2. Descriptive Stats

March 7, 2024

1 2. Descriptive Statistics

Load Data and set SQL function

```
[1]: from pandasql import sqldf
     import pandas as pd
     import plotly.express as px
```

```
[2]: df1 = pd.read_csv("athlete_events_file.csv")
     df2 = pd.read_csv("noc_regions.csv")

     df = pd.merge(df1, df2, on = "NOC")

     pysqldf = lambda q: sqldf(q, globals())
```

Number of athletes

```
[3]: pysqldf("""
             SELECT Season,
             COUNT(DISTINCT(NAME)) AS Athletes
             FROM df
             GROUP BY Season
             """)
```

```
[3]:    Season  Athletes
     0  Summer    115956
     1  Winter     18923
```

```
[4]: pysqldf("""
             SELECT Sex,
             COUNT(DISTINCT(NAME)) AS Athletes
             FROM df
             GROUP BY Sex
             """)
```

```
[4]:   Sex  Athletes
     0   F     33755
     1   M    100866
```

```
[5]: pysqldf("""
            SELECT Sex, Season,
            COUNT(DISTINCT(NAME)) AS Athletes
            FROM df
            GROUP BY Sex, Season
            """)
```

```
[5]:   Sex  Season  Athletes
     0   F  Summer     28668
     1   F  Winter      5159
     2   M  Summer     87335
     3   M  Winter     13766
```

Max, **Min** and **Average** athletes' age

```
[6]: pysqldf("""
            SELECT Season,
            MAX(Age) AS Max_Age,
            MIN(Age) AS Min_Age,
            AVG(Age) AS Average_Age
            FROM df
            GROUP BY Season
            """)
```

```
[6]:    Season  Max_Age  Min_Age  Average_Age
     0  Summer     97.0     10.0    25.677776
     1  Winter     58.0     11.0    25.039147
```

```
[7]: pysqldf("""
            SELECT *
            FROM df
            WHERE Age IN (10, 97) AND Season = "Summer"

            UNION

            SELECT *
            FROM df
            WHERE Age IN (11, 58) AND Season = "Winter"

            ORDER by Age
            """)
```

```
[7]:       ID                                    Name Sex   Age  Height  \
     0  71691                      Dimitrios Loundras   M  10.0     NaN
     1  22411               Magdalena Cecilia Colledge   F  11.0   152.0
     2  47618     Sonja Henie (-Topping, -Gardiner, -Onstad)   F  11.0   155.0
     3  51268                          Beatrice Hutiu   F  11.0   151.0
```

```
4   52070                                   Etsuko Inada   F  11.0      NaN
5   70616                                     Liu Luyang   F  11.0      NaN
6   76675                              Marcelle Matthews   F  11.0      NaN
7  118925  Megan Olwen Devenish Taylor (-Mandeville-Ellis)   F  11.0   157.0
8   64263                    Carl August Verner Kronlund   M  58.0      NaN
9  128719                         John Quincy Adams Ward   M  97.0      NaN

   Weight                           Team  NOC        Games  Year  Season  \
0     NaN  Ethnikos Gymnastikos Syllogos  GRE  1896 Summer  1896  Summer
1     NaN                  Great Britain  GBR  1932 Winter  1932  Winter
2    45.0                         Norway  NOR  1924 Winter  1924  Winter
3    38.0                        Romania  ROU  1968 Winter  1968  Winter
4     NaN                          Japan  JPN  1936 Winter  1936  Winter
5     NaN                          China  CHN  1988 Winter  1988  Winter
6     NaN                   South Africa  RSA  1960 Winter  1960  Winter
7     NaN                  Great Britain  GBR  1932 Winter  1932  Winter
8     NaN                         Sweden  SWE  1924 Winter  1924  Winter
9     NaN                  United States  USA  1928 Summer  1928  Summer

                     City             Sport  \
0                  Athina        Gymnastics
1             Lake Placid    Figure Skating
2                Chamonix    Figure Skating
3                Grenoble    Figure Skating
4  Garmisch-Partenkirchen    Figure Skating
5                 Calgary    Figure Skating
6            Squaw Valley    Figure Skating
7             Lake Placid    Figure Skating
8                Chamonix           Curling
9               Amsterdam  Art Competitions

                                         Event   Medal        region notes
0        Gymnastics Men's Parallel Bars, Teams  Bronze        Greece  None
1               Figure Skating Women's Singles    None            UK  None
2               Figure Skating Women's Singles    None        Norway  None
3               Figure Skating Women's Singles    None       Romania  None
4               Figure Skating Women's Singles    None         Japan  None
5             Figure Skating Mixed Ice Dancing    None         China  None
6                   Figure Skating Mixed Pairs    None  South Africa  None
7               Figure Skating Women's Singles    None            UK  None
8                        Curling Men's Curling  Silver        Sweden  None
9  Art Competitions Mixed Sculpturing, Statues    None           USA  None
```

