COSC3000 - REPORT
Visualisation

Teanlouise

May 3, 2020

Contents

1 Introduction

The topic of this project is the Modern Olympics. The games have been a global competition since 1896 with both Summer and Winter sports. The goal is to analyse the patterns of a medal winner depending on their physical characteristics (weight, height, age, sex), home country (athleticism, GDP, population) and the games in which they compete (location). The Olympics are supposed to be a celebration of peace, inclusion and human persistence. It is an opportunity for people to be proud of their country, and be in awe of the feats of athletes. By exploring the above topics it may be possible to determine whether there is a fair representation at the Olympics, and whether the winners are too predictable. If this is the case than the Olympics are no longer serving their purpose.

2 About the data

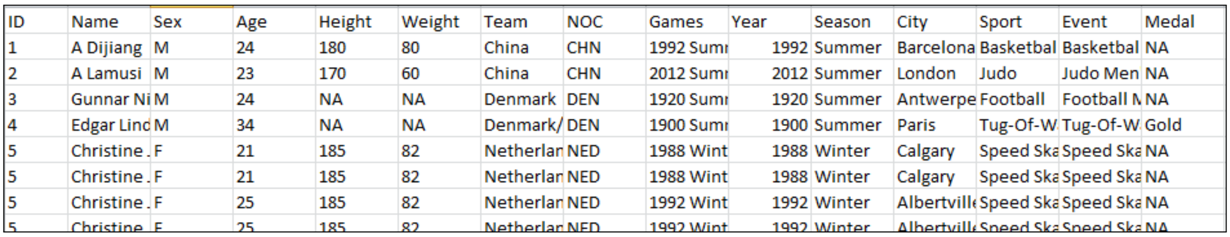
To explore and understand how the Olympics has changed over time, a variety of data was collected from numerous sources. There are three main sources broken up over five datasets.

2.1 Data Sources

2.1.1 Athlete Information

The first set of data that needs to be collected relates to the Athlete's information. This includes their physical characteristics (height, weight, age), their role in the Olympics (sport, medal, country) and when they competed (season, year). This information is available for public download on Kaggle under the title '120 years of Olympic history (1896 - 2018)'. This dataset was created by scraping from www.sport-reference.com. The data is broken down into two files:

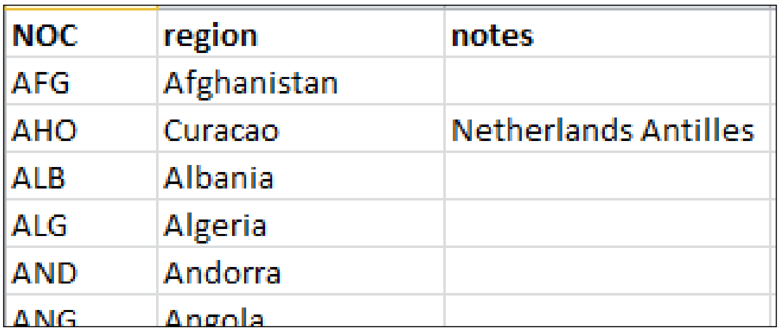
1. Athlete and Events - This file contains all of the information recorded about the athlete from all Modern Olympics. The variables of interest are ID, Sex, Age, Height, Weight, NOC, Year, Season and Medal.



ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	Sport	Event	Medal
1	A Dijiang	M	24	180	80	China	CHN	1992 Sum	1992	Summer	Barcelona	Basketbal	Basketbal	NA
2	A Lamusi	M	23	170	60	China	CHN	2012 Sum	2012	Summer	London	Judo	Judo Men	NA
3	Gunnar Ni	M	24	NA	NA	Denmark	DEN	1920 Sum	1920	Summer	Antwerpe	Football	Football M	NA
4	Edgar Linc	M	34	NA	NA	Denmark/	DEN	1900 Sum	1900	Summer	Paris	Tug-Of-W	Tug-Of-W	Gold
5	Christine .	F	21	185	82	Netherlan	NED	1988 Wint	1988	Winter	Calgary	Speed Ska	Speed Ska	NA
5	Christine .	F	21	185	82	Netherlan	NED	1988 Wint	1988	Winter	Calgary	Speed Ska	Speed Ska	NA
5	Christine .	F	25	185	82	Netherlan	NED	1992 Wint	1992	Winter	Albertvill	Speed Ska	Speed Ska	NA
5	Christine .	F	25	185	82	Netherlan	NED	1992 Wint	1992	Winter	Albertvill	Speed Ska	Speed Ska	NA

Figure 1: athlete_events.csv

2. NOC regions - A list of the countries and their NOC code. It is important to note some countries changed their code in the data. This is noted in Appendix A.



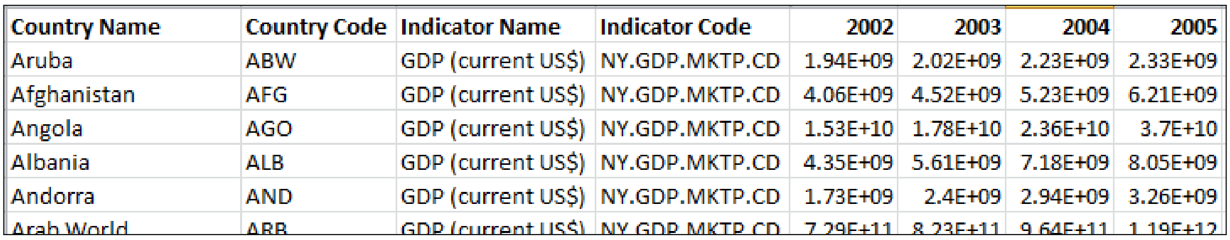
NOC	region	notes
AFG	Afghanistan	
AHO	Curacao	Netherlands Antilles
ALB	Albania	
ALG	Algeria	
AND	Andorra	
ANG	Angola	

Figure 2: noc_regions.csv

2.1.2 Country Information

The second set of data relates to the information about each country, including their GDP and population. The most trustworthy source for this data publicly available from World Bank national accounts data, and OECD National Accounts data files. The data is available from 1960 to present, and is accessed as separate files.

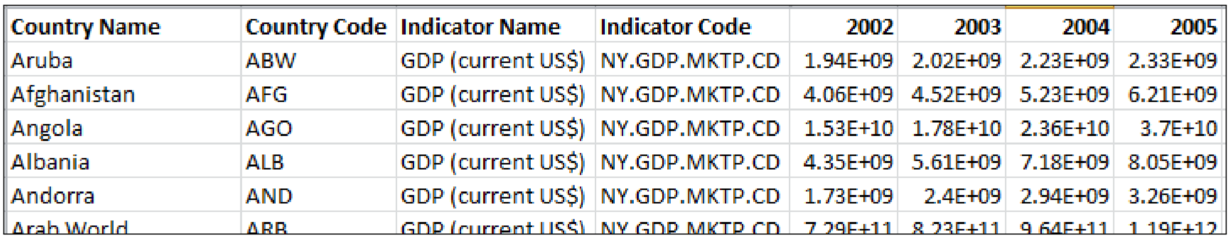
1. GDP - The GDP for all countries, represented in current US$.



Country Name	Country Code	Indicator Name	Indicator Code	2002	2003	2004	2005
Aruba	ABW	GDP (current US$)	NY.GDP.MKTP.CD	1.94E+09	2.02E+09	2.23E+09	2.33E+09
Afghanistan	AFG	GDP (current US$)	NY.GDP.MKTP.CD	4.06E+09	4.52E+09	5.23E+09	6.21E+09
Angola	AGO	GDP (current US$)	NY.GDP.MKTP.CD	1.53E+10	1.78E+10	2.36E+10	3.7E+10
Albania	ALB	GDP (current US$)	NY.GDP.MKTP.CD	4.35E+09	5.61E+09	7.18E+09	8.05E+09
Andorra	AND	GDP (current US$)	NY.GDP.MKTP.CD	1.73E+09	2.4E+09	2.94E+09	3.26E+09
Arab World	ARB	GDP (current US$)	NY.GDP.MKTP.CD	7.29E+11	8.23E+11	9.64E+11	1.19E+12

Figure 3: worldbank_gdp.csv

2. Population - The total population of all countries.

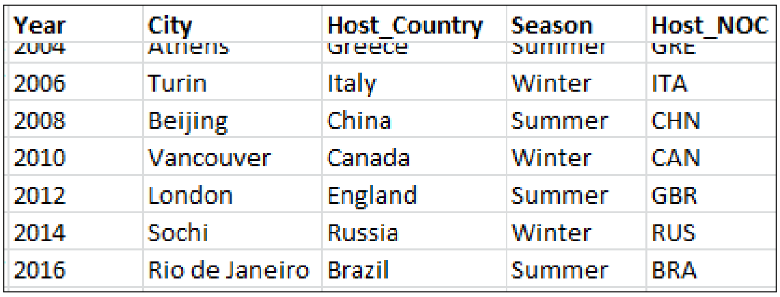


Country Name	Country Code	Indicator Name	Indicator Code	2002	2003	2004	2005
Aruba	ABW	GDP (current US$)	NY.GDP.MKTP.CD	1.94E+09	2.02E+09	2.23E+09	2.33E+09
Afghanistan	AFG	GDP (current US$)	NY.GDP.MKTP.CD	4.06E+09	4.52E+09	5.23E+09	6.21E+09
Angola	AGO	GDP (current US$)	NY.GDP.MKTP.CD	1.53E+10	1.78E+10	2.36E+10	3.7E+10
Albania	ALB	GDP (current US$)	NY.GDP.MKTP.CD	4.35E+09	5.61E+09	7.18E+09	8.05E+09
Andorra	AND	GDP (current US$)	NY.GDP.MKTP.CD	1.73E+09	2.4E+09	2.94E+09	3.26E+09
Arab World	ARB	GDP (current US$)	NY.GDP.MKTP.CD	7.29E+11	8.23E+11	9.64E+11	1.19E+12

Figure 4: worldbank_gdp.csv

2.1.3 Host Cities

The finally set of data is location of each of the games. The 'City' is included as a column in 'athlete_events.csv', however it is not paired with a country which is needed to compare an athlete's country with where they are competing. This data was not readily available as a data file but the information was found on https://architectureofthegames.net/olympic-host-cities/. The data was copied into two separate text files as is; summer and winter. Using python the files were read, reformatted and combined to create a csv file. The NOC was also added as an additional column manually using noc_regions.csv. The file contains the year, city, country, NOC and season of each Olympic games. The code is in Appendix A.1.



Year	City	Host_Country	Season	Host_NOC
2004	Athens	Greece	Summer	GRE
2006	Turin	Italy	Winter	ITA
2008	Beijing	China	Summer	CHN
2010	Vancouver	Canada	Winter	CAN
2012	London	England	Summer	GBR
2014	Sochi	Russia	Winter	RUS
2016	Rio de Janeiro	Brazil	Summer	BRA

Figure 5: host_countries.csv

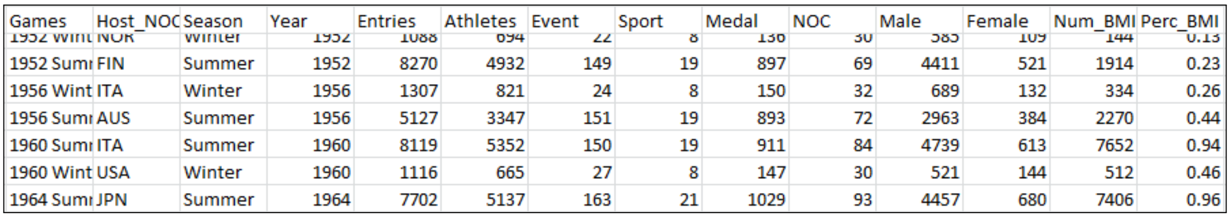
2.2 Data Parsing

2.2.1 Combined Dataset

From all of the above files, a new dataset was created using pandas (and article research) to refine the data selection, remove redundancies, combine related variables and update incorrect data to ensure a cleaner dataset for the visualisations. The code can be found in Appendix A.2. These steps were taken to combine the athlete_events.csv, host_countries.csv and noc_regions.csv:

1. Remove 'Art Competitions'
2. Remove 'Name', 'Team' from athlete_events.csv
   - 'Name' - the identification of the athletes is not important
   - 'Team' - sometimes contradicts NOC/Country
3. Make NOC codes consistent for countries that have changed.
   - Singapore (SIN): Stored as SGP in athlete_events
   - Russia (RUS): URS (1952-1988), EUN (1992), RUS (1994-2018)
   - Taiwan (TPE): ROC (1952-1976), TPE(1984-2018)
   - China (CHN): ROC (1924-1948), CHN (1980-2018)
   - Germany (GER): GER (1896-2018), EUA (1956-1964), FRG & GDR (1968-1988)
   - Czech Republic (CZE): CZE (1994-2018), TCH (1920-1992), BOH (1900-1912)
   - Serbia (SRB): SCG (2004-2006), SRB (1912, 2008-2018), YUG (1920-2002)
4. Add column COUNTRY by matching 'NOC' with the same from noc_regions.csv
5. Update host CITY in athlete_events.csv to match more common names used in host_cities.csv
   - Athina to Athens
   - Roma to Rome
   - Antwerpen to Antwerp
   - Moskva to Moscow
   - Torino to Turin
   - Sankt Moritz to St Moritz
6. Add column HOST NOC by matching 'NOC' with same from host_city.csv
7. Add column for BMI using the formula Weight (kg) / Height$\hat{2}$ (m)
   - Height is in centimetres in the dataset so needs to be converted to metres (Height / 100)
8. Add column with Boolean value corresponding to whether an athlete is a medal winner

9. Add column GDP by matching ‘country’ in gdp.csv
    - File listed with years as columns
    - ’Melt’ the table read in to create row entry for each NOC and year
    - Convert values to billions (divide by 1,000,000,000)
10. Add column POPULATION by matching ‘country’ in population.csv
    - The same procedure as GDP
    - Convert the values to millions (divide by 1,000,000)



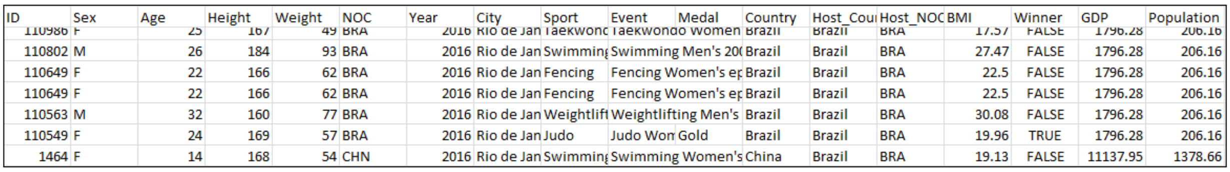
Games	Host_NOC	Season	Year	Entries	Athletes	Event	Sport	Medal	NOC	Male	Female	Num_BMI	Perc_BMI
1952 Wint	NOR	Winter	1952	1088	694	22	8	136	30	585	109	144	0.13
1952 Sumr	FIN	Summer	1952	8270	4932	149	19	897	69	4411	521	1914	0.23
1956 Wint	ITA	Winter	1956	1307	821	24	8	150	32	689	132	334	0.26
1956 Sumr	AUS	Summer	1956	5127	3347	151	19	893	72	2963	384	2270	0.44
1960 Sumr	ITA	Summer	1960	8119	5352	150	19	911	84	4739	613	7652	0.94
1960 Wint	USA	Winter	1960	1116	665	27	8	147	30	521	144	512	0.46
1964 Sumr	JPN	Summer	1964	7702	5137	163	21	1029	93	4457	680	7406	0.96

Figure 6: games_total_draft.csv

At this point, the data was looking at more intensely and some preliminary graphs were tested. After reviewing this information, there were a few additional changes to be made.

1. Remove Years 1896-1952 from athlete_events.csv
    - Women didn’t compete in 1896
    - Winter Games didn’t commence until 1924
    - China and USSR joined in 1952
    - The recording of more than 40% of weight and height started in 1956. Over 90% since 1960.
    - The GDP and population is not available until 1960
2. Remove ’Winter’ season, as not only will this overcomplicate the results and comparisons, but also Summer has a lot more data which will skew the outputs.
    - Summer has 2.3 times more countries each year
    - Summer has 2.3 times more sports each year
    - Summer has 4 times more athletes competing
    - summer has 3 times more events
3. Remove ’Season’ and ’Games’ columns
    - Season is no longer relevant since only looking at Summer
    - Games was unique identifier between seasons, without season comparison it is just a duplicate of Year and Season

With these changes, a new table was created with just Summer Olympics from 1956 with the above data. The code for this process is in Appendix A.2.



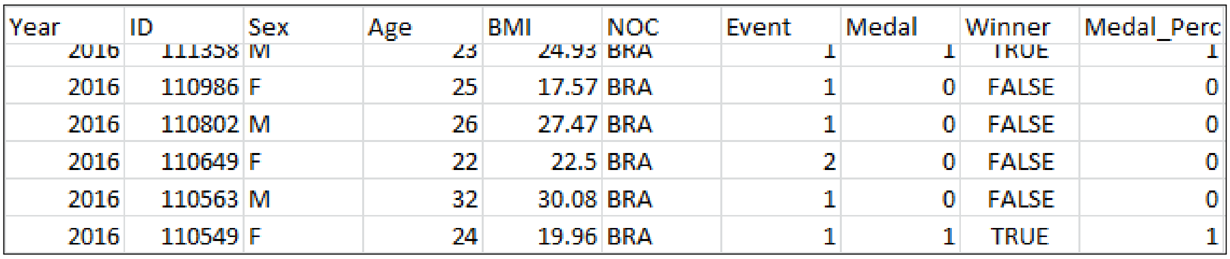
ID	Sex	Age	Height	Weight	NOC	Year	City	Sport	Event	Medal	Country	Host_Cou	Host_NOC	BMI	Winner	GDP	Population
110986	F	25	167	49	BRA	2016	Rio de Jan	Taekwonc	Taekwondo Women		Brazil	Brazil	BRA	17.57	FALSE	1796.28	206.16
110802	M	26	184	93	BRA	2016	Rio de Jan	Swimming	Swimming Men's 20(		Brazil	Brazil	BRA	27.47	FALSE	1796.28	206.16
110649	F	22	166	62	BRA	2016	Rio de Jan	Fencing	Fencing Women's ep		Brazil	Brazil	BRA	22.5	FALSE	1796.28	206.16
110649	F	22	166	62	BRA	2016	Rio de Jan	Fencing	Fencing Women's ep		Brazil	Brazil	BRA	22.5	FALSE	1796.28	206.16
110563	M	32	160	77	BRA	2016	Rio de Jan	Weightlift	Weightlifting Men's		Brazil	Brazil	BRA	30.08	FALSE	1796.28	206.16
110549	F	24	169	57	BRA	2016	Rio de Jan	Judo	Judo Won	Gold	Brazil	Brazil	BRA	19.96	TRUE	1796.28	206.16
1464	F	14	168	54	CHN	2016	Rio de Jan	Swimming	Swimming Women's		China	Brazil	BRA	19.13	FALSE	11137.95	1378.66

Figure 7: all_data.csv

2.2.2 Totals

To adequately explore the summer dataset there were a number of aggregations that needed to be performed to calculate the total values of certain variables. Using panda data frames new subsets of the data were created to allow repeated access to these aggregations. The code is in Appendix A.3.

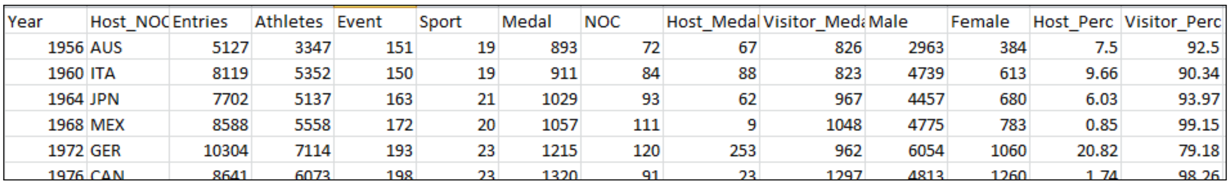
1. **The Athletes** - This subset was created to refine the information relating to the individual athletes. This dataset is not considered with the type of sports or events the athlete participates in nor the year.
    - Group the athletes by their ID so there is only one row per athlete, rather than a row for each entry by that athlete
    - Count the number of different sports the athlete competes in
    - Count the number of entries the athlete has in the summer dataset



Year	ID	Sex	Age	BMI	NOC	Event	Medal	Winner	Medal_Perc
2016	111358	M	23	24.93	BRA	1	1	TRUE	1
2016	110986	F	25	17.57	BRA	1	0	FALSE	0
2016	110802	M	26	27.47	BRA	1	0	FALSE	0
2016	110649	F	22	22.5	BRA	2	0	FALSE	0
2016	110563	M	32	30.08	BRA	1	0	FALSE	0
2016	110549	F	24	19.96	BRA	1	1	TRUE	1

Figure 8: athlete_total.csv

2. **The Games** - This table breaks down the data into one entry per Games. It records the year and host country as well as the number of entries, events, sports, athletes (also broken down into Male and Female) and medals (also split into host and visitor amounts and percentages.)
    - Group the entries by the year, so there is only one entry per games
    - Keep column with Host Country Code
    - Count the number of entries for that year, store as 'Entries'
    - Count the number of athletes for that year (only one entry per athlete ID) as 'Athletes'
    - Count the number of unique events held that year as 'Event'
    - Count the number of unique sports hosted that year as 'Sport'
    - Count the number of medals awarded that year as 'Medal'
    - Add column 'Host_Medal' to record the number of medals awarded to the host country
    - Add column 'Visitor_Medal' to records all medals not awarded to host (total - host)
    - Count how many male athletes entered as 'Male'
    - Count how many female athletes entered as 'Female'
    - Calculate the percentage of medals awarded to the host
    - Calculate the percentage of medals awarded to all others



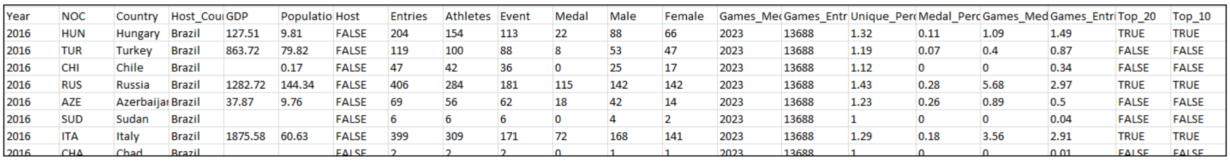
Year	Host_NOC	Entries	Athletes	Event	Sport	Medal	NOC	Host_Medal	Visitor_Medi	Male	Female	Host_Perc	Visitor_Perc
1956	AUS	5127	3347	151	19	893	72	67	826	2963	384	7.5	92.5
1960	ITA	8119	5352	150	19	911	84	88	823	4739	613	9.66	90.34
1964	JPN	7702	5137	163	21	1029	93	62	967	4457	680	6.03	93.97
1968	MEX	8588	5558	172	20	1057	111	9	1048	4775	783	0.85	99.15
1972	GER	10304	7114	193	23	1215	120	253	962	6054	1060	20.82	79.18
1976	CAN	8641	6073	198	23	1320	91	23	1297	4813	1260	1.74	98.26

Figure 9: games_total.csv

3. **The Countries** - Finally, this dataset looks at the data from the perspective of the

countries participating. There is an entry for each country and each games they compete it. As well as their code and name, this table also includes the GDP and population for the year, whether they were the host that year, as well as the same information as the games, except country specific.

- Group the entries by the year and country code
- Keep column with country code, name, GDP, population and whether they were the host
- Count the number of entries for that year, store as 'Entries'
- Count the number of athletes for that year (only one entry per athlete ID) as 'Athletes'
- Count the number of unique events held that year as 'Event'
- Count the number of unique sports hosted that year as 'Sport'
- Count the number of medals awarded that year as 'Medal'
- Count how many male athletes entered as 'Male'
- Count how many female athletes entered as 'Female'
- Include number of medals and entries for each games
- Calculate the percentage of medals awarded to the country from the total
- Calculate the percentage of entries from the country compared to the total
- Calculate the average number of events per athlete to determine uniqueness
- Add column of Boolean whether country is in top 20 of total medals since 1956
- Add column of Boolean whether country is in top 10 of total medals since 1956



Year	NOC	Country	Host_Cou	GDP	Populatio	Host	Entries	Athletes	Event	Medal	Male	Female	Games_Me	Games_Entr	Unique_Per	Medal_Perc	Games_Med	Games_Entri	Top_20	Top_10
2016	HUN	Hungary	Brazil	127.51	9.81	FALSE	204	154	113	22	88	66	2023	13688	1.32	0.11	1.09	1.49	TRUE	TRUE
2016	TUR	Turkey	Brazil	863.72	79.82	FALSE	119	100	88	8	53	47	2023	13688	1.19	0.07	0.4	0.87	FALSE	FALSE
2016	CHI	Chile	Brazil		0.17	FALSE	47	42	36	0	25	17	2023	13688	1.12	0	0	0.34	FALSE	FALSE
2016	RUS	Russia	Brazil	1282.72	144.34	FALSE	406	284	181	115	142	142	2023	13688	1.43	0.28	5.68	2.97	TRUE	TRUE
2016	AZE	Azerbaijar	Brazil	37.87	9.76	FALSE	69	56	62	18	42	14	2023	13688	1.23	0.26	0.89	0.5	FALSE	FALSE
2016	SUD	Sudan	Brazil			FALSE	6	6	6	0	4	2	2023	13688	1	0	0	0.04	FALSE	FALSE
2016	ITA	Italy	Brazil	1875.58	60.63	FALSE	399	309	171	72	168	141	2023	13688	1.29	0.18	3.56	2.91	TRUE	TRUE
2016	CHA	Chad	Brazil			FALSE	2	2	2	0	1	1	2023	13688	1	0	0	0.01	FALSE	FALSE

Figure 10: noc_total.csv

3 Discussion

The prediction of medal winners will be explored through a number of topics. Due to the difference n data size and added complexity, as explained in the data section, only data from the Summer Olympics since 1956 will be considered. The following questions will be explored and discussed.

- How have the games changed?
- What are the characteristics of an Olympic Athlete?
- Is there a difference in physicality between athletes and winners?
- Which countries are the best at the Olympics and how do they differ?
- Is the competition fair?

3.1 The Games

There are many factors associated with the Olympic games including the number of entries, athletes (male and female), countries, events, sports and medals. The below histogram provides a sense of the distribution of these factors for all Modern Summer Olympics. A histogram is used to show how many occurrences there are of a single variable and grouping the data into bins (i.e. sets) to give a preliminary idea of the data. Additionally, the data was separated into two subgroups; games before and after 1955

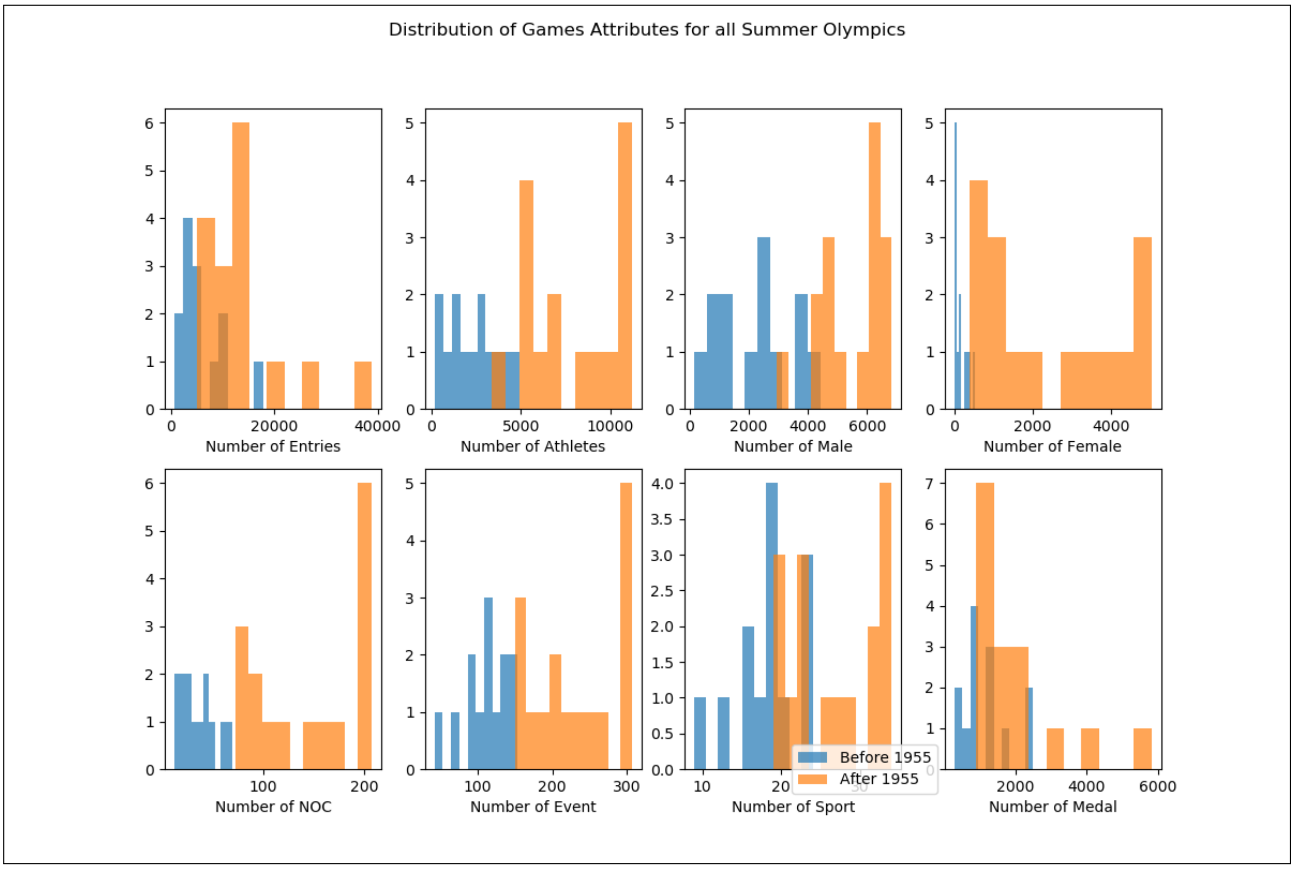


Figure 11: Distribution of variables in Games

By doing so it is evident that the games have changed since the first 60 years of competition. All of the factors are considerably higher after 1955. Notably the number of women prior to 1955 was never higher than 500 competitors, but after this time, at least half of the games has seen more than 3,000 female competitors. Also, the number of countries competing has more than doubled with 6 of the games in the last 60 years seeing approximately 200 countries competing. From this graph it is evident that the Summer Olympic games have significantly diversified and become more accessible to more athletes around the world.

3.2 The Athletes

Without the athletes there would be no Olympic Games, so naturally the next topic to explore is the characteristics of the Olympiads themselves. Besides their country and their sport of choice, the defining characteristics of an athlete are their Age and BMI. Other interesting factors include how many events they compete in and how many medals on average an athlete wins.

Firstly, the below boxplot shows the central tendency of athlete's age over the last 60 years of games. Additionally, the data are split by an athlete's sex to show the difference between male and female over time. A boxplot shows the spread of data as a compact alternative to a histogram, that highlights the general nature of the data. It identifies the median, 25th percentile and 75th percentile as the box, as well as low and high adjusters with outliers consistently.

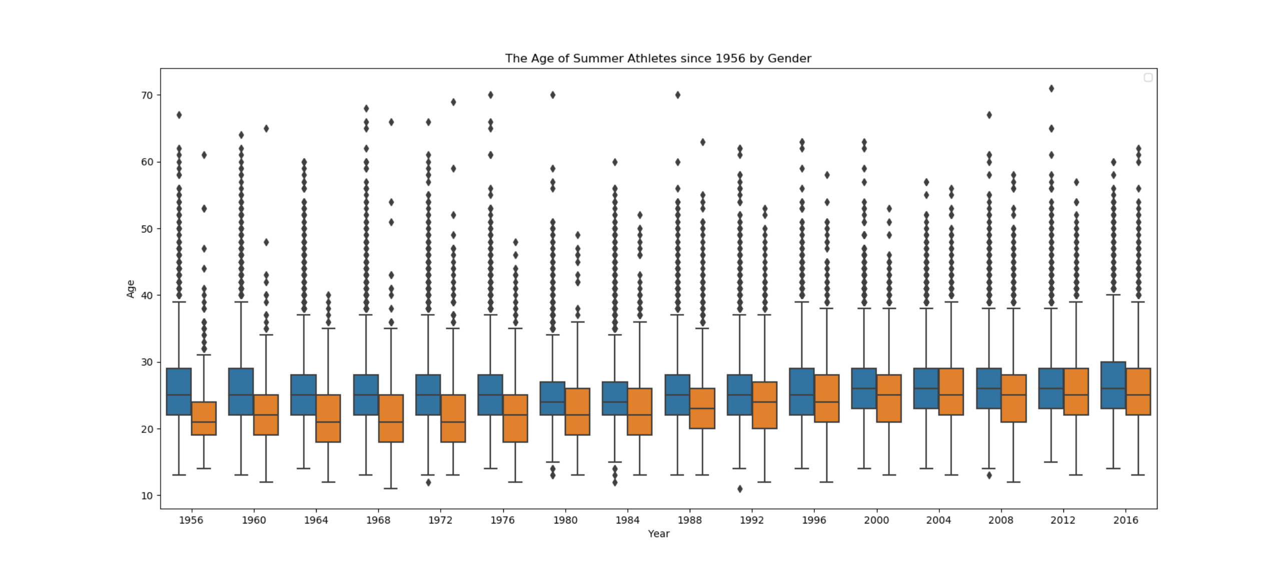


Figure 12: Distribution of Age of Athletes by Year

From this boxplot it is evident that the age and BMI of athletes has remaining fairly consistent through ....

3.3 The Athletes

3.4 The Countries

- The distribution of the number of events and medals [Histogram (#events, #medal)]
- The change in age for medal and non-medal winners [Turkey box (year v. age)]
- The BMI of medal and non-medal winners [q-q plot (medal v. non-medal BMI)]
- The proportion of medal winners to number athletes [scatter (#athletes v. #medals)]
- The proportion of men and women competing [scatter (#females v. #males)]
- The difference in % of medals when competing at host [scatter (% hosting v. visiting)]
- The effect of population and GDP on number of medals [multi pop, GDP, #medals]

Appendices

A Important notes about the data

1. athlete_events.csv - Possible factors that may affect results of each Olympics
   - 1924: Winter games commence
   - 1928: Women now compete in more than 2 sports
   - 1932: Low attendance due to Great Depression
   - 1940 & 1944: Cancelled due to WW2
   - 1948: Art sports (architecture, literature, music, painting, sculpture) removed
   - 1952: USSR/Russia starts competing, Republic of China (ROC) discontinued
   - 1956: Boycotts by 8 nations, including China
   - 1960: Height and Weight measured consistently from now
   - 1976: Boycotts by 25 nations (mostly from Africa)
   - 1980: Boycotts by 66 nations, including US
   - 2000: Summer Olympics capped at 28 sports, 300 events, 10,000 athletes
2. noc_regions.csv - The following countries are recorded under multiple codes:
   - Australia: AUS, ANZ (New Zealand, 19081912)
   - Russia: URS (19521988), EUN (1992), RUS (19942018)
   - China: ROC (19241948), CHN (19522018), HKG (Hong Kong, 19522018)
   - Germany: GER (18962018), EUA (19561964), FRG & GDR (19681988)
   - Czech Republic: CZE (19942018), TCH (19201992), BOH (19001912)
   - Serbia: SCG (20042006), SRB (1912, 20082018), YUG (19202002)

Appendices

A About the data

A.1 Host Cities

```
import pandas as pd

def get_season_df(file_name, season):
    host_data = []
    with open(file_name) as file_var:
        for line in file_var.readlines():
                year = line[:4]
                location = line[6:-1].split(',␣')
                city = location[0]
                country = location[1]
                host_data.append([year, city, country, season])

    season_df = pd.DataFrame(host_data, columns=['Year',
                                                  'City',
                                                  'Host_Country',
                                                  'Season'])
    return season_df

# Read in the data for each seasons
summer_df = get_season_df(
    'F:/TEAN/Portfolio/olympics/code/host_summer.txt', 'Summer')
winter_df = get_season_df(
    'F:/TEAN/Portfolio/olympics/code/host_winter.txt', 'Winter')
# Combine to create 1 DF
host_df = pd.merge(summer_df, winter_df, how='outer')\
    .reset_index(drop = True)

#Check if all cities accounted for:
athlete_df = pd.read_csv(
            'F:/TEAN/Portfolio/olympics/data/athlete_events.csv')

for host_city in host_df.City.unique():
    for athlete_city in athlete_df.City.unique():
        if (host_city not in host_df.City.unique()) \
                and (athlete_city not in host_df.City.unique()):
            print("Host␣City:␣", host_city)
            print("Athlete␣City:␣", athlete_city)

# Write to CSV
host_df.to_csv('F:/TEAN/Portfolio/olympics/data/host_countries.csv')
```

A.2 Combined Data

```
import pandas as pd
import numpy as np


# COMPARE SUBSETS OF DATA WITH MAIN AS CHANGES
def check_item_not_in(df1, df2):
    item_list = []
    count = 0
    if df1.nunique() != df2.nunique():
        for item in df1.unique():
            if item not in df2.unique():
                item_list.append(item)
                count+=1
    return item_list,count

# PRINT CHECKS OF HOW DATA CHANGING
def checkpoint(action, all, bool=False, lost=None):
    print("{action}:␣\n␣Unique␣NOC:␣{num_noc}␣\
␣␣␣␣␣␣␣␣\n␣Unique␣Athletes:␣{num_athletes}."
        .format(action=action,
                num_noc=all.NOC.nunique(),
                num_athletes=all.ID.nunique()))

    if bool:
        print("\tLost␣NOC:␣{}␣\t␣Lost␣Athletes:␣{}"
                .format(check_item_not_in(lost.NOC, all.NOC)[0],
                check_item_not_in(lost.ID, all.ID)[1]))

########### READ IN DATASETS ##########
noc_df = pd.read_csv(
    'F:/TEAN/Portfolio/olympics/data/noc_regions.csv')
host_df = pd.read_csv(
    'F:/TEAN/Portfolio/olympics/data/host_countries.csv')
athlete_df = pd.read_csv(
    'F:/TEAN/Portfolio/olympics/data/athlete_events.csv')
worldbank_gdp = pd.read_csv(
    'F:/TEAN/Portfolio/olympics/data/worldbank_gdp.csv', index_col=0).reset_i
worldbank_pop = pd.read_csv(
    'F:/TEAN/Portfolio/olympics/data/worldbank_pop.csv', index_col=0).reset_i

############### START ##############
all_df = athlete_df
#checkpoint('START', all_df)

# 1. Remove art competitions
art_df = all_df[all_df['Sport'] == 'Art␣Competitions']
all_df = all_df.drop(art_df.index)
#checkpoint('REMOVE ART', all_df, True, art_df)
```

```
# 2. Remove irrelevant columns
extra_df = all_df
all_df = all_df.drop(["Name", "Team"], axis=1)
#checkpoint('REMOVE EXTRA', all_df, True, extra_df)

# 3. Make NOC codes consistent for countries that have changed.
noc_unique = all_df.NOC.unique()
all_df.loc[(all_df.NOC == 'TCH'),'NOC'] = 'CZE'
all_df.loc[(all_df.NOC == 'SGP'),'NOC'] = 'SIN'
all_df.loc[(all_df.NOC == 'EUN')
                  | (all_df.NOC == 'URS'),'NOC'] = 'RUS'
all_df.loc[(all_df.NOC == 'FRG')
                  | (all_df.NOC == 'GDR'),'NOC'] = 'GER'
all_df.loc[(all_df.NOC == 'SCG')
                  | (all_df.NOC == 'YUG'),'NOC'] = 'SRB'
#checkpoint('UPDATE NOC', all_df)

# 4. Add Country Column to match NOC
all_df = all_df.merge(noc_df[['region', 'NOC']]
                        .rename(columns={'region':'Country'})) \
                        .reset_index(drop = True)
#checkpoint('ADD COUNTRY', all_df)

# 5. Update Host City names that don't match
all_df.loc[(all_df.City == 'Athina'),'City'] = 'Athens'
all_df.loc[(all_df.City == 'Roma'),'City'] = 'Rome'
all_df.loc[(all_df.City == 'Antwerpen'),'City'] = 'Antwerp'
all_df.loc[(all_df.City == 'Moskva'),'City'] = 'Moscow'
all_df.loc[(all_df.City == 'Torino'),'City'] = 'Turin'
all_df.loc[(all_df.City == 'Sankt␣Moritz'),'City'] = 'St.␣Moritz'
#checkpoint('UPDATE HOST', all_df)

# 6. Add Host Country NOC
all_df = all_df.merge(host_df[['Host_Country', 'Host_NOC', 'City']]) \
                              .sort_values("Year") \
                              .reset_index(drop = True)
#checkpoint('ADD HOST COUNTRY', all_df)

# 7. Add BMI columns [Weight (kg) / Height^2 (m)]
all_df['BMI'] = all_df.apply(lambda x: round(x.Weight/((x.Height/100)**2), 2)

# 8. Add boolean to mark who is a medal winner
all_df['Winner'] = all_df.Medal.notna()


# 9. Add GDP
# Get GDP and merge with noc_total, divide by 1billion
worldbank_gdp = worldbank_gdp.drop(['Indicator␣Name', 'Indicator␣Code', 'Unna
worldbank_gdp = worldbank_gdp.melt(id_vars="Country␣Code",
        var_name=”Year”,
        value_name=”GDP”)
worldbank_gdp.columns = ([’NOC’, ’Year’, ’GDP’])
worldbank_gdp.sort_values(’Year’)
worldbank_gdp[’Year’] = pd.to_numeric(worldbank_gdp.Year)
worldbank_gdp[’GDP’] = round(worldbank_gdp[’GDP’].divide(1000000000), 2)

all_df = all_df.merge(worldbank_gdp, how=’left’)

# 10. Add Population
worldbank_pop = worldbank_pop.drop([’Indicator␣Name’, ’Indicator␣Code’, ’Unna
worldbank_pop = worldbank_pop.melt(id_vars=”Country␣Code”,
        var_name=”Year”,
        value_name=”Population”)
worldbank_pop.columns = ([’NOC’, ’Year’, ’Population’])
worldbank_pop.sort_values(’Year’)
worldbank_pop[’Year’] = pd.to_numeric(worldbank_pop.Year)
worldbank_pop[’Population’] = round(worldbank_pop[’Population’].divide(100000
all_df = all_df.merge(worldbank_pop, how=’left’)

# Summer 1956 Olympics Equistrian events in Sweden − update to reflect actua
all_df.loc[(all_df.Host_NOC == ’SWE’),’Host_NOC’] = ’AUS’
all_df.loc[(all_df.City == ’Stockholm’),’City’] = ’Melbourne’
all_df.loc[(all_df.Host_Country == ’Sweden’),’Host_Country’] = ’Australia’


#############TEST THE DATA###############
## LOOK AT OVERVIEW OF GAMES DATA
games_total_ath = all_df.groupby([’Games’]).ID.count().reset_index()
games_total_ath.columns = [’Games’, ’Entries’]
games_athletes = all_df.groupby([’Games’]).ID.nunique().reset_index()
games_athletes.columns = [’Games’, ’Athletes’]
games_events = all_df.groupby([’Games’]).Event.nunique().reset_index()
games_sports = all_df.groupby([’Games’]).Sport.nunique().reset_index()
games_medals = all_df.groupby([’Games’]).Medal.count().reset_index()
games_countries = all_df.groupby([’Games’]).NOC.nunique().reset_index()
games_male = all_df[all_df[’Sex’] == ’M’].groupby(’Games’).ID.nunique().reset
games_male.columns = [’Games’, ’Male’]
games_female = all_df[all_df[’Sex’] == ’F’].groupby(’Games’).ID.nunique().res
games_female.columns = [’Games’, ’Female’]
games_BMI = all_df[~all_df[’BMI’].isna()].groupby(’Games’, as_index=False).ID
games_BMI.columns = [’Games’, ’Num_BMI’]

games_host = all_df[all_df[’NOC’] == all_df[’Host_NOC’]].groupby(’Games’).Me
games_host.columns = [’Games’, ’Host_Medal’]
games_visitor = all_df[all_df[’NOC’] != all_df[’Host_NOC’]].groupby(’Games’).
games_visitor.columns = [’Games’, ’Visitor_Medal’]
games_total_df = all_df[[’Games’, ’Host_NOC’, ’Season’, ’Year’]]
games_total_df = games_total_df.drop_duplicates()
games_total_df = games_total_df.merge(games_total_ath, how=’outer’) \
                                    .merge(games_athletes, how='outer')\
                                    .merge(games_events, how='outer')\
                                    .merge(games_sports, how='outer')\
                                    .merge(games_medals, how='outer')\
                                    .merge(games_countries, how='outer')\
                                    .merge(games_male, how='outer')\
                                    .merge(games_female, how='outer')\
                                     .merge(games_BMI, how='outer')
# Check the percentage of weight and height recorded for each athlete
games_total_df['Perc_BMI'] = round(games_total_df.Num_BMI / games_total_df.E
print(games_total_df)

# Write to file before further changes for pre 1956 data
games_total_df.to_csv('F:/TEAN/Portfolio/olympics/data/games_total_draft.csv'



######### UPDATE DATA FOR SUMMER ONLY FROM 1956 ##########

# 11. Remove winter
winter_df = all_df[all_df['Season'] == 'Winter']
all_df = all_df.drop(winter_df.index)
#checkpoint('REMOVE WINTER', all_df, True, winter_df)

# 12. Remove years
years_df = all_df[all_df['Year'].isin(range(1896,1955))]
all_df = all_df.drop(years_df.index)
#checkpoint('REMOVE YEARS', all_df, True, years_df)

# 13. Remove irrelevant columns
extra_df = all_df
all_df = all_df.drop(["Season", 'Games'], axis=1) #add season, games
#checkpoint('REMOVE EXTRA', all_df, True, extra_df)




######## WRITE TO FILE ##########
all_df.to_csv('F:/TEAN/Portfolio/olympics/data/all_data.csv')
```

A.3 Combined Data

```
import pandas as pd

# Read in summer data only
all_df = pd.read_csv(
    'F:/TEAN/Portfolio/olympics/data/summer_data.csv', index_col=0).reset_ind

######## THE ATHLETE #######
# Athlete Totals (Games, Year, ID, Sex, Age, BMI, Season, NOC, #Events, #Med
athlete_events = all_df.groupby(['Year', 'ID']).Event.count().reset_index()
athlete_medals = all_df.groupby(['Year', 'ID']).Medal.count().reset_index()
athlete_total_df = all_df[['Year', 'ID', 'Sex', 'Age', 'BMI', 'NOC']] #Remove
athlete_total_df = athlete_total_df.drop_duplicates()
athlete_total_df = athlete_total_df.merge(athlete_events, how='outer') \
                                    .merge(athlete_medals, how='outer')
athlete_total_df['Winner'] = athlete_total_df['Medal'] != 0
athlete_total_df['Medal_Perc'] = round((athlete_total_df.Medal / athlete_tota

print(athlete_total_df)
athlete_total_df.to_csv('F:/TEAN/Portfolio/olympics/data/athlete_total.csv')

######## THE GAMES #######
# Games totals (Games, Year, #Athletes, #Medals, #Male, #Female, #Events)
games_total_ath = all_df.groupby(['Year']).ID.count().reset_index()
games_total_ath.columns = ['Year', 'Entries']
games_athletes = all_df.groupby(['Year']).ID.nunique().reset_index()
games_athletes.columns = ['Year', 'Athletes']
games_events = all_df.groupby(['Year']).Event.nunique().reset_index()
games_sports = all_df.groupby(['Year']).Sport.nunique().reset_index()
games_medals = all_df.groupby(['Year']).Medal.count().reset_index()
games_countries = all_df.groupby(['Year']).NOC.nunique().reset_index()
games_male = all_df[all_df['Sex'] == 'M'].groupby('Year').ID.nunique().reset_
games_male.columns = ['Year', 'Male']
games_female = all_df[all_df['Sex'] == 'F'].groupby('Year').ID.nunique().rese
games_female.columns = ['Year', 'Female']
# Add column for number of medals awarded to host country and the visitors
games_host = all_df[all_df['NOC'] == all_df['Host_NOC']].groupby('Year').Meda
games_host.columns = ['Year', 'Host_Medal']
games_visitor = all_df[all_df['NOC'] != all_df['Host_NOC']].groupby('Year').M
games_visitor.columns = ['Year', 'Visitor_Medal']
# Merge seperate together
games_total_df = all_df[['Year', 'Host_NOC']]
games_total_df = games_total_df.drop_duplicates()
games_total_df = games_total_df.merge(games_total_ath, how='outer') \
                                .merge(games_athletes, how='outer')\
                                .merge(games_events, how='outer')\
                                .merge(games_sports, how='outer')\
                                .merge(games_medals, how='outer')\
                                .merge(games_countries, how='outer')\
                                .merge(games_host, how='outer')\
                                .merge(games_visitor, how='outer')\
                                .merge(games_male, how='outer')\
                               .merge(games_female, how='outer')
# Add percentage of medals awarded to host and visitors
games_total_df['Host_Medal_Perc'] = round((games_total_df.Host_Medal / games
games_total_df['Visitor_Medal_Perc'] = round((games_total_df.Visitor_Medal /

print(games_total_df)
games_total_df.to_csv('F:/TEAN/Portfolio/olympics/data/games_total.csv')


######## THE COUNTRIES #######
# Group all by Year and NOC to create seperate tallies [Total Athletes, Tota
noc_total_ath = all_df.groupby(['Year', 'NOC']).ID.count().reset_index()
noc_total_ath.columns = ['Year', 'NOC', 'Entries']
noc_athletes = all_df.groupby(['Year', 'NOC']).ID.nunique().reset_index()
noc_athletes.columns = ['Year', 'NOC', 'Athletes']
noc_events = all_df.groupby(['Year', 'NOC']).Event.nunique().reset_index()
noc_medals = all_df.groupby(['Year', 'NOC']).Medal.count().reset_index()
noc_male = all_df[all_df['Sex'] == 'M'].groupby(['Year', 'NOC']).ID.nunique()
noc_male.columns = ['Year', 'NOC', 'Male']
noc_female = all_df[all_df['Sex'] == 'F'].groupby(['Year', 'NOC']).ID.nunique
noc_female.columns = ['Year', 'NOC', 'Female']
# Add the number of medals and entries from each games
games_medals = games_total_df[['Year', 'Medal']]
games_medals.columns = ['Year', 'Games_Medals']
games_athletes = games_total_df[['Year', 'Entries']]
games_athletes.columns = ['Year', 'Games_Entries']
# Merge all seperate together
noc_total_df = all_df[['Year', 'NOC', 'Country', 'Host_Country', 'GDP', 'Pop
noc_total_df = noc_total_df.drop_duplicates()
noc_total_df['Host']  = noc_total_df['Country'] == noc_total_df['Host_Country
noc_total_df = noc_total_df.merge(noc_total_ath, how='outer') \
                           .merge(noc_athletes, how='outer') \
                           .merge(noc_events, how='outer') \
                           .merge(noc_medals, how='outer') \
                           .merge(noc_male, how='outer') \
                           .merge(noc_female, how='outer') \
                           .merge(games_medals, how='outer') \
                           .merge(games_athletes, how='outer')
# Add percentage calculations of how noc did at games
noc_total_df['Unique_Perc'] = round((noc_total_df.Entries / noc_total_df.Ath
noc_total_df['Medal_Perc'] = round((noc_total_df.Medal / noc_total_df.Entries
noc_total_df['Games_Medal_Perc'] = round((noc_total_df.Medal / noc_total_df.G
noc_total_df['Games_Entries_Perc'] = round((noc_total_df.Entries / noc_total_
# Add column denoting whether they were in top 20 or top 10
top_20 = noc_total_df.groupby(['NOC'], as_index=False)['Medal'].sum().sort_va
noc_total_df['Top_20'] = noc_total_df['NOC'].isin(top_20)
noc_total_df['Top_10'] = noc_total_df['NOC'].isin(top_20[:10])
```

```
print(noc_total_df)
noc_total_df.to_csv('F:/TEAN/Portfolio/olympics/data/noc_total.csv')
```