COSC3000 - VISUALISATION REPORT

—

Modern Summer Olympics (1956-2016)

Tean-louise Cunningham (42637460)

May 6, 2020

Contents

List of Figures

1 Introduction

The topic of this project is data analysis of the Modern Summer Olympics (1956-2016). Historically, the games have been a global competition since 1896 with both Summer and Winter sports. The goal is to analyse the patterns of medal winners depending on their physical characteristics (weight, height, age, sex) and their home country (participation behaviour, GDP, population). 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.

In addition to this report a presentation can be found here. Also, the raw data and code can be found here.

2 The Data

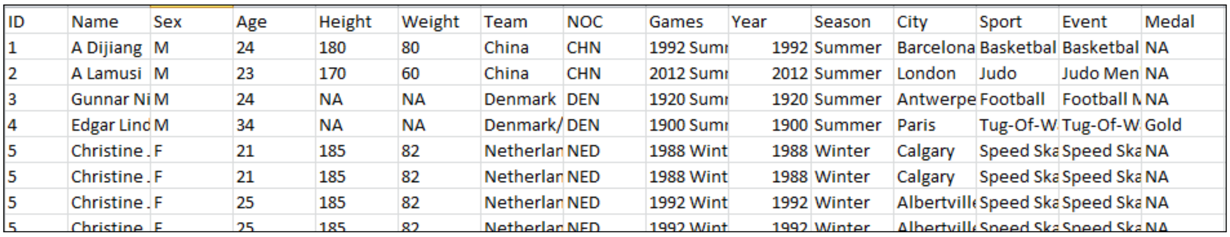
2.1 Data Sources

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.

2.1.1 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:

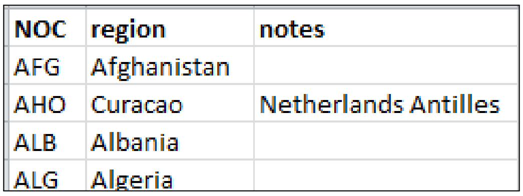
1. Athlete and Events - This file contains all of the information recorded about the athlete from all Modern Olympics. The variables of interest are ID, Sex, Age, Height, Weight, NOC, Year, Season and Medal.



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 Sumr	1992	Summer	Barcelona	Basketbal	Basketbal	NA
2	A Lamusi	M	23	170	60	China	CHN	2012 Sumr	2012	Summer	London	Judo	Judo Men	NA
3	Gunnar Ni	M	24	NA	NA	Denmark	DEN	1920 Sumr	1920	Summer	Antwerpe	Football	Football M	NA
4	Edgar Linc	M	34	NA	NA	Denmark/	DEN	1900 Sumr	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.



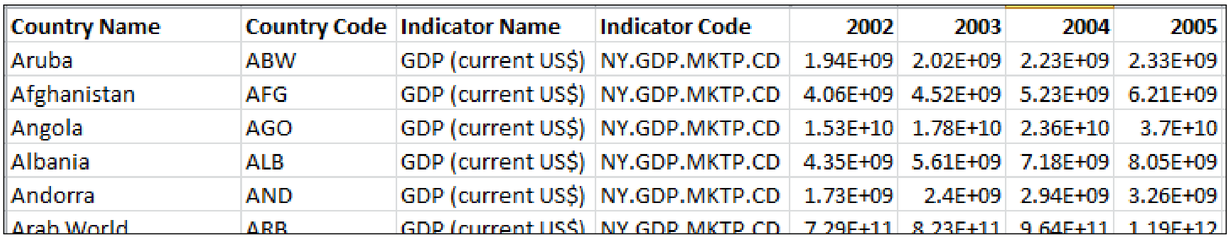
NOC	region	notes
AFG	Afghanistan	
AHO	Curacao	Netherlands Antilles
ALB	Albania	
ALG	Algeria	

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 (Worldbank, 2020).

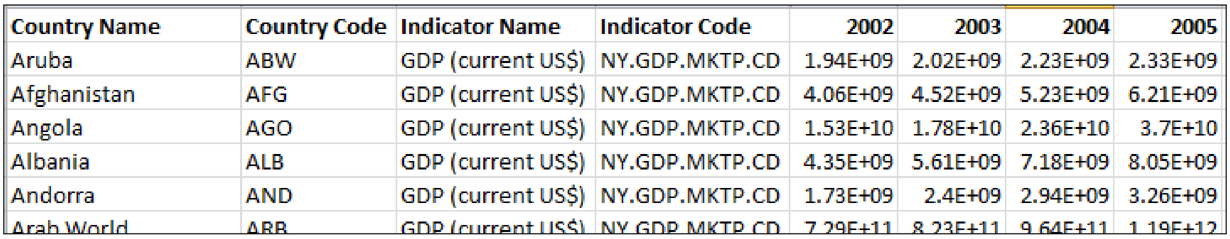
1. GDP - The GDP for all countries, represented in current US$.



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.

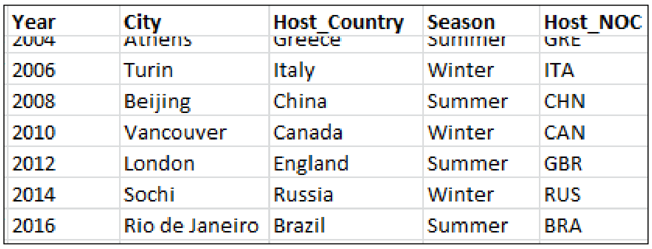


Country Name	Country Code	Indicator Name	Indicator Code	2002	2003	2004	2005
Aruba	ABW	GDP (current US$)	NY.GDP.MKTP.CD	1.94E+09	2.02E+09	2.23E+09	2.33E+09
Afghanistan	AFG	GDP (current US$)	NY.GDP.MKTP.CD	4.06E+09	4.52E+09	5.23E+09	6.21E+09
Angola	AGO	GDP (current US$)	NY.GDP.MKTP.CD	1.53E+10	1.78E+10	2.36E+10	3.7E+10
Albania	ALB	GDP (current US$)	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 final 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 Processing

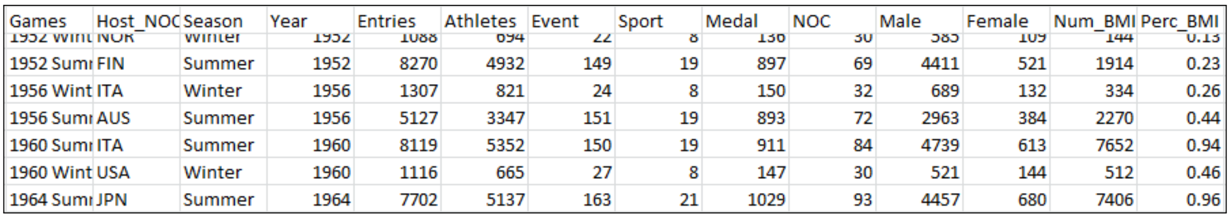
Once all the data was collected it needed to be processed in such a way that it could be used to create effective and accurate graphs. First, the datasets needed to be combined and secondly a number of aggregations needed to be done for graphs to access. Python and its pandas library was used for this process.

2.2.1 Combined Dataset

From all of the above files, a new dataset was created using the python pandas library (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**2 (m)
    - Height is centimetres in dataset, 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 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)



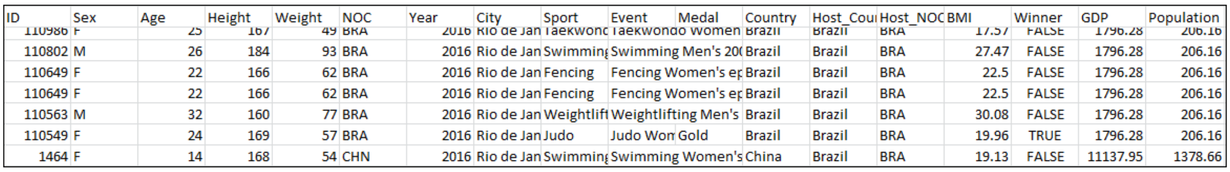
Games	Host_NOC	Season	Year	Entries	Athletes	Event	Sport	Medal	NOC	Male	Female	Num_BMI	Perc_BMI
1952 Wint	NOR	Winter	1952	1088	694	22	8	136	30	585	109	144	0.13
1952 Sumr	FIN	Summer	1952	8270	4932	149	19	897	69	4411	521	1914	0.23
1956 Wint	ITA	Winter	1956	1307	821	24	8	150	32	689	132	334	0.26
1956 Sumr	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 looked at more intensely and some preliminary graphs were tested. After reviewing this information, there were a few additional changes to be made. The data at this stage was saved to a separate file.

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
    - Since 1960, the recording of weight and height is more than 90%
    - 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.



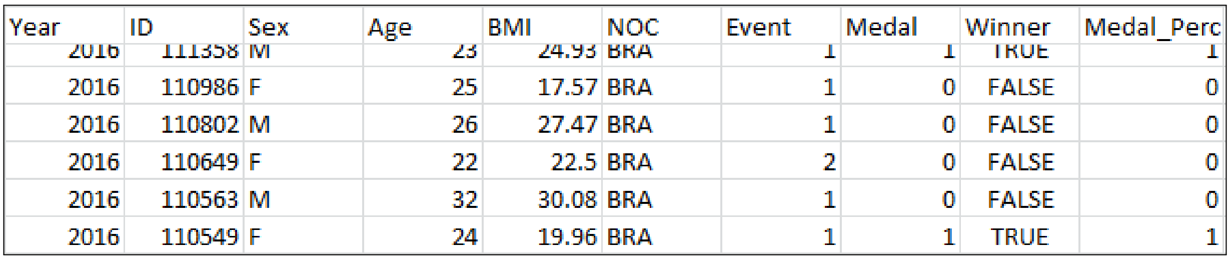
ID	Sex	Age	Height	Weight	NOC	Year	City	Sport	Event	Medal	Country	Host_Cou	Host_NOC	BMI	Winner	GDP	Population
110986	F	25	167	49	BRA	2016	Rio de Jan	Taekwonc	Taekwondo Women		Brazil	Brazil	BRA	17.57	FALSE	1796.28	206.16
110802	M	26	184	93	BRA	2016	Rio de Jan	Swimming	Swimming Men's 20(		Brazil	Brazil	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.

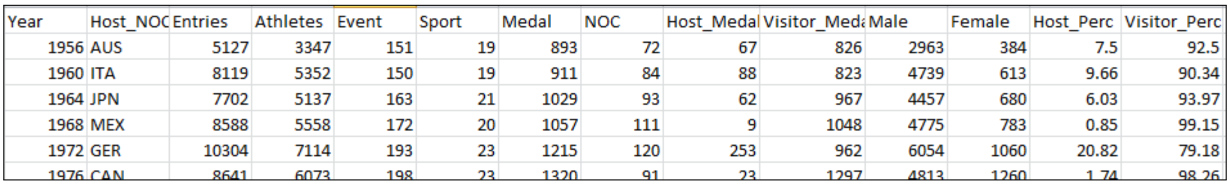
1. **The Athletes** - This subset was created to refine the information relating to the individual athletes. This dataset is not considered with the type of sports or events the athlete participates in nor the year.
    - Group the athletes by their ID so there is only one row per athlete, rather than 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
    - Update the sex labels from 'M' to 'Male' and 'F' to 'Female'



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 (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 number of medals awarded to 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



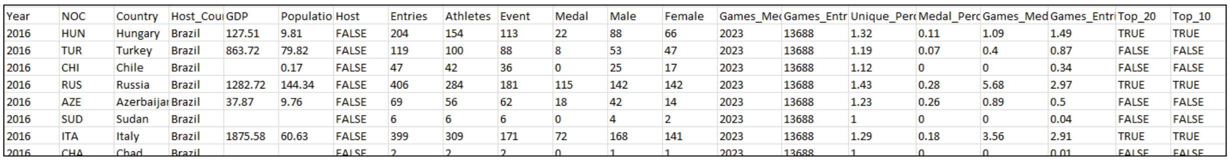
Year	Host_NOC	Entries	Athletes	Event	Sport	Medal	NOC	Host_Medal	Visitor_Meda	Male	Female	Host_Perc	Visitor_Perc
1956	AUS	5127	3347	151	19	893	72	67	826	2963	384	7.5	92.5
1960	ITA	8119	5352	150	19	911	84	88	823	4739	613	9.66	90.34
1964	JPN	7702	5137	163	21	1029	93	62	967	4457	680	6.03	93.97
1968	MEX	8588	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 columns for country code, name, GDP, population and host
- Count the number of entries for that year, store as ’Entries’
- Count the number of athletes for that year (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 in 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?
- Does the competition provide equal opportunity to all countries?

All of these graphs were created from the ’totals’ data files created during the data processing stage using python’s pandas library. All of the graphs are created using a combination of python’s matplotlib and seaborn libraries. Some additional computation use the SciPy and numpy libraries as well.

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 data used is from the games_totals.csv and games_total_before.csv files. The code for this section is in Appendix B.1.

3.1.1 Overall Distribution

To show how the Olympics has changed over time, in this instance the data set of interest (after 1955) will be compared with the data before 1955. This is the only graph that includes data before 1955. 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.

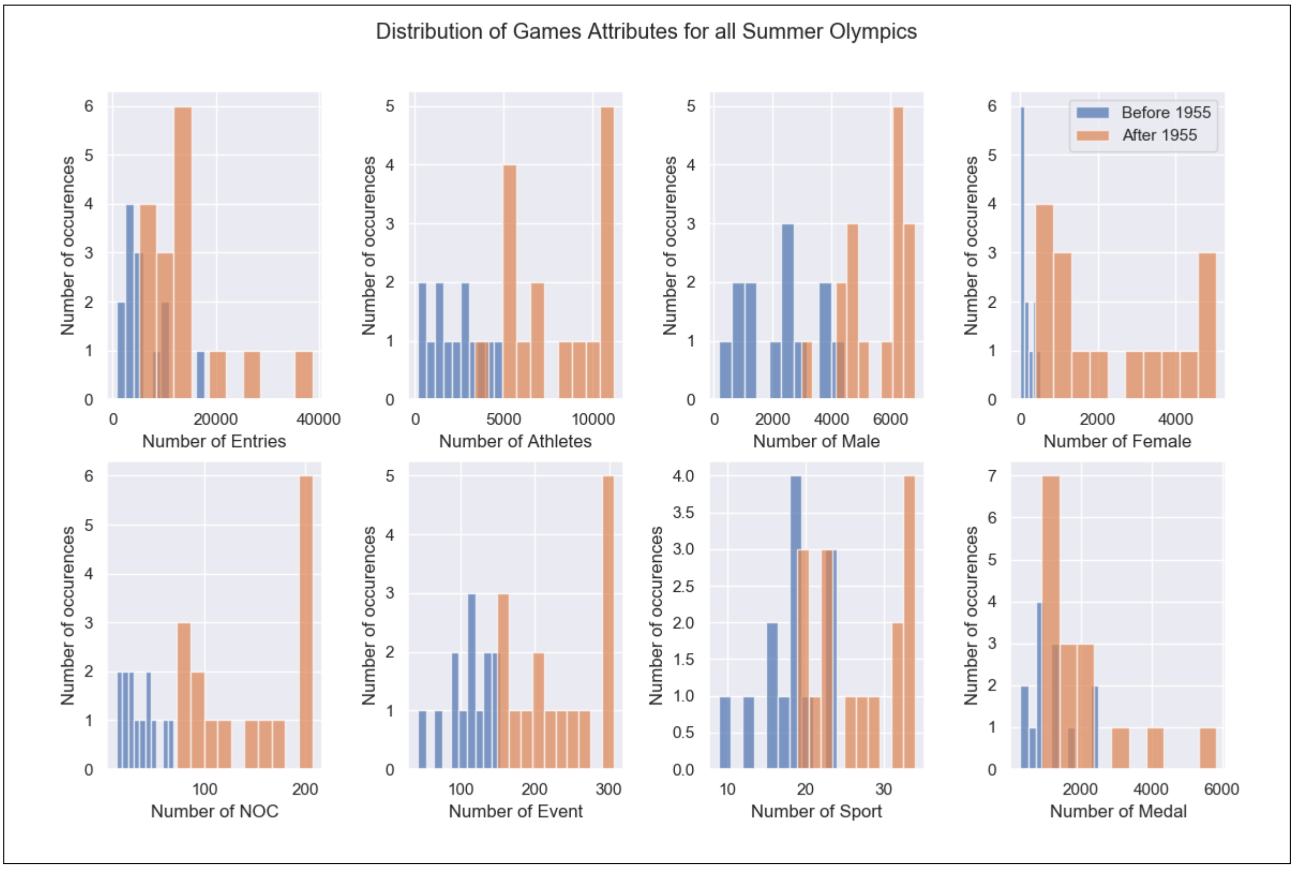


Figure 11: Distribution of Games Attributes for all Summer Olympics

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 are the characteristics of the Olympians 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. All of these graphs use the data of the totals for each unique athlete from athlete_total.csv. The total for this section is in Appendix B.2.

3.2.1 Age

To explore the change in age of athletes the below boxplot shows the central tendency of athlete's age over the last 60 years of games. Additionally, the data is 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, which highlights the general nature of the data. It identifies the median and interquartile range (25th to 75th) percentile as the box, as well as low and high adjusters with outliers consistently. Due to the high number of athletes, a boxplot was the ultimate choice to display this data as the median and interquartile range are not impacted by outliers allowing a clear view of the general trends of the athletes (matplotlib, 2020; Nieminen, 2020; Doshi, 2018)

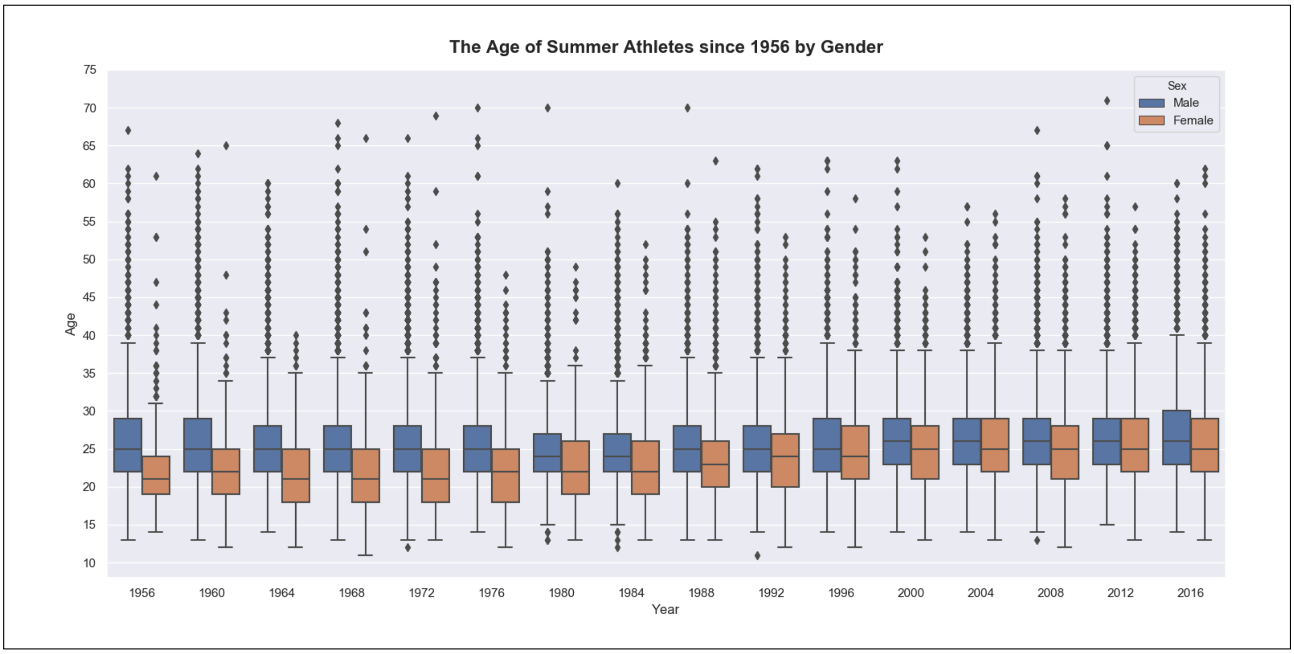


Figure 12: Box Plot - The Age of Summer Athletes since 1956 by Gender

Male: From this boxplot it is evident that the age of male athletes has remained fairly consistent. Overall the median age for males has fluctuated between 24 and 26 years old (where it currently stands) and the interquartile range has simply moved up one year.

- Median:
  - 25 years old: 1956 to 1976, then dropped slightly to,
  - 24 years old: 1980 to 1984, before returning to,
  - 25 years old: 1988 to 1996, then increasing to
  - 26 years old: 2000 to 2016. same.
- Interquartile Range:
  - The 25th percentile value over time has only increased from 22 to 23 years old.
  - The 75th percentile has increased from 29 to 30 years old

Female: From the boxplot, it is clear that female athletes in general are younger than men with lower percentile values. The median age of females has increased by four years from 21 to 25 years old. Also, the interquartile range for females has seen a more dramatic change from 19 and 24, to 22 and 29. Not only has the range shifted up by 3 years but also expanded by 2 years. As the histogram showed that the number of women competing has increased then it can also be assumed that more older woman are becoming eligible to compete as well.

- Median:
  - 21 years old: 1956, then fluctuating to
  - 22 years old: 1960 to 1984, then increase today
  - 23 years old: 1988, then increase again to
  - 24 years old: 1992 to 1996, then increase again to
  - 25 years old: 2000 to 2016.
- Interquartile range:
  - The 25th percentile decreased from 19 to 18 until 1976, before continuing to increase to 22 in 2004.
  - The 75th percentile value has continued to increase from 24 to 29.

Another interesting observation is that male athletes always appear to have more outliers in their age, however the number of outliers for women is increasing.

3.2.2 BMI

The BMI of an athlete is determined by athlete's height and weight. The equation is weight (kg) multiplied by the height (m) squared. To explore another method of viewing the central tendency distribution of a variable the change in athlete's BMI has been shown with a violin plot. Similar to the boxplot, it shows the median, interquartile range and outliers. The main difference is that instead of clear guidelines for the ranges and specific points representing the outliers, the violin plot represents this data with a kernel density estimation to show very general trends and skewness. This makes it perfect options for BMI as the interquartile range is small, compared to the extension of the outliers. The graph is also split between gender to see the difference (seaborn, 2020; Chakure, 2019)

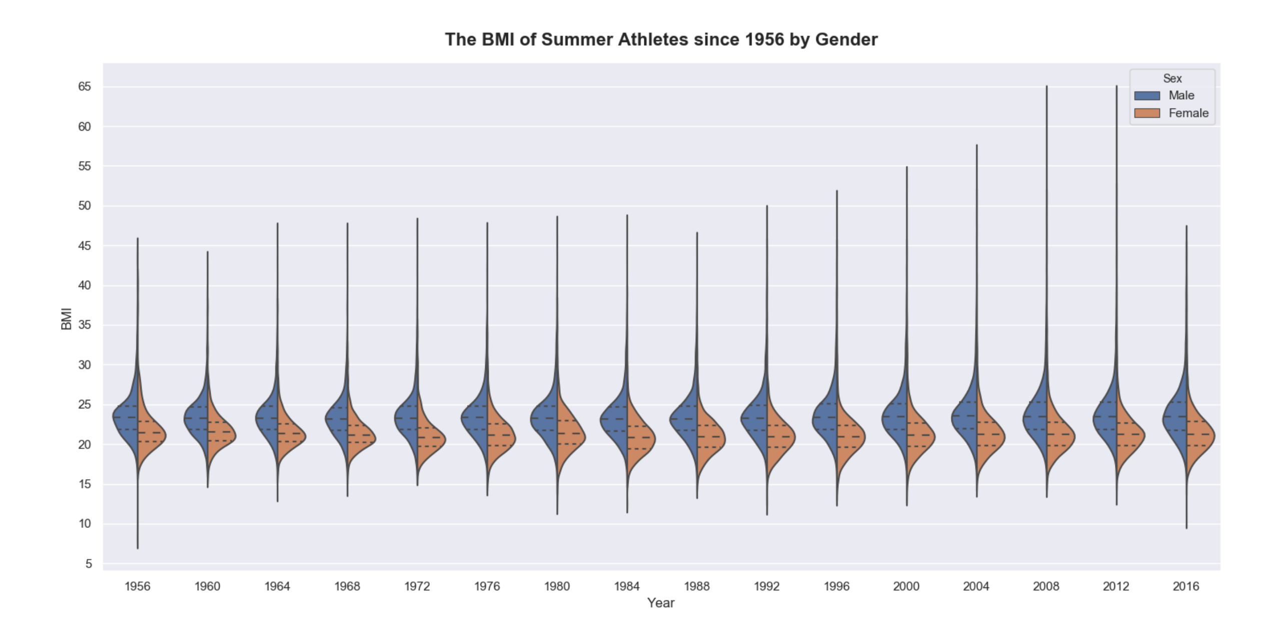


Figure 13: Violin Plot - The BMI of Summer athletes since 1956 by gender

From the plot, it is again evident that the values for females are slightly lower. It is also clear that the general trend of BMI within the male and female groups are the roughly the same, with similar unimodal density shapes. The median and interquartile ranges for both genders has remained almost consistent for the last 60 years. Males range from 22 to 25, with a median of approximately 23.5, whilst females range from 20 to 23 with a median of approximately 21.5. The biggest variation over time has been the number of outliers. It appears in the last 5 games the variation of outlier has increased since 1992 from 50 to 65. This graph suggests that the BMI of most athletes has not changed in the last 60 years, though in some events a higher BMI is advantageous.

3.2.3 Events

The next aspect of an athlete is the number of events they participate in. By exploring this topic, some insights into how diverse the athlete is can be gained. To plot this data a bar graph was used, due to the very small variation in the behaviour of most athletes and the extremes of the outliers. A bar graph calculates the mean and a confidence interval around this estimate which is displayed with error bars (seaborn, 2020; Lin, 2018)

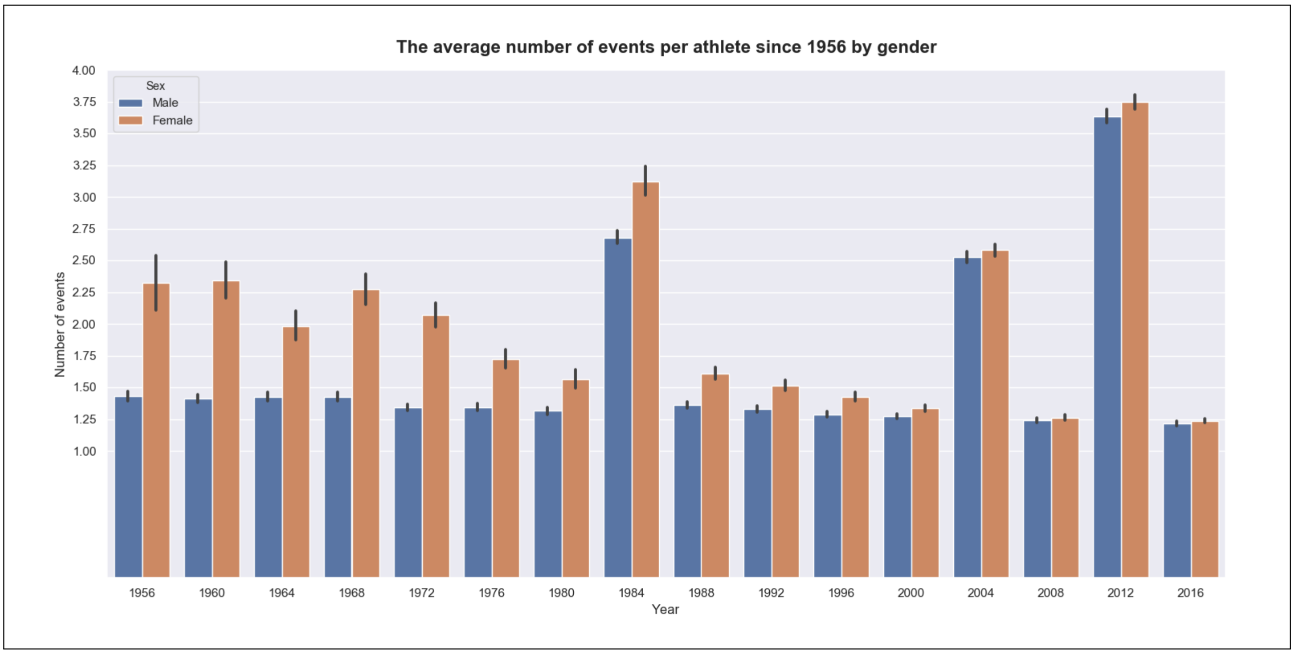


Figure 14: Bar Plot - The average number of events per Summer athlete since 1956 by gender

The first observation to note is that the number of events competed in by each athlete increase by almost 1.5 events from the previous year in 1984, 2004 and 2002. Taking another look at the data and games histogram, it is evident that there are a least 3 games that experience at very large increase in number of entries and medals awarded.

Males: The number of events on average of only changed from approximately 1.4 events to 1.25 over the last 60 years, not considering the three outlying years. The error bar has also remained the same and small.

Females: From 1956 to 1960, the number of events per female is almost 1 event extra than males. The trend of being higher continues for the rest of the games, although it decreases considerably by 2000. Ignoring the outlying years, the number of events per female has decreased from 2.3 (almost 1.5 events more than males) to be almost the same average as males of 1.25 events per athlete. Also, the error bar decreases greatly to be consistent with males from 1988. These trends could be related to the greatly increased number of female athletes and opportunities for more diversified athletes, as per the histogram.

3.2.4 Medals

Another aspect of an athletes participation behaviour at the Olympics, and potentially the most important to them, is the number of medals awarded to each athlete. By exploring this data it may provide insight into whether the medals are being awarded to a variety of athletes, or if the games rely on 'super' athletes winning all of the medals. This data is displayed using a point plot. Similar to the bar plot it computes the mean and a confidence interval with error bars, additionally it joins the points together to help visualise a pattern over time (seaborn, 2020; Nieminen, 2020; Chakure, 2019)

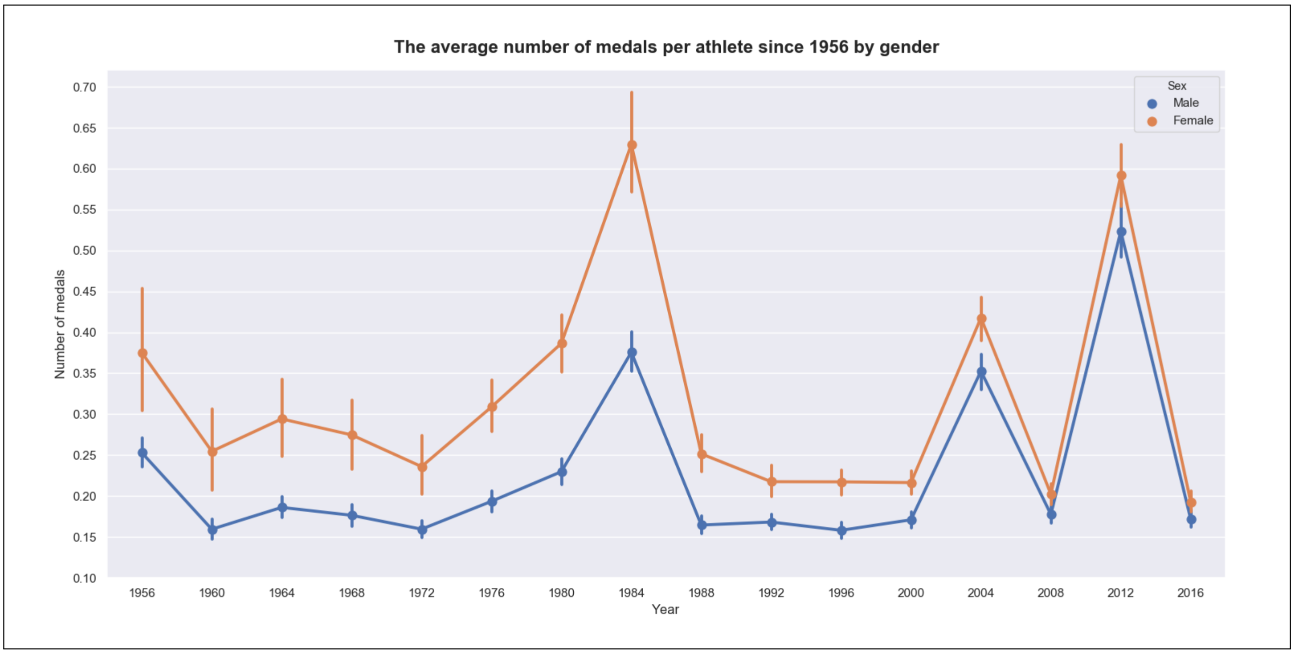


Figure 15: Point Plot - The average number of medals per Summer athlete since 1956 by gender

Again this plot shows a spike for athletes in 1984, 2004 and 2008, which is explained previously. Additionally, the higher values in 1956, 1976 and 1980 maybe be explained by boycotts by various nations during these year, decreasing the competition in events and allowing an individual athlete more opportunity to win medals. On average an event will have 8 entries, 3 of which will receive a medal. This means if each entry at the Olympics was filled by a different athlete the average number of medals should be 0.375 for completely equal distribution.

Males: Not accounting for the above outlying years, the average number of medals has remained around 0.15 per athlete. Also, the error bars are greater than those seen in the average number of events plot, suggesting more variation from the average. Since the average number of events per male athlete is 1.25 and as mentioned the average number of medals for equal distribution is 0.375, this suggests that more athletes are not winning a medal, whilst a smaller group are dominating the medal count.

Females: On the graph it is apparent that female athletes on average win more medals, though the gap between male and female as been closing since 1988. The error bars are also much greater than that of male athletes. The average, ignoring outlying years, for number of medals awarded to each athletes has decreased from 0.25 to 0.18. The difference between males and females follows the same pattern as the average number of events, suggesting that the same females are winning the medals across a number of events.

3.2.5 Medallists

Naturally the next topic to discuss is whether the physical characteristics of athletes who win a medal and athletes who don't differ. Since the average number of medals per athlete is only approximately 0.15 there is a big difference in the amount of data points for medallists and non medal athletes. Therefore, to be able to view the relationship between the two the below are two QQ (Quantile-Quantile) plots for age and BMI respectively. The percentiles of each dataset are calculated and then the corresponding percentiles are plotted against each other. This creates an even number of points for each dataset to be able to compare and view the general trend of their relationship. The standard 45 degree angle is included on the plots to show the linear line if the two datasets had equal percentiles. Additionally the median (50th percentile) and interquartile range (between 25th and 75th percentile) have been included for reference (Nieminen, 2020; Dash, 2020).

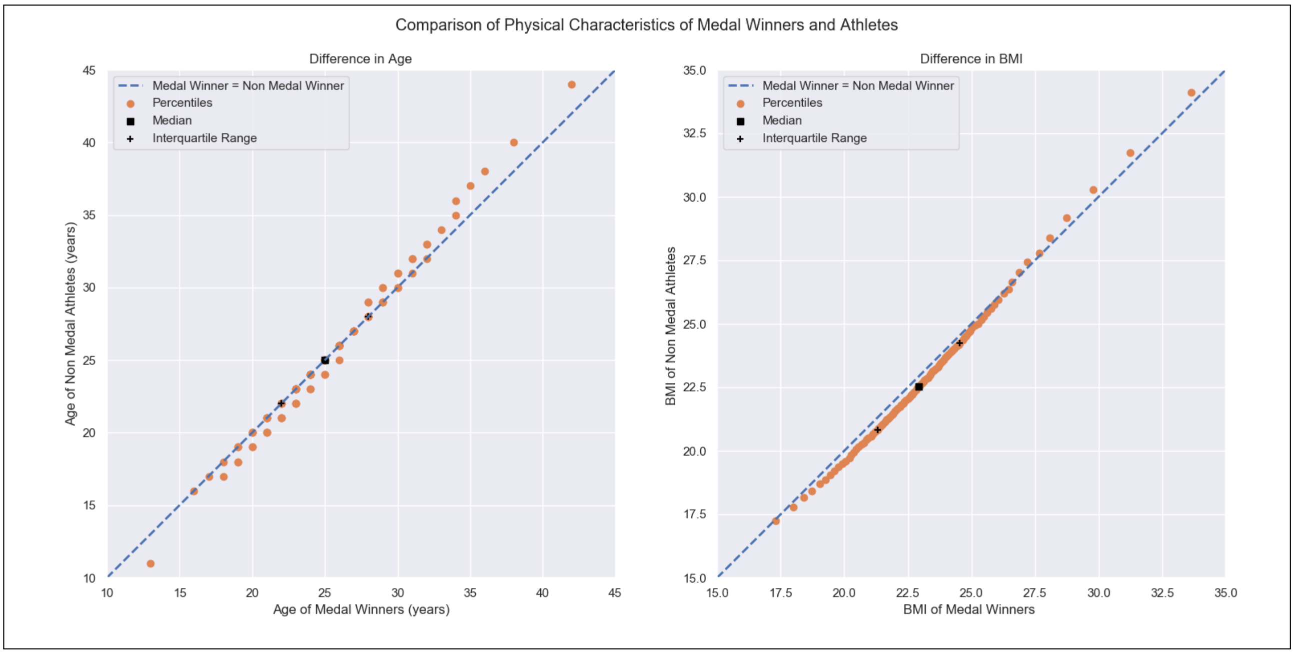


Figure 16: QQ Plot - Comparison of Age and BMI for Medallists and Non-Medal Athletes

First, looking at the comparison of age on the left. From the age of 18 to 26 years old the percentiles are either equal or slightly toward the medal winners, meaning the corresponding percentile of the medal winners is slightly higher. This reverses from this point until around age 35, meaning the medal winners are slightly younger during this range. Then from the age of 35 the data points of the non-medal percentiles start to pull away more to a difference of about 2 years. Therefore, medal winners have more athletes older than 18 and younger than 35 years old.

Now looking at the BMI on the right, the percentile points are obviously more compact with less variation. It appears that almost 90% of the data points sit between a BMI of 18.5 and 26.5. There is little deviation from the 45 degree angle, though medal winners appear to have a slightly higher BMI than other athletes. Together these graphs show that the percentiles of each dataset are almost identical. This indicates that the age and BMI of an athlete does not determine how competitive they are going to be at their chosen event.

It is also important to note that since there is much less data on medal winners than non-medal athletes the percentiles may have some larger gaps, especially for the medallists.

3.3 The Countries

The number of countries participating at the Olympics has now risen to 207 nations. The Olympics are a time to show national pride and represent your nation proudly. There are many different countries with different cultures, physical strengths, resources and opportunities to athletes. This section explores how the average country participates in the Summer Olympics. The data is sources from the aggregations found in noc_totals.csv and host_difference.csv. The code for this section is in Appendix B.3.

3.3.1 Hosting

Over the last 60 years, there have been 16 Summer Olympics hosted by 14 nations, with both United States and Australia hosting twice. The Olympics have been historically very popular with fans around the world travelling or tuning in to watch the action. It is a well known trope that sporting teams do better when they have the home advantage. The below scatterplot explores whether the same can be said about the Summer Olympics. One of the advantages of a scatterplot is being able to compare two variables and view the relationship between them. The average percentage of medals won by each host country during games when they didn't host, are compared against the percentage (average for AUS and USA) won when hosting. Each point represents a year at the Olympic Games. Included on the plot is a 45 degree angle line to show the points at which the two percentages are equal i.e. hosting has no effect on the percentage of medals won. Additionally on the graph is a robust linear regression model to de-weight outliers with a 95% confidence interval (seaborn, 2020; Nieminen, 2020; Dash, 2020).

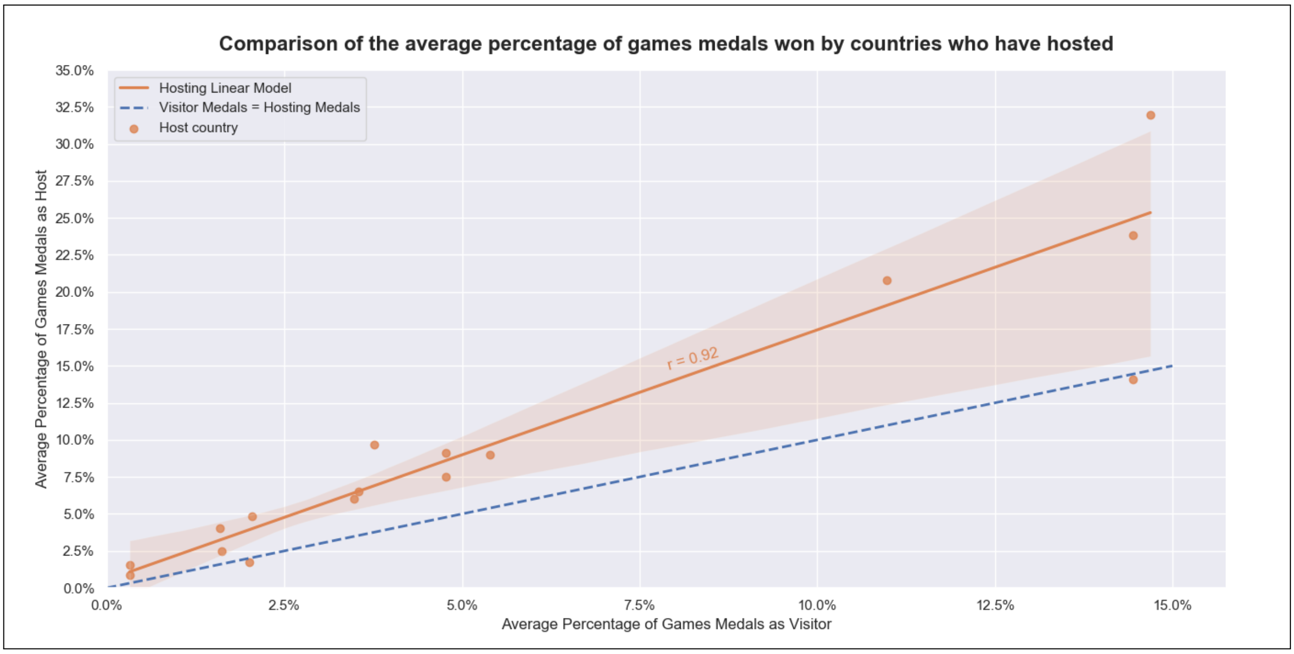


Figure 17: Scatter Plot - Comparison of the average percentage of games medals won by host countries

From the graph it is very clear that when a country hosts they win a higher percentage of medals compared to when they are a visitor. There are two instances where the percentage is only slightly less, once from a country whom only generally wins around 1.9% of medals and another country around 14.5%. Besides these two cases the performance of countries when hosting is far greater than when visiting. The linear regression model seems to be a good fit up until when the confidence interval becomes greater for countries who win more than 5.5% of medals normally. The outlying values may be explained by boycotts. However, according to the correlation coefficient of 0.92, calculated using Pearson's, the percentage of medals when hosting and visiting are positively correlated and 92% of the variation in the percentage of hosting can be explained by the linear model fitted.

3.3.2 Participation Behaviour

Besides hosting, there are a number of other attributes that effect how a country performs at the Olympics. Due to the high number of characteristics and the interest in how each relate to the number of medals awarded to a country, a heatmap is shown. The heatmap shows the Pearson's correlation coefficient for each pair of variables and colour codes it to match. This creates a simple grid to be able to gain quick insights into what variables may have a correlation (seaborn, 2020; Lin, 2018; EliteDataScience, 2019).

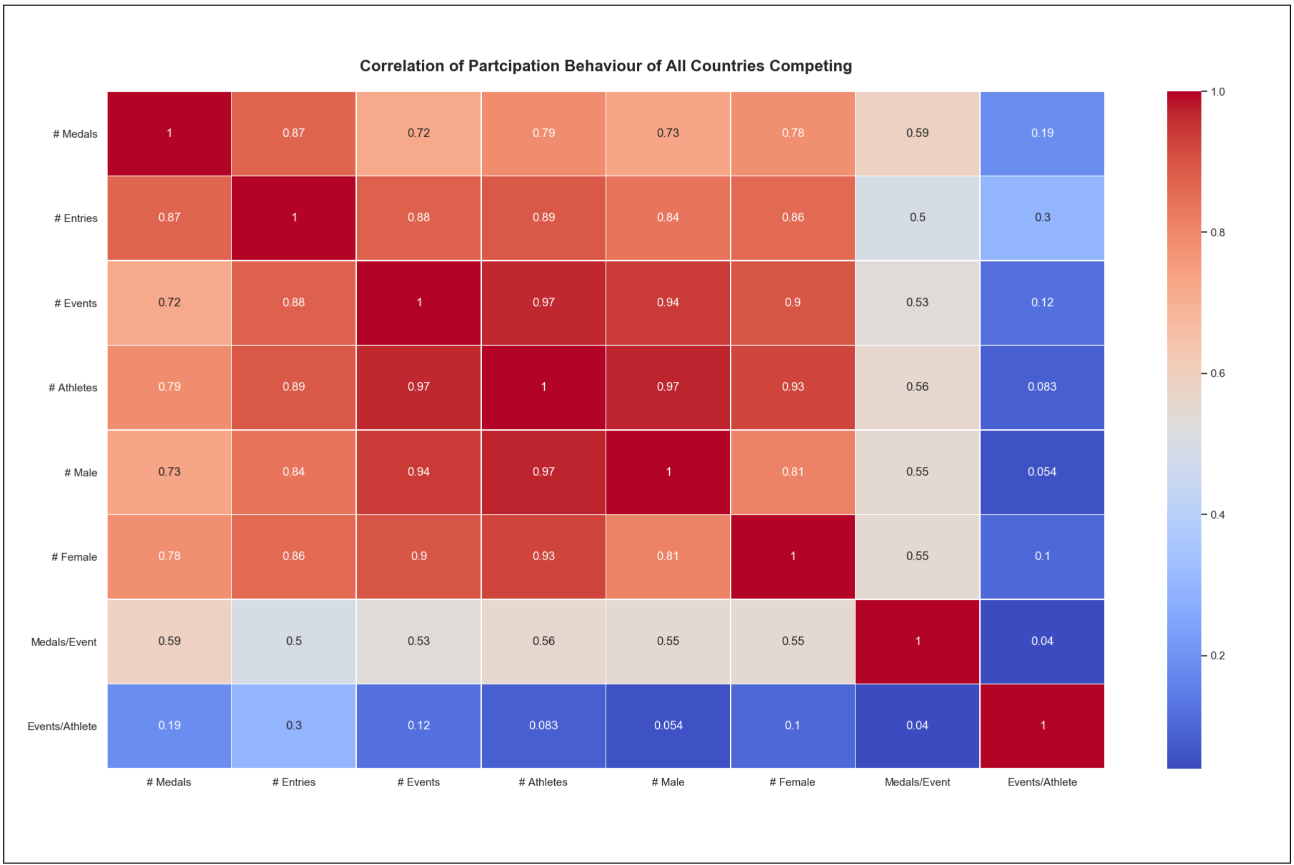


Figure 18: Heatmap - Correlation of participation behaviour attributes of all competing countries

The strongest correlation to number of medals for a country is the the number of entries (0.87). This intuitively makes sense as the number of entries you have is the same as the opportunities you have to win medals. If there is no difference in general physical characteristics for athletes than simply filling a position increases chances, and the more athletes from the same country in the same event the higher the chance for that country to win at least one of the three medals. The next highest correlation is the number of athletes, with male and female closely following. These correlations suggests that the representation of both male and females impact the number of medals that will be awarded to a country.

The most surprising observation, in the writers opinion, is that the number of events per athlete seems to have little to no linear correlation to the number of medals (this doesn't change using polynomial fits). Originally the hypothesis was that a country's ability to enter 'super athletes' would correlate well with how many medals would be achieved. But as also supported by the bar graph on the average number of events per athlete which only averaged at 1.25 events, countries too seem to have not much difference in diversification of their athletes.

3.3.3 Indicators

The next topic to explore is whether barriers exists for countries to fill more entries and hence win more medals. Two of the indicators that may have an impact are the population and GDP of the countries. To demonstrate this multivariate data the below 3D plot shows the relationship of the number of medals, the GDP and population of all countries (with all the available data) for each year from 1960. A 3D graphs allows multiple variables to be displayed to show not only the relationship of variables as pairs but also all together. Shading of points is used to show depth, and colouring based on the number of medals for assisted readability (matplotlib, 2020; Nieminen, 2020; Zhong, 2019; Lin, 2019).

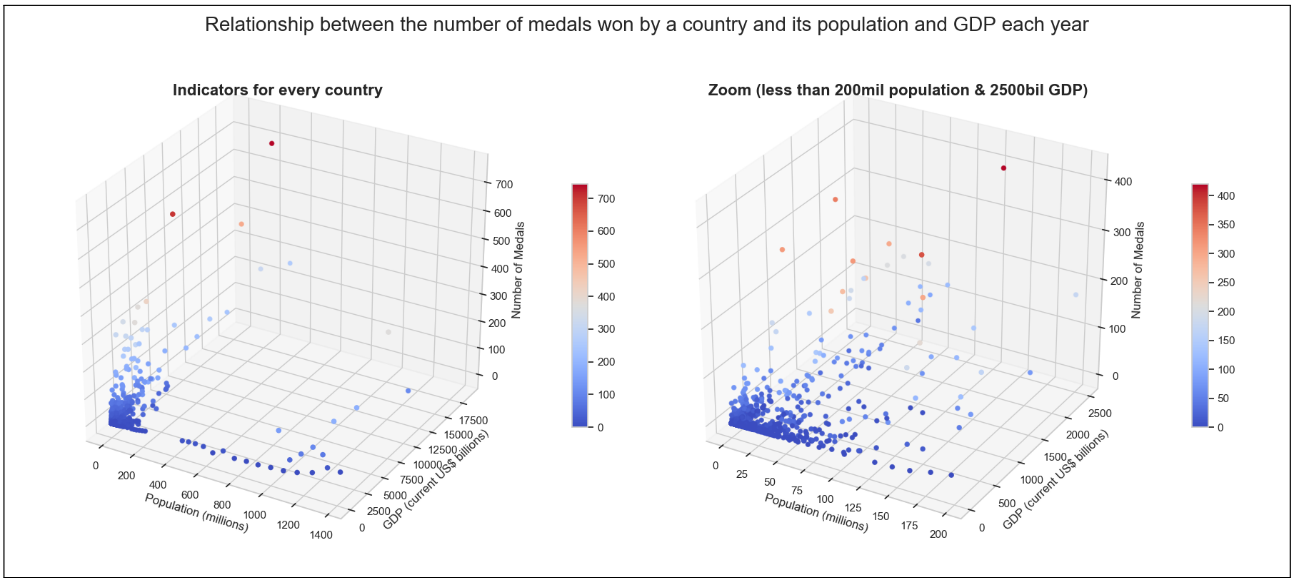


Figure 19: 3D Scatter Plot -Relationship between the number of medals won by a country and its population and GDP each year

Most of the data points lie in a cluster with a population less than 200 million and GDP less than 2500 US$ billion. Outside of this cluster it appears that the population and GDP have little to do with winning more medals that year. Zooming in on this cluster however shows a general trend of increased number of medals associated with higher GDP and population. Many of the countries are still very clustered and there is not much variation.

3.4 Representation

In this section, the aim is to identify whether there is an equal representation of athletes and equal distribution of medals awarded to the countries that participate at the Olympics. The top 10 countries in terms of total number of medals won since 1956 are the USA, Russia, Germany, Australia, China, Great Britain, Italy, Japan, France and Hungary in order. These graphs use the aggregations found in noc_totals.csv and games_total.csv The code for this section is in Appendix B.4.

3.4.1 Entries

Due to the strong correlation between the number of entries and the number of medals, the distribution of the number of entries from the top 20 countries with the most number of medals since 1956 will be explored. To account for outliers, the percentage of total entries per game by each of the top 10 countries and the average for each year of the top 11 to 20 countries will be plotted in a swarm plot. A swarm plot groups a variable into categories and then plots the spread of the data with each dot representing a data point (in this case each year). Rather than overlapping points that are the same, the points fan out next to each other (seaborn, 2020; Sharma, 2018)

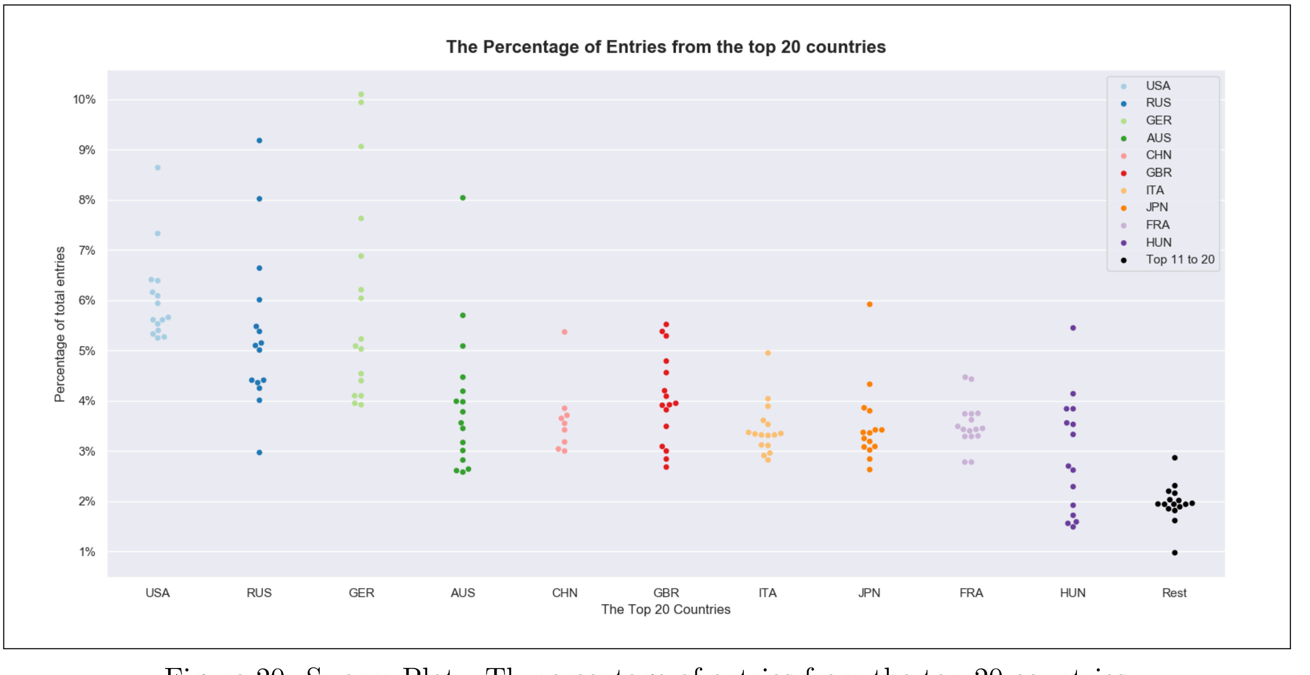


Figure 20: Swarm Plot - Th percentage of entries from the top 20 countries

This graph shows that except for Hungary the percentage of entries from each of the top 10 countries each year is always higher than the average of the top 11 to 20 nations. Besides boycott years, the USA always accounts for at least 5% of entries, Germany at least 4% and the remaining except for Hungary account for at least 3% each. This means that just looking at their minimum contributions, the top 10 nations always account for at least 35% of the entries at the Olympics. On average the next 10 nations range from 1 to 3% of the entries. Adding and average of 2% from these nations, results in the top 20 nations filling at least 55% of the positions at the Olympics. The 2016 Summer Olympics saw 207 nations competing and 13,688 entries. Therefore, whilst 20 nations enter approximately 7,500 of the available positions, the remaining 187 nations enter only 6,100 positions.

3.4.2 Medal Distribution

It is clear that the top 10 countries have much higher percentage of entries than other countries. The below stacked bar chart shows how much of the total medals are awarded to the top 10 countries. As mentioned previously a bar chart is in a easy way to represent the count of a variable. In this case, to effectively represent the distribution of 11 groups, a stacked chart clearly shows the distribution and changes of each country (seaborn, 2020; Zwitch, 2014)

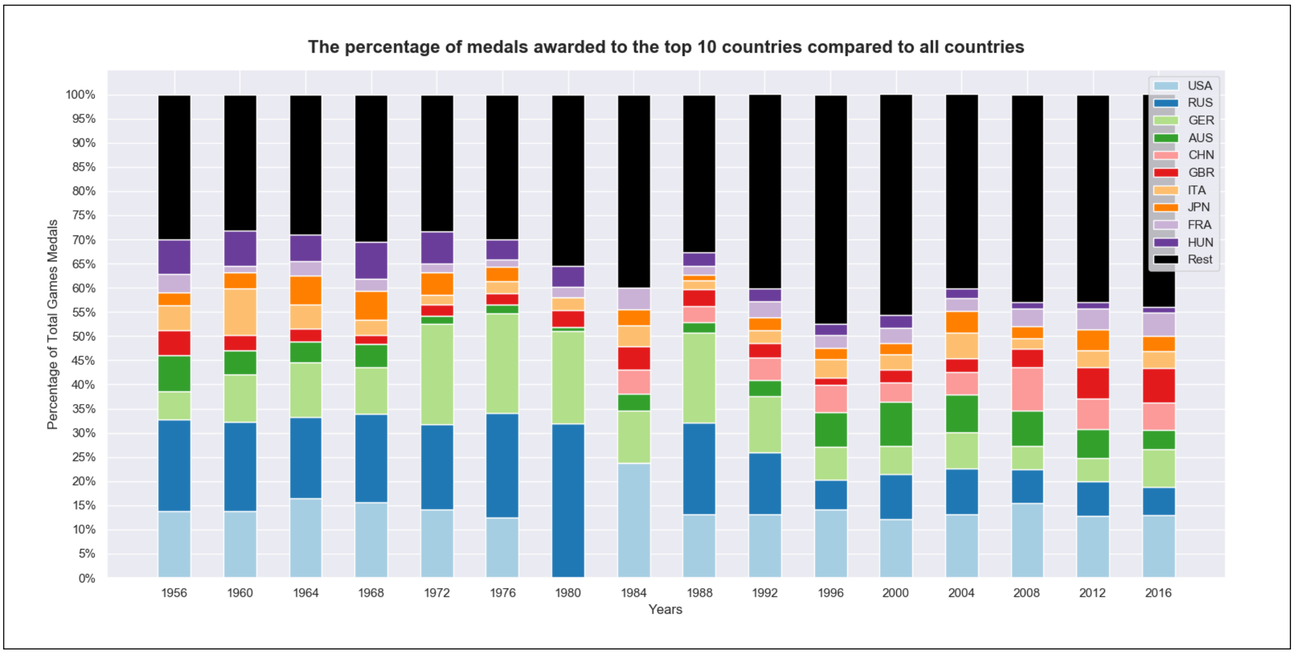


Figure 21: Stacked Bar Chart - The percentage of medals awarded to the top 10 countries compared to all countries

Rather than representing this chart with total number of medals, a more interesting comparison is to show the percentage of total medals awarded to each country over the years. Using the percentage eliminates outlying years and provides a more consistent representation of the performance of countries. From the chart it is evident that the top 10 countries have consistently dominated the Olympic games, though the domination is easing. The top 10 countries account for at most 70% in 1956 and at least 52.5% in 1996 of the total medals awarded. All other nations only account for the remaining percentage of medals.

Individually, there are some interesting observations in the top 10 countries. Reminder that during 1980 Games, Russia hosted and there were boycotts from various nations including USA, China and Japan (some athletes from Australia, Great Britain and France), as well as a returned boycott in 1984 where USA hosted and Russia boycotted. China also boycotted the 1956 and 1964 games. The absence on the graph from these countries during these year, prompted the further inspection into these years where this knowledge of boycotts was gained.

- The percentage of medals awarded to Russia has dropped from over 15% to just around 5% (ignoring 1980). Interestingly, during 1980 Russia simply accounted for the USA percentage, with little difference to the other top 10 values.
- The USA have consistently received at least 12.5% of the Olympics medals each game.
- China is seen entering with more considerable dominance from 1980 increasing to receive 7.5% of medals (during their year of hosting).
- Great Britain, Japan and Italy have fluctuated without much pattern.
- Hungary has continued to decrease from around 6% to less than 1%.

- Germany has seen some drastic changes in their percentage of medals, ranging from 5% to 20% and back again.

3.4.3 Events

The previous heatmap used Pearson's correlation process to determine whether a linear relationship existed amongst the variables. Therefore the possibility of non-linear relationships is yet to be explored. The most interesting relationship using non-linear fits was the number of medals and the number of events. The below graph shows a scatter plot of the top 20 countries with the most medals (the top 10 split individually) and their relationship between the number of medals won each year and the number of different events their athletes competed in. A polynomial fit for all countries can be seen on the graph with a confidence interval of 95% and to the right are the residuals of this fit (Nieminen, 2020; Doshi, 2019; numpy, 2019; SciPy, 2020)

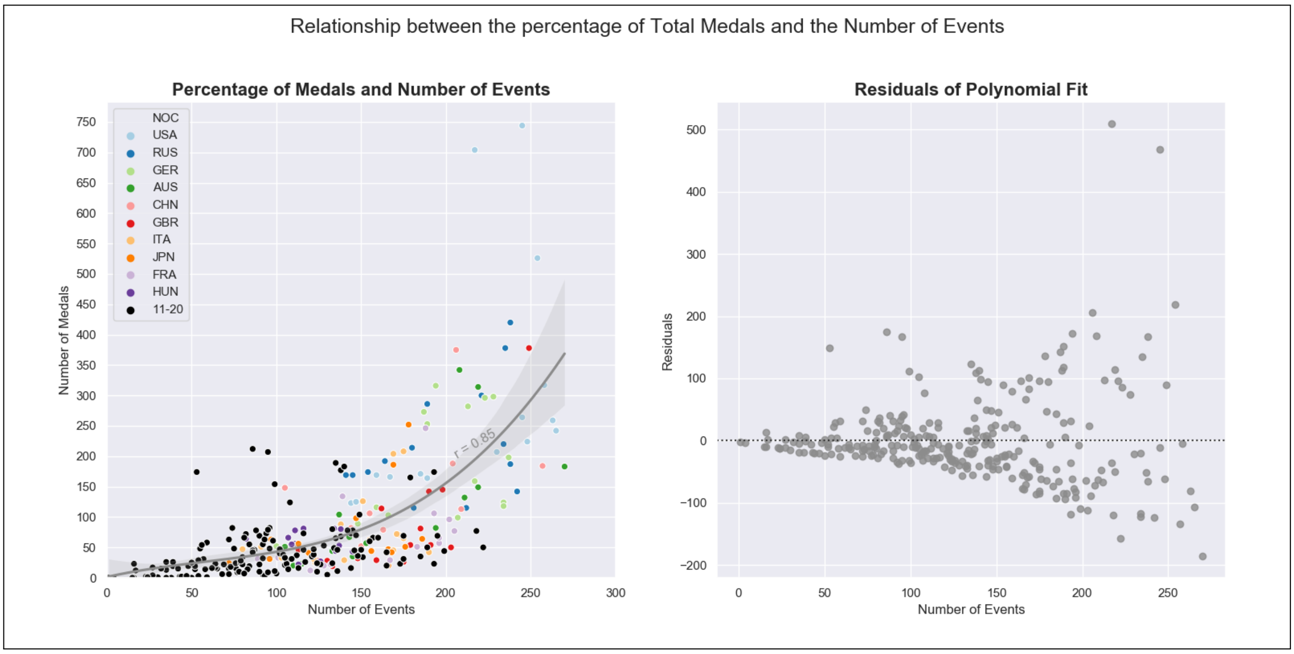


Figure 22: Polynomial Fit & Residuals - The relationship of the percentage of total medals and number of events of the top 20 countries

For this relationship on the heatmap, the Pearson's correlation coefficient was 0.72, whilst using Spearman's the calculation was 0.79, suggesting a non-linear relationship may be stronger (). Using polynomial fits, and least squares calculations, when the order is increased to 2 (quadratic) the coefficient increases to 0.8. So now instead of 72% of the data explained, 80% of the data is explained. However, a quadratic fit for this data causes issues as the fit on the graph showed that if you enter 0 events you can have almost 50 medals which is impossible. Moving up one more order to 3, the least squares coefficient is now 0.85 which not only means these variables are more positively correlated with this fit but also provides a much better and more logical visual fit. Going up another order makes no difference again and so the best fit, without getting any further into complicated algorithms, is a cubic fit.

Looking at the residual on the right this pattern is much more scattered that that of the linear plot. However, besides clusters of uniformed scatter, there appears to be unequal scatter overall or heteroscedasticity which may be due to the extremes between the top 10 countries and the rest of the nations (Frost, 2019).

4 Conclusion

In conclusion, it is not clear what makes an athlete great or a country 'the best'. Though what is clear from this report is that the opportunity to win a medal at the Olympics has never been better. The games have more sports, events, athletes and countries than ever. There appears to be no difference in the physical characteristics of winners and non-medal athletes. The range for females has increased immensely to be almost on par with males all round. The number of medals a country wins is positively correlated with the number of entries, and diversification of athletes seems to be key. GDP and population of home countries do not appear to be a barrier to most nations winning medals, and whilst there are 10 countries who have won up to 70% of the medals historically this dominance is decreasing. While no strict conclusions can be made about the equal distribution of the Olympics or whether it is 'fair' the visualisations provide a unique and powerful insight into the last 60 years of the Summer Olympics.

5 Reflection

Overall, I feel I learnt and implemented a wide range of visualisations that were each fitted for their intended purpose. I analysed univariate, bivariate and multivariate data in different ways to demonstrate my understanding of the multiple ways in which data can be displayed. I learnt and displayed the importance of design choices (such as colour) and accurate labelling. With some basic knowledge of python and little knowledge about the pandas, matplotlib and seaborn libraries I gained some powerful knowledge on these libraries in terms of manipulating data and graphing it appropriately. I believe my report and presentation deserve a high distinction as I demonstrated evidence of exceptional achievement of course learning outcomes. I have described core principles and conceptual frameworks that are used for scientific visualisation, used different data visualisation techniques to meaningfully represent data, showed understanding in the application of different techniques appropriately to the type of data and performed basic statistical analysis to identify trends. I showed in-depth knowledge with effective communication, creativity and critical judgment.

5.1 Design Choices

For all of the graphs there was considerable time spent on the design to ensure easy viewing as well as being appealing to the eye. I feel I demonstrated the knowledge I have gained from lectures and research about the importance that style has on a user's ability to interpret a graph accurately.

- The same colours used throughout for consistency and flow.
- A grid is displayed on all graphs to assist with reading the values on the graphs as the data is numeric and the specific values are important for analysis.
- A grey background is used so that the grid does not take away from the actual points on the graph. Often there are many data points and having a white background with the grid made the grid take away the attention of the points.
- A qualitative palette is used for pair comparisons (i.e. male vs. female). The muted version of the default color palette was chosen as it has higher luminance and middle saturation to contrast well with the grey background (seaborn, 2020). This keeps the

colours bright but not overpowering. The colours chosen were orange and blue as these contrast well with each other as well as being colour-blind friendly (tableau, 2020)

- Categorical colour scheme used for the top 10 countries to assist with representing their order and pairing position 1 with 2, and 3 with 4, etc.
- For the heatmap and 3D plot where the difference in extremes of values are displayed a diverging palette is used. I chose 'coolwarm' palette ranging between blue and red to show this range, as well as not stray from the orange and blue used throughout the rest of the report and compliment the background.
- Titles on all graphs as well as an additional title for the whole figure if there is more than one plot in the figure.
- Legends and axis labels on every graph.
- Add units on all axis where needed, including percentage symbols and years in brackets.
- Add additional ticks to axis for more accurate reading.
- Add frame around each graph to contain and help it stand out amongst the report text.

5.2 Limitations

My number one limitation for this project was time. Whilst I spent a lot of time tweaking and rethinking the graphs there is always so much more to do. Some other graphs I would have liked to look into were:

- All of the same graphs for Winter to see the distribution.
- Whether an athlete winning one medal increases their likelihood to win more.
- Other factors that assist with determining what makes the top 10 countries stand out i.e. spending on sports, average income, contributions by government etc.
- Geographical map with animation showing countries being added each year.
- A contour and/or surface plot.
- Multiple linear regression models.

Another limitation I ran into with python plotting was the lack of functionality for logarithmic and exponential graphs. There were some workarounds but they were cumbersome and didn't really produce the results I was looking for and so weren't included.

5.3 Key Takeaways

1. Don't always trust the data - When looking at the relationship between the number of medals per athlete for a country, and the number of medals they win, I noticed three very distinct groupings. I checked multiple times I was selecting the correct data and hadn't made any coding errors but there they still were. I wondered if maybe there were just three distinct groups of behaviour from countries on how good they were at winning medals. However, on closer inspection I came to notice that in fact there were three years of Olympics where there were a lot more events and medals being offered (partially due to new team sports being introduced like golf) which was skewing the data incorrectly.

2. Reminding myself that correlation does NOT imply causation - Often I found myself trapped in the thinking, 'Well having more athletes means more medals right?' When actually all I was looking at was that the number of athletes positively correlated with the number medals.

3. Hypotheses don't always come true - There were many times when I thought for sure I knew what the answer would be, 'There must be a difference between medal winners and other athletes', 'A graph with GDP and Population will show that only the rich countries get to compete!' But, data analysis doesn't always work out that way and I learnt the hard way. Graphs that I thought be easy and super clear ended up being confusing with little insight.

5.4 Decision Process

Originally when proposing this project, I wanted to look at both the Summer and Winter Olympics for all modern Olympic years to compare if there was much difference between the two games. The first thing I did was make some preliminary graphs of the univariate data of the athletes; their age, BMI, number of events and number of medals. I was overwhelmed with the number of graphs that it seemed I needed to present the information. Graph with hues to show the difference of a variable depending on season, sex and whether they were a medal winner. But then what about the difference between gender for the season separately, that's another graph each again.

Also the difference in the amount of data between the seasons was found to be very large, so was comparing the two actually fair? The Summer Olympics saw at least 4 times more athletes each year and this means thousands of more data points used to calculate medians, means and percentiles. It became obvious that the data was skewed. Additionally, when it came to plotting over time, since the Winter Olympics used to be the same year and then switched to alternate years, the graph was squished with an overload of data on the left to then stretching out to the right. This overcrowded the graph and left no room to be able to compare any additional variables.

Secondly, when it came to plotting over time, it became apparent that over 100 years of Games was too much to plot at one time. When deciding what year to instead condense the data to I looked at a few factors and decided to take the dataset from 1956 to show the games over a period of 60 years (16 Olympiads).

- 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
- 1952: USSR/Russia starts competing
- 1956: Weight and height is recorded for more than 40% (over 90% the next year)
- 1960: GDP and Population data is available from worldbank
- 1984: Mainland China joins the Olympics

A comparison, despite style, of the readability and knowledge gained from the condensed graphs to what I started with, I think, shows my development in the understanding of sometimes less is more. It is better for users to be able to understand the graph then overload it with all of the information.

5.5 Proposal Updates

Originally my proposal suggested 7 plots for these project, and in addition to the smaller selection of data mentioned earlier, there were a number of refinements made to the suggested graphs as part of the learning process.

These graphs were implemented:

- The distribution of the number of events and medals [Histogram (#events, #medal] - Included in addition to other attributes.
- The BMI of medal and non-medal winners [q-q plot (medal v. non-medal BMI)] - This was implemented.
- The difference in % of medals when competing at host [scatter (% hosting v. visiting)] - This was implemented with interesting linear regression model added.
- The effect of population and GDP on number of medals [multi - pop, GDP, #medals] - Implemented with less than desired results, but still demonstrates some key insight.

These are the changes that were made:

- The change in age for medal and non-medal winners [Turkey box (year v. age)] - There was no difference in the age so the boxplot was just the same data presented twice next to each other. A more effective graph instead was a QQ plot used in this report.
- The proportion of medal winners to number athletes [scatter (#athletes v. #medals)] - This was investigated but rather than the number of athletes, the percentage of entries was interesting and more effective representation of the impact of countries at the Olympics.
- The proportion of men and women competing [scatter (#females v. #males)] - From the histogram, it was clear that men were going to clearly out way woman, and rather the increase in the number of females competing is a more interesting metric as show on the histogram.

These graphs were added:

- Number of events on average per athlete (Bar Graph)
- Number of medals on average per athlete (Point Plot)
- Correlation between participation behaviour of countries (Heatmap)
- Comparison of percentage of medals won by the top 10 countries (Stacked Bar Chart)
- Closer look at affect of number of events to number of medals by countries (Polynomial)

References

5 tips on designing colorblind-friendly visualizations. (2020, May 5). Tableau Software. https://www.tableau.com/about/blog/2016/4/examining-data-viz-rules-dont-use-red-green-together-53463

Building structured multi-plot grids—Seaborn 0.10.1 documentation. (2020, May 6). https://seaborn.pydata.org/tutorial/axis_grids.html

Chakure, A. (2019, November 1). *Data Visualization*. Medium. https://towardsdatascience.com/data-visualization-a6dccf643fbb

Choosing color palettes—Seaborn 0.10.1 documentation. (2020, May 5). https://seaborn.pydata.org/tutorial/color_palettes.html

Data Visualization with Seaborn (Part #2). (2018, April 8). Jovian's Notes. https://jovianlin.io/data-visualization-seaborn-part-2/

Heteroscedasticity in Regression Analysis. (2017, August 13). *Statistics By Jim*. http://statisticsbyjim.com/regression/heteroscedasticity-regression/

Koehrsen, W. (2018, April 6). *Visualizing Data with Pairs Plots in Python*. Medium. https://towardsdatascience.com/visualizing-data-with-pair-plots-in-python-f228cf529166

Learn Python, Data Viz, Pandas & More | Tutorials | Kaggle. (2020, May 6). https://www.kaggle.com/learn/overview

Nieminen, T. (2020). *Visulisation—Lecture Notes*.

numpy.polyfit—NumPy v1.17 Manual. (2020, May 5). https://docs.scipy.org/doc/numpy/reference/generated/numpy.polyfit.html

Randyzwitch.com | Creating A Stacked Bar Chart in Seaborn. (2020, May 6). https://randyzwitch.com/creating-stacked-bar-chart-seaborn/

seaborn.lmplot—Seaborn 0.10.1 documentation. (2020, May 6).

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(
    './data/host_summer.txt', 'Summer')
winter_df = get_season_df(
    './data/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(
            './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('./data/host_countries.csv')
```

A.2 Combined Data

```
import pandas as pd, 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(
    './data/noc_regions.csv')
host_df = pd.read_csv(
    './data/host_countries.csv')
athlete_df = pd.read_csv(
    './data/athlete_events.csv')
worldbank_gdp = pd.read_csv(
    './data/worldbank_gdp.csv',
    index_col=0).reset_index(drop = True)
worldbank_pop = pd.read_csv(
    './data/worldbank_pop.csv',
    index_col=0).reset_index(drop = True)

############### 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), axis=1)

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

# 9. Add GDP - merge with noc_total, divide by 1billion
worldbank_gdp = worldbank_gdp\
                .drop(['Indicator␣Name', 'Indicator␣Code',
                      'Unnamed:␣64'], axis=1)
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',
                                     'Unnamed:␣64'], axis=1)
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(1000000), 2)
all_df = all_df.merge(worldbank_pop, how='left')

# Summer 1956 Olympics Equistrian events in Sweden
# Update to reflect actual host Australia
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_index()
games_male.columns = ['Games', 'Male']
games_female = all_df[all_df['Sex'] == 'F']\
                    .groupby('Games').ID.nunique().reset_index()
games_female.columns = ['Games', 'Female']
games_BMI = all_df[~all_df['BMI'].isna()]\
                    .groupby('Games', as_index=False).ID.count()
games_BMI.columns = ['Games', 'Num_BMI']
games_host = all_df[all_df['NOC'] == all_df['Host_NOC']]\
                    .groupby('Games').Medal.count().reset_index()
```

```
games_host.columns = ['Games', 'Host_Medal']
games_visitor = all_df[all_df['NOC'] != all_df['Host_NOC']]\
                    .groupby('Games').Medal.count().reset_index()
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
games_total_df['Perc_BMI'] = round(games_total_df.Num_BMI
                                / games_total_df.Entries, 2)
# Write to file before further changes for pre 1956 data
games_total_df.to_csv('./data/games_total_before.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)
#checkpoint('REMOVE EXTRA', all_df, True, extra_df)

######## WRITE TO FILE ##########
all_df.to_csv('./data/all_data.csv')
```

A.3 Totals

```
import pandas as pd

# Read in summer data only
all_df = pd.read_csv('./data/summer_data.csv', index_col=0)\
            .reset_index(drop = True)

######## THE ATHLETE #######
# Athlete Totals (Year, ID, Sex, Age, BMI, NOC, #Events, #Medals)
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']]
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_total_df.Event), 2)
athlete_total_df['Sex'] = athlete_total_df.Sex.replace('M', 'Male')
athlete_total_df['Sex'] = athlete_total_df.Sex.replace('F', 'Female')
athlete_total_df.to_csv('./data/athlete_total.csv')

######## THE GAMES #######
# Games totals (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_index()
games_male.columns = ['Year', 'Male']
games_female = all_df[all_df['Sex'] == 'F']\
                    .groupby('Year').ID.nunique().reset_index()
games_female.columns = ['Year', 'Female']
# Add column for number of medals awarded to host country
games_host = all_df[all_df['NOC'] == all_df['Host_NOC']]\
                .groupby('Year').Medal.count().reset_index()
games_host.columns = ['Year', 'Host_Medal']
games_visitor = all_df[all_df['NOC'] != all_df['Host_NOC']]\
                .groupby('Year').Medal.count().reset_index()
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_total_df.Medal)*100, 2)
games_total_df['Visitor_Medal_Perc'] = round(
                                    (games_total_df.Visitor_Medal
                                    / games_total_df.Medal)*100, 2)
games_total_df.to_csv('./data/games_total.csv')

######## THE COUNTRIES #######
# Group all by Year and NOC to create seperate tallies
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().reset_index()
noc_male.columns = ['Year', 'NOC', 'Male']
noc_female = all_df[all_df['Sex'] == 'F']\
                .groupby(['Year', 'NOC']).ID.nunique().reset_index()
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', 'Population']]
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.Athletes), 2)
noc_total_df['Medal_Perc'] = round((noc_total_df.Medal
                              / noc_total_df.Entries), 2)
noc_total_df['Games_Medal_Perc'] = round((noc_total_df.Medal
                          / noc_total_df.Games_Medals)*100, 2)
noc_total_df['Games_Entries_Perc'] = round((noc_total_df.Entries
                          / noc_total_df.Games_Entries)*100, 2)
# 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_values(by='Medal', ascending=False)\
                      .head(20).NOC.tolist()
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])
noc_total_df.to_csv('F:/TEAN/Portfolio/olympics/data/noc_total.csv')


#### Host Medal Percentage vs. Average Percentage ####
# Create host percentage and regular percentage df
medals_all = all_df.groupby(['Year', 'NOC', 'Country'])\
                  .Medal.count().reset_index()
medals_all.columns=['Year','NOC', 'Country', 'Total_Medal']
medals_all = medals_all.merge(games_total_df[['Year', 'Medal']],
                              how='outer')
medals_all['Medal_Perc'] = round((medals_all.Total_Medal
                          / medals_all.Medal)*100, 2)
medals_all = round(medals_all.groupby(['NOC', 'Country'])
                  .Medal_Perc.mean(),2).reset_index()
medals_all.columns=['NOC', 'Country', 'Medal_Perc']
host_medals = games_total_df[['Year', 'Host_NOC',
                          'Host_Medal_Perc']]
host_medals.columns=['Year', 'NOC', 'Host_Medal_Perc']
host_difference = pd.merge(host_medals, medals_all, how='left')
host_difference.to_csv('F:/TEAN/Portfolio/olympics/data/host_difference.csv')
```

B Discussion

B.1 The Games

```
import pandas as pd, seaborn as sns, numpy as np
from matplotlib import pyplot as plt
# Read data
games_total_df = pd.read_csv(
            './data/games_total.csv', index_col=0)
games_total_before = pd.read_csv(
            './data/games_total_before.csv', index_col=0)\
            .reset_index(drop = True)
# Set style
sns.set()
title_dict = {'fontsize': 16, 'fontweight': 'bold'}

##### GRAPHS######
# Histogram - Distribution of games before and after 1955
games_var_list = ['Entries', 'Athletes',
                    'Male', 'Female',
                    'NOC', 'Event',
                    'Sport', 'Medal']

games_total_before = games_total_before[
                        (games_total_before['Season'] == 'Summer')
                        & (games_total_before['Year']<1955)]
plt.figure(figsize=[12,8])
plt.gcf().suptitle('Distribution of Games Attributes for all Summer Olympics'
            fontdict=title_dict)
plot = [2,4,0]
bins = 10
dfs = [games_total_before, 'Before 1955'], \
      [games_total_df, 'After 1955']
for var in games_var_list:
    plot[2] += 1
    plt.subplot(plot[0], plot[1], plot[2])
    for df in dfs:
        if (plot[2] == 4) and (df[1] =='Before 1955'):
            bins=5
        plt.hist(df[0][var], label=df[1], histtype='bar',
                    alpha=0.7, bins=bins)
        bins=10
        if plot[2] == 4:
            plt.legend()
    plt.xlabel('Number of {var}'.format(var=var))
    plt.ylabel('Number of occurences')
plt.subplots_adjust(top=0.9, left=0.08, right=0.95,wspace=0.4)
plt.savefig('./images/graph/games_histogram.png')
plt.show()
```

B.2 The Athletes

```
import pandas as pd, seaborn as sns, numpy as np
from matplotlib import pyplot as plt
# Read the data
athlete_total_df = pd.read_csv(
            './data/athlete_total.csv', index_col=0)
#Set the style
sns.set()
year_fig = [16,8]
title_dict = {'fontsize': 16, 'fontweight': 'bold'}

### GRAPHS ###
## Boxplot - Age
plt.figure(figsize=year_fig)
ax = plt.subplot()
sns.boxplot(x='Year' , y='Age', data=athlete_total_df, hue='Sex')
plt.title('The␣Age␣of␣Summer␣Athletes␣since␣1956␣by␣Gender',
            fontdict=title_dict, pad=15)
ax.set_yticks(range(10,76,5))
plt.subplots_adjust(top=0.9, left=0.08, right=0.95)
plt.savefig('./images/graph/athlete_age_boxplot.png')
plt.show()

# Violin plot - BMI
plt.figure(figsize=year_fig)
ax = plt.subplot()
sns.violinplot(x='Year', y='BMI', data=athlete_total_df,
                hue='Sex', split=True, inner='quartile')
plt.title('The␣BMI␣of␣Summer␣Athletes␣since␣1956␣by␣Gender',
            fontdict=title_dict, pad=15)
ax.set_yticks(range(5,68, 5))
plt.subplots_adjust(top=0.9, left=0.08, right=0.95)
plt.savefig('./images/graph/athlete_bmi_violinplot.png')
plt.show()

# Barplot - Number of events per athlete
plt.figure(figsize=year_fig)
ax = plt.subplot()
sns.barplot(x='Year', y='Event', data=athlete_total_df, hue='Sex')
ax.set_yticks(np.arange(1,4.1,0.25))
plt.title('The␣average␣number␣of␣events␣per␣athlete␣since␣1956␣by␣gender',
            fontdict=title_dict, pad=15)
plt.ylabel('Number␣of␣events')
plt.subplots_adjust(top=0.9, left=0.08, right=0.95)
plt.savefig('./images/graph/athlete_event_barplot.png',
                fontdict=title_dict, pad=15)
plt.show()

# Pointplot - Number of medals per athlete
```

```
plt.figure(figsize=year_fig)
ax = plt.subplot()
sns.pointplot(x='Year', y='Medal', data=athlete_total_df, hue='Sex')
ax.set_yticks(np.arange(0.1, 0.71, 0.05))
plt.title('The␣average␣number␣of␣medals␣per␣athlete␣since␣1956␣by␣gender',
          fontdict=title_dict, pad=15)
plt.ylabel('Number␣of␣medals')
plt.subplots_adjust(top=0.9, left=0.08, right=0.95)
plt.savefig('./images/graph/athlete_medal_pointplot.png')
plt.show()

# QQplot - Difference of age and BMI for medal winners
athlete_var_list = [['Age', [10, 45], '(years)'], ['BMI', [15,35], '']]
medal_athlete = athlete_total_df\
                [athlete_total_df['Winner']]
non_medal_athlete = athlete_total_df\
                [athlete_total_df['Winner'] == False]
plt.figure(figsize=year_fig)
plot = [1, 2, 0]
for var in athlete_var_list:
    plot[2] += 1
    ax = plt.subplot(plot[0], plot[1], plot[2])
    medal_percentile = medal_athlete[var[0]].quantile(np.arange(0,1,0.01))
    non_medal_percentile = non_medal_athlete[var[0]]\
                              .quantile(np.arange(0,1,0.01))
    plt.scatter(medal_percentile, non_medal_percentile, color='C1')
    plt.scatter(medal_percentile[0.49], non_medal_percentile[0.49],
                color='black', marker='s')
    plt.scatter(medal_percentile[0.24], non_medal_percentile[0.24],
                color='black', marker='+')
    plt.scatter(medal_percentile[0.74],
                non_medal_percentile[0.74],
                color='black', marker='+')
    plt.plot(var[1],var[1], color='C0', linewidth=2, linestyle='dashed')
    plt.title("Difference␣in␣{var}".format(var=var[0]))
    plt.xlabel('{var}␣of␣Medal␣Winners␣{units}'
                .format(var=var[0], units=var[2]))
    plt.ylabel('{var}␣of␣Non␣Medal␣Athletes␣{units}'
                .format(var=var[0], units=var[2]))
    ax.set_xlim(left=var[1][0], right=var[1][1])
    ax.set_ylim(bottom=var[1][0], top=var[1][1])
    plt.legend(['Medal␣Winner␣=␣Non␣Medal␣Winner',
                'Percentiles', 'Median', 'Interquartile␣Range'])
plt.subplots_adjust(top=0.9, left=0.08, right=0.95)
plt.gcf().suptitle('Comparison␣of␣Physical␣Characteristics␣of␣Medal␣Winners␣a
          fontdict=title_dict)
plt.savefig('./images/graph/athlete_difference_qqplot.png')
plt.show()
```

B.3 The Countries

```
import pandas as pd, scipy.stats as stats, seaborn as sns, numpy as np
from matplotlib import pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
# Read in files
noc_total_df = pd.read_csv(
            './data/noc_total.csv', index_col=0)
host_difference = pd.read_csv(
            './data/host_difference.csv', index_col=0)
# Set style
sns.set()
title_dict = {'fontsize': 16, 'fontweight': 'bold'}

#### GRAPHS ####
# Hosting - Scatter with regression
sns.set_palette(['C1', 'C0'])
facet = sns.lmplot(data=host_difference,
            x='Medal_Perc', y='Host_Medal_Perc',
            robust=True, palette=['C0'], height=7, aspect=2)
plt.plot([0,15],[0,15], color='C1', linewidth=2, linestyle='dashed')
facet.ax.set_xticks(np.arange(0,16,2.5))
facet.ax.set_yticks(np.arange(0,36,2.5))
facet.ax.set_xticklabels(['{}%'.format(x)
                        for x in facet.ax.get_xticks()])
facet.ax.set_yticklabels(['{}%'.format(x)
                        for x in facet.ax.get_yticks()])
r, _ = stats.pearsonr(host_difference['Host_Medal_Perc'],
                        host_difference['Medal_Perc'])
facet.ax.annotate("r␣=␣{:.2f}".format(r), xy=(.5, .42),
                    xycoords=facet.ax.transAxes,
                    rotation=15, color='C0')
facet.ax.set_xlim(left=0)
facet.ax.set_ylim(bottom=0)
plt.subplots_adjust(top=0.9, left=0.08, right=0.95)
plt.xlabel('Average␣Percentage␣of␣Games␣Medals␣as␣Visitor')
plt.ylabel('Average␣Percentage␣of␣Games␣Medals␣as␣Host')
plt.title('Comparison␣of␣the␣average␣percentage␣of␣games␣medals␣won␣by␣count
            fontdict=title_dict, pad=15)
plt.legend(['Hosting␣Linear␣Model',
            'Visitor␣Medals␣=␣Hosting␣Medals',
            'Host␣country'])
plt.savefig('./images/graph/countries_host_lmplot.png')
plt.show()

# Heatmap of stats for all countries
plt.figure(figsize=[18,12])
noc_labels = ['#␣Medals', '#␣Entries', '#␣Events',
                '#␣Athletes', '#␣Male', '#␣Female',
                'Medals/Event', 'Events/Athlete']
```

```
corr = noc_total_df[['Medal', 'Entries', 'Event',
                     'Athletes', 'Male', 'Female',
                     'Medal_Perc', 'Unique_Perc']].corr()
sns.heatmap(corr, annot=True, xticklabels=noc_labels,
            yticklabels=noc_labels, linewidths=0.5,
            cmap='coolwarm')
plt.title("Correlation␣of␣Partcipation␣Behaviour␣of␣All␣Countries␣Competing",
            fontdict=title_dict, pad=20)
plt.yticks(rotation = 0)
plt.xticks(rotation = 0)
plt.subplots_adjust(top=0.9, left=0.08, right=1.05)
plt.savefig('./images/graph/countries_stats_heatmap.png')
plt.show()

# 3D plot - Population, GDP and medals
sns.set_style("whitegrid")
fig = plt.figure(figsize=[18,8])
ax = fig.add_subplot(121, projection='3d')
df = noc_total_df
z =df.Medal
x =df.Population
y =df.GDP
surf = ax.scatter(x, y, z, marker='o', c=z, cmap='coolwarm')
ax.set_xlabel('Population␣(millions)')
ax.set_ylabel('GDP␣(current␣US$␣billions)')
ax.set_zlabel('Number␣of␣Medals')
plt.title('Indicators␣for␣every␣country', fontdict=title_dict)
fig.colorbar(surf, shrink=0.5, aspect=15)
ax2 = fig.add_subplot(122, projection='3d')
df = df[(df['GDP'] < 2500) & (df['Population'] < 200)]
z =df.Medal
x =df.Population
y =df.GDP
surf2 = ax2.scatter(x, y, z, marker='o', c=z, cmap='coolwarm')
ax2.set_xlabel('Population␣(millions)')
ax2.set_ylabel('GDP␣(current␣US$␣billions)')
ax2.set_zlabel('Number␣of␣Medals')
plt.title('Zoom␣(less␣than␣200mil␣population␣&␣2500bil␣GDP)',
            fontdict=title_dict)
fig.colorbar(surf2, shrink=0.5, aspect=15)
plt.subplots_adjust(top=0.9, left=0.01,
        right=1, bottom=0.05, wspace = -0.05)
plt.gcf().suptitle('Relationship␣between␣the␣number␣of␣medals␣won␣by␣a␣countr
                     fontsize=20)
plt.savefig('./images/graph/countries_pop_gdp_3d.png')
plt.show()
```

B.4 Representation

```
import pandas as pd, seaborn as sns, scipy.stats as stats, numpy as np
from matplotlib import pyplot as plt
# Read in the data
noc_total_df = pd.read_csv(
            './data/noc_total.csv', index_col=0)
# Set style
sns.set()
title_dict = {'fontsize': 16, 'fontweight': 'bold'}
#Set the colors for top 10 countries and 'OTHER'
noc_colors = sns.color_palette("Paired", n_colors=11)
noc_colors[-1] = (0.0, 0.0, 0.0)
# SET UP LISTS FOR TOP 10 and 20 COUNTRIES
# Get the top 10 and top 20 countries
top_10 = noc_total_df[noc_total_df['Top_10'] == True]
top_20 = noc_total_df[(noc_total_df['Top_20'] == True) & (noc_total_df['Top_1
top_20['NOC'] = '11-20'
# Get the median of 11-20 countries
top_20_med = top_20.groupby('Year').median()
top_20_med['NOC'] = 'Rest'
top_20_med_all = pd.merge(top_10, top_20_med, how='outer')
# Sum the values of all countries not in top 10
not_top_10 = noc_total_df[noc_total_df['Top_10'] == False]
not_top_10['NOC'] = 'Rest'
not_top_10_sum = not_top_10.groupby('Year').sum().reset_index()
not_top_10_sum['NOC'] = 'Rest'
all_count = top_10.merge(not_top_10_sum, how='outer')
# Set the order of the top 10
top_summer_order = top_10.groupby(['NOC'], as_index=False)['Medal']\
        .sum().sort_values(by='Medal', ascending=False).NOC.tolist()
top_summer_order.append('Rest')

#### GRAPHS ####
# Scatterplots - Top 20 medals against athlete and events
hue_order = top_summer_order[:-1]
hue_order.append('11-20')
top_20_all = top_10.merge(top_20, how='outer')
df = top_20_all[top_20_all['Year'] != 1980]
y = 'Medal'
plt.figure(figsize=(16,8))
order = 3
plt.subplot(1,2,1)
ax2 = sns.scatterplot(data=df, y=y, x='Event', hue='NOC',
                    hue_order=hue_order, palette=noc_colors)
ax2 = sns.regplot(data=df, y=y, x='Event', scatter=False, color='C7', order=o
ax2.set_ylabel('Number␣of␣Medals')
ax2.set_yticks(np.arange(0,751,50))
ax2.set_xticks(range(0,301,50))
ax2.set_xticklabels(range(0,301,50))
```

```
ax2.set_xlim(left=0)
ax2.set_ylim(bottom=0)
ax2.set_xlabel('Number of Events')
x2 = noc_total_df['Medal']
y2 = noc_total_df['Event']
quad = np.polyfit(x2,y2,order)
v2 = np.polyval(quad, x2)
_,_,least2,p,_ = stats.linregress(v2,y2)
ax2.annotate("r = {:.2f}".format(least2), xy=(.68, .25),
                xycoords=ax2.transAxes, rotation=30, color='C7')
plt.title('Percentage of Medals and Number of Events', fontdict=title_dict)
ax4 = plt.subplot(1,2,2)
sns.residplot('Event', y, data=df, order=order, color='C7')
ax4.set_ylabel('Residuals')
ax4.set_xlabel('Number of Events')
plt.title('Residuals of Polynomial Fit', fontdict=title_dict)
plt.subplots_adjust(top=0.85, left=0.08, right=0.95)
plt.gcf().suptitle('Relationship between the percentage of Total Medals and t
plt.savefig('./images/graph/countries_medals_resid.png')
plt.show()

# Stacked bar chart - Medal Percentage of NOC
years = noc_total_df.Year.unique().tolist()
plt.figure(figsize=[16,8])
ax = plt.subplot()
bottom = [0]*len(years)
color = 0
for noc in top_summer_order:
    country = all_count[all_count['NOC']==noc]
    noc_perc = country.Games_Medal_Perc.tolist()
    for year in years:
        if year not in country.Year.unique():
            noc_perc.insert(years.index(year),0)
    plt.bar(years, noc_perc, bottom=bottom, color=noc_colors[color],
                label=noc, width=2, align='center')
    bottom = [sum(i) for i in zip(bottom, noc_perc)]
    color += 1
ax.set_xticks(years)
ax.set_yticks(range(0,101,5))
plt.xlabel('Years')
plt.ylabel('Percentage of Total Games Medals')
ax.set_yticklabels(['{}%'.format(x) for x in ax.get_yticks()])
plt.legend()
plt.subplots_adjust(top=0.9, left=0.08, right=0.95)
plt.title('The percentage of medals awarded to the top 10 countries compared 
            fontdict=title_dict, pad=15)
plt.savefig('./images/graph/countries_medals_stacked.png')
plt.show()

#Swarmplots - Percentage of total entries
```

```
plt.figure(figsize=[16,8])
ax = plt.subplot()
sns.swarmplot(data=top_20_med_all, x='NOC', y='Games_Entries_Perc',
              order=top_summer_order, palette=noc_colors)
plt.xlabel('The Top 20 Countries')
plt.ylabel('Percentage of Total Games Entries')
ax.set_yticks(range(1,11))
ax.set_yticklabels(['{}%'.format(x) for x in ax.get_yticks()])
plt.xlabel('The Top 20 Countries')
plt.ylabel('Percentage of total entries')
plt.title('The Percentage of Entries from the top 20 countries',
          fontdict=title_dict, pad=15)
legend = top_summer_order[:-1]
legend.append('Top 11 to 20')
plt.legend(legend)
plt.subplots_adjust(top=0.9, left=0.08, right=0.95)
plt.savefig('./images/graph/countries_entryperc_swarm.png')
plt.show()
```

C 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)