[8]:
```
fig = px.box(df, x = "Age", y = "Season", color= "Sex")
fig.show()
```

Max, **Min** and **Average** athletes' height

[9]:

```
pysqldf("""
        SELECT Season,
        MAX(Height) AS Max_Height,
        MIN(Height) AS Min_Height,
        AVG(Height) AS Average_Height
        FROM df
        GROUP BY Season
        """)
```

[9]:

```
   Season  Max_Height  Min_Height  Average_Height
0  Summer       226.0       127.0      175.522349
1  Winter       211.0       137.0      174.590112
```

[10]:

```
fig = px.box(df, x = "Height", y = "Season", color= "Sex")
fig.show()
```

Max, **Min** and **Average** athletes' weight

[11]:

```
pysqldf("""
        SELECT Season,
        MAX(Weight) AS Max_Weight,
        MIN(Weight) AS Min_Weight,
        AVG(Weight) AS Average_Weight
        FROM df
        GROUP BY Season
        """)
```

[11]:

```
   Season  Max_Weight  Min_Weight  Average_Weight
0  Summer       214.0        25.0       70.697843
1  Winter       145.0        32.0       70.759275
```

[12]:

```
fig = px.box(df, x = "Weight", y = "Season", color= "Sex")
fig.show()
```

Countries with most **Number** of Athletes in Summer Olympics

[13]:

```
pysqldf("""
        SELECT Season,
        NOC,
        TEAM,
        COUNT(Name) AS Count
        FROM df
        WHERE Season = "Summer"
        GROUP BY NOC
        ORDER BY Count DESC
        LIMIT 3
        """)
```

```
[13]:    Season  NOC           Team  Count
      0  Summer  USA  United States  15064
      1  Summer  GBR  Great Britain  10917
      2  Summer  FRA         France  10633
```

```
[14]: pysqldf("""
              SELECT Season,
              NOC,
              TEAM,
              COUNT(Name) AS Count
              FROM df
              WHERE Season = "Summer"
              GROUP BY NOC
              ORDER BY Count ASC
              LIMIT 3
              """)
```

```
[14]:    Season  NOC          Team  Count
      0  Summer  NFL  Newfoundland      1
      1  Summer  NBO  North Borneo      2
      2  Summer  UNK       Unknown      2
```

```
[15]: fig = px.scatter_geo(df[df["Season"] == "Summer"][["NOC","region"]].
        ↪value_counts().reset_index(),
                           locations = "NOC",
                           size = "count",
                           color_discrete_sequence = ['#F9AE19'],
                           projection = "natural earth",
                           hover_name = "region",
                           title = "Summer Athletes")
      fig.show()
```

Countries with most **Number** of Athletes in Winter Olympics

```
[16]: pysqldf("""
              SELECT Season,
              NOC,
              TEAM,
              COUNT(Name) AS Count
              FROM df
              WHERE Season = "Winter"
              GROUP BY NOC
              ORDER BY Count DESC
              LIMIT 3
              """)
```

```
[16]:    Season  NOC           Team  Count
      0  Winter  USA  United States   3789
      1  Winter  CAN         Canada   2873
      2  Winter  ITA          Italy   2498
```

```
[17]: pysqldf("""
              SELECT Season,
              NOC,
              TEAM,
              COUNT(Name) AS Count
              FROM df
              WHERE Season = "Winter"
              GROUP BY NOC
              ORDER BY Count ASC
              LIMIT 3
              """)
```

```
[17]:    Season  NOC      Team  Count
      0  Winter  DMA  Dominica      1
      1  Winter  GHA     Ghana      1
      2  Winter  GUM      Guam      1
```

```
[18]: fig = px.scatter_geo(df[df["Season"] == "Winter"][["NOC","region"]].
        ↪value_counts().reset_index(),
                           locations = "NOC",
                           size = "count",
                           color_discrete_sequence = ['#0BD1D4'],
                           projection = "natural earth",
                           hover_name = "region",
                           title = "Winter Athletes")

      fig.show()
```

Medals

```
[19]: df_summer_medals = pd.pivot_table(df[df["Season"] == "Summer"][["NOC","Medal"]].
        ↪value_counts().reset_index(), index = "NOC", columns = "Medal", values =␣
        ↪"count")
      df_summer_medals["Sum"] = df_summer_medals[["Gold", "Silver", "Bronze"]].
        ↪sum(axis = 1)
      df_summer_medals = df_summer_medals.sort_values(by = ["Sum"], ascending = False)
      df_summer_medals.reset_index(inplace  = True)
      df_summer_medals = pd.merge(df_summer_medals, df2[["NOC", "region"]], on =␣
        ↪"NOC")
      df_summer_medals = df_summer_medals.groupby("region").sum().reset_index().
        ↪sort_values(by = ["Sum"], ascending = False)
```

```
athletes_summer = pd.DataFrame(df[df["Season"] == "Summer"][["region", "Name"]].
 ↪value_counts().groupby("region").sum()).rename(columns = {"count" :␣
 ↪"Athletes"}).sort_values(by = ["Athletes"], ascending = False).reset_index()
df_summer_medals = pd.merge(df_summer_medals, athletes_summer, on = "region")

df_summer_medals["medals/athlete"] = df_summer_medals["Sum"]/
 ↪df_summer_medals["Athletes"]

df_summer_medals.head()
```

[19]:

```
    region        NOC  Bronze    Gold  Silver     Sum  Athletes  \
0      USA        USA  1197.0  2472.0  1333.0  5002.0     15064
1   Russia  URSRUSEUN   994.0  1220.0   974.0  3188.0      8855
2  Germany  GERGDRFRG  1064.0  1075.0   987.0  3126.0     12377
3       UK        GBR   620.0   636.0   729.0  1985.0     10917
4   France        FRA   587.0   465.0   575.0  1627.0     10633

   medals/athlete
0        0.332050
1        0.360023
2        0.252565
3        0.181827
4        0.153014
```

[20]:

```
fig = px.bar(df_summer_medals[:20],
             x = "region",
             y = ["Gold", "Silver", "Bronze"],
             title = "Summer Medals",
             color_discrete_map = {'Gold':'#FFD700',
                                   'Silver': '#C0C0C0',
                                   'Bronze': '#CD7F32'},
            category_orders  = {"Medal": ['Gold', 'Silver', "Bronze"]})
fig.show()
```

[21]:

```
fig = px.bar(df_summer_medals.sort_values(by = ["medals/athlete"], ascending =␣
 ↪False)[:20],
             x = "region",
             y = "medals/athlete",
             title = "Summer Medals per Athlete",
             color_discrete_sequence = ['#F9AE19'])
fig.show()
```

[22]:

```
pd.DataFrame(df[(df["Medal"] == "Gold") & (df["Season"] ==␣
 ↪"Summer")][["region", "Sport"]].value_counts()).reset_index().sort_values(by␣
 ↪= ["count"], ascending = False)[:6]
```

[22]:
```
     region       Sport  count
0       USA    Swimming    649
1       USA   Athletics    542
2       USA  Basketball    281
3   Germany      Rowing    272
4       USA      Rowing    186
5    Russia  Gymnastics    176
```

[23]:
```
pd.DataFrame(df[(df["Medal"] == "Silver") & (df["Season"] ==␣
 ↪"Summer")][["region", "Sport"]].value_counts()).reset_index().sort_values(by␣
 ↪= ["count"], ascending = False)[:6]
```

[23]:
```
      region       Sport  count
0        USA   Athletics    317
1        USA    Swimming    254
2  Australia    Swimming    165
3     Russia  Gymnastics    142
4    Germany    Swimming    137
5      Italy     Fencing    136
```

[24]:
```
pd.DataFrame(df[(df["Medal"] == "Bronze") & (df["Season"] ==␣
 ↪"Summer")][["region", "Sport"]].value_counts()).reset_index().sort_values(by␣
 ↪= ["count"], ascending = False)[:6]
```

[24]:
```
      region      Sport  count
0        USA  Athletics    221
1        USA   Swimming    175
2    Germany   Swimming    148
3    Germany  Athletics    143
4         UK  Athletics    127
5  Australia   Swimming    124
```

[25]:
```
df_winter_medals = pd.pivot_table(df[df["Season"] == "Winter"][["NOC","Medal"]].
 ↪value_counts().reset_index(), index = "NOC", columns = "Medal", values =␣
 ↪"count")
df_winter_medals["Sum"] = df_winter_medals[["Gold", "Silver", "Bronze"]].
 ↪sum(axis = 1)
df_winter_medals = df_winter_medals.sort_values(by = ["Sum"], ascending = False)
df_winter_medals.reset_index(inplace  = True)
df_winter_medals = pd.merge(df_winter_medals, df2[["NOC", "region"]], on =␣
 ↪"NOC")
df_winter_medals = df_winter_medals.groupby("region").sum().reset_index().
 ↪sort_values(by = ["Sum"], ascending = False)

athletes_winter = pd.DataFrame(df[df["Season"] == "Winter"][["region", "Name"]].
 ↪value_counts().groupby("region").sum()).rename(columns = {"count" :␣
 ↪"Athletes"}).sort_values(by = ["Athletes"], ascending = False).reset_index()
```

```
df_winter_medals = pd.merge(df_winter_medals, athletes_winter, on = "region")

df_winter_medals["medals/athlete"] = df_winter_medals["Sum"]/
 ↪df_winter_medals["Athletes"]

df_winter_medals.head()
```

```
[25]:    region        NOC  Bronze   Gold  Silver    Sum  Athletes  medals/athlete
      0   Russia  URSRUSEUN   184.0  379.0   196.0  759.0      2837        0.267536
      1      USA        USA   161.0  166.0   308.0  635.0      3789        0.167590
      2  Germany  GERGDRFRG   196.0  226.0   208.0  630.0      3506        0.179692
      3   Canada        CAN   107.0  305.0   199.0  611.0      2873        0.212670
      4   Norway        NOR   127.0  151.0   165.0  443.0      2362        0.187553
```

```
[26]: fig = px.bar(df_winter_medals[:20],
                   x = "region",
                   y = ["Gold", "Silver", "Bronze"],
                   title = "Winter Medals",
                   color_discrete_map = {'Gold':'#FFD700',
                                         'Silver': '#C0C0C0',
                                         'Bronze': '#CD7F32'},
                  category_orders  = {"Medal": ['Gold', 'Silver', "Bronze"]})
      fig.show()
```

```
[27]: fig = px.bar(df_winter_medals.sort_values(by = ["medals/athlete"], ascending =␣
       ↪False)[:20],
                   x = "region",
                   y = "medals/athlete",
                   title = "Winter Medals per Athlete",
                   color_discrete_sequence = ['#0BD1D4'])
      fig.show()
```

```
[28]: df[(~df["Medal"].isna()) & (df["region"] == "India") & (df["Season"] ==␣
       ↪"Winter")]["Sport"].unique()
```

```
[28]: array(['Alpinism'], dtype=object)
```

```
[29]: df[(~df["Medal"].isna()) & (df["region"] == "Finland") & (df["Season"] ==␣
       ↪"Winter")]["Sport"].unique()
```

```
[29]: array(['Ice Hockey', 'Ski Jumping', 'Cross Country Skiing',
             'Military Ski Patrol', 'Biathlon', 'Speed Skating',
             'Nordic Combined', 'Figure Skating', 'Curling', 'Snowboarding',
             'Freestyle Skiing', 'Alpine Skiing'], dtype=object)
```

```
[30]:
```

```
pd.DataFrame(df[(df["Medal"] == "Gold") & (df["Season"] ==␣
 ↪"Winter")][["region", "Sport"]].value_counts()).reset_index().sort_values(by␣
 ↪= ["count"], ascending = False)[:6]
```

[30]:

```
   region                Sport  count
0  Canada           Ice Hockey    212
1  Russia           Ice Hockey    153
2  Russia  Cross Country Skiing     73
3     USA           Ice Hockey     56
4  Russia       Figure Skating     56
5  Norway  Cross Country Skiing     54
```

[31]:

```
pd.DataFrame(df[(df["Medal"] == "Silver") & (df["Season"] ==␣
 ↪"Winter")][["region", "Sport"]].value_counts()).reset_index().sort_values(by␣
 ↪= ["count"], ascending = False)[:6]
```

[31]:

```
           region                 Sport  count
0             USA            Ice Hockey    178
1          Canada            Ice Hockey     93
2  Czech Republic            Ice Hockey     74
3          Norway  Cross Country Skiing     73
4          Sweden            Ice Hockey     72
5          Russia  Cross Country Skiing     51
```

[32]:

```
pd.DataFrame(df[(df["Medal"] == "Bronze") & (df["Season"] ==␣
 ↪"Winter")][["region", "Sport"]].value_counts()).reset_index().sort_values(by␣
 ↪= ["count"], ascending = False)[:6]
```

[32]:

```
           region                 Sport  count
0         Finland            Ice Hockey    129
1          Sweden            Ice Hockey     99
2  Czech Republic            Ice Hockey     81
3         Finland  Cross Country Skiing     64
4          Russia  Cross Country Skiing     48
5     Switzerland            Ice Hockey     48
```

2 2. Key Points

In my analysis of the Olympics dataset spanning various years, several intriguing patterns and statistics have emerged, shedding light on the diverse dynamics of the games across different seasons and countries.

Firstly, it's evident that the Summer Olympics attract significantly more athletes than their Winter counterpart, with a ratio of 6:1. Moreover, there's a notable gender skew, with male athletes outnumbering female athletes by a ratio of 3:1 across both seasons.

Age-wise, the spectrum of participants ranges widely, from the youngest at 10 years in Gymnastics for the Summer Olympics to 11 years in Figure Skating for the Winter Games. Conversely, the oldest competitors clock in at 97 years for the Summer Olympics (in Art Competitions) and 58 years for the Winter Olympics (in Curling).

Interestingly, statistical analysis reveals that male athletes tend to be older, taller, and heavier compared to their female counterparts, regardless of the season.

When it comes to national representation, the United States, Great Britain, and France boast the highest number of athletes in the Summer Olympics, while the Winter Olympics see dominance from the United States, Canada, and Italy.

Geographically, most Winter Olympic athletes hail from the northern hemisphere, though outliers like Argentina, Australia, and New Zealand also participate actively.

In terms of medal tallies, the top three countries in the Summer Olympics are the USA, Russia, and Germany. Meanwhile, in the Winter Olympics, Russia leads the pack, followed closely by the USA and Germany.

An intriguing metric emerges when examining the ratio of total medals to total athletes per country. While Russia, the USA, and Germany maintain their dominance in the Summer Olympics, the Winter Games witness an unexpected shift, with Russia, India, and Finland boasting the most efficient medal-per-athlete ratios. India's prowess in winter sports, particularly Alpinism, stands out, while Finland's strength lies in traditional winter disciplines like Ice Hockey, Ski Jumping, and Cross Country Skiing.

Finally, the top three countries in terms of gold medals reflect their prowess in specific sports. In the Summer Olympics, the USA dominates in Swimming, Germany excels in Rowing, and Russia shines in Gymnastics. Meanwhile, in the Winter Olympics, Canada, Russia, and the USA dominate the podium in Ice Hockey.

3 3. Hypothesis Analysis

Hypothesis `Certain countries may exhibit a consistent dominance in specific sports across both summer and winter Olympic games. For example, countries with a strong tradition in winter sports may perform exceptionally well in disciplines like skiing or ice hockey during the Winter Olympics`

The hypothesis that certain countries maintain a consistent dominance in specific sports across both the Summer and Winter Olympics is indeed supported by the data analysis conducted. This pattern highlights the deep-rooted traditions and strengths that certain nations possess in particular athletic endeavors.

One of the most prominent examples of this phenomenon is evident in the realm of winter sports. Countries with a rich heritage and infrastructure in winter sports, such as Finland, Canada, and Russia, consistently excel in disciplines like Ice Hockey, Ski Jumping, and Cross Country Skiing. This dominance extends across multiple editions of the Winter Olympics, showcasing the enduring prowess of these nations in these specialized events.

Similarly, in the Summer Olympics, we observe certain countries consistently leading the medal tables in specific sports. The United States, for instance, has historically been dominant in Swimming, while Germany has showcased remarkable strength in Rowing. Russia's stronghold in Gymnastics further exemplifies this trend.

This consistency in performance can be attributed to various factors, including robust training programs, cultural emphasis on certain sports, and investment in infrastructure. For countries with a strong tradition in winter sports, the availability of suitable terrain and facilities plays a crucial role in nurturing talent and fostering excellence.

Furthermore, the specialization of athletes and coaches in particular sports over generations contributes to the sustained success of these nations. By focusing resources and efforts on disciplines where they have a competitive advantage, countries can maintain their position at the forefront of Olympic competition.

Overall, the data analysis supports the hypothesis that certain countries exhibit a consistent dominance in specific sports across both the Summer and Winter Olympics. This trend underscores the enduring legacy of athletic excellence and the profound impact of cultural heritage and investment in sports development.

4 4. Additional Questions

1. **Correlation between age and performance**: Is there a correlation between the age of athletes and their performance in different sports?
2. **Gender disparity in sports participation**: Despite the overall gender ratio of athletes being 3:1, are there specific sports or countries where the gender gap is narrower or wider?
3. **Long-term trends in Olympic performance**: How has the performance of countries in the Olympics evolved over time? Are there any noticeable trends in terms of the rise or decline of certain nations in specific sports or across seasons?