Week 3 SportStats Analysis

June 25, 2023

```python
[52]: import pandas as pd
      import pandasql as ps
      import matplotlib.pyplot as plt
```

```python
[53]: noc_regions = pd.read_csv('Data/noc_regions.csv')
```

```python
[54]: athlete_events = pd.read_csv('Data/athlete_events.csv')
```

```python
[55]: athlete_events.head()
```

```
[55]:    ID                    Name Sex   Age  Height  Weight            Team  \
       0   1              A Dijiang   M  24.0   180.0    80.0           China
       1   2               A Lamusi   M  23.0   170.0    60.0           China
       2   3     Gunnar Nielsen Aaby   M  24.0     NaN     NaN         Denmark
       3   4    Edgar Lindenau Aabye   M  34.0     NaN     NaN  Denmark/Sweden
       4   5  Christine Jacoba Aaftink   F  21.0   185.0    82.0     Netherlands

          NOC        Games  Year  Season       City          Sport  \
       0  CHN  1992 Summer  1992  Summer  Barcelona     Basketball
       1  CHN  2012 Summer  2012  Summer     London          Judo
       2  DEN  1920 Summer  1920  Summer  Antwerpen       Football
       3  DEN  1900 Summer  1900  Summer      Paris     Tug-Of-War
       4  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
```

```python
[56]: noc_regions.head()
```

```
[56]:    NOC      region                 notes
       0  AFG  Afghanistan                   NaN
       1  AHO      Curacao  Netherlands Antilles
       2  ALB      Albania                   NaN
```

```
3  ALG      Algeria                 NaN
4  AND      Andorra                 NaN
```

[85]:
```
query1 = "SELECT Sport, SUM(CASE WHEN Medal != 'NaN' THEN 1 ELSE 0 END) AS␣
↪'Total Medal', SUM(CASE WHEN Medal = 'Gold' THEN 1 ELSE 0 END)  AS 'Gold␣
↪Medal', SUM(CASE WHEN Medal == 'Silver' THEN 1 ELSE 0 END) AS 'Silver␣
↪Medal', SUM(CASE WHEN Medal == 'Bronze' THEN 1 ELSE 0 END) AS 'Bronze Medal'␣
↪FROM athlete_events GROUP BY Sport ORDER BY COUNT(Medal) DESC LIMIT 6"
```

[86]:
```
sport_medal_count = ps.sqldf(query1, locals())
sport_medal_count
```

[86]:
```
        Sport  Total Medal  Gold Medal  Silver Medal  Bronze Medal
0    Athletics         3969        1339          1334          1296
1     Swimming         3048        1099           993           956
2       Rowing         2945         978           977           990
3    Gymnastics        2256         791           746           719
4      Fencing         1743         594           583           566
5     Football         1571         515           513           543
```

[57]:
```
query2 = "SELECT Name, Sport, MAX(Medal) AS 'Total Medal', Gold AS 'Gold␣
↪Medal', Silver AS 'Silver Medal', Bronze AS 'Bronze Medal' FROM (SELECT␣
↪DISTINCT Name, Sport, SUM(CASE WHEN Medal != 'NaN' THEN 1 ELSE 0 END) AS␣
↪'Medal', SUM(CASE WHEN Medal IS 'Gold' THEN 1 ELSE 0 END) AS Gold, SUM(CASE␣
↪WHEN Medal = 'Silver' THEN 1 ELSE 0 END) AS Silver, SUM(CASE WHEN Medal =␣
↪'Bronze' THEN 1 ELSE 0 END) AS Bronze FROM athlete_events GROUP BY Name)␣
↪GROUP BY Sport ORDER BY MAX(Medal) DESC"
```

[58]:
```
player_medal_count = ps.sqldf(query2, locals())
player_medal_count
```

[58]:
```
                                Name           Sport  Total Medal  \
0              Michael Fred Phelps, II        Swimming           28
1    Larysa Semenivna Latynina (Diriy-)     Gymnastics           18
2                  Edoardo Mangiarotti        Fencing           13
3                   Ole Einar Bjrndalen       Biathlon           13
4              Birgit Fischer-Schmidt        Canoeing           12
..                                 ...             ...          ...
61                       A. M. Woods        Lacrosse            1
62               Alfred James Bowerman         Cricket            1
63       Francisco Villota y Baquiola  Basque Pelota            1
64                     Antarge Sherpa        Alpinism            1
65                   Hermann Schreiber     Aeronautics           1

     Gold Medal  Silver Medal  Bronze Medal
0            23             3             2
1             9             5             4
```

```
2              6              5              2
3              8              4              1
4              8              4              0
..             …              …              …
61             0              1              0
62             1              0              0
63             1              0              0
64             1              0              0
65             1              0              0

[66 rows x 6 columns]
```

[77]:
```
query3 = "SELECT Region, Team, COUNT(DISTINCT Name) AS Athletes, AVG(Age) AS␣
→'Average Age', AVG(Height) AS 'Average Height', AVG(Weight) AS 'Average␣
→Weight', (AVG(Weight)/(AVG(Height)*AVG(Height)))*10000 AS BMI FROM␣
→athlete_events INNER JOIN noc_regions ON athlete_events.NOC = noc_regions.
→NOC GROUP BY Region ORDER BY COUNT(DISTINCT Name) DESC lIMIT 11"
```

[78]:
```
country_athlete_production = ps.sqldf(query3,locals())
country_athlete_production
```

[78]:
```
      region          Team  Athletes  Average Age  Average Height  \
0        USA  United States      9652    26.050606      176.886903
1    Germany        Germany      7541    25.687842      177.060998
2         UK  Great Britain      6273    26.925491      175.722488
3     France         France      6161    26.795863      175.254745
4     Russia         Russia      5597    25.097097      175.729666
5      Italy          Italy      4921    25.877047      175.298943
6     Canada         Canada      4810    25.092806      174.978103
7      Japan          Japan      4036    24.476522      168.228163
8  Australia      Australia      3868    24.934017      176.873466
9     Sweden         Sweden      3782    26.824214      177.942064
10    Poland         Poland      2964    25.683794      175.265043

    Average Weight        BMI
0        72.631871  23.213246
1        71.973115  22.957494
2        70.856799  22.947049
3        69.607815  22.663054
4        71.670728  23.208745
5        70.955604  23.090222
6        70.546604  23.041392
7        63.305896  22.369010
8        72.337996  23.122836
9        73.103839  23.087821
10       71.129381  23.155727
```

```
[ ]:
```