SQL Capstone Project Milestone 1 - Project Proposal and Data Selection, Preparation

October 2, 2022

1 Preparation

1.0.1 1. Client/Dataset

I chose the SportsStats Olympics dataset. I enjoy watching many sporting events, especially the Olympics, and I am interested in investigating what factors might be associated with success in its different sports.

1.0.2 2. Importing/Cleaning

I imported the following modules for reading/exploring the data:

[1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from scipy import stats
from pandasql import sqldf
pysqldf = lambda q: sqldf(q, globals())
```

The data came in two .csv files:

[2]:
```
event_data = pd.read_csv('athlete_events.csv')
event_data.head()
```

[2]:

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

```
                              Event Medal
0       Basketball Men's Basketball   NaN
1      Judo Men's Extra-Lightweight   NaN
2           Football Men's Football   NaN
3       Tug-Of-War Men's Tug-Of-War  Gold
4  Speed Skating Women's 500 metres   NaN
```

[3]:
```
region_data = pd.read_csv('noc_regions.csv')
region_data.head()
```

[3]:
```
   NOC       region                 notes
0  AFG  Afghanistan                   NaN
1  AHO      Curacao  Netherlands Antilles
2  ALB      Albania                   NaN
3  ALG      Algeria                   NaN
4  AND      Andorra                   NaN
```

The 'NOC' and 'notes' columns reflect territories as they were defined at the time an athlete competed at an Olympics. The 'region' column indicates what territories these areas belong to as of 2016.

For example, row 1 of the above output reflects the dissolution of the Netherlands Antilles 2010, as Curacao and Sint Maarten became autonomous territories of the Kingdom of the Netherlands.

Here is a list of all territory changes reflected in this data:

[4]:
```
bool_series = pd.notnull(region_data["notes"])
region_data[bool_series]
```

[4]:
```
     NOC                       region                         notes
1    AHO                      Curacao          Netherlands Antilles
6    ANT                      Antigua           Antigua and Barbuda
7    ANZ                    Australia                    Australasia
26   BOH                Czech Republic                       Bohemia
51   CRT                       Greece                        Crete
88   HKG                        China                    Hong Kong
93   IOA  Individual Olympic Athletes  Individual Olympic Athletes
99   ISV            Virgin Islands, US                Virgin Islands
143  NBO                      Malaysia                  North Borneo
147  NFL                       Canada                  Newfoundland
168  ROT                          NaN          Refugee Olympic Team
175  SCG                       Serbia         Serbia and Montenegro
179  SKN                   Saint Kitts      Turks and Caicos Islands
205  TTO                      Trinidad           Trinidad and Tobago
208  TUV                          NaN                        Tuvalu
210  UAR                        Syria          United Arab Republic
213  UNK                          NaN                       Unknown
223  WIF                      Trinidad        West Indies Federation
```

```
224  YAR                        Yemen                North Yemen
226  YMD                        Yemen                South Yemen
227  YUG                       Serbia                 Yugoslavia
```

To avoid confusion, I'd like to use athletes' present-day territories (in the "region" column) to conduct analysis based on countries.

The result of merging the two datasets is:

[5]:
```
full_data = event_data.merge(region_data, on="NOC")
full_data.head()
```

[5]:
```
     ID           Name Sex   Age  Height  Weight   Team  NOC        Games  \
0     1      A Dijiang   M  24.0   180.0    80.0  China  CHN  1992 Summer
1     2       A Lamusi   M  23.0   170.0    60.0  China  CHN  2012 Summer
2   602  Abudoureheman   M  22.0   182.0    75.0  China  CHN  2000 Summer
3  1463      Ai Linuer   M  25.0   160.0    62.0  China  CHN  2004 Summer
4  1464      Ai Yanhan   F  14.0   168.0    54.0  China  CHN  2016 Summer

   Year  Season            City       Sport  \
0  1992  Summer       Barcelona  Basketball
1  2012  Summer          London        Judo
2  2000  Summer          Sydney      Boxing
3  2004  Summer          Athina   Wrestling
4  2016  Summer  Rio de Janeiro    Swimming

