Import the packages

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

Read the data

In [4]:
```
path=r"C:\Users\omkar\OneDrive\Documents\Data science\Naresh IT\Datafiles\V
visa_df=pd.read_csv(path)
visa_df.head(3)
```

Out[4]:

	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	

Reading a specific column

In [5]:
```
visa_df['continent'] # series type
```

Out[5]:
```
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 [6]:
```
visa_df[['continent']] # data frame
```

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]:
```
visa_df.continent # series
```

Out[7]:
```
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 [ ]:
```
visa_df['continent'] # series
visa_df.continent    # series
visa_df[['continent']] # df
```

In [8]:
```
visa_df.columns
```

Out[8]:
```
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 [9]:
```
cols=['continent','education_of_employee']
visa_df[cols]
```

Out[9]:

	continent	education_of_employee
0	Asia	High School
1	Asia	Master's
2	Asia	Bachelor's
3	Asia	Bachelor's
4	Africa	Master's
...	...	...
25475	Asia	Bachelor's
25476	Asia	High School
25477	Asia	Master's
25478	Asia	Master's
25479	Asia	Bachelor's

25480 rows × 2 columns

In [11]:
```
visa_df.values

# list of all the samples
# list of all the observations
# list of all the tuples
```

Out[11]:
```
array([['EZYV01', 'Asia', 'High School', ..., 'Hour', 'Y', 'Denied'],
       ['EZYV02', 'Asia', "Master's", ..., 'Year', 'Y', 'Certified'],
       ['EZYV03', 'Asia', "Bachelor's", ..., 'Year', 'Y', 'Denied'],
       ...,
       ['EZYV25478', 'Asia', "Master's", ..., 'Year', 'N', 'Certified'],
       ['EZYV25479', 'Asia', "Master's", ..., 'Year', 'Y', 'Certified'],
       ['EZYV25480', 'Asia', "Bachelor's", ..., 'Year', 'Y', 'Certifie
d']],
      dtype=object)
```

In [ ]:
```
# if i give list ==== df
# if i give df ==== list
```

continent

In [16]:
```
l1=[1,2,3]
l2=['A','B','C']
l=[l1,l2]
l
pd.DataFrame(l)
```

Out[16]:

	0	1	2
0	1	2	3
1	A	B	C

In [17]:
```
col=['continent']
visa_df[col]
```

Out[17]:

	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

unique

In [18]:
```
# how many unique labels are there
visa_df['continent'].unique()
```

Out[18]:
```
array(['Asia', 'Africa', 'North America', 'Europe', 'South America',
       'Oceania'], dtype=object)
```

In [19]:
```
# python basic logics
l1=['A','A','B','C']   # ['A','B','C']
set(l1)
```

Out[19]:
```
{'A', 'B', 'C'}
```

In [21]:
```
set(visa_df['continent'].values)
```

Out[21]:
```
{'Africa', 'Asia', 'Europe', 'North America', 'Oceania', 'South America'}
```

nunique

In [22]:
```
visa_df['continent'].nunique()
# number of unique elements
```

Out[22]:
```
6
```

in the contienent column only 7 elements repeated

{'Africa', 'Asia', 'Europe', 'North America', 'Oceania', 'South America'}

Q1)out of total observations How many asia observations are there?

In [26]:
```
con=visa_df['continent']=='Asia'  # True and False
visa_df[con]
```

Out[26]:

	case_id	continent	education_of_employee	has_job_experience	requires_job_traini
0	EZYV01	Asia	High School	N	
1	EZYV02	Asia	Master's	Y	
2	EZYV03	Asia	Bachelor's	N	
3	EZYV04	Asia	Bachelor's	N	
5	EZYV06	Asia	Master's	Y	
...	...	...	...	...	
25475	EZYV25476	Asia	Bachelor's	Y	
25476	EZYV25477	Asia	High School	Y	
25477	EZYV25478	Asia	Master's	Y	
25478	EZYV25479	Asia	Master's	Y	
25479	EZYV25480	Asia	Bachelor's	Y	

16861 rows × 12 columns

In [27]:
```
con=visa_df['continent']=='Asia'  # True and False
len(visa_df[con])
```

Out[27]: 16861

In [28]:
```
con=visa_df['continent']=='Africa'  # True and False
len(visa_df[con])
```

Out[28]: 551

In [31]:
```
unique_labels= visa_df['continent'].unique()
for i in unique_labels:
    con=visa_df['continent']==i  # True and False
    print(i,":",len(visa_df[con]))
```

```
Asia : 16861
Africa : 551
North America : 3292
Europe : 3732
South America : 852
Oceania : 192
```

Frequency table

In [35]:
```
unique_labels= visa_df['continent'].unique()
count=[]
for i in unique_labels:
    con=visa_df['continent']==i  # True and False
    count.append(len(visa_df[con]))

continent_df=pd.DataFrame(zip(unique_labels,count),
                          columns=['Continent','Count'])
continent_df.to_csv('continent_df.csv',index=False)
```

In [ ]:
```
visa_df  # Total data frame
visa_df['continent'] # specific column
visa_df['continent']=='Asia' # Specific lable
#######################################################
len(visa_df[visa_df['continent']=='Asia'])

##################################################

unique_labels= visa_df['continent'].unique()
count=[]
for i in unique_labels:
    con=visa_df['continent']==i  # True and False
    count.append(len(visa_df[con]))

#######################################################
continent_df=pd.DataFrame(zip(unique_labels,count),
                          columns=['Continent','Count'])

#########################################################
continent_df.to_csv('continent_df.csv',index=False)
```

In [36]:
```
continent_df
```

Out[36]:

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

value-counts

In [38]:
```
continent_vc=visa_df['continent'].value_counts() # series
continent_vc
```

Out[38]:
```
continent
Asia             16861
Europe            3732
North America     3292
South America      852
Africa             551
Oceania            192
Name: count, dtype: int64
```

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

In [39]:
```
continent_vc.keys()
```

Out[39]:
```
Index(['Asia', 'Europe', 'North America', 'South America', 'Africa',
       'Oceania'],
      dtype='object', name='continent')
```

In [41]:
```
continent_vc.values
```

Out[41]:
```
array([16861,  3732,  3292,   852,   551,   192], dtype=int64)
```

In [43]:
```
continent_vc=visa_df['continent'].value_counts() # series
l1=continent_vc.keys()
l2=continent_vc.values
continent_vc_df=pd.DataFrame(zip(l1,l2),
                             columns=['continent','count'])

continent_vc_df
```

Out[43]:

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

In [46]:
```
visa_df  # Total data frame
visa_df['continent'] # specific column
visa_df['continent']=='Asia' # Specific lable
#####################################################
len(visa_df[visa_df['continent']=='Asia'])
len(visa_df[visa_df['continent']=='Africa'])
len(visa_df[visa_df['continent']=='Europe'])
len(visa_df[visa_df['continent']=='North America'])
len(visa_df[visa_df['continent']=='South America'])
len(visa_df[visa_df['continent']=='Oceania'])

########-------Method-1---------#########################

unique_labels= visa_df['continent'].unique()
count=[]
for i in unique_labels:
    con=visa_df['continent']==i  # True and False
    count.append(len(visa_df[con]))

continent_df=pd.DataFrame(zip(unique_labels,count),
                          columns=['Continent','Count'])


print(continent_df)

##################------ M-2----(Value counts)----##########################
continent_vc=visa_df['continent'].value_counts() # series
l1=continent_vc.keys()
l2=continent_vc.values
continent_vc_df=pd.DataFrame(zip(l1,l2),
                             columns=['continent','count'])

print(continent_vc_df)
```

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

In [47]:
```
continent_vc
```

Out[47]:
```
continent
Asia             16861
Europe            3732
North America     3292
South America      852
Africa             551
Oceania            192
Name: count, dtype: int64
```

In [48]:
```
continent_df
```

Out[48]:

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

Bar chart

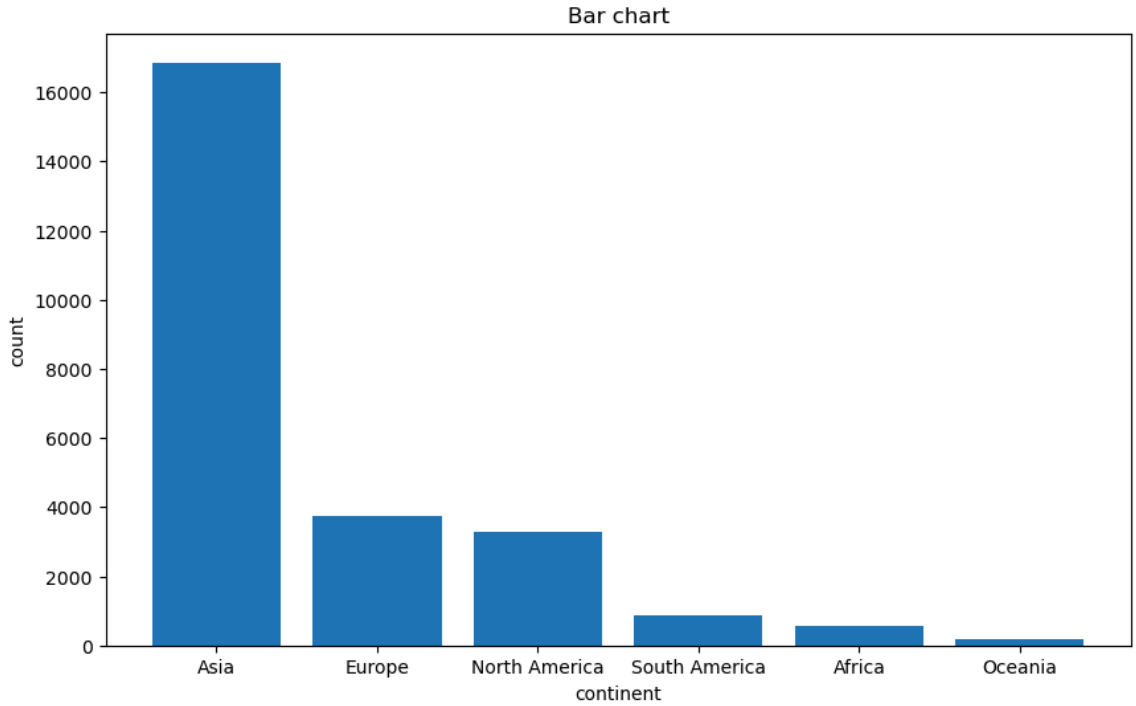
- in order to draw bar chart
- we required one categorical colun
- we required one numerical column
- package: matplotlib
- dataframe: continent_vc_df

In [51]:
```
#plt.bar(<cat>,<numer>,<data>)
continent_vc_df
```

Out[51]:

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

In [61]:
```
plt.figure(figsize=(10,6)) # to incease the plot size
plt.bar('continent',
        'count',
        data=continent_vc_df)
plt.xlabel("continent") # x-axis name
plt.ylabel('count') # y-axis name
plt.title("Bar chart") # title of the chart
plt.savefig('continent_bar.jpg')
plt.show()
```



- we read the data
- we read categorical column
- we made frequency table by using value counts
- we plot the bar chart using matplotlib
- But matplotlib required 3 arguments
  - x label: categorical column (width)
  - y label: numerical column (height)
  - data (frquency table name)

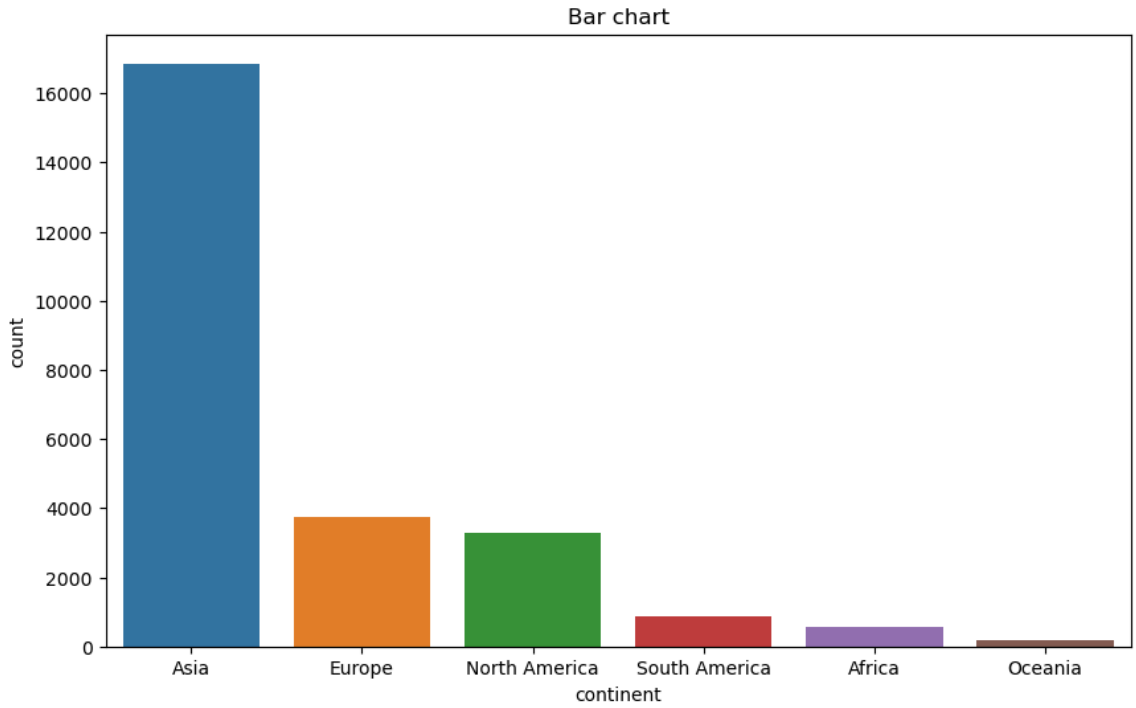
Count plot

- count plot can use bt seaborn package
- It requires only **entire dataframe** and **categorical column**
- entire dataframe name: **Visadf**
- categorical column name: **contnent**
- order: In which order you want plot

In [65]:
```
visa_df['continent'].value_counts().keys()
```

Out[65]:
```
Index(['Asia', 'Europe', 'North America', 'South America', 'Africa',
       'Oceania'],
      dtype='object', name='continent')
```

In [70]:
```
plt.figure(figsize=(10,6))
# l=['Asia', 'Oceania', 'North America', 'South America', 'Africa',
#        'Europe']
l=visa_df['continent'].value_counts().keys() # order provide automatically
sns.countplot(data=visa_df,
              x='continent',
             order=l)
plt.xlabel("continent") # x-axis name
plt.ylabel('count') # y-axis name
plt.title("Bar chart") # title of the chart
plt.savefig('continent_bar.jpg')
plt.show()
```



In [ ]:
```
# perform the same analysis on education employee
# show me the plots in whatsapp group
# take a screenshot and post in the group
```

In [1]:
```
# Import packages
# and read 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]:

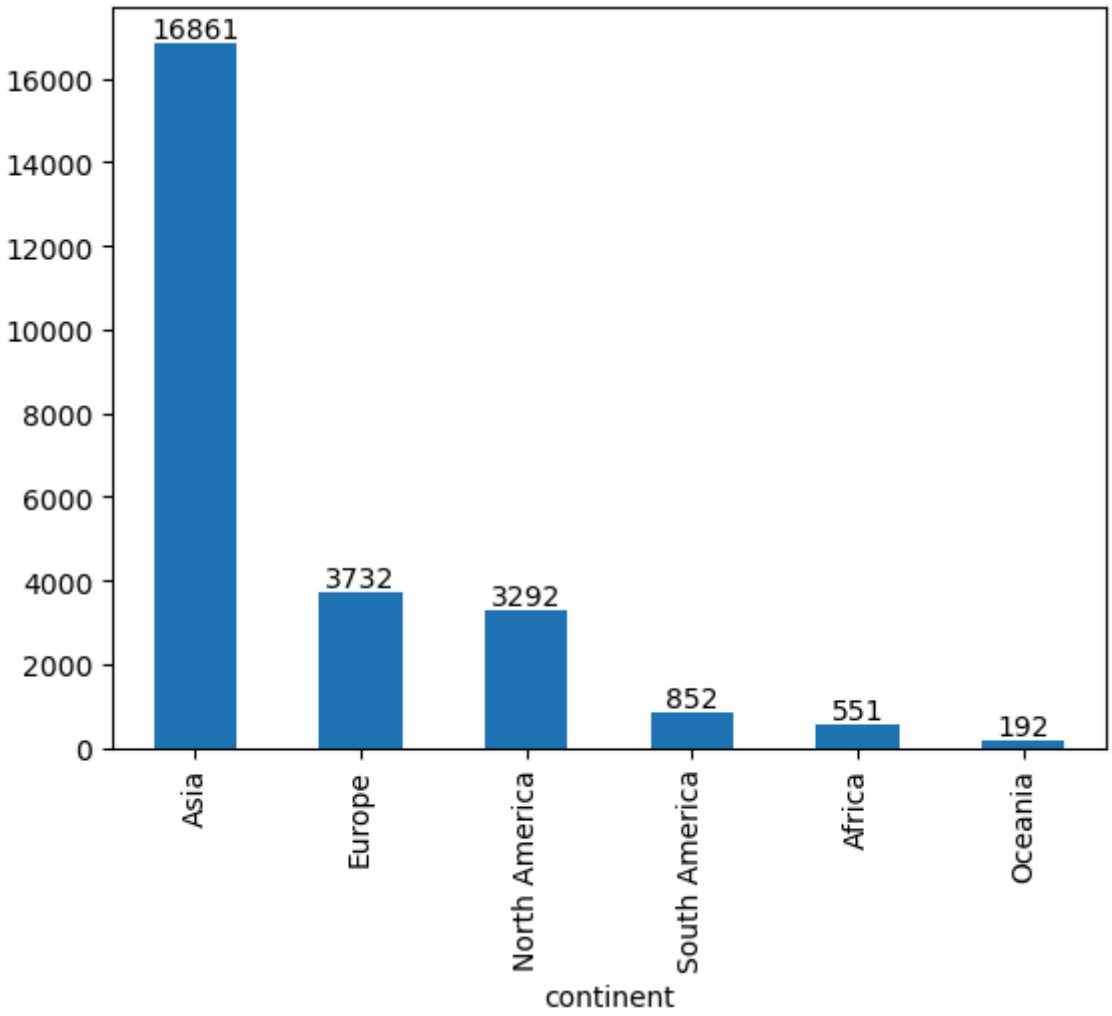
	case_id	continent	education_of_employee	has_job_experience	requires_job_training	no_
0	EZYV01	Asia	High School	N	N	
1	EZYV02	Asia	Master's	Y	N	
2	EZYV03	Asia	Bachelor's	N	Y	

$Method - 3$

- we created a frequency table : matplotlib
- we created bar chart using seaborn
  - main dataframe
  - column name
- by using value counts

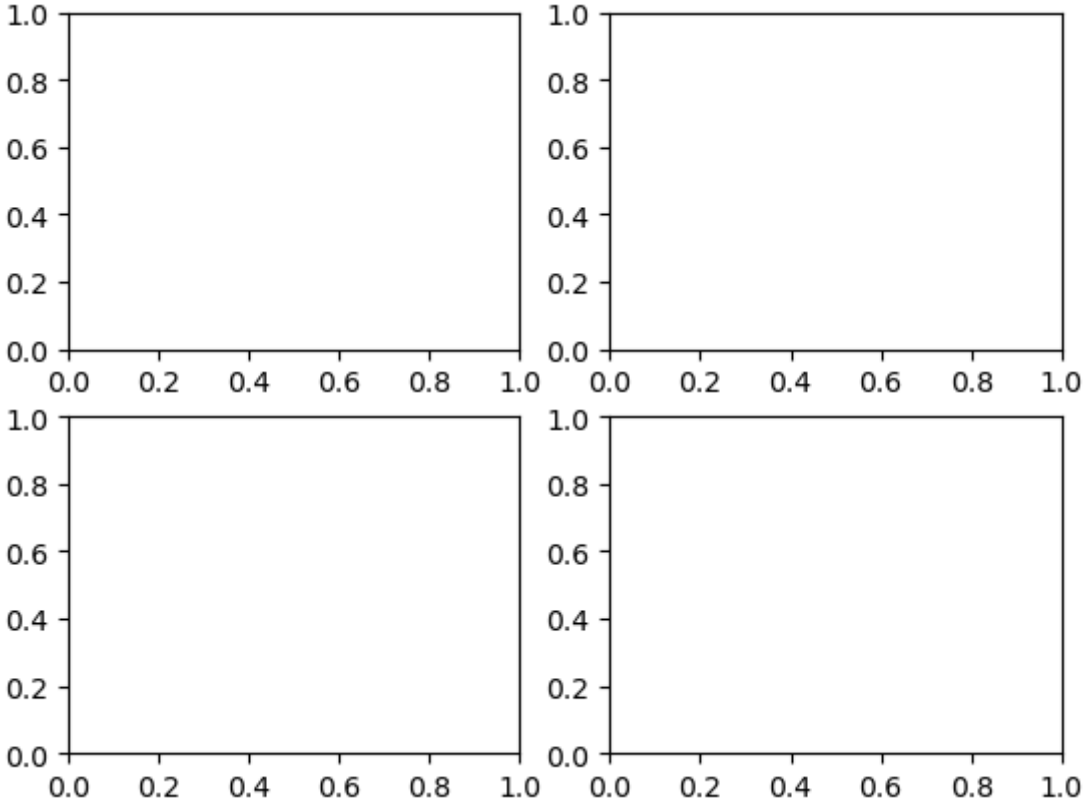
In [14]:
```
values=visa_df['continent'].value_counts()
ax=values.plot(kind='bar')
ax.bar_label(ax.containers[0])
```

Out[14]:
```
[Text(0, 0, '16861'),
 Text(0, 0, '3732'),
 Text(0, 0, '3292'),
 Text(0, 0, '852'),
 Text(0, 0, '551'),
 Text(0, 0, '192')]
```



In [15]:
```
plt.subplot(2,2,1)
plt.subplot(2,2,2)
plt.subplot(2,2,3)
plt.subplot(2,2,4)
```

Out[15]: <Axes: >



In [ ]:
```
######################### M-1 ##############################################
plt.figure(figsize=(10,6)) # to incease the plot size
plt.bar('continent',
        'count',
        data=continent_vc_df)
plt.xlabel("continent") # x-axis name
plt.ylabel('count') # y-axis name
plt.title("Bar chart") # title of the chart
plt.savefig('continent_bar.jpg')
plt.show()

######################### M-2 ##############################################

plt.figure(figsize=(10,6))
# l=['Asia', 'Oceania', 'North America', 'South America', 'Africa',
#        'Europe']
l=visa_df['continent'].value_counts().keys() # order provide automatically
sns.countplot(data=visa_df,
              x='continent',
             order=l)
plt.xlabel("continent") # x-axis name
plt.ylabel('count') # y-axis name
plt.title("Bar chart") # title of the chart
plt.savefig('continent_bar.jpg')
plt.show()

############################## M-3 ##############################################

values=visa_df['continent'].value_counts()
ax=values.plot(kind='bar')
ax.bar_label(ax.containers[0])
```

Relative frequency

In [18]:
```
visa_df['continent'].value_counts(normalize=True)
```

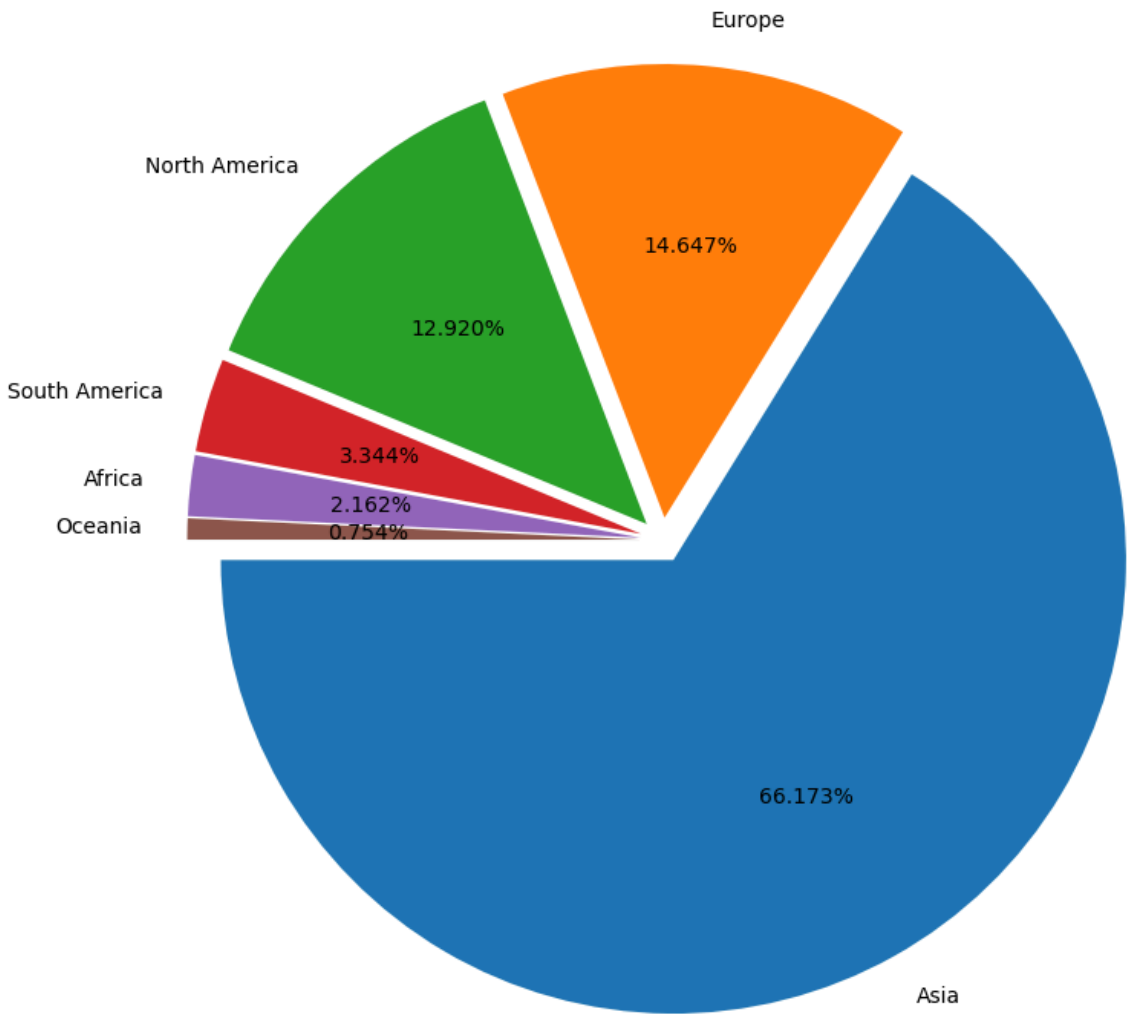
Out[18]:
```
continent
Asia             0.661735
Europe           0.146468
North America    0.129199
South America    0.033438
Africa           0.021625
Oceania          0.007535
Name: proportion, dtype: float64
```

```
$Pie$ $chart$

- pie chart will automatically convert values to percentages

- will take value count help with out normalize

- x is data in the form list

- labels also in the form of list
```

In [22]:
```
keys=visa_df['continent'].value_counts().keys()
values=visa_df['continent'].value_counts().values
values
```

Out[22]:
```
array([16861,  3732,  3292,   852,   551,   192], dtype=int64)
```

In [34]:
```
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) # rotation
plt.show()
```



In [ ]: