Data exploration the association between Olympic datasets and economy of countries all over the world in recent 16 years

❖ Introduction:

This report illustrates the datasets of Olympic winning countries and their economy. Explore if there is any association between Olympic medallist achievement of countries and the increments rate in their economy. There are many articles take this topic in multiple aspects. Martin and Tomas have published a paper about 'Olympic Medals, Economy, Geography and Politics from Sydney to Rio' [1]. Their study focusing on economy, politics and geography factors and their affection on the economy of Sydney and their Olympic medal achievements.

Is Olympic medallist achievement a reflection of a country economic? This report will look at the countries that won medals in recent years between 2000 and 2016, and contrast that data with the economic change rate between 2000 and 2016; to find if there is any correlation between those two datasets.

❖ Datasets:

➢ 120 years of Olympic history: athletes and results:

this data from Kaggle which has been scraped from www.sports-reference.com in may 2018. It is a real historical dataset on the modern Olympic Games from Athens 1896 to Rio 2016. It consists of two datasets [2]:

- **Athlete events "athlete_events.csv" file:**
  It has 271116 entries and 15 columns with different type.

ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year
Int64	Object	Object	Float64	Float64	Float64	Object	Object	Object	Int64

Season	City	Sport	Event	Medal
Object	Object	Object	Object	Object

- **National Olympic Committee NOC regions "noc_regions.csv" file:**
  It has 230 entries and 3 columns.

NOC	Region	Note
Object	Object	Object

➢ Gross domestic product GDP growth, world bank national accounts data:

this live data from the world bank data which has been scraped from https://data.worldbank.org in March 2019. It is a real GDP growth for the economy of world countries from 1960 to 2018. It has 264 entries and 63 columns [3].

Country Name	Country Code	Indicator Name	Indicator Code	1960	1961	….to….	2017	2018
Object	Object	Object	Object	Float64	Float64	Float64	Float64	Float64

❖ Process:

➢ Tools:

Jupyter notebook, python language, Excel program and libraries such as : numpy, pandas, matplotlib, seaborn, sklearn and plotly.

➢ Cleaning:

- Olympic athlete events dataset

```
In [9]: data1.head(21)
Out[9]:
```

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	Sport	Event	Medal
0	1	A Dijiang	M	24.0	180.0	80.0	China	CHN	1992 Summer	1992	Summer	Barcelona	Basketball	Basketball Men's Basketball	NaN
1	2	A Lamusi	M	23.0	170.0	60.0	China	CHN	2012 Summer	2012	Summer	London	Judo	Judo Men's Extra-Lightweight	NaN
2	3	Gunnar Nielsen Aaby	M	24.0	NaN	NaN	Denmark	DEN	1920 Summer	1920	Summer	Antwerpen	Football	Football Men's Football	NaN
3	4	Edgar Lindenau Aabye	M	34.0	NaN	NaN	Denmark/Sweden	DEN	1900 Summer	1900	Summer	Paris	Tug-Of-War	Tug-Of-War Men's Tug-Of-War	Gold
4	5	Christine Jacoba Aaftink	F	21.0	185.0	82.0	Netherlands	NED	1988 Winter	1988	Winter	Calgary	Speed Skating	Speed Skating Women's 500 metres	NaN
5	5	Christine Jacoba Aaftink	F	21.0	185.0	82.0	Netherlands	NED	1988 Winter	1988	Winter	Calgary	Speed Skating	Speed Skating Women's 1,000 metres	NaN

Figure 1:Olympic athlete events dataset

Figure1, shows the original form of Olympic dataset before cleaning. From this figure, you can notice how preprocessing this data needs. There is a duplication data with a lot of NaN values rather than unnecessary data for this report. Following these steps will increase the quality of the data and make it more clean:

1. Remove duplicated value by "**drop_duplicates()**" function.
2. NaN value of the Medal column means not a winner just a participant person who did not win to gain a medal. So using this function "**fillna('not a winner', inplace=True)**" for replacing NaN value to have more meaningful.
3. NaN values in Age, height and weight columns have less meaning to this report. Because this report is focusing on country achievement which is "NOC" and "Medal" regardless of the participant's information itself. By using "**dropna()**" function the rows become 206146 entries.
4. This study is between the recent 16 years 2000 and 2016. Using this function to customize it: **data[(data['Year'] >= 2000) & (data['Year'] <= 2016)]**
5. Now data rows become 83926 entries. For more organization use "**sort_values(by ='Year')**" function to sort the entries between 2000 and 2016.
6. NOC regions dataset just to know the meaning of NOC region name. like USA is a NOC National Olympic Committee a short of United State America. It is already merged with athlete events dataset

- Economy of world bank dataset

```
In [55]: #Data Source, World Development Indicators, Last update 21/03/2019
         #this data from : https://data.worldbank.org/indicator/NY.GDP.MKTP.KD.ZG?locations=RU

         dataframe_economy = pd.read_csv(r'../mac/DM&V_Datasets/country_economy_data.csv',encoding = "ISO-8859-1")
         dataframe_economy.head(21)
Out[55]:
```

	Country Name	Country Code	Indicator Name	Indicator Code	1960	1961	1962	1963	1964	1965	...	2009	2010	2011
0	Aruba	ABW	GDP growth (annual %)	NY.GDP.MKTP.KD.ZG	NaN	NaN	NaN	NaN	NaN	NaN	...	-10.519748	-3.685030	3.446055
1	Afghanistan	AFG	GDP growth (annual %)	NY.GDP.MKTP.KD.ZG	NaN	NaN	NaN	NaN	NaN	NaN	...	21.390528	14.362441	0.426355
2	Angola	AGO	GDP growth (annual %)	NY.GDP.MKTP.KD.ZG	NaN	NaN	NaN	NaN	NaN	NaN	...	0.858713	4.859220	3.471981
3	Albania	ALB	GDP growth (annual %)	NY.GDP.MKTP.KD.ZG	NaN	NaN	NaN	NaN	NaN	NaN	...	3.350000	3.710000	2.550000
4	Andorra	AND	GDP growth (annual %)	NY.GDP.MKTP.KD.ZG	NaN	NaN	NaN	NaN	NaN	NaN	...	-3.690654	-5.358826	-4.646543
5	Arab World	ARB	GDP growth (annual %)	NY.GDP.MKTP.KD.ZG	NaN	NaN	NaN	NaN	NaN	NaN	...	0.465712	4.749403	3.608109

Figure 2: World Bank national accounts data, and OECD National Accounts data files.

Figure2, shows the original form of economy dataset before cleaning. The format of this data was unreadable and unwritable csv. So using this statement " **encoding = 'ISO-8859-1'** " help the jupyter notebook to read the data without any errors. Unwritable has been solved by transform the data to writable csv file by save as "MS-DOS Comma Separated (.csv)". There is a lot of noisy data and nan values dealing with it like the Olympic athlete events dataset as mentioned previously.

➢ Visualization

- First graph : Olympic Medallists of Top 20 countries between 2000 and 2016

  Visualizing a horizontal bar chart express the Olympic medal achievements of countries between 2000 and 2016 and the number of medals each country had. Through following these steps:

  1. Delete all rows who have not a medal by :

     data=data[data.Medal != 'not a winner']

  2. Extract the column "NOC and Medal" by:

     data[['NOC','Medal']]

  3. Now we have a data consist of NOC and Medal, but the graph need to know how many medals each NOC have.

     d = d.sort_values(by ='NOC') #rearrangement

     d = d.groupby('NOC').size() #new col. count medal for each NOC

     using **"d.to_csv"** to extract NOC and new the column "The number of Medals each NOC have". Sort value by the winning which has the highest number of medals through the years period using "**.sort_values()**". Finally, by using **plotly** library for interactive chart and some changes to add labels and title and other features to optimize the visualization result [4].

  4. Still the graph is not very clear because we have 118 entries to represent. Therefore, represent the top 20 wining country is much enough for this study. Using this code:

     Top20_dataframe = exdataframe_NOC_Medal.head(20)

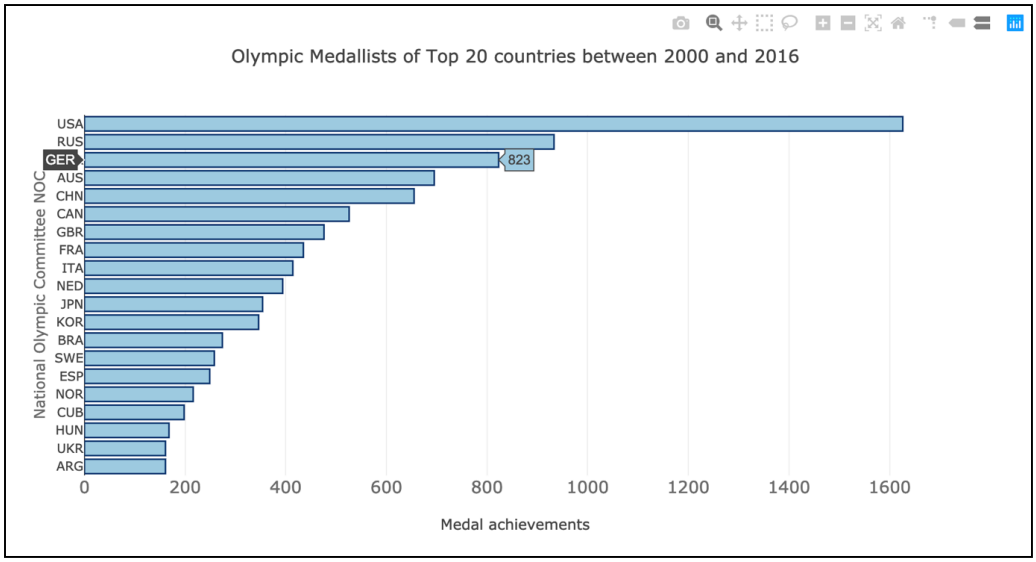

Figure 3: Interactive graph of Olympic Medalists of Top 20 countries between 2000 and 2016.

- Second graph: Top 20 countries for medal achievements

  Stacked horizontal bar chart illustrate the amount of each Medals' type of country had acquired between 2000 and 2016. The graph in figure3 has been done via these steps:

  1. Extract top 20 wining countries from data with their medals over the years period:
     usadf=dataframe_NOC_Medal.loc[dataframe_NOC_Medal['NOC'] == 'USA']
  2. Then collect the 20 countries in one dataset using panda library:
     test_data = [USA_df, Russia_df, German_df, Australia_df,…]
     result = pd.concat(test_data)
  3. Extract the "**result**" by "**.to_csv**" function to complete the process through Excel program.
  4. In Excel count the Gold, Silver and Bronze for each of 20 countries by:
     =COUNTIF(B:B,"Gold")
  5. Read the Excel process result using **panda** library "**pd.read_csv()**"

```
In [139]: dataframe_NOC_GSB = pd.read_csv('../mac/DM&V_Datasets/Stacked_Medal.csv')
          print(dataframe_NOC_GSB)
          #dataset of: NOC, Gold, Silver, Bronze
```

	NOC	Gold	Silver	Bronze
0	USA	698	519	409
1	RUS	314	266	353
2	GER	287	238	298
3	AUS	186	262	247
4	CHN	283	190	182
5	CAN	231	124	171
6	GBR	186	160	130
7	FRA	132	166	137
8	ITA	94	145	175
9	NED	123	167	104
10	JPN	73	132	149
11	KOR	129	101	116
12	BRA	82	106	86
13	SWE	67	126	65
14	ESP	28	135	86
15	NOR	97	42	77
16	CUB	62	81	55
17	HUN	89	55	24
18	UKR	36	45	80
19	ARG	70	33	58

Figure 4: Read and print the excel table of NOC and the medals' type

Using **plotly** library for interactive graph and edit on some features to be suitable with the excellent visualization criteria [5].

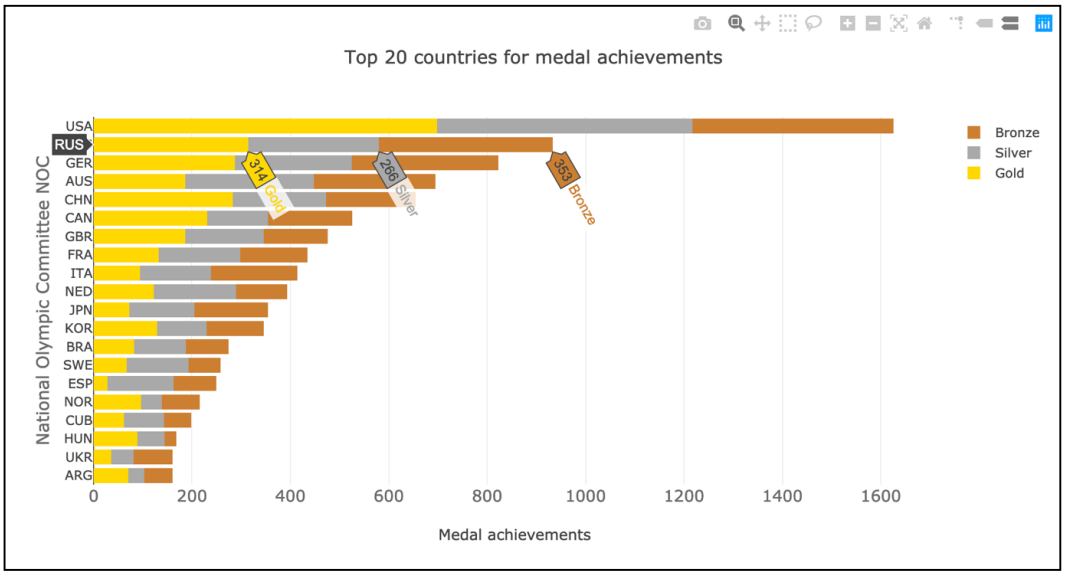

Figure 5: Interactive graph of Top 20 countries for medal achievements

- Third graph: United States and Argentina economic growth over 2000 and 2016

  United States (USA) has achieved 1626 medals over 2000 and 2016. And Argentina (ARG) has achieved 161 over the same period. USA consider the highest of the Top 20 winning countries and ARG has the lowest number of medals. Therefor, comparing with them will help this study to find the answer if there is any association between earning medals and increment change of economic growth. Extract USA and ARG via the same steps have mentioned before gets this result by using **Plotly** library interactive graph [6].

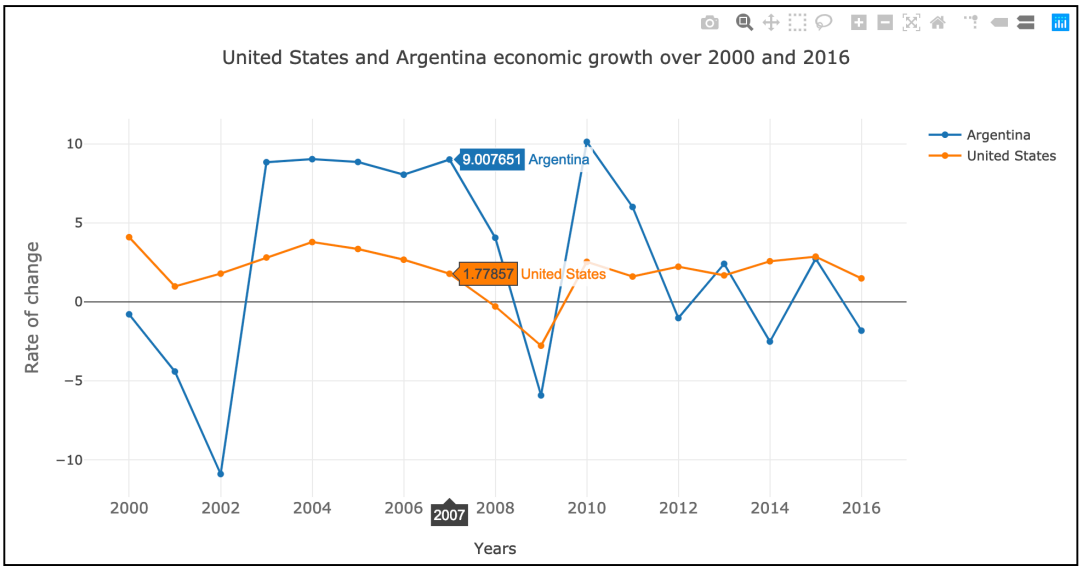

Figure 6: Interactive graph of United States and Argentina economic growth over 2000 and 2016

- Fourth graph: Argentina Olympic medallist achievement and economic between 2000 and 2016

  This angled text bar chart has made to compare between Olympic achievement of ARG and its economy growth over 2000 and 2016. Using **Plotly** library and extra features [7].

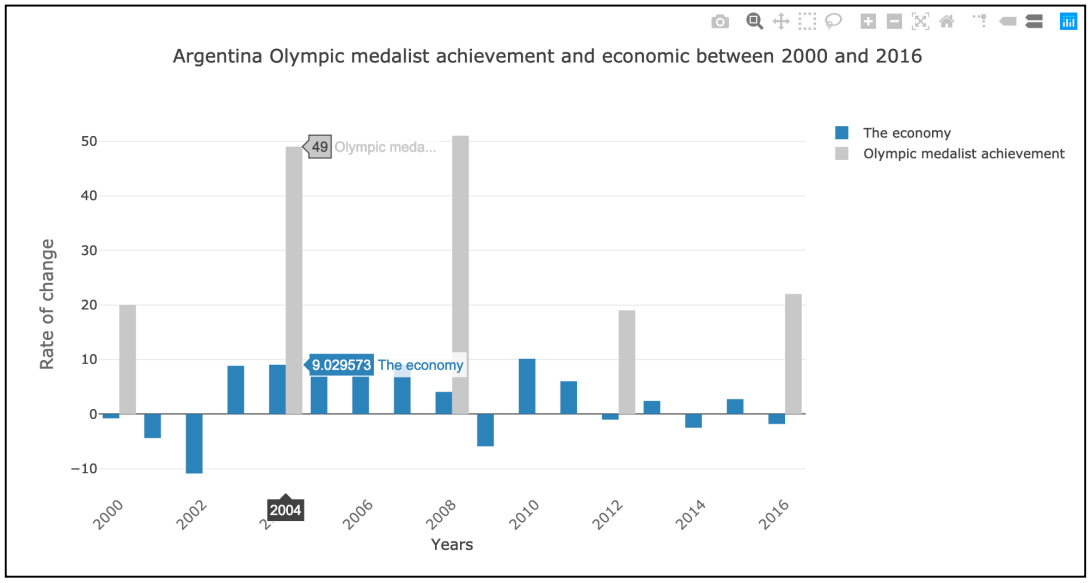

Figure 7: Interactive graph of Argentina Olympic medalist achievement and economic between 2000 and 2016.

❖ Result:

Theas visualizations have been done to illustrate if there is any affection between Olympic wining country and their economy. In figure5, it is obvious that strong countries like USA, Russia, German, Australia, China and other strong country has a strong positive economy growth achieved the highest number of medals. Using Plotly library interactive graph, such as angled text bar at figure7 with a legible color, legend and Clear labels will lead to clear illation.

After processing the dataset of Olympic and economy of a countries all over the world between 2000 and 2016, It is clear that there is no association or strong relationship between medals achievement of United States or Argentina or any other countries and their economy. Figure7 proves clearly that illation, ARG acquired 20 medals in 2000 but with slightly descent rate of the ARG economy. The same illation got from USA graph. Which leads to consider that there are more strong factors affect the economy growth of a country like political and business factors rather than Olympic medals.

❖ Appendix:

FIGURES:

WORKS CITED:

[1] T. D. Martin Grančay* 1, "Olympic Medals, Economy, Geography and Politics from Sydney to Rio," 12 September 2017.

[2] r. U. account, "120 years of Olympic history: athletes and results," May 2018. [Online]. Available: https://www.kaggle.com/heesoo37/120-years-of-olympic-history-athletes-and-results#athlete_events.csv .

[3] "GDP growth (annual %)," [Online]. Available: https://data.worldbank.org/indicator/NY.GDP.MKTP.KD.ZG?locations=RU .

[4] "Bar Charts in Python," [Online]. Available: https://plot.ly/python/bar-charts/.

[5] "Horizontal Bar Charts in Python," [Online]. Available: https://plot.ly/python/horizontal-bar-charts/ .

[6] "Line Charts in Python," [Online]. Available: https://plot.ly/python/line-charts/ .

[7] "Bar Charts in Python," [Online]. Available: https://plot.ly/python/bar-charts/ .