In [1]:

```
import pandas as pd
import numpy as np
```

In [2]:

```
%pwd
```

Out[2]:

```
'C:\\Users\\HSA'
```

In [3]:

```
country=pd.read_csv(r'C:\Users\HSA\Country.csv')
notes=pd.read_csv(r'C:\Users\HSA\CountryNotes.csv')
indicators=pd.read_csv(r'C:\Users\HSA\Indicators.csv')
series=pd.read_csv(r'C:\Users\HSA\Series.csv')
```

In [4]:

```
merge1=pd.merge(country, indicators,how='outer', left_on='CountryCode',right_on='Country
Code')
```

In [5]:

```
merge1.shape
```

Out[5]:

```
(5656458, 36)
```

In [6]:

```
merge1.dtypes
```

Out[6]:

```
CountryCode                                   object
ShortName                                     object
TableName                                     object
LongName                                      object
Alpha2Code                                    object
CurrencyUnit                                  object
SpecialNotes                                  object
Region                                        object
IncomeGroup                                   object
Wb2Code                                       object
NationalAccountsBaseYear                      object
NationalAccountsReferenceYear                 object
SnaPriceValuation                             object
LendingCategory                               object
OtherGroups                                   object
SystemOfNationalAccounts                      object
AlternativeConversionFactor                   object
PppSurveyYear                                 object
BalanceOfPaymentsManualInUse                  object
ExternalDebtReportingStatus                   object
SystemOfTrade                                 object
GovernmentAccountingConcept                   object
ImfDataDisseminationStandard                  object
LatestPopulationCensus                        object
LatestHouseholdSurvey                         object
SourceOfMostRecentIncomeAndExpenditureData    object
VitalRegistrationComplete                     object
LatestAgriculturalCensus                      object
LatestIndustrialData                         float64
LatestTradeData                              float64
LatestWaterWithdrawalData                    float64
```

```
CountryName                                    object
IndicatorName                                  object
IndicatorCode                                  object
Year                                            int64
Value                                         float64
dtype: object
```

In [7]:

```
#1.Checking for NULL Values
#2.Checking for Header Errors
#3.Checking for Incorrect datatype
```

In [8]:

```
merge1.isnull().sum()
```

Out[8]:

```
CountryCode                                       0
ShortName                                         0
TableName                                         0
LongName                                          0
Alpha2Code                                    28108
CurrencyUnit                                 730124
SpecialNotes                                1761144
Region                                       730124
IncomeGroup                                  730124
Wb2Code                                       18518
NationalAccountsBaseYear                     757415
NationalAccountsReferenceYear               4454857
SnaPriceValuation                            791642
LendingCategory                             1856100
OtherGroups                                 4067260
SystemOfNationalAccounts                     730124
AlternativeConversionFactor                 4442953
PppSurveyYear                                962097
BalanceOfPaymentsManualInUse                 984659
ExternalDebtReportingStatus                 2154174
SystemOfTrade                                823746
GovernmentAccountingConcept                 1429354
ImfDataDisseminationStandard                 989644
LatestPopulationCensus                       730696
LatestHouseholdSurvey                       1822872
SourceOfMostRecentIncomeAndExpenditureData  1471283
VitalRegistrationComplete                   3395425
LatestAgriculturalCensus                    1993993
LatestIndustrialData                        2697671
LatestTradeData                             1013327
LatestWaterWithdrawalData                   1012343
CountryName                                       0
IndicatorName                                     0
IndicatorCode                                     0
Year                                              0
Value                                             0
dtype: int64
```

In [9]:

```
merge1.columns
```

Out[9]:

```
Index(['CountryCode', 'ShortName', 'TableName', 'LongName', 'Alpha2Code',
       'CurrencyUnit', 'SpecialNotes', 'Region', 'IncomeGroup', 'Wb2Code',
       'NationalAccountsBaseYear', 'NationalAccountsReferenceYear',
       'SnaPriceValuation', 'LendingCategory', 'OtherGroups',
       'SystemOfNationalAccounts', 'AlternativeConversionFactor',
       'PppSurveyYear', 'BalanceOfPaymentsManualInUse',
       'ExternalDebtReportingStatus', 'SystemOfTrade',
       'GovernmentAccountingConcept', 'ImfDataDisseminationStandard',
       'LatestPopulationCensus', 'LatestHouseholdSurvey',
```

```
       'SourceOfMostRecentIncomeAndExpenditureData',
       'VitalRegistrationComplete', 'LatestAgriculturalCensus',
       'LatestIndustrialData', 'LatestTradeData', 'LatestWaterWithdrawalData',
       'CountryName', 'IndicatorName', 'IndicatorCode', 'Year', 'Value'],
      dtype='object')
```

In [10]:

```
merge1.dtypes
```

Out[10]:

```
CountryCode                                 object
ShortName                                   object
TableName                                   object
LongName                                    object
Alpha2Code                                  object
CurrencyUnit                                object
SpecialNotes                                object
Region                                      object
IncomeGroup                                 object
Wb2Code                                     object
NationalAccountsBaseYear                    object
NationalAccountsReferenceYear               object
SnaPriceValuation                           object
LendingCategory                             object
OtherGroups                                 object
SystemOfNationalAccounts                    object
AlternativeConversionFactor                 object
PppSurveyYear                               object
BalanceOfPaymentsManualInUse                object
ExternalDebtReportingStatus                 object
SystemOfTrade                               object
GovernmentAccountingConcept                 object
ImfDataDisseminationStandard                object
LatestPopulationCensus                      object
LatestHouseholdSurvey                       object
SourceOfMostRecentIncomeAndExpenditureData  object
VitalRegistrationComplete                   object
LatestAgriculturalCensus                    object
LatestIndustrialData                       float64
LatestTradeData                            float64
LatestWaterWithdrawalData                  float64
CountryName                                 object
IndicatorName                               object
IndicatorCode                               object
Year                                         int64
Value                                      float64
dtype: object
```

4. Funtion for Checking Errors in Dataframe

In [11]:

```
def nullvalues(x):
    return sum(x.isnull())
```

In [12]:

```
merge1.apply(nullvalues, axis=0)
```

Out[12]:

```
CountryCode                                     0
ShortName                                       0
TableName                                       0
LongName                                        0
Alpha2Code                                  28108
CurrencyUnit                               730124
SpecialNotes                              1761144
Region                                     730124
```

```
IncomeGroup                                730124
Wb2Code                                     18518
NationalAccountsBaseYear                   757415
NationalAccountsReferenceYear             4454857
SnaPriceValuation                          791642
LendingCategory                           1856100
OtherGroups                               4067260
SystemOfNationalAccounts                   730124
AlternativeConversionFactor               4442953
PppSurveyYear                              962097
BalanceOfPaymentsManualInUse               984659
ExternalDebtReportingStatus               2154174
SystemOfTrade                              823746
GovernmentAccountingConcept               1429354
ImfDataDisseminationStandard               989644
LatestPopulationCensus                     730696
LatestHouseholdSurvey                     1822872
SourceOfMostRecentIncomeAndExpenditureData 1471283
VitalRegistrationComplete                 3395425
LatestAgriculturalCensus                  1993993
LatestIndustrialData                      2697671
LatestTradeData                           1013327
LatestWaterWithdrawalData                 1012343
CountryName                                     0
IndicatorName                                   0
IndicatorCode                                   0
Year                                            0
Value                                           0
dtype: int64
```

In [13]:

```
def datatype(x):
    return (x.dtype)
```

In [14]:

```
merge1.apply(datatype, axis=0)
```

Out[14]:

```
CountryCode                                object
ShortName                                  object
TableName                                  object
LongName                                   object
Alpha2Code                                 object
CurrencyUnit                               object
SpecialNotes                               object
Region                                     object
IncomeGroup                                object
Wb2Code                                    object
NationalAccountsBaseYear                   object
NationalAccountsReferenceYear              object
SnaPriceValuation                          object
LendingCategory                            object
OtherGroups                                object
SystemOfNationalAccounts                   object
AlternativeConversionFactor                object
PppSurveyYear                              object
BalanceOfPaymentsManualInUse               object
ExternalDebtReportingStatus                object
SystemOfTrade                              object
GovernmentAccountingConcept                object
ImfDataDisseminationStandard               object
LatestPopulationCensus                     object
LatestHouseholdSurvey                      object
SourceOfMostRecentIncomeAndExpenditureData object
VitalRegistrationComplete                  object
LatestAgriculturalCensus                   object
LatestIndustrialData                       object
LatestTradeData                            object
LatestWaterWithdrawalData                  object
```

```
CountryName                                    object
IndicatorName                                  object
IndicatorCode                                  object
Year                                           object
Value                                          object
dtype: object
```

In [15]:

```
def headererrors(x):
     return list(x.columns)
```

In [16]:

```
headererrors(merge1)
```

Out[16]:

```
['CountryCode',
 'ShortName',
 'TableName',
 'LongName',
 'Alpha2Code',
 'CurrencyUnit',
 'SpecialNotes',
 'Region',
 'IncomeGroup',
 'Wb2Code',
 'NationalAccountsBaseYear',
 'NationalAccountsReferenceYear',
 'SnaPriceValuation',
 'LendingCategory',
 'OtherGroups',
 'SystemOfNationalAccounts',
 'AlternativeConversionFactor',
 'PppSurveyYear',
 'BalanceOfPaymentsManualInUse',
 'ExternalDebtReportingStatus',
 'SystemOfTrade',
 'GovernmentAccountingConcept',
 'ImfDataDisseminationStandard',
 'LatestPopulationCensus',
 'LatestHouseholdSurvey',
 'SourceOfMostRecentIncomeAndExpenditureData',
 'VitalRegistrationComplete',
 'LatestAgriculturalCensus',
 'LatestIndustrialData',
 'LatestTradeData',
 'LatestWaterWithdrawalData',
 'CountryName',
 'IndicatorName',
 'IndicatorCode',
 'Year',
 'Value']
```

Combining all 3 above functions

In [17]:

```
def check4errors(x):
    return print('Checking for Null Values\n',x.isnull().any(),
                 '\n\nChecking for Datatype Erros\n',
                 x.dtypes,'\n\nChecking for Header errors\n',
                 list(x.columns))
```

In [18]:

```
check4errors(merge1)
```

```
Checking for Null Values
```

```
 CountryCode                                 False
ShortName                                   False
TableName                                   False
LongName                                    False
Alpha2Code                                   True
CurrencyUnit                                 True
SpecialNotes                                 True
Region                                       True
IncomeGroup                                  True
Wb2Code                                      True
NationalAccountsBaseYear                     True
NationalAccountsReferenceYear                True
SnaPriceValuation                            True
LendingCategory                              True
OtherGroups                                  True
SystemOfNationalAccounts                     True
AlternativeConversionFactor                  True
PppSurveyYear                                True
BalanceOfPaymentsManualInUse                 True
ExternalDebtReportingStatus                  True
SystemOfTrade                                True
GovernmentAccountingConcept                  True
ImfDataDisseminationStandard                 True
LatestPopulationCensus                       True
LatestHouseholdSurvey                        True
SourceOfMostRecentIncomeAndExpenditureData   True
VitalRegistrationComplete                    True
LatestAgriculturalCensus                     True
LatestIndustrialData                         True
LatestTradeData                              True
LatestWaterWithdrawalData                    True
CountryName                                 False
IndicatorName                               False
IndicatorCode                               False
Year                                        False
Value                                       False
dtype: bool

Checking for Datatype Erros
 CountryCode                                 object
ShortName                                   object
TableName                                   object
LongName                                    object
Alpha2Code                                  object
CurrencyUnit                                object
SpecialNotes                                object
Region                                      object
IncomeGroup                                 object
Wb2Code                                     object
NationalAccountsBaseYear                    object
NationalAccountsReferenceYear               object
SnaPriceValuation                           object
LendingCategory                             object
OtherGroups                                 object
SystemOfNationalAccounts                    object
AlternativeConversionFactor                 object
PppSurveyYear                               object
BalanceOfPaymentsManualInUse                object
ExternalDebtReportingStatus                 object
SystemOfTrade                               object
GovernmentAccountingConcept                 object
ImfDataDisseminationStandard                object
LatestPopulationCensus                      object
LatestHouseholdSurvey                       object
SourceOfMostRecentIncomeAndExpenditureData  object
VitalRegistrationComplete                   object
LatestAgriculturalCensus                    object
LatestIndustrialData                       float64
LatestTradeData                            float64
LatestWaterWithdrawalData                  float64
CountryName                                 object
IndicatorName                               object
```

```
IndicatorCode                                    object
Year                                              int64
Value                                           float64
dtype: object

Checking for Header errors
 ['CountryCode', 'ShortName', 'TableName', 'LongName', 'Alpha2Code', 'CurrencyUnit', 'Spe
cialNotes', 'Region', 'IncomeGroup', 'Wb2Code', 'NationalAccountsBaseYear', 'NationalAcco
untsReferenceYear', 'SnaPriceValuation', 'LendingCategory', 'OtherGroups', 'SystemOfNatio
nalAccounts', 'AlternativeConversionFactor', 'PppSurveyYear', 'BalanceOfPaymentsManualInU
se', 'ExternalDebtReportingStatus', 'SystemOfTrade', 'GovernmentAccountingConcept', 'ImfD
ataDisseminationStandard', 'LatestPopulationCensus', 'LatestHouseholdSurvey', 'SourceOfMo
stRecentIncomeAndExpenditureData', 'VitalRegistrationComplete', 'LatestAgriculturalCensus
', 'LatestIndustrialData', 'LatestTradeData', 'LatestWaterWithdrawalData', 'CountryName',
'IndicatorName', 'IndicatorCode', 'Year', 'Value']
```

In [19]:

```
pwd
```

Out[19]:

```
'C:\\Users\\HSA'
```

Subsetting data based on criteria

In [20]:

```
subset=merge1[merge1['ShortName'].isin(['China','Denmark','Finland','Italy'])]
```

In [21]:

```
subset['ShortName'].unique()
```

Out[21]:

```
array(['China', 'Denmark', 'Finland', 'Italy'], dtype=object)
```

New Index

In [22]:

```
subset.reset_index(inplace=True)
```

2nd merge resulatant DF with series

In [23]:

```
Merge2= pd.merge(subset,series[['SeriesCode','Topic','AggregationMethod','LimitationsAnd
Exceptions','UnitOfMeasure']], how = 'outer', left_on ='IndicatorCode', right_on='SeriesC
ode')
```

In [24]:

```
Merge2.head()
```

Out[24]:

	index	CountryCode	ShortName	TableName	LongName	Alpha2Code	CurrencyUnit	SpecialNotes	Region	IncomeGrou
0	930603.0	CHN	China	China	People's Republic of China	CN	Chinese yuan	On 1 July 1997 China resumed its exercise of s...	East Asia & Pacific	Upper midd incon
								On 1 July		

	index	CountryCode	ShortName	TableName	LongName	Alpha2Code	CurrencyUnit	SpecialNotes	Region	IncomeGroup
1	930723.0	CHN	China	China	People's Republic of China	CN	Chinese yuan	1997 China resumed its exercise of s...	East Asia & Pacific	Upper middle incon
2	930884.0	CHN	China	China	People's Republic of China	CN	Chinese yuan	On 1 July 1997 China resumed its exercise of s...	East Asia & Pacific	Upper middle incon
3	931029.0	CHN	China	China	People's Republic of China	CN	Chinese yuan	On 1 July 1997 China resumed its exercise of s...	East Asia & Pacific	Upper middle incon
4	931170.0	CHN	China	China	People's Republic of China	CN	Chinese yuan	On 1 July 1997 China resumed its exercise of s...	East Asia & Pacific	Upper middle incon

5 rows × 42 columns

In [25]:

```
Merge2.columns
```

Out[25]:

```
Index(['index', 'CountryCode', 'ShortName', 'TableName', 'LongName',
       'Alpha2Code', 'CurrencyUnit', 'SpecialNotes', 'Region', 'IncomeGroup',
       'Wb2Code', 'NationalAccountsBaseYear', 'NationalAccountsReferenceYear',
       'SnaPriceValuation', 'LendingCategory', 'OtherGroups',
       'SystemOfNationalAccounts', 'AlternativeConversionFactor',
       'PppSurveyYear', 'BalanceOfPaymentsManualInUse',
       'ExternalDebtReportingStatus', 'SystemOfTrade',
       'GovernmentAccountingConcept', 'ImfDataDisseminationStandard',
       'LatestPopulationCensus', 'LatestHouseholdSurvey',
       'SourceOfMostRecentIncomeAndExpenditureData',
       'VitalRegistrationComplete', 'LatestAgriculturalCensus',
       'LatestIndustrialData', 'LatestTradeData', 'LatestWaterWithdrawalData',
       'CountryName', 'IndicatorName', 'IndicatorCode', 'Year', 'Value',
       'SeriesCode', 'Topic', 'AggregationMethod', 'LimitationsAndExceptions',
       'UnitOfMeasure'],
      dtype='object')
```

Subsetting based on Column Names

In [26]:

```
Subset3=Merge2[['CountryCode', 'SeriesCode','ShortName','LongName','Region','SystemOfTrade','Value','Year','LimitationsAndExceptions','UnitOfMeasure']]
```

In [27]:

```
Subset3.shape
```

Out[27]:

```
(107573, 10)
```

Filtering DataFrame based on Years (2014-2015)

In [28]:

```
Subsetyear=Subset3[(Subset3.Year >=2014)]
```

In [29]:

```
Subsetyear.Year.unique()
```

Out[29]:

```
array([2014., 2015.])
```

In [30]:

```
Subsetyear.shape
```

Out[30]:

```
(2130, 10)
```

In [31]:

```
Subsetyear.reset_index(inplace=True)
```

Merging with Country_notes.csv

In [32]:

```
notes.columns
```

Out[32]:

```
Index(['Countrycode', 'Seriescode', 'Description'], dtype='object')
```

In [33]:

```
Merge3=Subset3.merge(notes, how='inner',left_on='SeriesCode',right_on='Seriescode')
```

In [34]:

```
Merge3.shape
```

Out[34]:

```
(697416, 13)
```

In [35]:

```
Merge3.columns
```

Out[35]:

```
Index(['CountryCode', 'SeriesCode', 'ShortName', 'LongName', 'Region',
       'SystemOfTrade', 'Value', 'Year', 'LimitationsAndExceptions',
       'UnitOfMeasure', 'Countrycode', 'Seriescode', 'Description'],
      dtype='object')
```

In [36]:

```
Merge3.shape
```

Out[36]:

```
(697416, 13)
```

In [37]:

```
Merge3=Merge3.drop(labels=['Countrycode','Seriescode'], axis=1)
```

In [38]:

```
Merge3.columns
```

Out[38]:

```
Index(['CountryCode', 'SeriesCode', 'ShortName', 'LongName', 'Region',
       'SystemOfTrade', 'Value', 'Year', 'LimitationsAndExceptions',
```

```
       'UnitOfMeasure', 'Description'],
      dtype='object')
```

In [39]:

```
Merge3.drop(labels=['LimitationsAndExceptions', 'UnitOfMeasure'], axis=1, inplace=True)
Merge3.shape
```

Out[39]:

```
(697416, 9)
```

In [46]:

```
Merge3.isnull().sum()
```

Out[46]:

```
CountryCode       3
SeriesCode        0
ShortName         3
LongName          3
Region            3
SystemOfTrade     3
Value             3
Year              3
Description       0
dtype: int64
```

12. Code to read the File as CSV

In [55]:

```
def csvfilefunc():
    print("Hello please follow the instructions below \nif yes then type y otherwise n "
)
    answer = input("Your option : ").upper()
    if answer == "Y":
        csvfile= input("File name : ").lower()
        value = "%s.csv " % (csvfile)
        Merge3.to_csv(value)
        print(value)

    else:
          print("Good Bye")
```

In [57]:

```
csvfilefunc()
```

```
Hello please follow the instructions below
if yes then type y otherwise n
Your option : y
File name : Merge3
merge3.csv
```

13 Graph Plotting

In [58]:

```
import matplotlib.pyplot as plt
%matplotlib inline
```

In [59]:

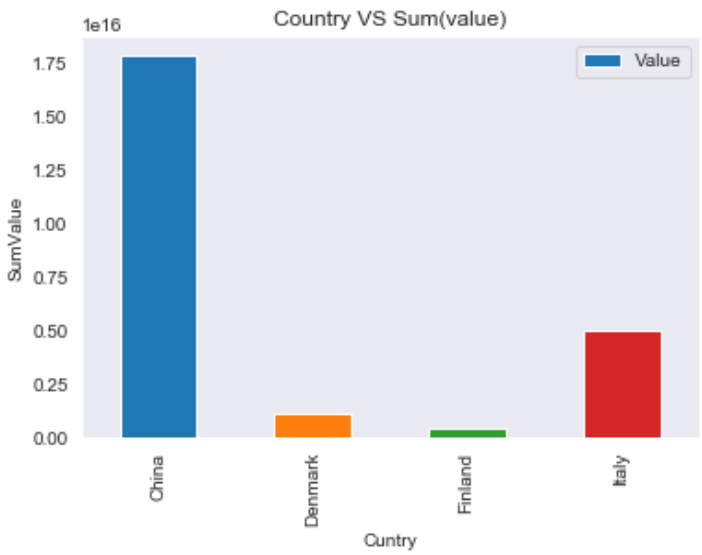
```
import seaborn as sns
sns.set_style('dark')
graphdata=Merge3.groupby(['ShortName'])['Value'].sum()
```

In [60]:

```
Merge3.groupby('ShortName').Value.sum().plot(kind = "bar")
plt.xlabel("Country")
plt.ylabel("SumValue")
plt.legend(loc='best')
plt.title("Country VS Sum(value)")
```

Out[60]:

```
Text(0.5, 1.0, 'Country VS Sum(value)')
```



In [ ]: