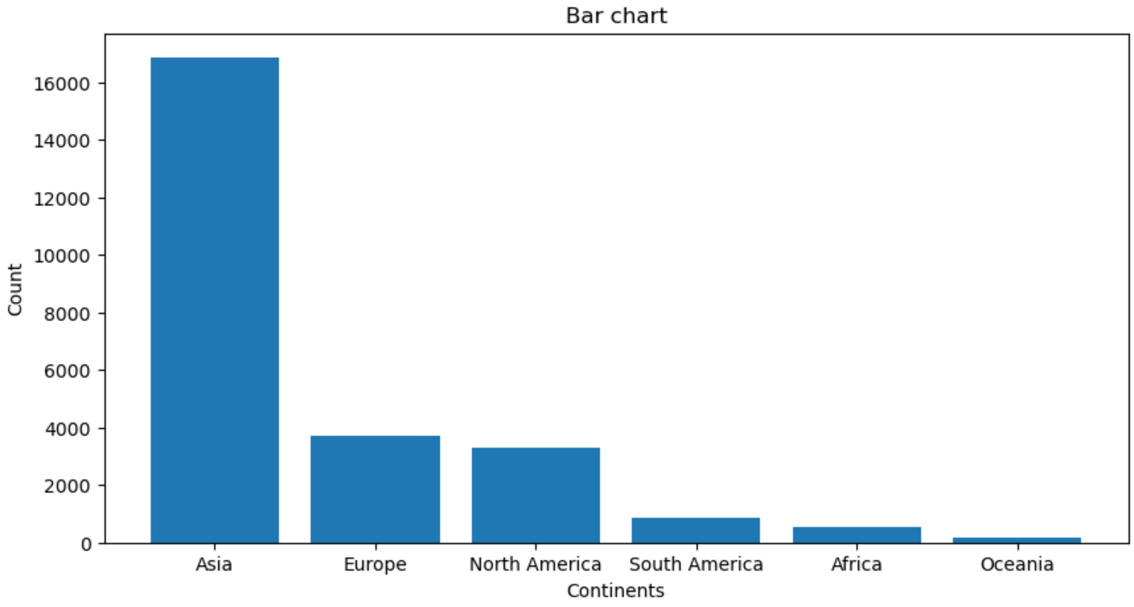


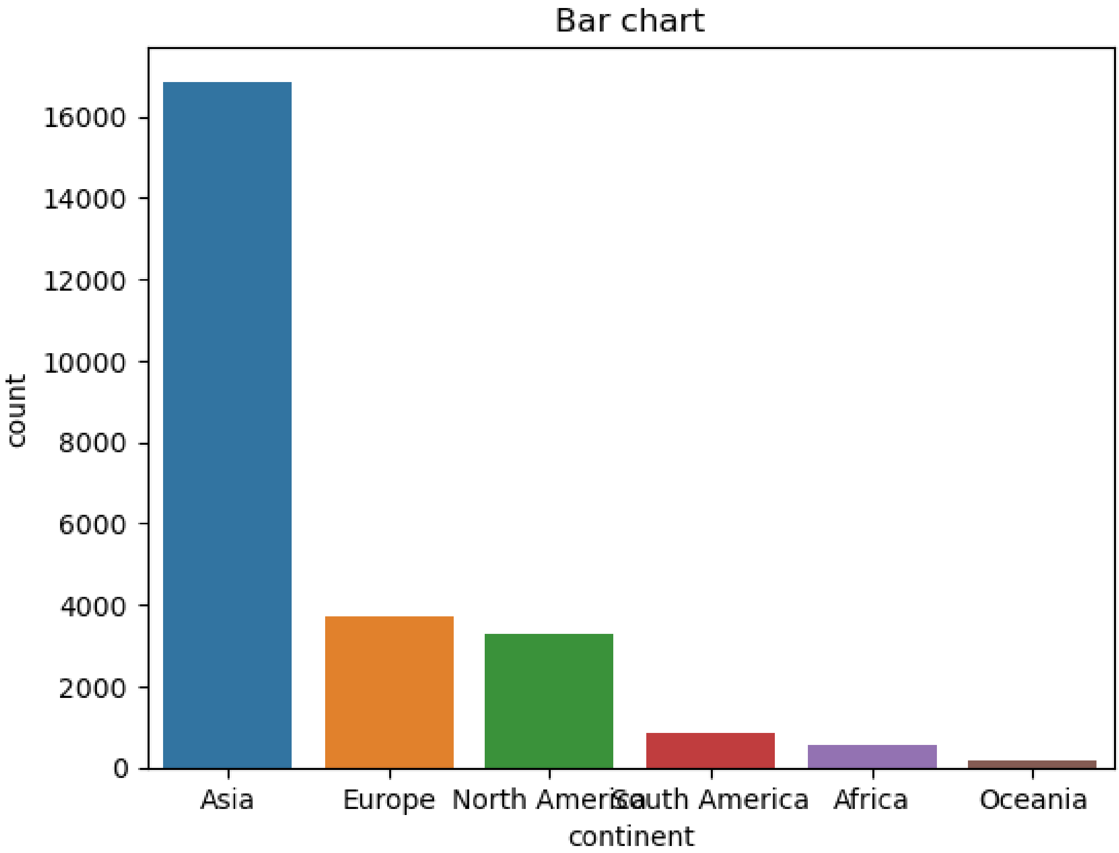
In [54]:

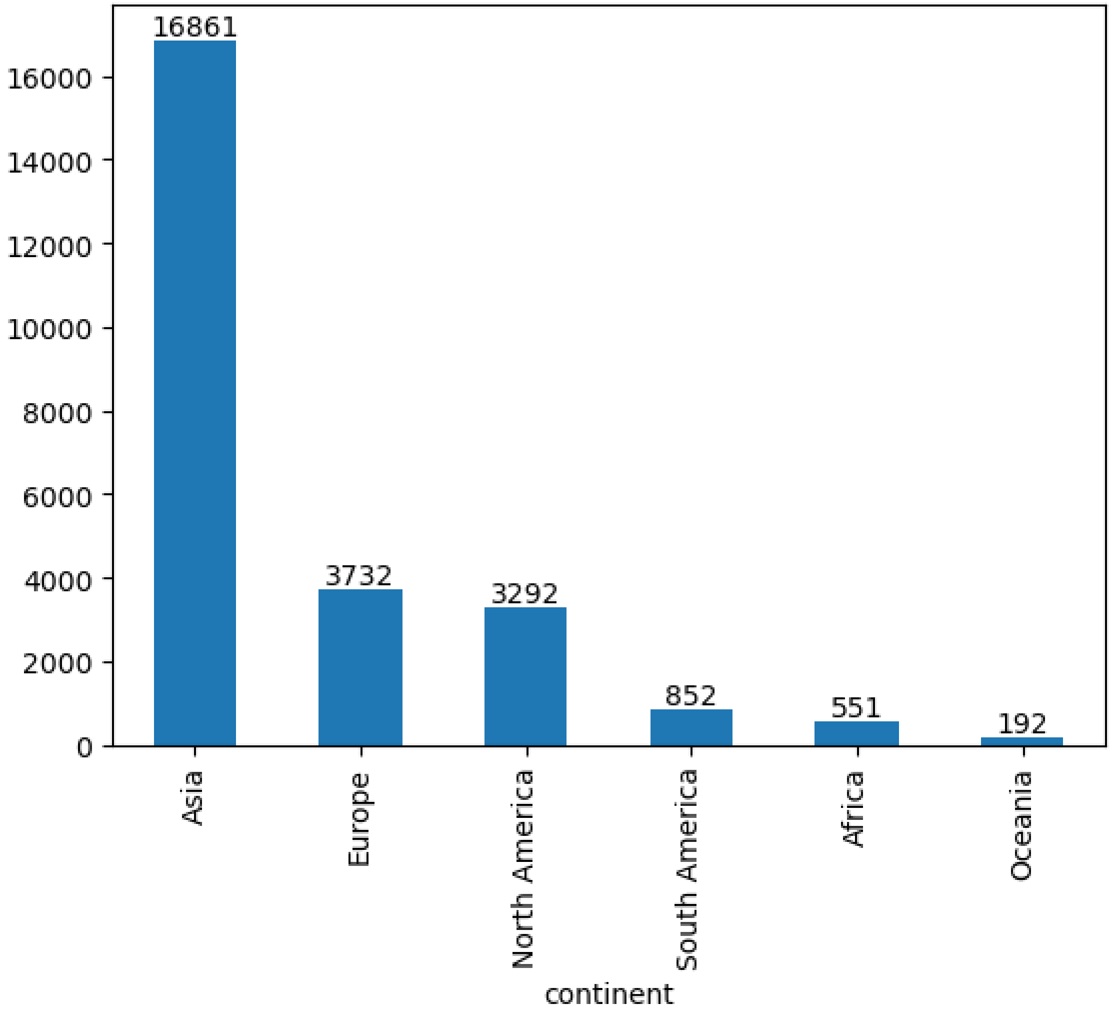
```
# Method-1:
########################### Matplotlib bar plot####################
plt.figure(figsize=(10,5))
plt.bar('Continent','Count',data=continent_df)
plt.xlabel("Continents")
plt.ylabel("Count")
plt.title("Bar chart")
plt.savefig("Continent_bar_chart.jpg")
plt.show()

# Method-2:
######################## Seaborn count plot###############
visa_df['continent'].value_counts()
keys=visa_df['continent'].value_counts().keys()  # keys
values=visa_df['continent'].value_counts().to_list() # values
sns.countplot(data=visa_df,
              x='continent',
             order=keys)
plt.title("Bar chart")
plt.show()

#Method-3:
############################ Value counts plot #######################
values=visa_df['continent'].value_counts()
ax=values.plot(kind='bar')
ax.bar_label(ax.containers[0])
plt.show()
```




Bar chart
count
0
2000
4000
6000
8000
10000
12000
14000
16000
Asia
Europe
North America
South America
Africa
Oceania
continent


16861
3732
3292
852
551
192
0
2000
4000
6000
8000
10000
12000
14000
16000
Asia
Europe
North America
South America
Africa
Oceania
continent

*Pie – chart*

In [56]:

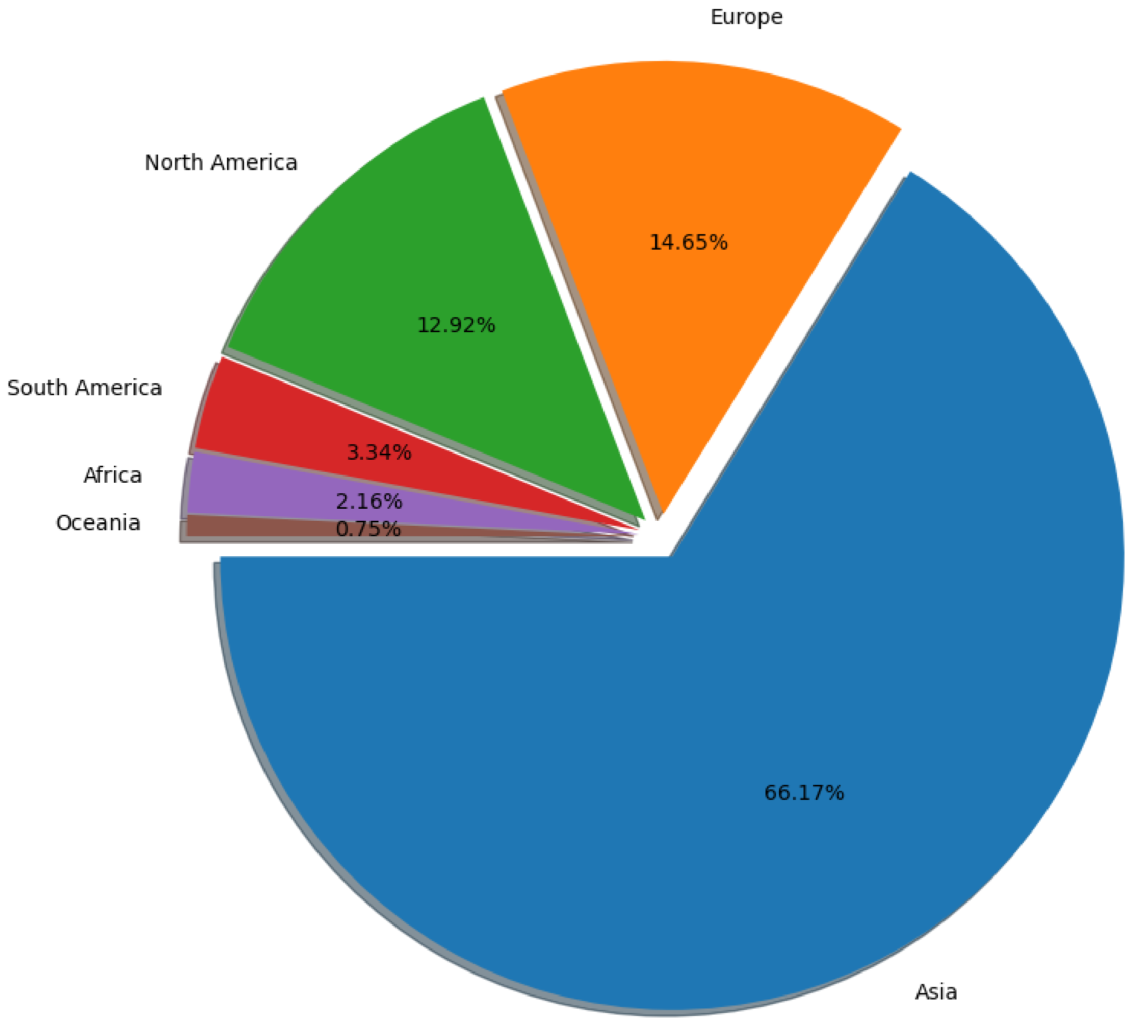
```
values=visa_df['continent'].value_counts(normalize=True)
values
```

Out[56]:

```
continent
Asia             0.661735
Europe           0.146468
North America    0.129199
South America    0.033438
Africa           0.021625
Oceania          0.007535
Name: proportion, dtype: float64
```

In [73]:

```
# x= values
# labels= keys
keys=visa_df['continent'].value_counts().keys()
values=visa_df['continent'].value_counts().to_list()
plt.pie(x=values,
       labels=keys,
       autopct="%0.2f%%",
       explode=[0.1,0.1,0.1,0.1,0.1,0.1],
       shadow=True,
       startangle=180,
       radius=2)
plt.show()
```



In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: