Assignment 02

April 8, 2021

1 Assignment 02: Evaluate the Summer Olympics, London 2012 dataset

The comments/sections provided are your cues to perform the assignment. You don't need to limit yourself to the number of rows/cells provided. You can add additional rows in each section to add more lines of code.

If at any point in time you need help on solving this assignment, view our demo video to understand the different steps of the code.

Happy coding!

---

[3]:
```
"""
Analyse London Olympics Dataset

DESCRIPTION

Problem:

Evaluate the dataset of the Summer Olympics, London 2012 to:

    Find and print the name of the country that won maximum gold medals,
    Find and print the countries who won more than 20 gold medals,
    Print the medal tally,
    Print each country name with the corresponding number of gold medals, and
    Print each country's name with the total number of medals won.
"""
```

[3]:
```
"\nAnalyse London Olympics Dataset\n\nDESCRIPTION\n\nProblem: \n\nEvaluate the
dataset of the Summer Olympics, London 2012 to:\n\n    Find and print the name
of the country that won maximum gold medals,\n    Find and print the countries
who won more than 20 gold medals,\n    Print the medal tally,\n    Print each
country name with the corresponding number of gold medals, and\n    Print each
country's name with the total number of medals won.\n"
```

1: View and add the dataset

[83]:
```
#Import the necessary library
import numpy as np
```

[125]:
```
#Manually add the Summer Olympics, London 2012 dataset as arrays
np_countries = np.array(['GBR','CHN','RUS','US','KOR','JPN','GER'])
np_Olympics2012_golds = np.array([29,38,24,46,13,7,11])
np_Olympics2012_silvers = np.array([17,28,25,28,8,14,11])
np_Olympics2012_bronzes = np.array([19,22,32,29,7,17,14])
```

Find the country with maximum gold medals

[126]:
```
#Use the argmax() method to find the highest number of gold medals
np_Olympics2012_max_golds=np_Olympics2012_golds.argmax()
```

[127]:
```
#Print the name of the country
print(np_countries[np_Olympics2012_max_golds])
```

```
US
```

Find the countries with more than 20 gold medals

[128]:
```
#Use Boolean indexing technique to find the required output
print (np_countries[np_Olympics2012_golds>20])
```

```
['GBR' 'CHN' 'RUS' 'US']
```

Evaluate the dataset and print the name of each country with its gold medals and total number of medals

[168]:
```
#Creating an array by Sorting the Golds in descending order.
#note: argsort() output gives the index positions only.
#Sort_by_Gold has the index positions of Golds in descending order.
Sort_by_Gold=np_Olympics2012_golds.argsort()[::-1]
#using Sort_by_Gold value to create an new array with reference to total no. of␣
 ↪Gold
Countries_by_Gold=np_countries[Sort_by_Gold]
Sorted_Gold_Tally=np_Olympics2012_golds[Sort_by_Gold]
Sorted_Silver_Tally=np_Olympics2012_silvers[Sort_by_Gold]
Sorted_Bronze_Tally=np_Olympics2012_bronzes[Sort_by_Gold]
print('Medal Tally by Gold')
#Use a for loop to create the required output
for i in range(len(np_countries)):
    total_medals=Gold_Tally[i]+Silver_Tally[i]+Bronze_Tally[i]
    print ('Rank: {} Country: {} Gold: {} Silver: {} Bronze: {} Total_Medals:␣
 ↪{}'.
 ↪format(i+1,Countries_by_Gold[i],Sorted_Gold_Tally[i],Sorted_Silver_Tally[i],Sorted_Bronze_T
```

```
Medal Tally by Gold
Rank: 1 Country: US Gold: 46 Silver: 28 Bronze: 29 Total_Medals: 103
Rank: 2 Country: CHN Gold: 38 Silver: 28 Bronze: 22 Total_Medals: 88
Rank: 3 Country: GBR Gold: 29 Silver: 17 Bronze: 19 Total_Medals: 65
```

```
Rank: 4 Country: RUS Gold: 24 Silver: 25 Bronze: 32 Total_Medals: 81
Rank: 5 Country: KOR Gold: 13 Silver: 8 Bronze: 7 Total_Medals: 28
Rank: 6 Country: GER Gold: 11 Silver: 11 Bronze: 14 Total_Medals: 36
Rank: 7 Country: JPN Gold: 7 Silver: 14 Bronze: 17 Total_Medals: 38
```

[ ]: