EXPLOLATORY DATA ANALYSIS OF INDIAS TRADE TRENDS FROM 2008-2018

November 13, 2019

```
In [46]: #STC PROJECT: WE TRY TO UNDERSTAND HOW INDIA TRADES USING DATA MANIPULATION AND VISUA
         # BY SHREYAS K 2017120056, VINAYAK JADHAV 2017120020, SOHAM PINGE 2017120047
```

```
In [47]: import numpy as np # linear algebra
         import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
         import matplotlib.pyplot as plt
         import seaborn as sns
         from pandas import DataFrame
```

```
In [48]: import_df=pd.read_csv(r"C:\Users\kailash\Desktop\import.csv")#importing the csv file
```

```
In [49]: export_df=pd.read_csv(r"C:\Users\kailash\Desktop\export.csv")#importing the csv file
```

```
In [50]: from statsmodels.graphics.tsaplots import plot_acf #Auto-Correlation Plots
         from statsmodels.graphics.tsaplots import plot_pacf #Partial-Auto Correlation Plots
```

```
In [51]: import_df.isnull().sum()# calculating the number of nulls
```

```
Out[51]: HSCode           0
         Commodity        0
         value        14027
         country          0
         year             0
         dtype: int64
```

```
In [52]: export_df = export_df.dropna()
         export_df = export_df.reset_index(drop=True)#drop the null values
```

```
In [53]: import_df = import_df.dropna()
         import_df = import_df.reset_index(drop=True)
```

```
In [54]: import_df.head(10)
```

```
Out[54]:    HSCode                                          Commodity   value  \
         0       5  PRODUCTS OF ANIMAL ORIGIN, NOT ELSEWHERE SPECI...    0.00
         1       7    EDIBLE VEGETABLES AND CERTAIN ROOTS AND TUBERS.   12.38
         2       8  EDIBLE FRUIT AND NUTS; PEEL OR CITRUS FRUIT OR...  268.60
         3       9                       COFFEE, TEA, MATE AND SPICES.   35.48
```

```
4      12  OIL SEEDS AND OLEA. FRUITS; MISC. GRAINS, SEED...    8.32
5      13  LAC; GUMS, RESINS AND OTHER VEGETABLE SAPS AND...  108.78
6      20  PREPARATIONS OF VEGETABLES, FRUIT, NUTS OR OTH...    0.65
7      25  SALT; SULPHUR; EARTHS AND STONE; PLASTERING MA...    0.05
8      27  MINERAL FUELS, MINERAL OILS AND PRODUCTS OF TH...    0.00
9      41  RAW HIDES AND SKINS (OTHER THAN FURSKINS) AND ...    0.00

           country  year
0  AFGHANISTAN TIS  2018
1  AFGHANISTAN TIS  2018
2  AFGHANISTAN TIS  2018
3  AFGHANISTAN TIS  2018
4  AFGHANISTAN TIS  2018
5  AFGHANISTAN TIS  2018
6  AFGHANISTAN TIS  2018
7  AFGHANISTAN TIS  2018
8  AFGHANISTAN TIS  2018
9  AFGHANISTAN TIS  2018
```

```
In [55]: export_df.head(10)
```

```
Out[55]:    HSCode                                          Commodity  value  \
         0       2                          MEAT AND EDIBLE MEAT OFFAL.   0.18
         1       3  FISH AND CRUSTACEANS, MOLLUSCS AND OTHER AQUAT...   0.00
         2       4  DAIRY PRODUCE; BIRDS' EGGS; NATURAL HONEY; EDI...  12.48
         3       6  LIVE TREES AND OTHER PLANTS; BULBS; ROOTS AND ...   0.00
         4       7    EDIBLE VEGETABLES AND CERTAIN ROOTS AND TUBERS.   1.89
         5       8  EDIBLE FRUIT AND NUTS; PEEL OR CITRUS FRUIT OR...  25.01
         6       9                        COFFEE, TEA, MATE AND SPICES.  13.75
         7      10                                           CEREALS.   0.75
         8      11  PRODUCTS OF THE MILLING INDUSTRY; MALT; STARCH...   0.01
         9      12  OIL SEEDS AND OLEA. FRUITS; MISC. GRAINS, SEED...  10.02

                    country  year
         0  AFGHANISTAN TIS  2018
         1  AFGHANISTAN TIS  2018
         2  AFGHANISTAN TIS  2018
         3  AFGHANISTAN TIS  2018
         4  AFGHANISTAN TIS  2018
         5  AFGHANISTAN TIS  2018
         6  AFGHANISTAN TIS  2018
         7  AFGHANISTAN TIS  2018
         8  AFGHANISTAN TIS  2018
         9  AFGHANISTAN TIS  2018
```

```
In [ ]: importing_countries=import_df[['country']].nunique()
        exporting_countries=export_df[['country']].nunique()# CODE TO FIND THE TOTAL NUMBER OF
        print("India imports from:",importing_countries,"countries")
        print("India exports to:",exporting_countries,"countries")
```

```
In [56]: import_group=import_df.groupby(['country','year']).agg({'value':'sum'})
         export_group=export_df.groupby(['country','year']).agg({'value':'sum'})
```

```
In [57]: import_group
```

```
Out[57]:                         value
         country         year
         AFGHANISTAN TIS 2010   146.01
                         2011   132.49
                         2012   159.53
                         2013   208.76
                         2014   261.92
                         2015   615.80
                         2016   292.90
                         2017   867.54
                         2018   870.88
         ALBANIA         2010     0.11
                         2011   141.13
                         2012    31.05
                         2013   198.79
                         2014    50.77
                         2015    34.36
                         2016     6.50
                         2017    77.70
                         2018    45.24
         ALGERIA         2010  1816.19
                         2011  2111.43
                         2012   683.54
                         2013   860.89
                         2014   551.81
                         2015   598.86
                         2016   605.11
                         2017  2521.92
                         2018  3393.88
         AMERI SAMOA     2010     0.15
                         2011     1.40
                         2012     0.78
         ...                       ...
         WALLIS F IS     2014     1.01
                         2015     0.24
                         2017     0.02
         YEMEN REPUBLC   2010  1743.89
                         2011   970.69
                         2012   958.92
                         2013   782.18
                         2014   540.70
                         2015     6.87
                         2016     4.81
```

```
                2017   306.10
                2018    23.50
ZAMBIA          2010    32.08
                2011   168.83
                2012   324.85
                2013   243.15
                2014   283.34
                2015   475.36
                2016   743.90
                2017  2189.96
                2018  1021.00
ZIMBABWE        2010    11.62
                2011     2.88
                2012    34.54
                2013    12.51
                2014    32.69
                2015    24.44
                2016    60.45
                2017   124.40
                2018    15.62

[2001 rows x 1 columns]
```

```
In [58]: export_group
```

Out[58]:

```
                        value
country         year
AFGHANISTAN TIS 2010   422.31
                2011   510.81
                2012   472.55
                2013   474.26
                2014   422.48
                2015   526.51
                2016   506.26
                2017   709.66
                2018   715.35
ALBANIA         2010    11.49
                2011    12.61
                2012    17.47
                2013    18.71
                2014    19.44
                2015    24.00
                2016    26.43
                2017    29.29
                2018    37.07
ALGERIA         2010   781.92
                2011   835.59
                2012  1088.67
                2013  1069.44
                2014  1063.65
                2015   787.74
                2016   841.81
                2017   823.18
                2018   940.24
AMERI SAMOA     2010     0.20
                2011     2.05
                2012     0.37
...                       ...
WALLIS F IS     2014     0.03
                2016     0.00
                2017     0.05
YEMEN REPUBLC   2010   513.98
                2011   730.53
                2012  1477.17
                2013  1306.89
                2014   992.02
                2015   399.69
                2016   446.01
                2017   563.37
                2018   741.25
ZAMBIA          2010   118.32
                2011   210.84
                2012   242.95
                2013   377.20
                2014   366.43
                2015   298.02
                2016   237.05
                2017   294.02
                2018   318.90
ZIMBABWE        2010   113.07
                2011   129.22
                2012   153.16
                2013   157.96
                2014   223.84
                2015   205.01
                2016   109.02
                2017   163.46
                2018   181.62

[2099 rows x 1 columns]
```

```
In [59]: export_group.groupby(['country'])
         import_temp=import_group.groupby(['country']).agg({'value':'sum'})
         export_temp=export_group.groupby(['country']).agg({'value':'sum'}).loc[import_temp.inc
         #GROUPING THE IMPORTS AND EXPORTS COUNTRY WISE AND AGGREGATING THE VALUE AND ARRANGIN
```

```
In [60]: data_1=import_group.groupby(['country']).agg({'value':'sum'}).sort_values(by='value')
```

```
        data_2=export_temp
        data_3=data_2-data_1
        #GETTING THE TOP 10 VALUES
```

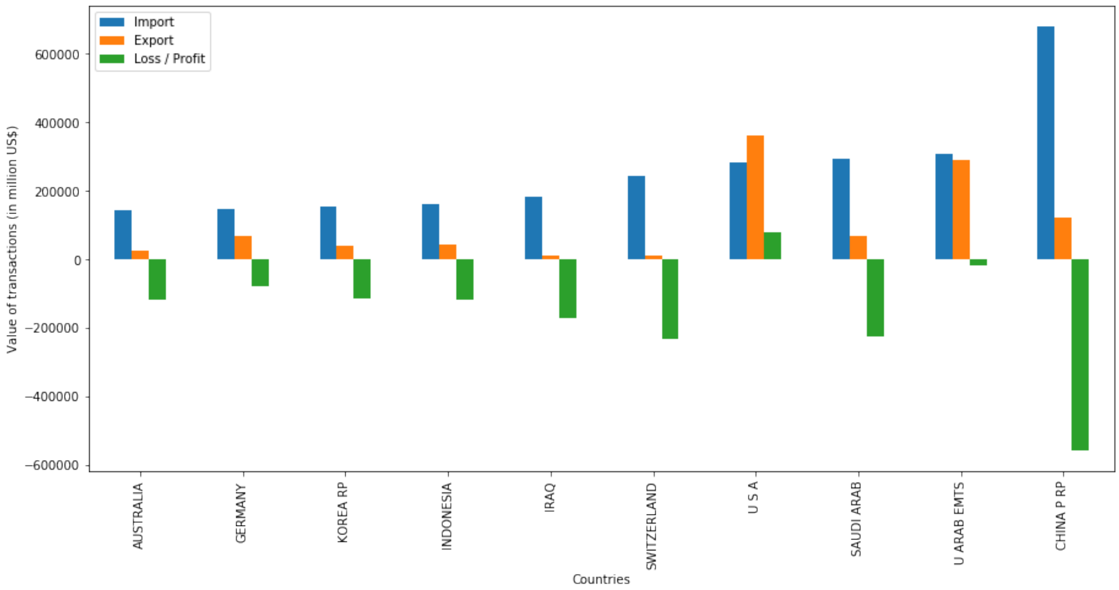
```
In [61]: data_1.columns=['Import']
         data_2.columns=['Export']
         data_3.columns=['Loss / Profit']
```

```
In [62]: df=pd.DataFrame(index=data_1.index.values)
         #df=pd.concat([data_1,data_2,data_3])
         df['Import']=data_1
         df['Export']=data_2
         df['Loss / Profit']=data_3

         df
         #IT GIVES US THE TOTAL AMOUNT OF MONEY INDIA IMPORTS AND EXPORTS IN IN INREASING ORDE
```

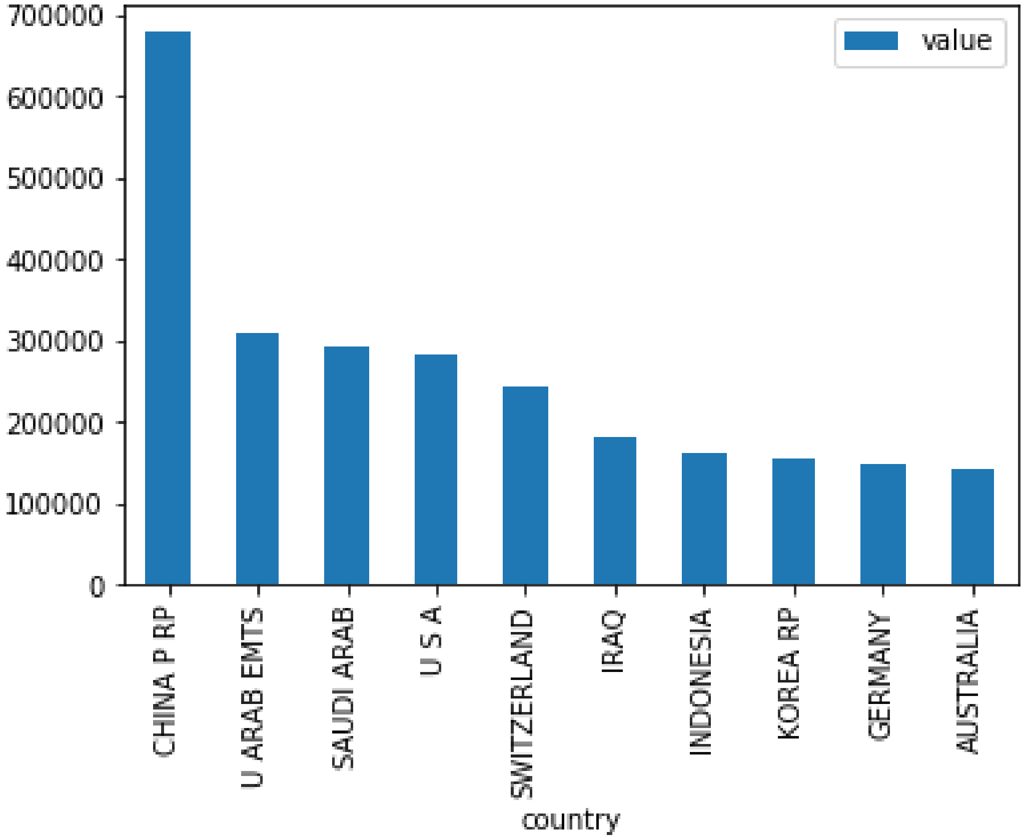
```
Out[62]:                 Import     Export  Loss / Profit
         AUSTRALIA    142723.67   25372.29     -117351.38
         GERMANY      148120.79   68855.45      -79265.34
         KOREA RP     154263.94   38023.02     -116240.92
         INDONESIA    160452.65   42148.84     -118303.81
         IRAQ         182464.57    9833.61     -172630.96
         SWITZERLAND  243739.33    9992.56     -233746.77
         U S A        281436.58  360609.57       79172.99
         SAUDI ARAB   292104.09   65995.99     -226108.10
         U ARAB EMTS  307810.15  289374.88      -18435.27
         CHINA P RP   678877.15  121806.00     -557071.15
```

```
In [63]: fig, ax = plt.subplots(figsize=(15,7))
         df.plot(kind='bar',ax=ax)
         ax.set_xlabel('Countries')
         ax.set_ylabel('Value of transactions (in million US$)')

         # PLOTS THE TOTAL IMPORT EXPORT AND TOTAL INCOME
```

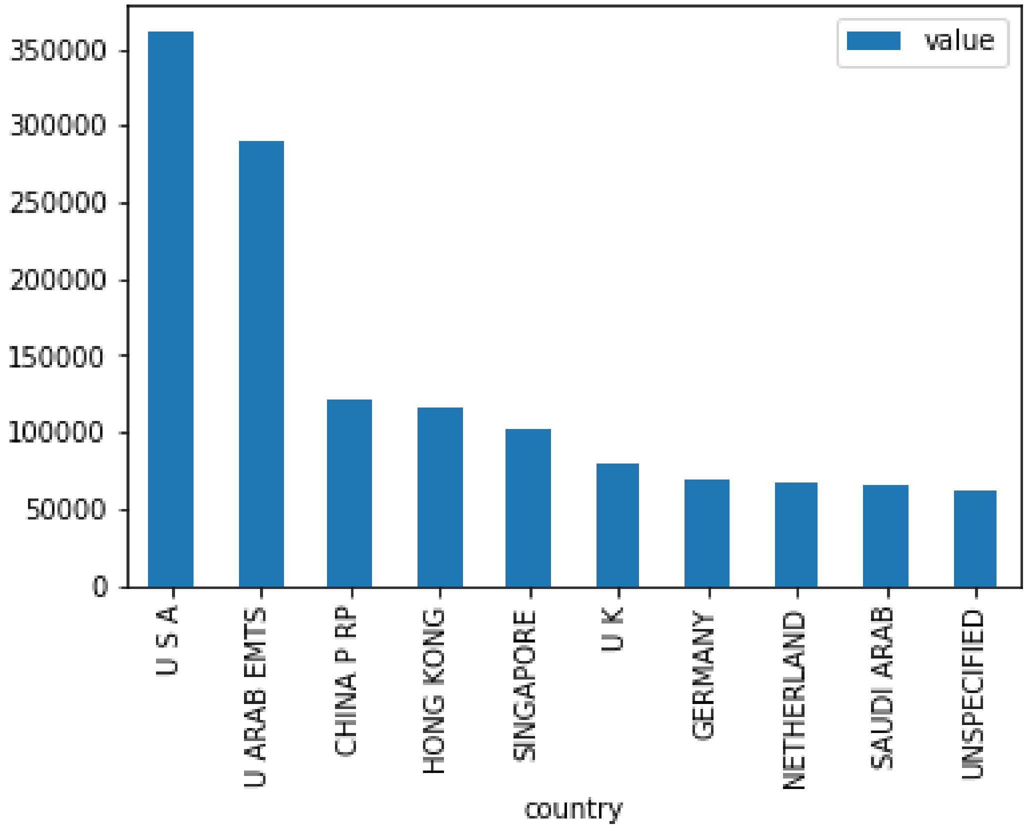
```
Out[63]: Text(0, 0.5, 'Value of transactions (in million US$)')
```



```
In [64]: df_import = import_df.groupby('country').agg({'value':'sum'}).sort_values(by='value',
         df_import.plot(kind='bar')
         #PLOTS HOW MUCH INDIA IMPORTS FROM ITS TOM 10 LENDERS

Out[64]: <matplotlib.axes._subplots.AxesSubplot at 0x19b290caf60>
```



```
In [65]: df_export = export_df.groupby('country').agg({'value':'sum'}).sort_values(by='value',
         df_export.plot(kind='bar')
         #PLOTS HOW MUCH INDIA EXPORTS TO ITS TOP 10 BUYERS

Out[65]: <matplotlib.axes._subplots.AxesSubplot at 0x19b28c73978>
```



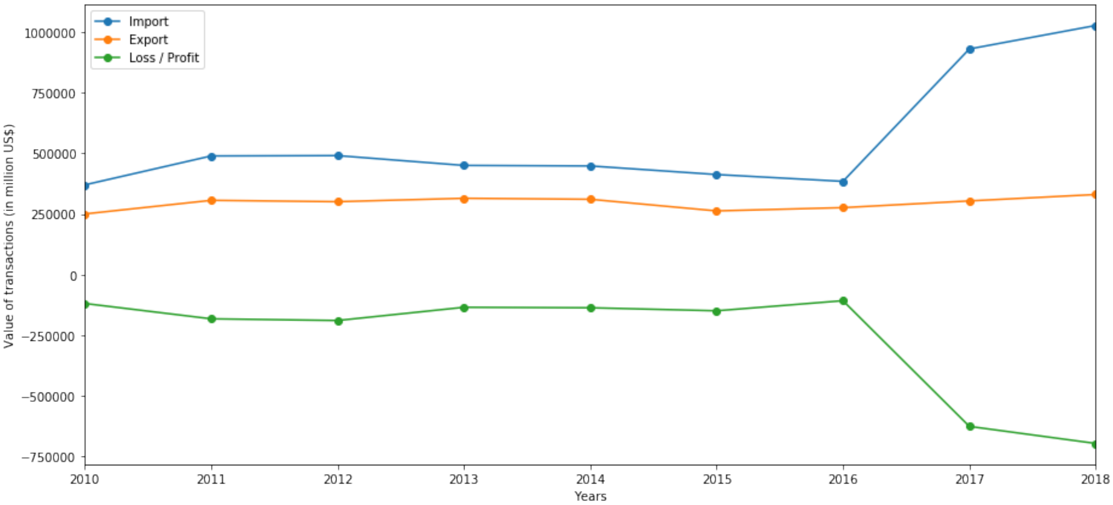
```
In [66]: Import_ =import_df.groupby(['year']).agg({'value':'sum'})
         Export_ =export_df.groupby(['year']).agg({'value':'sum'})
         Deficit_=Export_ -Import_
         Time_Series=pd.DataFrame(index=Import_.index.values)
         Time_Series['Import']=Import_
         Time_Series['Export']=Export_
         Time_Series['Loss / Profit']=Deficit_
         #YEAR WISE ANALYSIS
```

```
In [67]: Time_Series
```

```
Out[67]:           Import     Export  Loss / Profit
         2010   369762.25  249801.18     -119961.07
         2011   489311.81  305948.28     -183363.53
         2012   490730.07  300384.32     -190345.75
         2013   450192.99  314388.61     -135804.38
         2014   448026.63  310321.02     -137705.61
         2015   412537.53  262274.30     -150263.23
         2016   384350.29  275835.27     -108515.02
         2017   931148.04  303507.85     -627640.19
         2018  1028142.66  330058.64     -698084.02
```

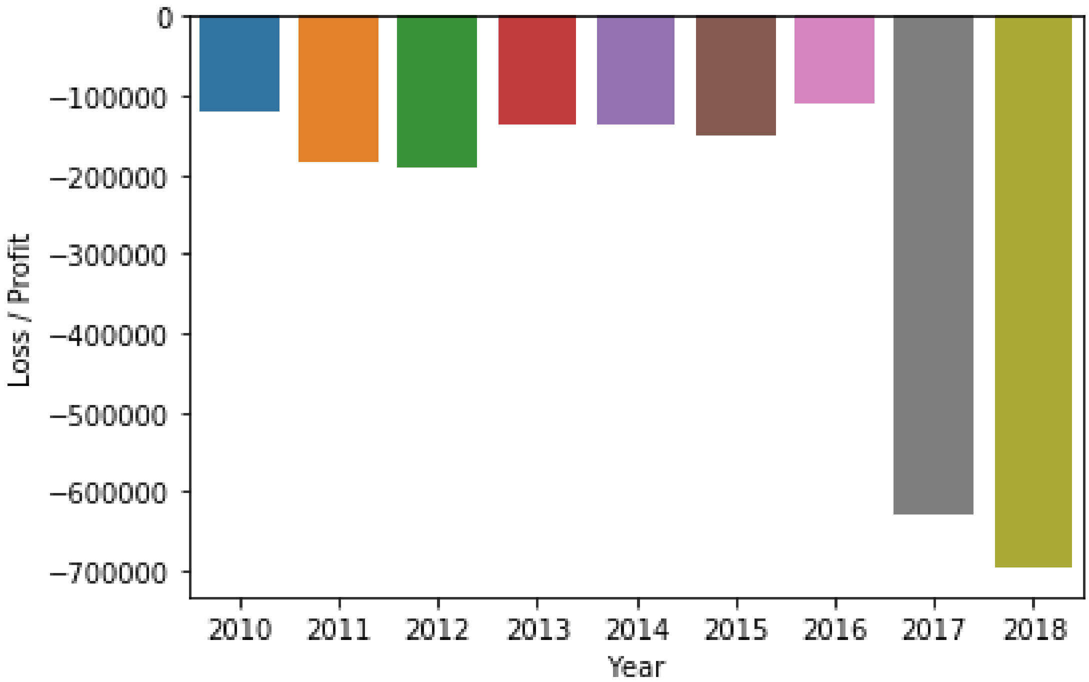
```
In [68]: fig, ax = plt.subplots(figsize=(15,7))
         Time_Series.plot(ax=ax,marker='o')
         ax.set_xlabel('Years')
         ax.set_ylabel('Value of transactions (in million US$)')
         # PLOTS HOW MUCH MONEY INDIA HAS MADE OVER THE LAST 10 YEARS FROM TRADE
```

```
Out[68]: Text(0, 0.5, 'Value of transactions (in million US$)')
```



```
In [69]: Time_Series.index.name = 'Year'
         Time_Series.reset_index(inplace=True)
         #LOSS INDIA HAS MADE FROM TRADE IN THE LAST 10 YEARS
```

```
In [70]: sns.barplot(x = 'Year', y = 'Loss / Profit', data = Time_Series)
         plt.show()#LOSS INDIA HAS MADE FROM TRADE IN THE LAST 10 YEARS
```



```
In [71]: import_group=import_df.groupby(['Commodity','year']).agg({'value':'sum'})
         export_group=export_df.groupby(['Commodity','year']).agg({'value':'sum'})
         #GROUP BY COMMODITY
```

```
In [72]: import_group.sort_values(by ='value',ascending = False).head(10)
```

```
Out[72]:                                                           value
         Commodity                                           year
         MINERAL FUELS, MINERAL OILS AND PRODUCTS OF THE...  2018  335743.56
                                                             2017  264589.04
                                                             2013  181382.57
                                                             2012  181344.64
                                                             2011  172753.87
                                                             2014  156399.99
         NATURAL OR CULTURED PEARLS,PRECIOUS OR SEMIPREC...  2017  149420.82
                                                             2018  129440.38
         MINERAL FUELS, MINERAL OILS AND PRODUCTS OF THE...  2010  115929.10
                                                             2015  107250.79
```

```
In [73]: import_temp=import_group.groupby(['Commodity']).agg({'value':'sum'})
         export_temp=export_group.groupby(['Commodity']).agg({'value':'sum'}).loc[import_temp.
```

```
In [74]: import_temp
```

```
Out[74]:                                                      value
         Commodity
```

```
AIRCRAFT, SPACECRAFT, AND PARTS THEREOF.            65212.62
ALBUMINOIDAL SUBSTANCES; MODIFIED STARCHES; GLU...    4097.18
ALUMINIUM AND ARTICLES THEREOF.                     42375.88
ANIMAL OR VEGETABLE FATS AND OILS AND THEIR CLE... 115462.40
ARMS AND AMMUNITION; PARTS AND ACCESSORIES THER...    414.10
ARTICLES OF APPAREL AND CLOTHING ACCESSORIES, K...   2884.66
ARTICLES OF APPAREL AND CLOTHING ACCESSORIES, N...   3991.15
ARTICLES OF IRON OR STEEL                           45781.95
ARTICLES OF LEATHER,SADDLERY AND HARNESS;TRAVEL...   3750.24
ARTICLES OF STONE, PLASTER, CEMENT, ASBESTOS, M...   7561.06
BEVERAGES, SPIRITS AND VINEGAR.                      6191.45
CARPETS AND OTHER TEXTILE FLOOR COVERINGS.           1041.29
CERAMIC PRODUCTS.                                    7077.66
CEREALS.                                             2830.67
CLOCKS AND WATCHES AND PARTS THEREOF.                3390.63
COCOA AND COCOA PREPARATIONS.                        2378.61
COFFEE, TEA, MATE AND SPICES.                        7257.63
COPPER AND ARTICLES THEREOF.                        39889.03
CORK AND ARTICLES OF CORK.                             65.21
COTTON.                                              9030.46
DAIRY PRODUCE; BIRDS' EGGS; NATURAL HONEY; EDIB...    807.00
EDIBLE FRUIT AND NUTS; PEEL OR CITRUS FRUIT OR ...  31111.30
EDIBLE VEGETABLES AND CERTAIN ROOTS AND TUBERS.     28442.52
ELECTRICAL MACHINERY AND EQUIPMENT AND PARTS TH... 427570.66
ESSENTIAL OILS AND RESINOIDS; PERFUMERY, COSMET...   8100.83
EXPLOSIVES; PYROTECHNIC PRODUCTS; MATCHES; PYRO...    109.04
FERTILISERS.                                        68717.35
FISH AND CRUSTACEANS, MOLLUSCS AND OTHER AQUATI...    896.99
FOOTWEAR, GAITERS AND THE LIKE; PARTS OF SUCH A...   5741.58
FURNITURE; BEDDING, MATTRESSES, MATTRESS SUPPOR...  15785.33
...                                                       ...
PREPARATIONS OF VEGETABLES, FRUIT, NUTS OR OTHE...    976.57
PREPARED FEATHERS AND DOWN AND ARTICLES MADE OF...    169.11
PRINTED BOOKDS, NEWSPAPERS, PICTURES AND OTHER ...   4782.72
PRODUCTS OF ANIMAL ORIGIN, NOT ELSEWHERE SPECIF...    457.45
PRODUCTS OF THE MILLING INDUSTRY; MALT; STARCHE...    662.15
PROJECT GOODS; SOME SPECIAL USES.                   43677.74
PULP OF WOOD OR OF OTHER FIBROUS CELLULOSIC MAT...  19670.78
RAILWAY OR TRAMWAY LOCOMOTIVES, ROLLING-STOCK A...   4808.38
RAW HIDES AND SKINS (OTHER THAN FURSKINS) AND L...   6638.67
RESIDUES AND WASTE FROM THE FOOD INDUSTRIES; PR...   4651.84
RUBBER AND ARTICLES THEREOF.                        37362.65
SALT; SULPHUR; EARTHS AND STONE; PLASTERING MAT...  28984.75
SHIPS, BOATS AND FLOATING STRUCTURES.               58294.94
SILK                                                 2791.80
SOAP, ORGANIC SURFACE-ACTIVE AGENTS, WASHING PR...   7500.32
SPECIAL WOVEN FABRICS; TUFTED TEXTILE FABRICS; ...   1943.68
SUGARS AND SUGAR CONFECTIONERY.                      8213.27
```

```
TANNING OR DYEING EXTRACTS; TANNINS AND THEIR D...   19103.44
TIN AND ARTICLES THEREOF.                             2277.85
TOBACCO AND MANUFACTURED TOBACCO SUBSTITUTES.          481.77
TOOLS IMPLEMENTS, CUTLERY, SPOONS AND FORKS, OF...   10568.39
TOYS, GAMES AND SPORTS REQUISITES; PARTS AND AC...    5734.05
UMBRELLAS, SUN UMBRELLAS, WALKING-STICKS, SEAT-...     310.79
VEGETABLE PLAITING MATERIALS; VEGETABLE PRODUCT...     308.94
VEHICLES OTHER THAN RAILWAY OR TRAMWAY ROLLING ...   57415.76
WADDING, FELT AND NONWOVENS; SPACIAL YARNS; TWI...    2891.36
WOOD AND ARTICLES OF WOOD; WOOD CHARCOAL.            25978.66
WOOL, FINE OR COARSE ANIMAL HAIR, HORSEHAIR YAR...    4290.27
WORKS OF ART COLLECTORS' PIECES AND ANTIQUES.         1186.38
ZINC AND ARTICLES THEREOF.                            6194.61

[98 rows x 1 columns]
```

```
In [ ]: HSCode=pd.DataFrame()
        HSCode['Start']=[1,6,15,16,25,28,39,41,44,47,50,64,68,71,72,84,86,90,93,94,97]
        HSCode['End']=[5,14,15,24,27,38,40,43,46,49,63,67,70,71,83,85,89,92,93,96,98]
        HSCode['Sections']=['Animals & Animal Products',
        'Vegetable Products',
        'Animal Or Vegetable Fats',
        'Prepared Foodstuffs',
        'Mineral Products',
        'Chemical Products',
        'Plastics & Rubber',
        'Hides & Skins',
        'Wood & Wood Products',
        'Wood Pulp Products',
        'Textiles & Textile Articles',
        'Footwear, Headgear',
        'Articles Of Stone, Plaster, Cement, Asbestos',
        'Pearls, Precious Or Semi-Precious Stones, Metals',
        'Base Metals & Articles Thereof',
        'Machinery & Mechanical Appliances',
        'Transportation Equipment',
        'Instruments - Measuring, Musical',
        'Arms & Ammunition',
        'Miscellaneous',
        'Works Of Art',]
```

```
In [78]: import_df['Sections']=import_df["HSCode"]
         export_df['Sections']=export_df["HSCode"]
         for i in range(0,len(HSCode)):
             import_df.loc[(import_df["Sections"] >= HSCode['Start'][i]) & (import_df["Section
             export_df.loc[(export_df["Sections"] >= HSCode['Start'][i]) & (export_df["Section
```

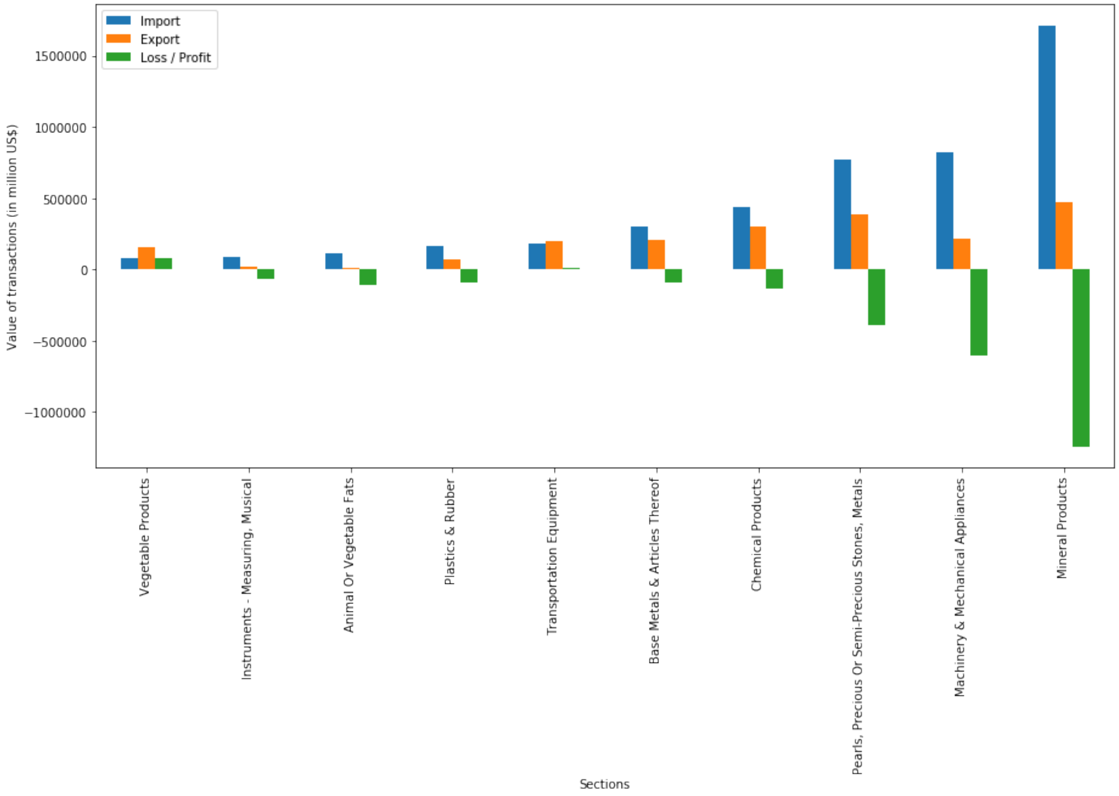
```
In [79]: import_group=import_df.groupby(['Sections','year']).agg({'value':'sum'})
```

```
         export_group=export_df.groupby(['Sections','year']).agg({'value':'sum'})

         import_temp=import_group.groupby(['Sections']).agg({'value':'sum'})
         export_temp=export_group.groupby(['Sections']).agg({'value':'sum'}).loc[import_temp.i
```

```
In [80]: data_1=import_group.groupby(['Sections']).agg({'value':'sum'}).sort_values(by='value'
         data_2=export_temp
         data_3=data_2-data_1
         data_1.columns=['Import']
         data_2.columns=['Export']
         data_3.columns=['Loss / Profit']
         df=pd.DataFrame(index=data_1.index.values)
         #df=pd.concat([data_1,data_2,data_3])
         df['Import']=data_1
         df['Export']=data_2
         df['Loss / Profit']=data_3
```

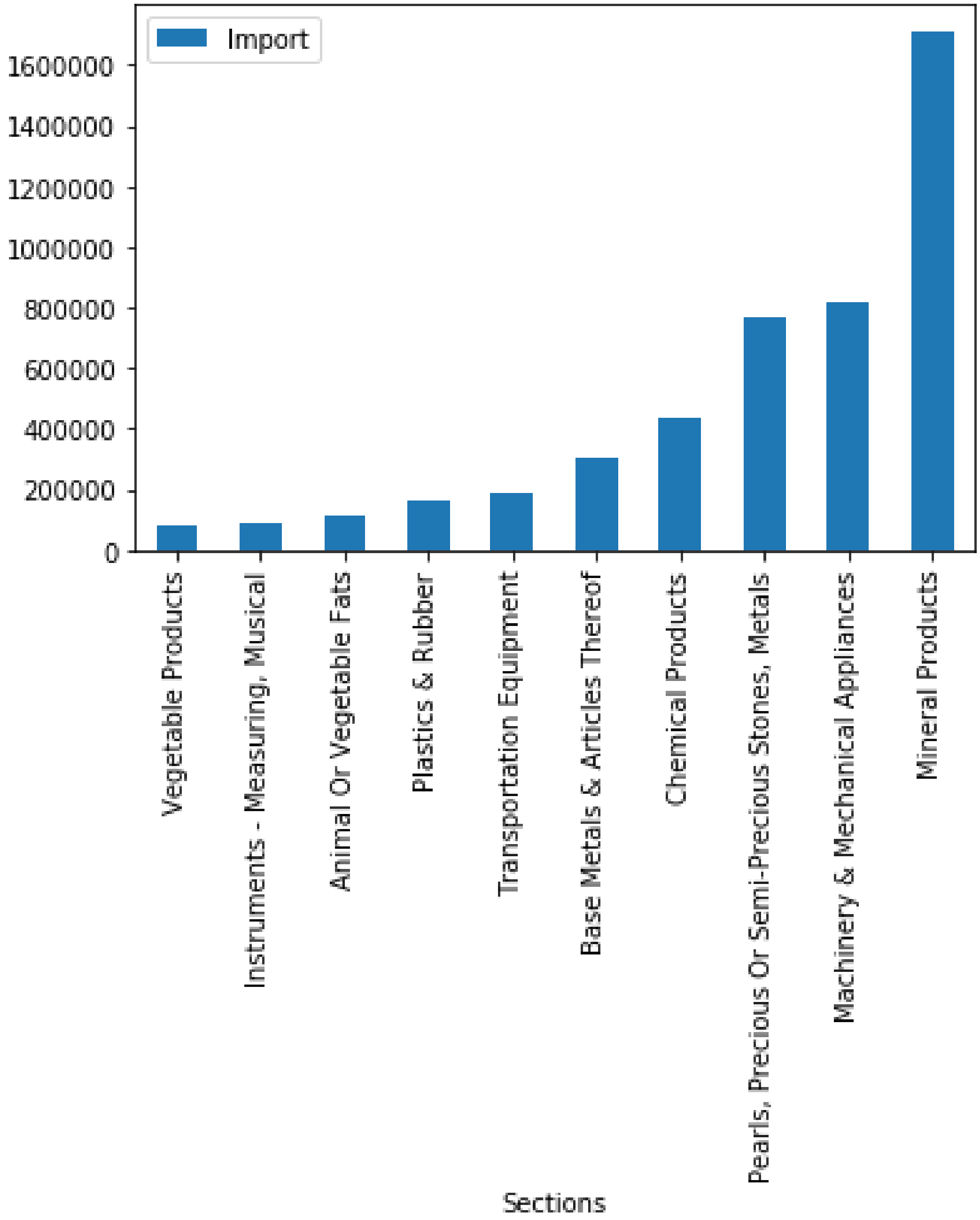
```
In [81]: df.index=HSCode['Sections'][data_1.index.values]
         fig, ax = plt.subplots(figsize=(15,7))
         df.plot(kind='bar',ax=ax)
         ax.set_xlabel('Sections')
         ax.set_ylabel('Value of transactions (in million US$)')
```

```
Out[81]: Text(0, 0.5, 'Value of transactions (in million US$)')
```



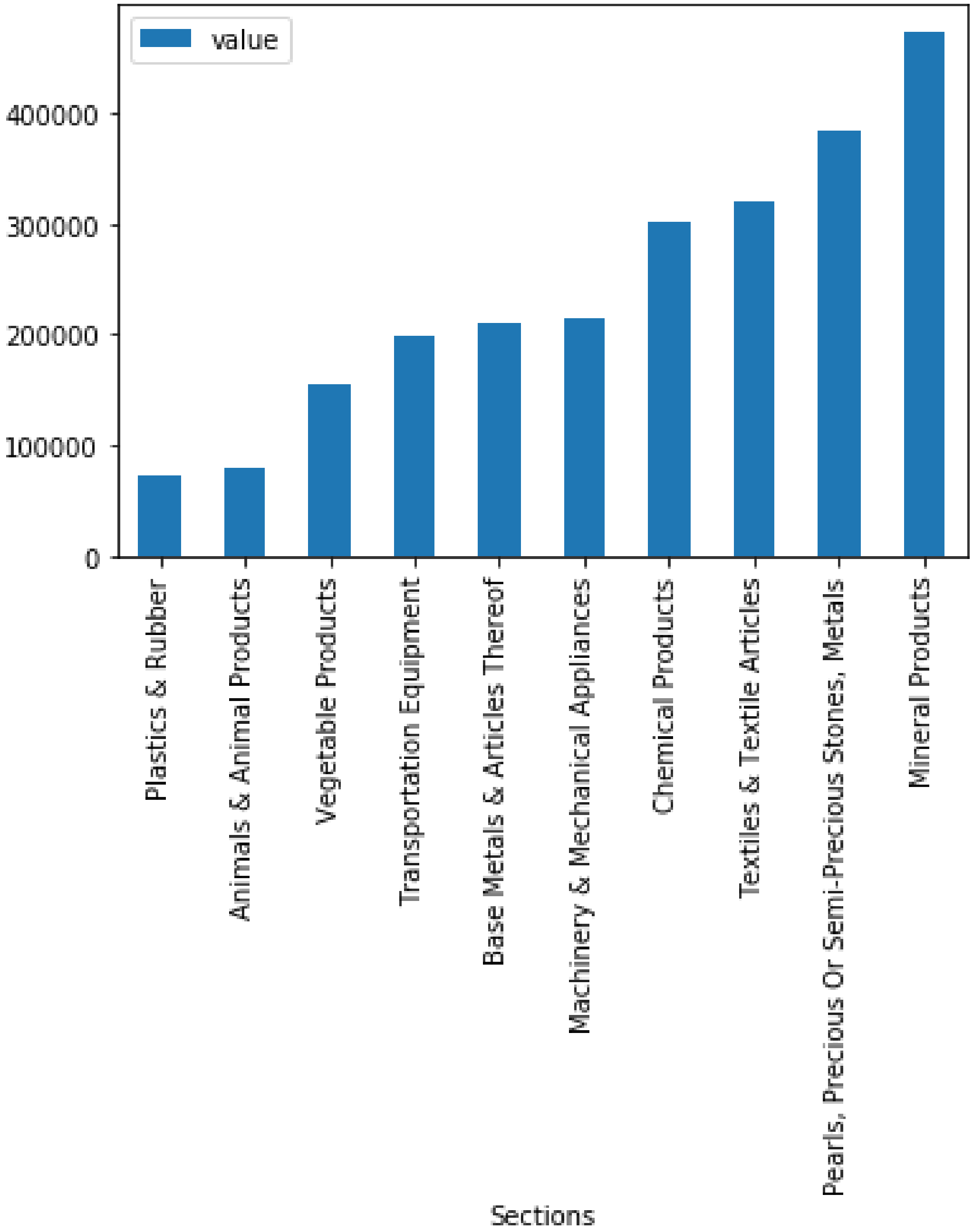
```
In [82]: data_1.index=HSCode['Sections'][data_1.index.values]
         data_1.plot(kind='bar')
```

```
Out[82]: <matplotlib.axes._subplots.AxesSubplot at 0x19b2a4b9438>
```



```
In [83]: data_2=export_group.groupby(['Sections']).agg({'value':'sum'}).sort_values(by='value'
         data_2.index=HSCode['Sections'][data_2.index.values]
         data_2.plot(kind='bar')
```

```
Out[83]: <matplotlib.axes._subplots.AxesSubplot at 0x19b2ac8e9e8>
```


value
400000
300000
200000
100000
0
Plastics & Rubber
Animals & Animal Products
Vegetable Products
Transportation Equipment
Base Metals & Articles Thereof
Machinery & Mechanical Appliances
Chemical Products
Textiles & Textile Articles
Pearls, Precious Or Semi-Precious Stones, Metals
Mineral Products
Sections

```
In [ ]:
```