                                      Event Medal region notes
0               Basketball Men's Basketball   NaN  China   NaN
1             Judo Men's Extra-Lightweight   NaN  China   NaN
2                Boxing Men's Middleweight   NaN  China   NaN
3  Wrestling Men's Lightweight, Greco-Roman   NaN  China   NaN
4     Swimming Women's 200 metres Freestyle   NaN  China   NaN
```

[6]:
```
full_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 270767 entries, 0 to 270766
Data columns (total 17 columns):
 #   Column  Non-Null Count   Dtype
---  ------  --------------   -----
 0   ID      270767 non-null  int64
 1   Name    270767 non-null  object
 2   Sex     270767 non-null  object
 3   Age     261305 non-null  float64
 4   Height  210684 non-null  float64
 5   Weight  207982 non-null  float64
 6   Team    270767 non-null  object
 7   NOC     270767 non-null  object
 8   Games   270767 non-null  object
```

```
 9   Year    270767 non-null  int64
 10  Season  270767 non-null  object
 11  City    270767 non-null  object
 12  Sport   270767 non-null  object
 13  Event   270767 non-null  object
 14  Medal   39774 non-null   object
 15  region  270746 non-null  object
 16  notes   5039 non-null    object
dtypes: float64(3), int64(2), object(12)
memory usage: 37.2+ MB
```

I created a new feature, Medal_Code, to represent whether an athlete won a medal, and if so, what color it was. (Representing losses with null values could make analysis difficult.)

[7]:
```
def medal_code(data):
    if(data['Medal']) == "Gold":
        return 1
    elif(data['Medal']) == "Silver":
        return 2
    elif(data['Medal']) == "Bronze":
        return 3
    else :
        return 4

full_data['Medal Code'] = full_data.apply(medal_code, axis=1)
full_data.head(10)
```

[7]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	\
0	1	A Dijiang	M	24.0	180.0	80.0	China	CHN	1992 Summer	
1	2	A Lamusi	M	23.0	170.0	60.0	China	CHN	2012 Summer	
2	602	Abudoureheman	M	22.0	182.0	75.0	China	CHN	2000 Summer	
3	1463	Ai Linuer	M	25.0	160.0	62.0	China	CHN	2004 Summer	
4	1464	Ai Yanhan	F	14.0	168.0	54.0	China	CHN	2016 Summer	
5	1464	Ai Yanhan	F	14.0	168.0	54.0	China	CHN	2016 Summer	
6	3605	An Weijiang	M	22.0	178.0	72.0	China	CHN	2006 Winter	
7	3605	An Weijiang	M	22.0	178.0	72.0	China	CHN	2006 Winter	
8	3610	An Yulong	M	19.0	173.0	70.0	China	CHN	1998 Winter	
9	3610	An Yulong	M	19.0	173.0	70.0	China	CHN	1998 Winter	

	Year	Season	City	Sport	\
0	1992	Summer	Barcelona	Basketball	
1	2012	Summer	London	Judo	
2	2000	Summer	Sydney	Boxing	
3	2004	Summer	Athina	Wrestling	
4	2016	Summer	Rio de Janeiro	Swimming	
5	2016	Summer	Rio de Janeiro	Swimming	
6	2006	Winter	Torino	Speed Skating	
7	2006	Winter	Torino	Speed Skating	

```
8  1998  Winter          Nagano  Short Track Speed Skating
9  1998  Winter          Nagano  Short Track Speed Skating

                                          Event   Medal region notes  \
0                     Basketball Men's Basketball     NaN  China   NaN
1                   Judo Men's Extra-Lightweight     NaN  China   NaN
2                      Boxing Men's Middleweight     NaN  China   NaN
3        Wrestling Men's Lightweight, Greco-Roman     NaN  China   NaN
4           Swimming Women's 200 metres Freestyle     NaN  China   NaN
5  Swimming Women's 4 x 200 metres Freestyle Relay     NaN  China   NaN
6                  Speed Skating Men's 500 metres     NaN  China   NaN
7               Speed Skating Men's 1,000 metres     NaN  China   NaN
8      Short Track Speed Skating Men's 500 metres  Silver  China   NaN
9    Short Track Speed Skating Men's 1,000 metres     NaN  China   NaN

   Medal Code
0           4
1           4
2           4
3           4
4           4
5           4
6           4
7           4
8           2
9           4
```

There appears to be some missing data for Age, Height, and Weight. These could be good to use as predictors, so I created another dataframe with the rows with missing values removed:

[8]:
```
reduced_data = full_data[full_data['Age'].notna()]
reduced_data = reduced_data[reduced_data['Height'].notna()]
reduced_data = reduced_data[reduced_data['Weight'].notna()]
```

I also added a BMI field to reduced_data:

[9]:
```
reduced_data['BMI'] = reduced_data['Weight']/pow(reduced_data['Height']/100, 2)
```

1.0.3 3. Initial exploration of data

The features and their datatypes are:

[10]:
```
reduced_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 205911 entries, 0 to 270766
Data columns (total 19 columns):
 #   Column      Non-Null Count   Dtype
---  ------      --------------   -----
```

```
 0   ID          205911 non-null  int64
 1   Name        205911 non-null  object
 2   Sex         205911 non-null  object
 3   Age         205911 non-null  float64
 4   Height      205911 non-null  float64
 5   Weight      205911 non-null  float64
 6   Team        205911 non-null  object
 7   NOC         205911 non-null  object
 8   Games       205911 non-null  object
 9   Year        205911 non-null  int64
 10  Season      205911 non-null  object
 11  City        205911 non-null  object
 12  Sport       205911 non-null  object
 13  Event       205911 non-null  object
 14  Medal       30172 non-null   object
 15  region      205895 non-null  object
 16  notes       3493 non-null    object
 17  Medal Code  205911 non-null  int64
 18  BMI         205911 non-null  float64
dtypes: float64(4), int64(3), object(12)
memory usage: 31.4+ MB
```

I found a link to the dataset on Kaggle (https://www.kaggle.com/datasets/heesoo37/120-years-of-olympic-history-athletes-and-results), which helped me fi: 1. Height is in centimeters

2. Weight is in kilograms
3. Only medals are tracked; 4th place or below is represented by a null value for Medal

The statistics for the numerical columns are:

[12]:
```
reduced_data.describe()
```

[12]:

	ID	Age	Height	Weight	\
count	205911.000000	205911.000000	205911.000000	205911.000000	
mean	68598.171924	25.057797	175.377425	70.695325	
std	38993.266878	5.481388	10.547082	14.342344	
min	1.000000	11.000000	127.000000	25.000000	
25%	35177.000000	21.000000	168.000000	60.000000	
50%	68607.000000	24.000000	175.000000	70.000000	
75%	102286.000000	28.000000	183.000000	79.000000	
max	135571.000000	71.000000	226.000000	214.000000	

	Year	Medal Code	BMI
count	205911.000000	205911.000000	205911.000000
mean	1989.663262	3.706835	22.785031
std	20.134386	0.774104	2.912365
min	1896.000000	1.000000	8.360954
25%	1976.000000	4.000000	20.957171

```
50%      1992.000000       4.000000      22.530864
75%      2006.000000       4.000000      24.212293
max      2016.000000       4.000000      63.901580
```

There are 205,911 records (excluding missing values). The number of distinct values for non-binary categorical variables are:

[13]:
```
pysqldf("""SELECT
           COUNT(DISTINCT Name),
           COUNT(DISTINCT Team),
           COUNT(DISTINCT NOC),
           COUNT(DISTINCT region),
           COUNT(DISTINCT Games),
           COUNT(DISTINCT Year),
           COUNT(DISTINCT City),
           COUNT(DISTINCT Sport),
           COUNT(DISTINCT Event)
           FROM reduced_data""")
```

[13]:
```
   COUNT(DISTINCT Name)  COUNT(DISTINCT Team)  COUNT(DISTINCT NOC)  \
0                 98445                   655                  225

   COUNT(DISTINCT region)  COUNT(DISTINCT Games)  COUNT(DISTINCT Year)  \
0                     205                     51                    35

   COUNT(DISTINCT City)  COUNT(DISTINCT Sport)  COUNT(DISTINCT Event)
0                    42                    56                   590
```

There are 51 games represented, but only 35 years. I expected there to be 70, since the Summer and Winter games occur in different years. However, as it turns out, they occurred in the same year until 1994; I checked Wikipedia to confirm this.

While on Wikipedia, I also noticed that one Olympics, the 1956 Summer Games, had two host cities. The main host city was Melbourne, Australia; however, due to the country's quarantine regulations on horses, the equestrian events were held in Stockholm, Sweden.

The countries with the top 10 medal counts are:

[14]:
```
pysqldf("""SELECT region, COUNT(*) AS medal_count
           FROM reduced_data WHERE ("Medal Code" = 1 OR "Medal Code" = 2 OR␣
 ↪"Medal Code" = 3)
           GROUP BY region
           ORDER BY medal_count DESC
           LIMIT 10""")
```

[14]:
```
      region  medal_count
0        USA         4383
1     Russia         3610
2    Germany         3189
```

```
3  Australia         1210
4      Italy         1060
5     Canada         1060
6         UK         1031
7      China          989
8     France          987
9      Japan          843
```

Strangely, these numbers seem much higher than the actual historical medal counts: for example, the USA has actually won only 2980 medals (as of 2022). This is probably because for team sports, all athletes on a team are counted. Perhaps I will account for this in my analysis.

Here are the first few records for Michael Phelps:

[15]:
```
pysqldf("""SELECT *
           FROM full_data
           WHERE Name = 'Michael Fred Phelps, II'
           LIMIT 5""")
```

[15]:
```
       ID                     Name Sex   Age  Height  Weight           Team  \
0  94406  Michael Fred Phelps, II   M  15.0   193.0    91.0  United States
1  94406  Michael Fred Phelps, II   M  19.0   193.0    91.0  United States
2  94406  Michael Fred Phelps, II   M  19.0   193.0    91.0  United States
3  94406  Michael Fred Phelps, II   M  19.0   193.0    91.0  United States
4  94406  Michael Fred Phelps, II   M  19.0   193.0    91.0  United States

   NOC        Games  Year  Season    City     Sport  \
0  USA  2000 Summer  2000  Summer  Sydney  Swimming
1  USA  2004 Summer  2004  Summer  Athina  Swimming
2  USA  2004 Summer  2004  Summer  Athina  Swimming
3  USA  2004 Summer  2004  Summer  Athina  Swimming
4  USA  2004 Summer  2004  Summer  Athina  Swimming

                                        Event   Medal region notes  \
0           Swimming Men's 200 metres Butterfly    None    USA  None
1           Swimming Men's 200 metres Freestyle  Bronze    USA  None
2  Swimming Men's 4 x 100 metres Freestyle Relay  Bronze    USA  None
3  Swimming Men's 4 x 200 metres Freestyle Relay    Gold    USA  None
4           Swimming Men's 100 metres Butterfly    Gold    USA  None

   Medal Code
0           4
1           3
2           3
3           1
4           1
```

It appears that the age column is updated to reflect Phelps' age at the time he competed, yet his height and weight are reported as the same across the years.

Let's verify this by looking at another athlete, Usain Bolt:

[16]:
```
pysqldf("""SELECT *
           FROM full_data
           WHERE Name = 'Usain St. Leo Bolt'
           LIMIT 5""")
```

[16]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	\
0	13029	Usain St. Leo Bolt	M	17.0	196.0	95.0	Jamaica	JAM	
1	13029	Usain St. Leo Bolt	M	21.0	196.0	95.0	Jamaica	JAM	
2	13029	Usain St. Leo Bolt	M	21.0	196.0	95.0	Jamaica	JAM	
3	13029	Usain St. Leo Bolt	M	21.0	196.0	95.0	Jamaica	JAM	
4	13029	Usain St. Leo Bolt	M	25.0	196.0	95.0	Jamaica	JAM	

	Games	Year	Season	City	Sport	\
0	2004 Summer	2004	Summer	Athina	Athletics	
1	2008 Summer	2008	Summer	Beijing	Athletics	
2	2008 Summer	2008	Summer	Beijing	Athletics	
3	2008 Summer	2008	Summer	Beijing	Athletics	
4	2012 Summer	2012	Summer	London	Athletics	

	Event	Medal	region	notes	Medal Code
0	Athletics Men's 200 metres	None	Jamaica	None	4
1	Athletics Men's 100 metres	Gold	Jamaica	None	1
2	Athletics Men's 200 metres	Gold	Jamaica	None	1
3	Athletics Men's 4 x 100 metres Relay	None	Jamaica	None	4
4	Athletics Men's 100 metres	Gold	Jamaica	None	1

So it appears that the height and weight column is the same value for all records for any particular athlete (perhaps it's an average).

Now let's look at some initial statistics, starting with average statistics for medal winners:

[17]:
```
pysqldf("""SELECT AVG(Age),
           AVG(Height),
           AVG(Weight),
           AVG(BMI)
           FROM reduced_data
           WHERE 'Medal Code' IS NOT 4
           """)
```

[17]:

	AVG(Age)	AVG(Height)	AVG(Weight)	AVG(BMI)
0	25.057797	175.377425	70.695325	22.785031

Now let's see averages among male and female medal winners:

[18]:
```
pysqldf("""SELECT Sex,
           AVG(Age),
           AVG(Height),
```

```
            AVG(Weight),
            AVG(BMI)
            FROM reduced_data
            WHERE 'Medal Code' IS NOT 4
            GROUP BY Sex
            """)
```

[18]:

	Sex	AVG(Age)	AVG(Height)	AVG(Weight)	AVG(BMI)
0	F	23.781552	167.867261	60.027223	21.193275
1	M	25.667887	178.967548	75.795053	23.545946

Values for females tend to be lower than those for males.

Now, let's look at medal winners in throwing events (a good example of an event type requiring strength) and in track events (a good example of an event type requiring speed):

[19]:
```
pysqldf("""SELECT AVG(Age),
            AVG(Height),
            AVG(Weight),
            AVG(BMI)
            FROM reduced_data
            WHERE 'Medal Code' IS NOT 4
            AND (Event LIKE '%Shot Put%'
            OR Event LIKE '%Discus%'
            OR Event LIKE '%Javelin%'
            OR Event LIKE '%Hammer%')
            """)
```

[19]:

	AVG(Age)	AVG(Height)	AVG(Weight)	AVG(BMI)
0	26.610873	183.246917	95.794991	28.308878

[20]:
```
pysqldf("""SELECT AVG(Age),
            AVG(Height),
            AVG(Weight),
            AVG(BMI)
            FROM reduced_data
            WHERE 'Medal Code' IS NOT 4
            AND Sport = 'Athletics'
            AND Event LIKE '%metres%'
            """)
```

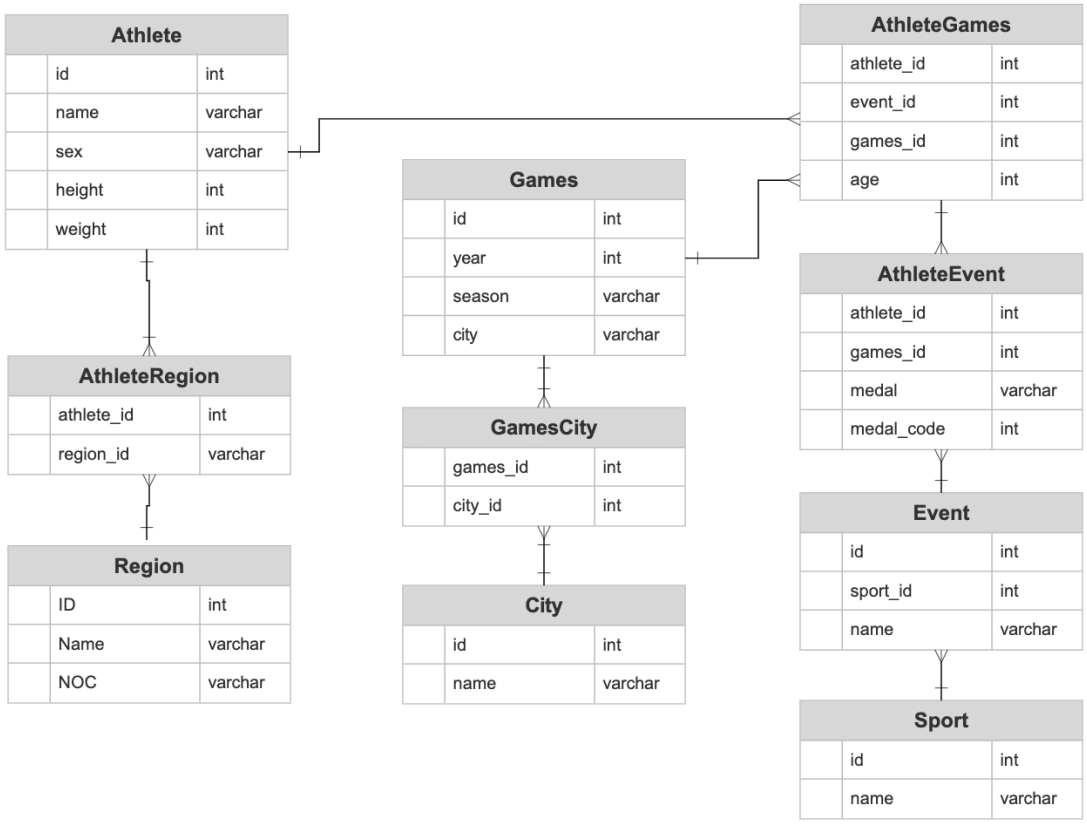
[20]:

	AVG(Age)	AVG(Height)	AVG(Weight)	AVG(BMI)
0	24.780823	174.664068	65.049201	21.223863

1.0.4 4. ERD Diagram

[21]:
```
from IPython.display import Image
from IPython.core.display import HTML
Image(filename = "ERDDiagram.png")
```

[21]:



2 Proposal

2.0.1 Description:

For my project, I am using the dataset entitled "Olympics Dataset - 120 years of data." The dataset includes records of all athletes who have competed in the modern Olympics as of 2016. Properties documented include their age at the time of a certain event, their (supposedly) average height and weight, and the medal they won (if any) in each event they participated in. I am interested in examining the effects that age, height, and weight have on performance, as well as how these effects differ between males and females and between various types of events. These insights would be beneficial to health professionals and personal trainers.

2.0.2 Questions:

1. What effect does age have on performance?

2. What effect do height, weight, and BMI (a way to express the ratio between them) have on performance?
3. How do these effects differ for men and women, and between different events?

2.0.3 Hypotheses

1. The best-performing athletes will be those in their mid-to-late 20's.
2. BMI of the best athletes will generally be near the middle of the "healthy" range (about 20-23)
3. Height and weight will be moderately to strongly positively correlated
4. Height, weight, and BMI in women will be less than those in men
5. In events requiring mostly strength, successful athletes will have higher height/weight/BMI values. In those requiring mostly speed/agility, they will have lower values.

2.0.4 Initial findings (from EDA above)

1. Avg age of medal winners is 25.7 for men, 23.8 for women. This calls hypothesis 1 into question: maybe early-to-mid 20's is in fact the optimal age range?
2. On average, among medal winners, BMI indeed seems lower for women than for men (21.2 vs. 23.5). Seems to support hypothesis 4, but perhaps calls hypothesis 2 into question for men…
3. In throwing sports (requiring strength), medal winners' weights average 95.8 kg, and their BMIs average 28.3. In track events (requiring speed), the average weight is 65 kg, and the average BMI is 21.2. Seems to support hypothesis 5.

2.0.5 Approach

To conduct my analysis, I will create various graphs. I will show the distribution of height, weight, age, and BMI with histograms. I will create such histograms for all the data, and also plan to create separate ones showing the distributions among men, women, medal winners, gold winners, or athletes in a certain sport only. I also might create graphs to visualize the distributions of average height, weight, age, and/or BMI between different sports. Depending on how much time I have, I might also be interested in investigating how these trends have changed over time, and/or country trends - for example, how athletes tend to perform in their home countries, and which countries are particularly dominant in given sports.

[ ]: