Categorical Analysis

Date-11-12-2023

$Step - 1$

Import required packages

In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

$Step - 2$

Read Data

In [3]:
```
file_path="C:\\Users\\kurre\\OneDrive\\Documents\\Naresh IT\\datafiles\\Vis
visa_df=pd.read_csv(file_path)
```

In [4]:
```
visa_df.head()
```

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

$step - 3$

read a column

In [4]:
```
visa_df.columns
```

Out[4]:
```
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 [5]:
```
type(visa_df.columns)
```

Out[5]:
```
pandas.core.indexes.base.Index
```

In [5]:
```
visa_df['continent']
```

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 [7]:
```
type(visa_df['continent'])
```

Out[7]:
```
pandas.core.series.Series
```

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

# visa_df['continent']============ series
# visa_df[['continent']] ======== dataframe(table)
```

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

Out[7]:
```
pandas.core.frame.DataFrame
```

In [8]:
```
type(visa_df[cols])
```

Out[8]:
```
pandas.core.frame.DataFrame
```

In [9]:
```
visa_df.continent
```

Out[9]:
```
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']==================> serise type
# visa_df[['continent']]================> dataframe
# visa_df.continent   ================> index type
```

In [10]:
```
# Two column at a time

cols=['continent','case_status']
visa_df[cols]
```

Out[10]:

	continent	case_status
0	Asia	Denied
1	Asia	Certified
2	Asia	Denied
3	Asia	Denied
4	Africa	Certified
...	...	...
25475	Asia	Certified
25476	Asia	Certified
25477	Asia	Certified
25478	Asia	Certified
25479	Asia	Certified

25480 rows × 2 columns

Unique

In [13]:
```
# first read the column
# the apply unique

# dont apply unique operation for dataframe : [[]]
# apply only for series :[]


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

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

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

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

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

Out[15]:
```
25480
```

nunique

In [16]:
```
visa_df['continent'].nunique()

# number of unique lables
```

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

In [17]:
```
visa_df[['continent','case_status']]
```

Out[17]:

	continent	case_status
0	Asia	Denied
1	Asia	Certified
2	Asia	Denied
3	Asia	Denied
4	Africa	Certified
...	...	...
25475	Asia	Certified
25476	Asia	Certified
25477	Asia	Certified
25478	Asia	Certified
25479	Asia	Certified

25480 rows × 2 columns

In [ ]:

```
# we read continent column
# we understood there 6 unique lables are there
# these 6 unique lables repaeting and total 25480 observations
# how many are 'asia' are there
# how many are 'africa' are there
```

In [19]:

```
visa_df['continent']
```

Out[19]:

```
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 [20]:

```
visa_df['continent']=='Asia'


# do you want to know how many True
# how many rows are satisfying condition
# how many observations are having continent as asia
```

Out[20]:

```
0         True
1         True
2         True
3         True
4        False
         ...  
25475     True
25476     True
25477     True
25478     True
25479     True
Name: continent, Length: 25480, dtype: bool
```

In [21]:

```
len(visa_df['continent']=='Asia')
```

Out[21]:

```
25480
```

In [22]:
```
visa_df[visa_df['continent']=='Asia']
```

Out[22]:

	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 [23]:
```
len(visa_df[visa_df['continent']=='Asia'])

# check the len only with  the dataframe
```

Out[23]:
```
16861
```

In [ ]:
```
visa_df                            # complete df
visa_df['continent']               # column
visa_df['continent']=='Asia'  # one label
visa_df[visa_df['continent']=='Asia'] # df
len(visa_df[visa_df['continent']=='Asia']) # len
```

In [24]:

```
############### BAD WAY################################

print(len(visa_df[visa_df['continent']=='Asia']))
print(len(visa_df[visa_df['continent']=='Africa']))
print(len(visa_df[visa_df['continent']=='North America']))
print(len(visa_df[visa_df['continent']=='Europe']))
print(len(visa_df[visa_df['continent']=='South America']))
print(len(visa_df[visa_df['continent']=='Oceania']))




continents=visa_df['continent'].unique()
# for loop
print(len(visa_df[visa_df['continent']==i]))
```

```
16861
551
3292
3732
852
192
```

In [27]:

```
# generlised expression

continents=visa_df['continent'].unique()
for i in continents:
    count=len(visa_df[visa_df['continent']==i])
    print(i,':',count)
```

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

In [28]:

```
count=[]
continents=visa_df['continent'].unique()
for i in continents:
    c=len(visa_df[visa_df['continent']==i])
    count.append(c)
count
```

Out[28]:

```
[16861, 551, 3292, 3732, 852, 192]
```

In [29]:

```
# list comprihention

continents=visa_df['continent'].unique()
count=[len(visa_df[visa_df['continent']==i]) for i in continents]

count,continents
```

Out[29]:

```
([16861, 551, 3292, 3732, 852, 192],
 array(['Asia', 'Africa', 'North America', 'Europe', 'South America',
        'Oceania'], dtype=object))
```

In [31]:
```
# create data frame and save in local

continents_df=pd.DataFrame(zip(continents,count),
                           columns=['Continents','Count'])

continents_df.to_csv('continents_info.csv',index=False)
```

In [ ]:
```
# improve step by step-


visa_df                     # complete df
visa_df['continent']        # column
visa_df['continent']=='Asia' # one label
visa_df[visa_df['continent']=='Asia'] # df
len(visa_df[visa_df['continent']=='Asia']) # len

###########################################################################
print(len(visa_df[visa_df['continent']=='Asia']))
print(len(visa_df[visa_df['continent']=='Africa']))
print(len(visa_df[visa_df['continent']=='North America']))
print(len(visa_df[visa_df['continent']=='Europe']))
print(len(visa_df[visa_df['continent']=='South America']))
print(len(visa_df[visa_df['continent']=='Oceania']))

###########################################################################
continents=visa_df['continent'].unique()
for i in continents:
    count=len(visa_df[visa_df['continent']==i])
    print(i,':',count)

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

continents=visa_df['continent'].unique()
count=[len(visa_df[visa_df['continent']==i]) for i in continents]

continents_df=pd.DataFrame(zip(continents,count),
                           columns=['Continents','Count'])

continents_df.to_csv('continetns_info.csv',index=False)
```

In [32]:
```
# one more method

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

Out[32]:
```
continent
Asia             16861
Europe            3732
North America     3292
South America      852
Africa             551
Oceania            192
Name: count, dtype: int64
```

$Value\text{-}Counts$

In [35]:
```
pd.Series(count)
```

Out[35]:
```
0    16861
1      551
2     3292
3     3732
4      852
5      192
dtype: int64
```

In [36]:
```
pd.Series(continents)
```

Out[36]:
```
0             Asia
1           Africa
2    North America
3           Europe
4    South America
5          Oceania
dtype: object
```

In [34]:
```
pd.Series(count,index=continents)
```

Out[34]:
```
Asia             16861
Africa             551
North America     3292
Europe            3732
South America      852
Oceania            192
dtype: int64
```

In [33]:
```
visa_df['continent'].value_counts()
```

Out[33]:
```
continent
Asia             16861
Europe            3732
North America     3292
South America      852
Africa             551
Oceania            192
Name: count, dtype: int64
```

In [ ]:
```
# Always Learn how a method a giving answer
# will im able to write same answer with out method
```

In [37]:

```
# How to create a dataframe using value counts
#      or   using series

visa_df['continent'].value_counts()

# in order to create a dataframe
# we need two list
# or one dictionary

# from value counts create two lists
# values
# keys
```

Out[37]:

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

In [42]:

```
# Method-1

continents=visa_df['continent'].value_counts().keys()
count=visa_df['continent'].value_counts().values
pd.DataFrame(zip(continents,count),columns=['continents','count'])
```

Out[42]:

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

In [43]:

```
# Method-2

dict1=dict(visa_df['continent'].value_counts())
print(dict1)

# 16861 is a scalar value, it is not in a list

pd.DataFrame(dict1,index=['count'])
```

```
{'Asia': 16861, 'Europe': 3732, 'North America': 3292, 'South America': 85
2, 'Africa': 551, 'Oceania': 192}
```

Out[43]:

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

In [44]:

```
# Method-3

dict1=dict(visa_df['continent'].value_counts())
keys=dict1.keys()
values=dict1.values()

pd.DataFrame(zip(keys,values),columns=['Continent','Count'])
```

Out[44]:

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

In [45]:

```
# this is a frequancy table

continents_df

# one column= categorical : Continents
#     column= numerical : Count
```

Out[45]:

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

- Bar plot
- pie chart

Bar-plot

- x-axis: categorical column
- y-axis: numerical column
- where you are taking the data: continents_df

In [ ]:

```
continents_df
# we are creating from scratch
```

In [ ]:

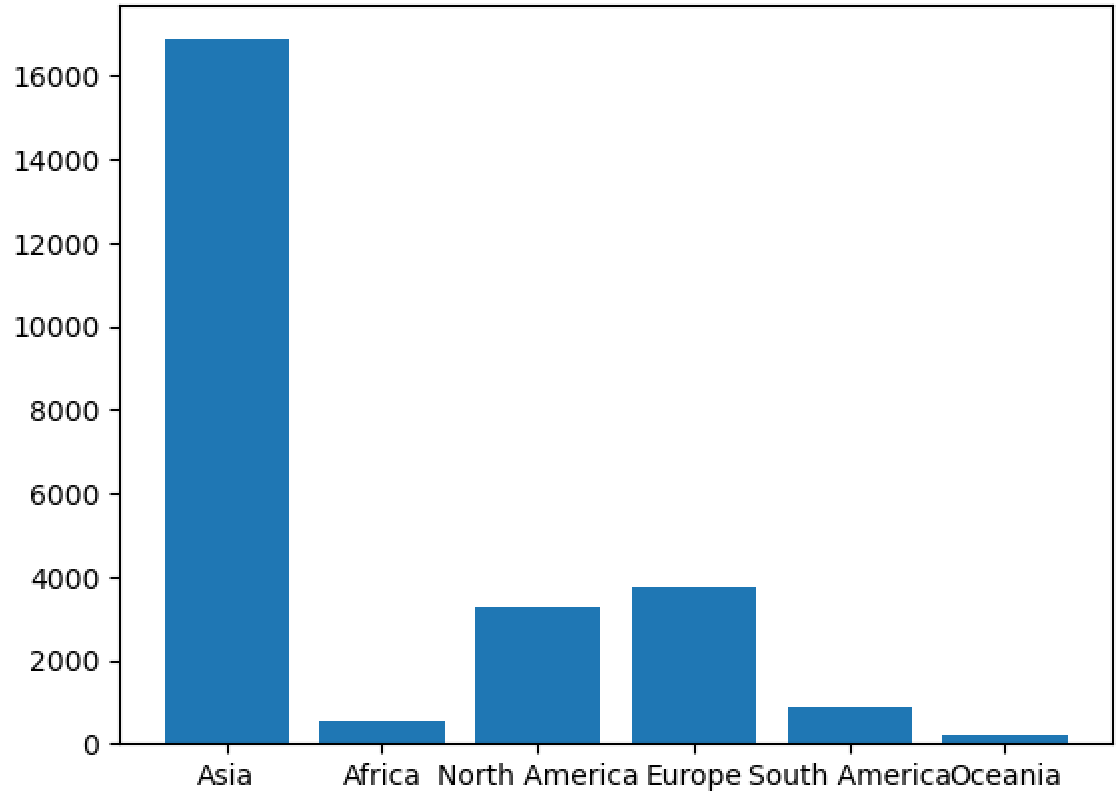
```
# one column= categorical : Continents  == x-axis
#     collumn= numerical : Count        == y-axis
```

In [46]:

```
plt.bar('Continents','Count',data=continents_df)
```

Out[46]:

```
<BarContainer object of 6 artists>
```



In [ ]:

```
# always take values count dataframe == for proper order
# dont take scratch level dataframe
```

In [51]:

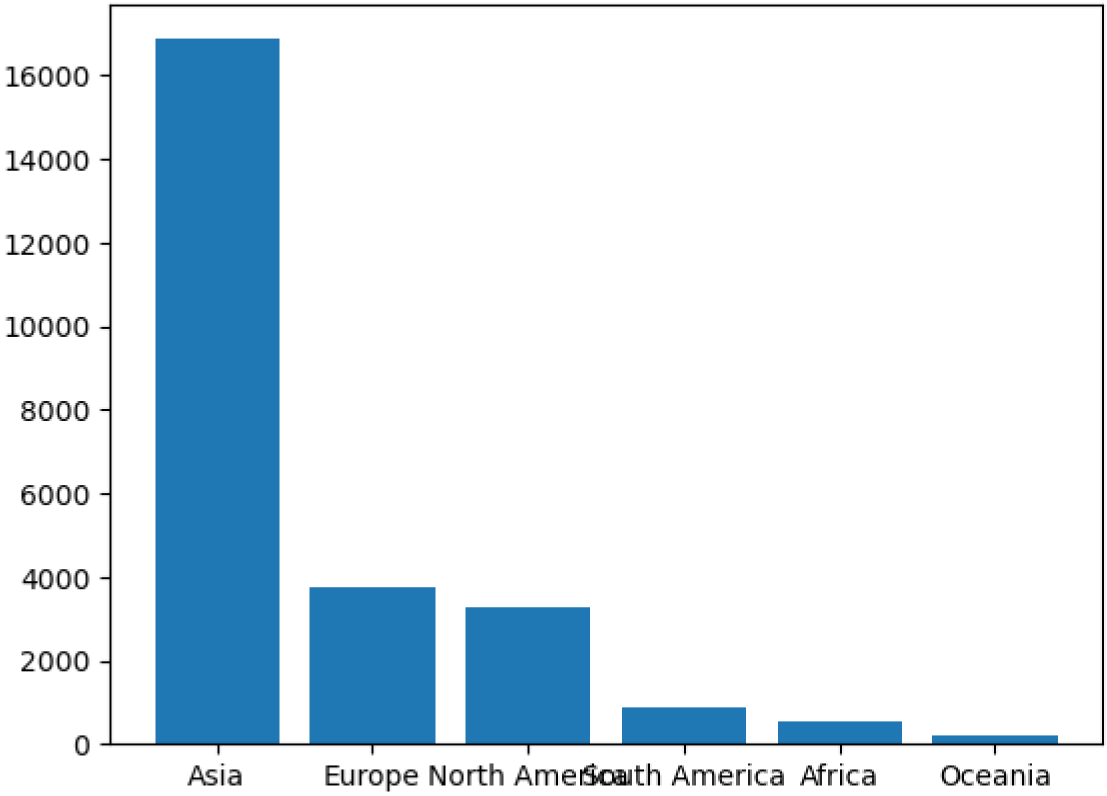
```
# Method-1:

visa_df['continent'].value_counts()
continents=visa_df['continent'].value_counts().keys()
count=visa_df['continent'].value_counts().values
contint_data=pd.DataFrame(zip(continents,count),
                          columns=['continetns','count'])

contint_data
```

Out[51]:

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

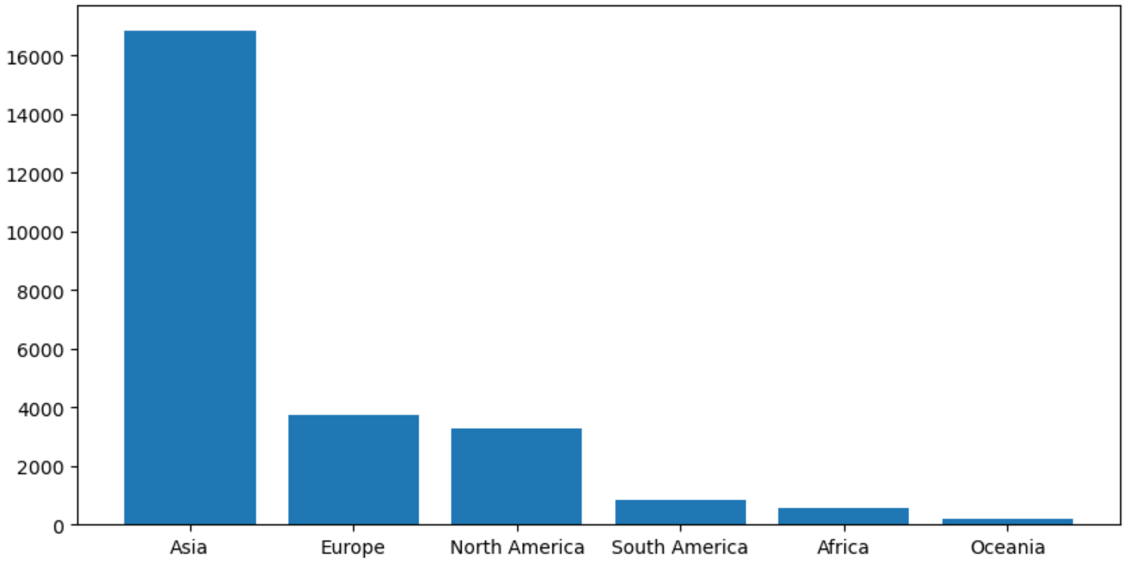
In [52]:
```
# contint_data

plt.bar('continetns','count',data=contint_data)
```

Out[52]:
```
<BarContainer object of 6 artists>
```

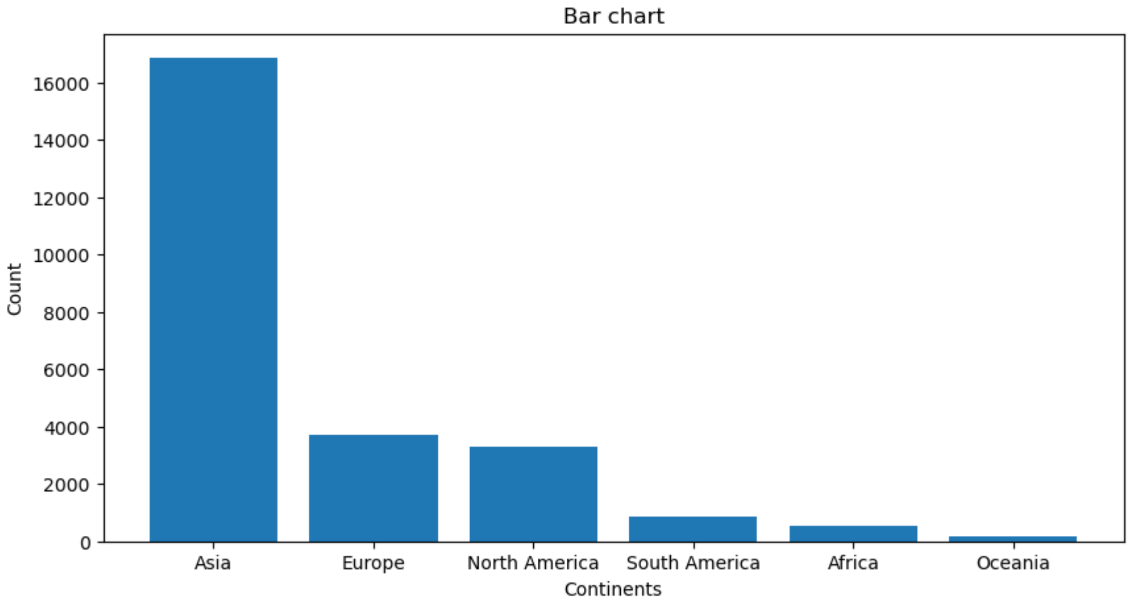


In [53]:
```
# for clear visulization of both axis

plt.figure(figsize=(10,5))
# 10= horizontal x
# 5= vertical  y

plt.bar('continetns','count',data=contint_data)
```

Out[53]:
```
<BarContainer object of 6 artists>
```



In [54]:
```
# provide titel

plt.figure(figsize=(10,5))
# 10= horizontal x
# 5= vertical  y

plt.bar('continetns','count',data=contint_data)
plt.title("Bar chart")
plt.xlabel("Continents")
plt.ylabel("Count")
plt.savefig("continents_bar.jpg")
plt.show()
```



- Reading a cat column
- unique
- nunique
- value counts
- frequency table (dataframe)
- barplot

In [ ]:
```
# value count is always in order so always take value count
```

Date-12-12-2023

In [ ]:
```
# Whenever you open notebook as fresh
# you need to run
# packages
# read the data
```

In [58]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [59]:

```
file_path="C:\\Users\\kurre\\OneDrive\\Documents\\Naresh IT\\datafiles\\Vis
visa_df=pd.read_csv(file_path)
```

bar plot using seaborn

In [ ]:

```
# in order to draw a bar chart we required frequency table
# continent column
# we created one more data frame
# haveing each lable frequncy
# asia   16k
# africa

# the above things are required, if you want to draw bar chart using matplo
```

In [ ]:

```
# saeborn will take directly the original column from original data frame
```

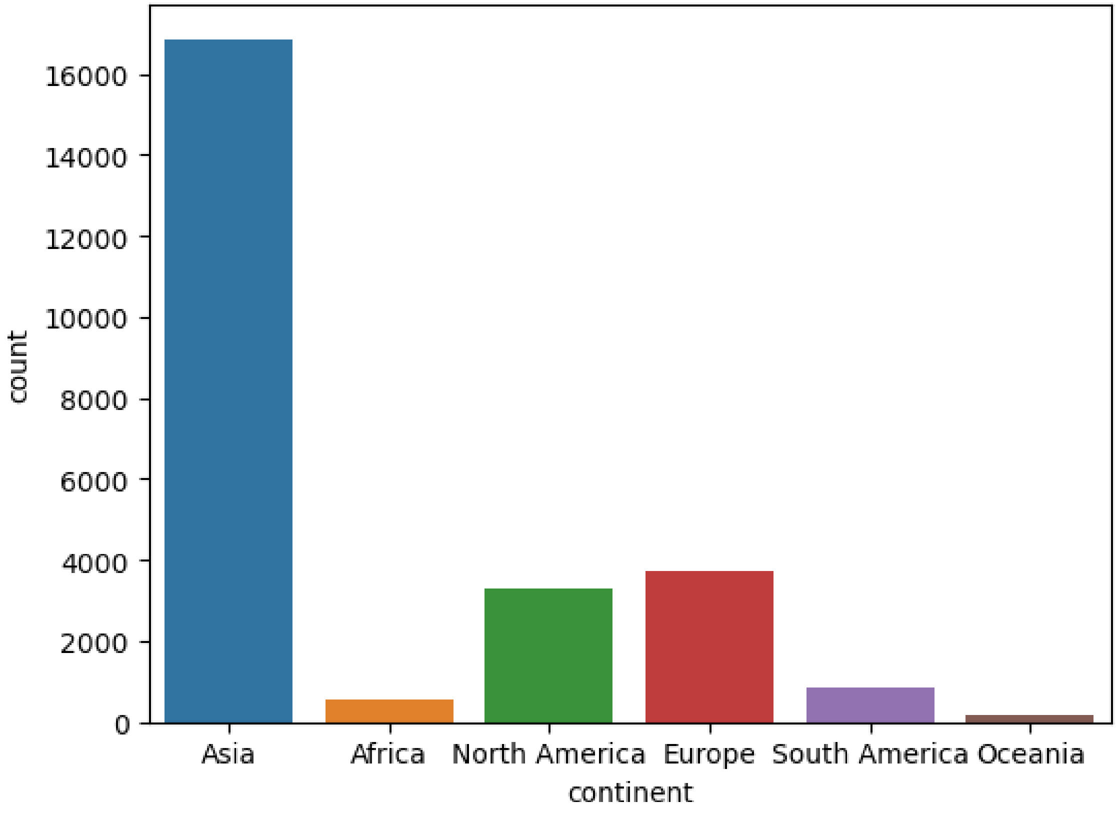
- seaborn requires 2 values
- data: original dataframe : visa_df
- x : original column name : 'continent'

In [60]:

```
import seaborn as sns
sns.countplot(data=visa_df,x='continent')
```

Out[60]:

```
<Axes: xlabel='continent', ylabel='count'>
```



- matplotlib bar chart requires 3 values
- x axis : categorical column
- y axis : numerical column
- data name

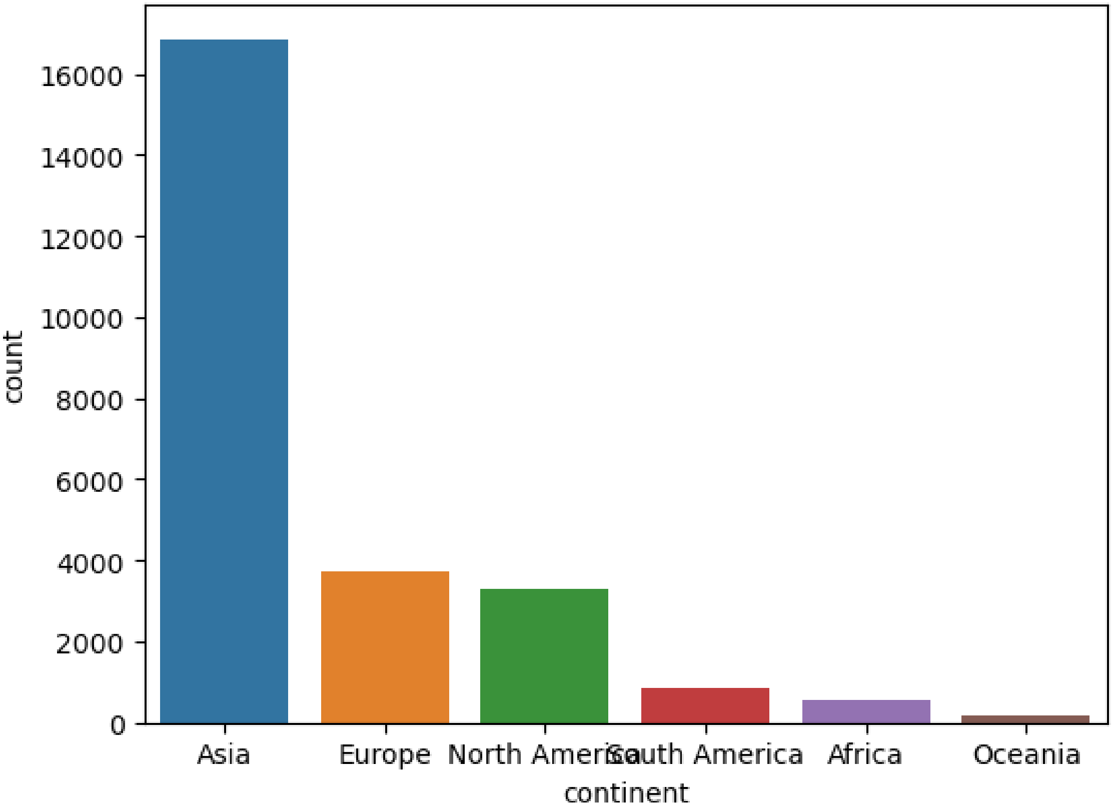
- from original data frame visadf , the original column is continent
- we created another table which has labels and its count

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

so we can not use mathplotlib

In [61]:
```
# make this as order = Method-1

import seaborn as sns
labels=['Asia','Europe','North America',
        'South America','Africa','Oceania']
sns.countplot(data=visa_df,
              x='continent',
              order=labels)
```

Out[61]:
```
<Axes: xlabel='continent', ylabel='count'>
```



In [64]:
```
visa_df['continent'].value_counts().keys()
```

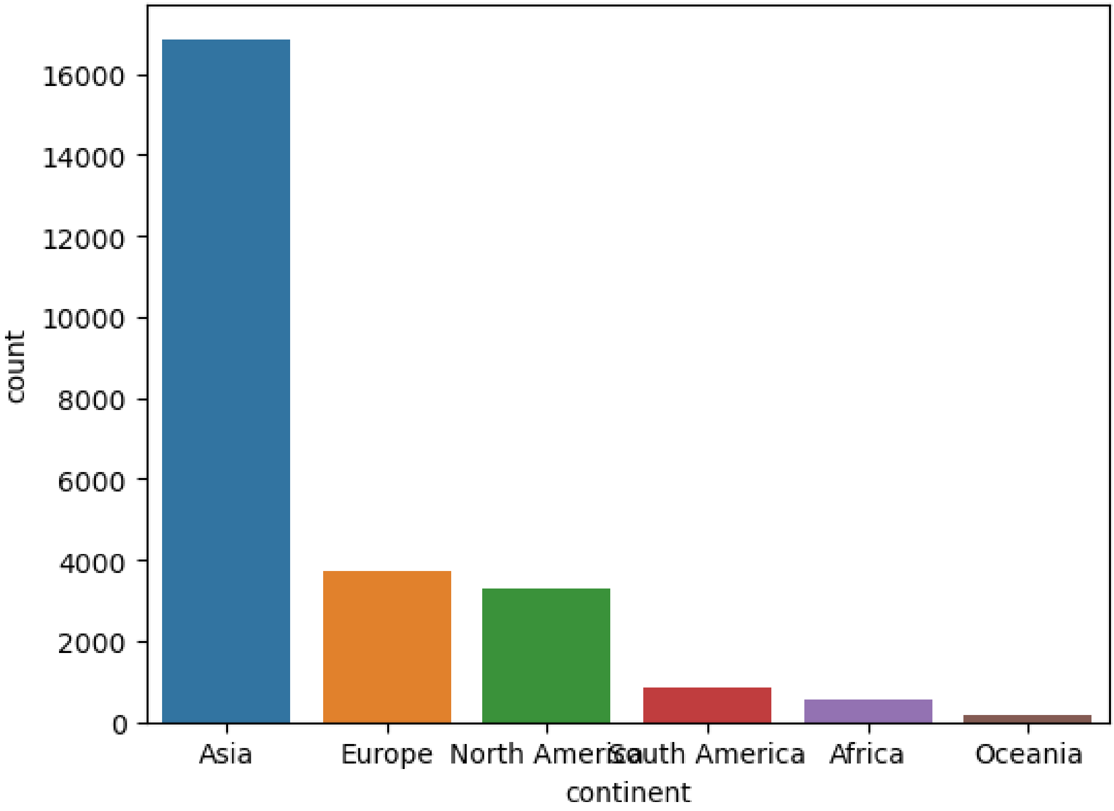
Out[64]:
```
Index(['Asia', 'Europe', 'North America', 'South America', 'Africa',
       'Oceania'],
      dtype='object', name='continent')
```

In [66]:

```
# method-2

import seaborn as sns
labels=visa_df['continent'].value_counts().keys()
sns.countplot(data=visa_df,
              x='continent',
              order=labels)
```

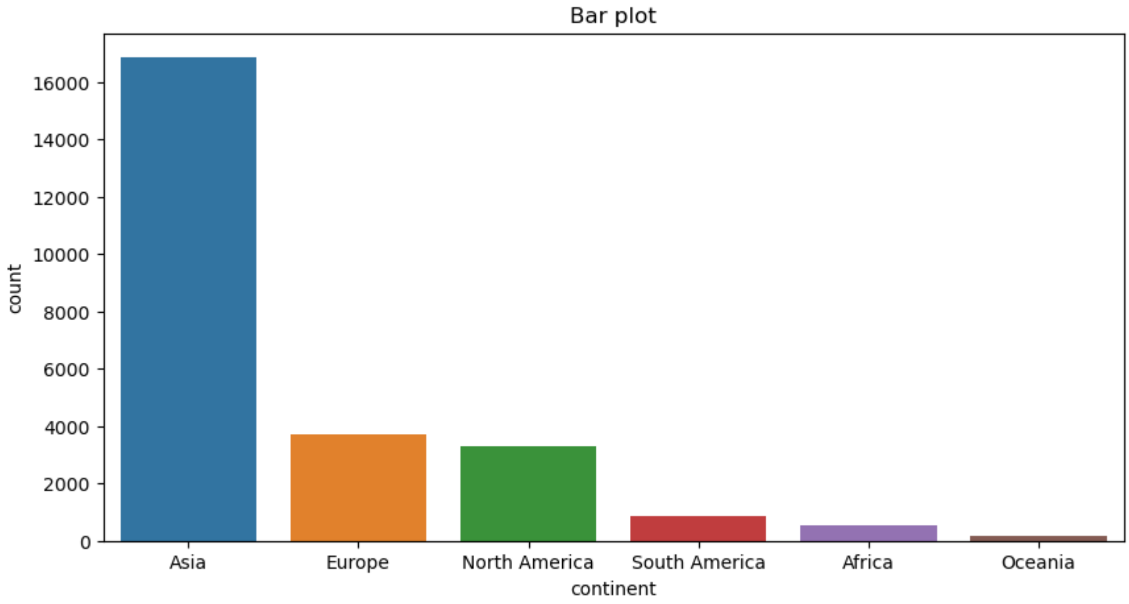
Out[66]:

```
<Axes: xlabel='continent', ylabel='count'>
```



In [67]:

```
# remove overlap x axis
# for titel


import seaborn as sns
labels=visa_df['continent'].value_counts().keys()
plt.figure(figsize=(10,5))
sns.countplot(data=visa_df,
              x='continent',
              order=labels)
plt.title("Bar plot")
plt.savefig("Continent_bar_seaborn")
plt.show()
```



In [ ]:

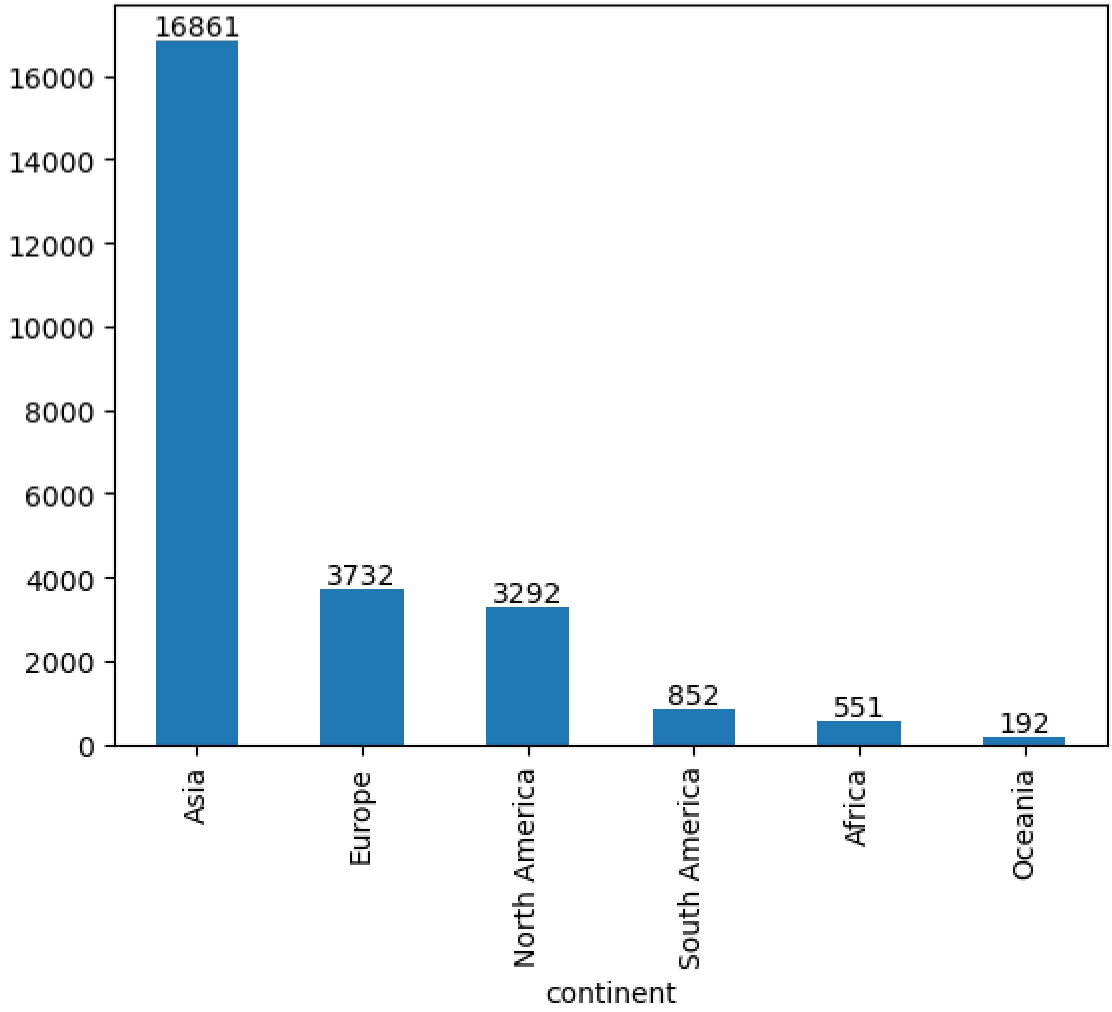
```
#Method-1: using matplotlib

################################## Reading the data #######################

file_path="C:\\Users\\omkar\\OneDrive\\Documents\\Data science\\Naresh IT\\

visa_df=pd.read_csv(file_path)
visa_df.head(2)


############################# Create a frequency table ####################

visa_df['continent'].value_counts()
continents=visa_df['continent'].value_counts().keys()
count=visa_df['continent'].value_counts().values
contint_data=pd.DataFrame(zip(continents,count),
                          columns=['continetns','count'])

contint_data

##################################### plot ################################
plt.figure(figsize=(10,5))
# 10= horizontal x
# 5= vertical  y
plt.bar('continetns','count',data=contint_data)
plt.title("Bar chart")
plt.xlabel("Continents")
plt.ylabel("Count")
plt.savefig("continents_bar.jpg")
plt.show()
```

In [ ]:

```
# Method-2: Seaborn

################################## Reading the data #######################

file_path="C:\\Users\\omkar\\OneDrive\\Documents\\Data science\\Naresh IT\\

visa_df=pd.read_csv(file_path)
visa_df.head(2)


######################################### plot ############################

import seaborn as sns
labels=visa_df['continent'].value_counts().keys()
plt.figure(figsize=(10,5))
sns.countplot(data=visa_df,
              x='continent',
              order=labels)
plt.title("Bar plot")
plt.savefig("Continent_bar_seaborn")
plt.show()
```

In [68]:

```
# Method-3: using value counts

count=visa_df['continent'].value_counts()
ax=count.plot(kind='bar')                    # ax= axis
ax.bar_label(ax.containers[0])
plt.show()
```



In [ ]:

pie-chart

In [69]:

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

Out[69]:

```
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 [70]:

```
#keys

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

Out[70]:

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

In [71]:

```
# values

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

Out[71]:

```
array([0.66173469, 0.14646782, 0.12919937, 0.03343799, 0.0216248 ,
       0.00753532])
```

In [73]:

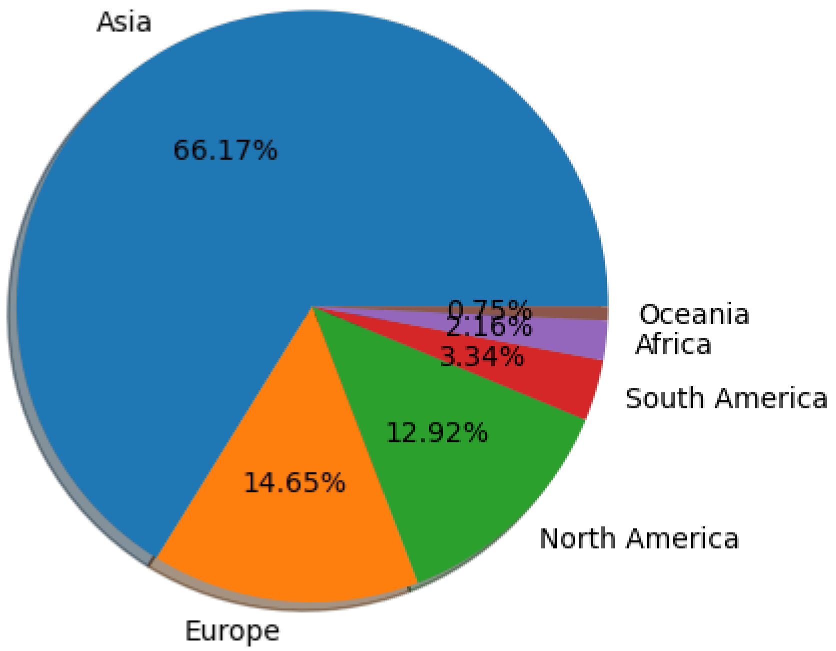
```
keys=visa_df['continent'].value_counts(normalize=True).keys()
values=visa_df['continent'].value_counts(normalize=True).values
pd.DataFrame(zip(keys,values),columns=['Continent','Relative frequency'])
```

Out[73]:

	Continent	Relative frequency
0	Asia	0.661735
1	Europe	0.146468
2	North America	0.129199
3	South America	0.033438
4	Africa	0.021625
5	Oceania	0.007535

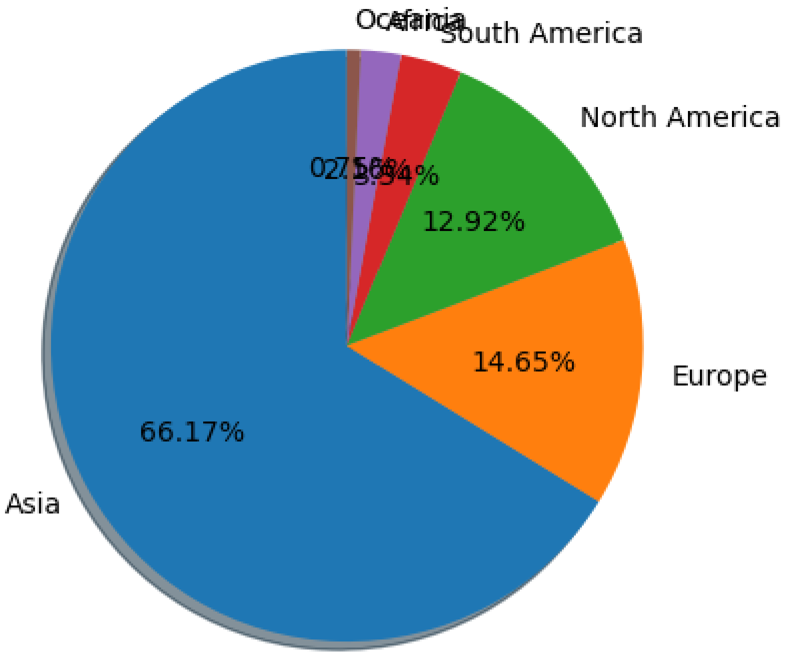
In [78]:

```
plt.pie(x=values,labels=keys,autopct="%0.2f%%",shadow=True)
plt.show()

# if you want two decimal after point than write 0.2
```



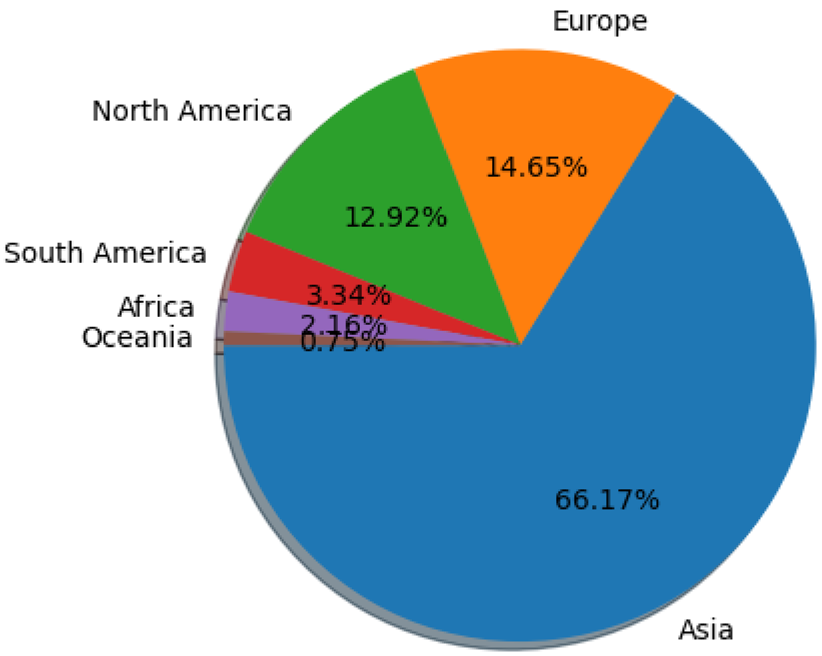
In [79]:

```
# rotete 90 d than using startangle

plt.pie(x=values,labels=keys,autopct="%0.2f%%",shadow=True,startangle=90)
plt.show()
```

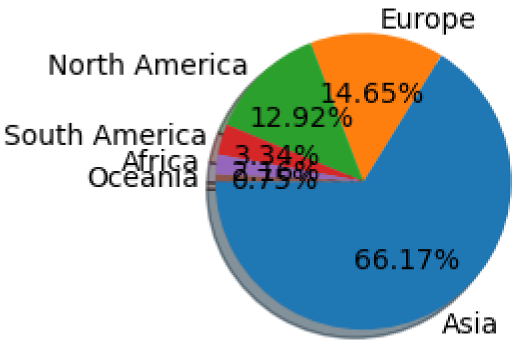


In [80]:

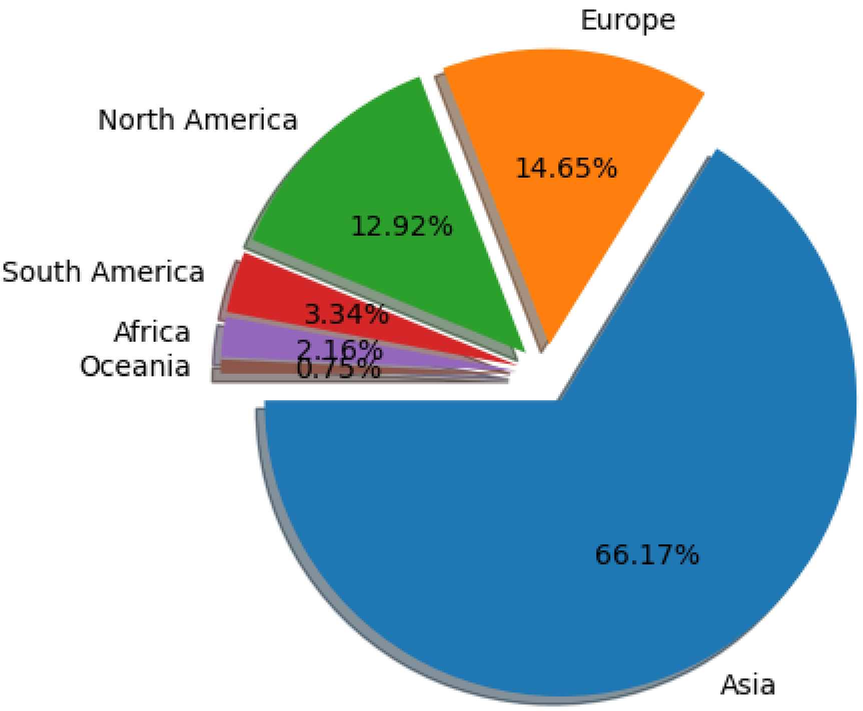
```
# 180 d

plt.pie(x=values,labels=keys,autopct="%0.2f%%",shadow=True,startangle=180)
plt.show()
```



In [81]:
```
# if you reduce the size


plt.pie(x=values,labels=keys,
        autopct="%0.2f%%",
        shadow=True,startangle=180,radius=0.5)  # defalut 1
plt.show()
```



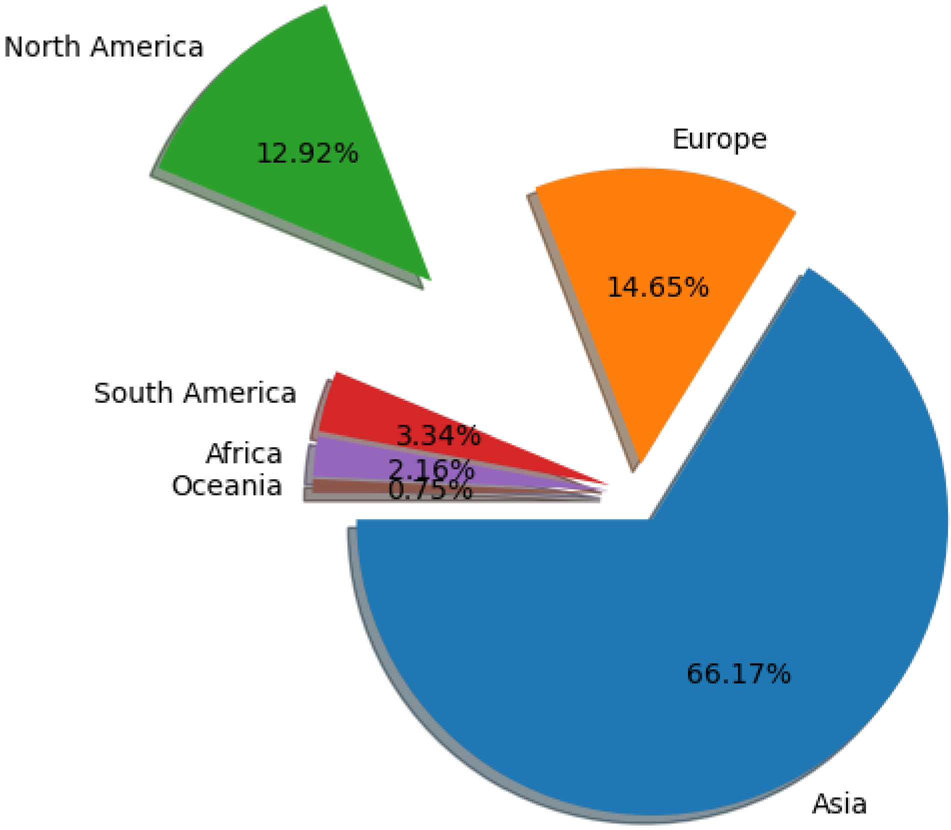
In [82]:
```
# how many pices

plt.pie(x=values,labels=keys,
        autopct="%0.2f%%",
        shadow=True,startangle=180
        ,radius=1,explode=[0.1,0.1,0.1,0.1,0.1,0.1])
plt.show()
```



In [84]:
```
keys
```

Out[84]:
```
Index(['Asia', 'Europe', 'North America', 'South America', 'Africa',
       'Oceania'],
      dtype='object', name='continent')
```

In [83]:
```
plt.pie(x=values,labels=keys,
        autopct="%0.2f%%",
        shadow=True,startangle=180
        ,radius=1,explode=[0.1,0.1,1,0.1,0.1,0.1])
plt.show()
```



In [87]:
```
data_types=dict(visa_df.dtypes)
cat=[i for i in data_types if data_types[i]=='O']
cat

# i want to use all data type so create a loop
```

Out[87]:
```
['case_id',
 'continent',
 'education_of_employee',
 'has_job_experience',
 'requires_job_training',
 'region_of_employment',
 'unit_of_wage',
 'full_time_position',
 'case_status']
```

In [89]:
```
for i in cat[1:]:
    visa_df[i].value_counts()
    value1=visa_df[i].value_counts().keys()
    value2=visa_df[i].value_counts().values
    data=pd.DataFrame(zip(value1,value2),
                                columns=[i,'count'])
    print(data)
```

```
       continent  count
0           Asia  16861
1         Europe   3732
2  North America   3292
3  South America    852
4         Africa    551
5        Oceania    192
  education_of_employee  count
0            Bachelor's  10234
1              Master's   9634
2           High School   3420
3             Doctorate   2192
  has_job_experience  count
0                  Y  14802
1                  N  10678
  requires_job_training  count
0                     N  22525
1                     Y   2955
  region_of_employment  count
0            Northeast   7195
1                South   7017
2                 West   6586
3              Midwest   4307
4               Island    375
  unit_of_wage  count
0         Year  22962
1         Hour   2157
2         Week    272
3        Month     89
  full_time_position  count
0                  Y  22773
1                  N   2707
  case_status  count
0   Certified  17018
1      Denied   8462
```

In [90]:
```
for i in cat[1:]:
    visa_df[i].value_counts()
    value1=visa_df[i].value_counts().keys()
    value2=visa_df[i].value_counts().values
    data=pd.DataFrame(zip(value1,value2),
                                columns=[i,'count'])

    data.to_csv('{}.csv'.format(i))

# this will save dataframes where python file existed
```

- follow this steps
- create a folder
- take the entire path

- add double slash at the end
- conactenate with your file name

In [94]:
```
file_path='C:\\Users\\kurre\\OneDrive\\Documents\\Naresh IT\\EDA_PYTHON\\da
```

In [95]:
```
file_path+'{}.csv'.format(i)

# This will save the data frames in a seperate folder

data_types=dict(visa_df.dtypes)
cat=[i for i in data_types if data_types[i]=='O']

for i in cat[1:]:
    visa_df[i].value_counts()
    value1=visa_df[i].value_counts().keys()
    value2=visa_df[i].value_counts().values
    data=pd.DataFrame(zip(value1,value2),
                                columns=[i,'count'])

    data.to_csv(file_path+'{}.csv'.format(i))
```

In [86]:
```
visa_df['continent'].value_counts()
continents=visa_df['continent'].value_counts().keys()
count=visa_df['continent'].value_counts().values
contint_data=pd.DataFrame(zip(continents,count),
                          columns=['continetns','count'])
contint_data
```

Out[86]:

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

In [91]:
```
# add 'case_ststus'

visa_df['case_status'].value_counts()
continents=visa_df['case_status'].value_counts().keys()
count=visa_df['case_status'].value_counts().values
contint_data=pd.DataFrame(zip(continents,count),
                          columns=['case_status','count'])

contint_data
```

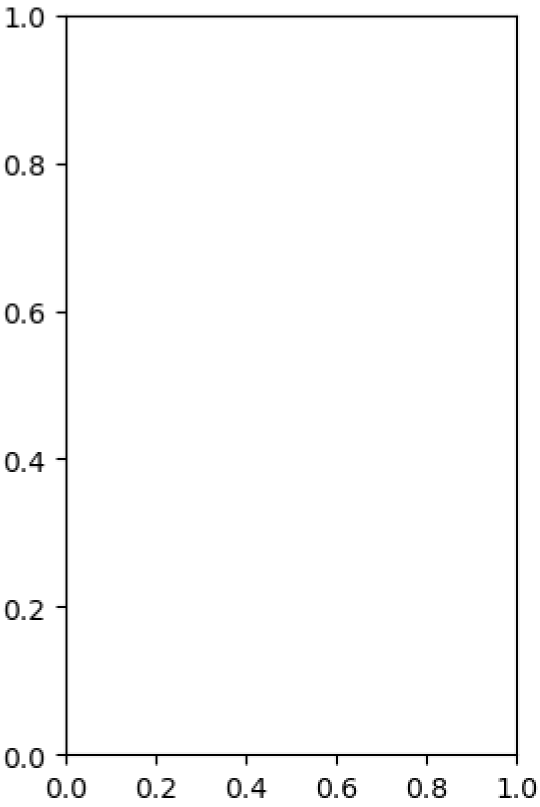
Out[91]:

	case_status	count
0	Certified	17018
1	Denied	8462

In [ ]:
```
# subplots
```

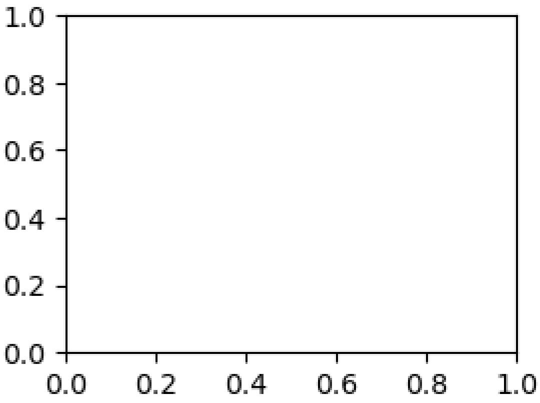
In [97]:
```
plt.subplot(1, 2, 2)
```

Out[97]:
```
<Axes: >
```



In [98]:
```
plt.subplot(2, 2, 2)
```

Out[98]:
```
<Axes: >
```

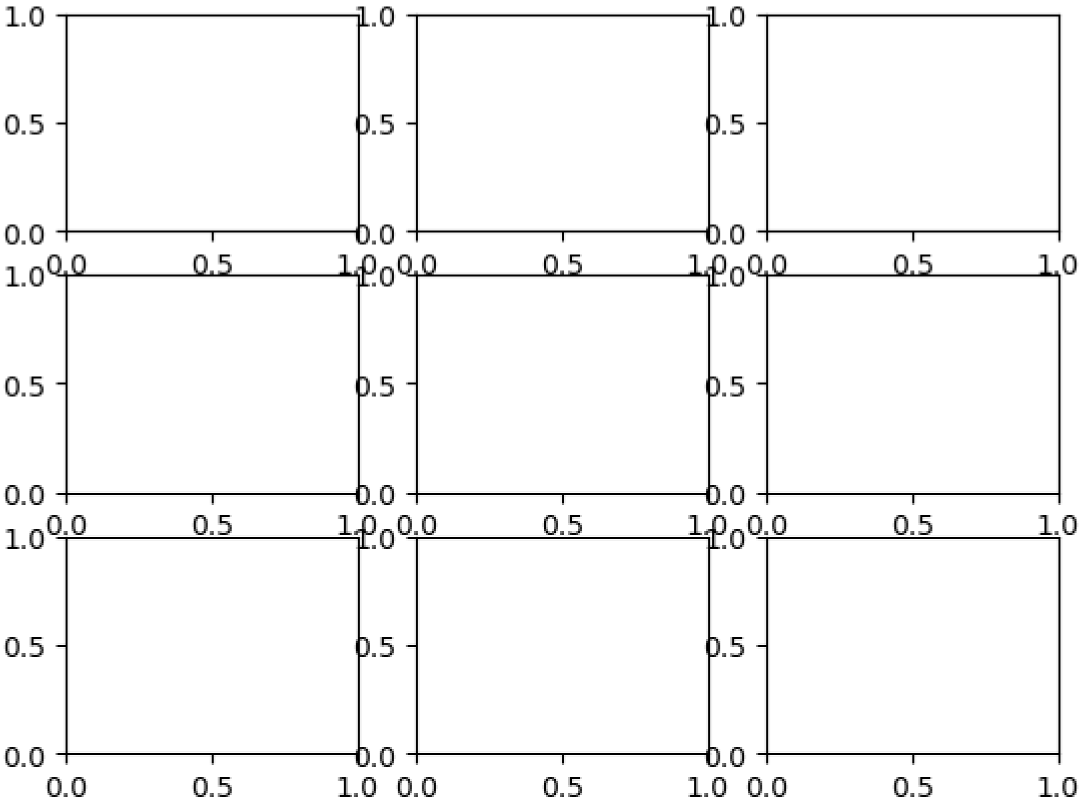


In [103]:

```
plt.subplot(3,3,1)

plt.subplot(3,3,2)
plt.subplot(3,3,3)
plt.subplot(3,3,4)
plt.subplot(3,3,5)
plt.subplot(3,3,6)
plt.subplot(3,3,7)
plt.subplot(3,3,8)
plt.subplot(3,3,9)
```

Out[103]:

```
<Axes: >
```



In [ ]:

In [ ]:

```
#list comprehention structure

#[<output> <forloop>]   =========> for loop condition

#[<o/p> <forloop> <if_condition>]  ==========> for loop and if condition bo


#[<if_output> <if_condition> else <else_ouput> <for loop>] ========> if els
```

In [ ]:

In [ ]:

In [ ]:

In [ ]: