FORE School of Management

New Delhi

Name: Vishal Kaushik

Roll no.-045058

Mentor-Amarnath mitra

Introduction

In the realm of sports and international competitions, the Commonwealth Games holds a distinctive place, bringing together nations from across the globe in a celebration of athletic excellence and camaraderie. The 2022 Commonwealth Games, hosted in Birmingham (England) witnessed exceptional performances by athletes representing 72 countries. As with any global sporting event, the medal standings serve as a tangible representation of a country's athletic prowess. This analysis endeavors to delve into the 2022 Commonwealth Games' medal standings, specifically focusing on the distribution and impact of gold, silver, and bronze medals on the rankings of participating countries. By scrutinizing the medal count and its correlation with each country's rank, we aim to uncover insights into the factors that contributed to success in the games. By undertaking this analysis, we seek to shed light on the intriguing dynamics of medal distribution and rank allocation, ultimately offering a deeper understanding of the outcomes of the 2022 Commonwealth Games. This understanding can serve as a valuable resource for sports enthusiasts, policymakers, and stakeholders seeking to assess and appreciate the achievements of nations on the global stage of athletics. In the subsequent sections, we will delve into the methodology used for data collection and analysis, present the findings, and engage in a meaningful discussion to draw conclusions and insights from the data. Additionally, we will provide recommendations and implications drawn from the results of this analysis. Let us embark on a journey to unravel the correlation between medals and rank, revealing the captivating story of the 2022 Commonwealth Games.

Objectives:

1.Medal Distribution Analysis:

- To comprehensively examine the distribution of gold, silver, and bronze medals among participating countries in the 2022 Commonwealth Games.
- To identify which countries excelled in specific medal categories and ascertain if there were any dominant performances.

2.Correlation Investigation:

- To investigate the correlation between a country's rank in the medal standings and its respective medal count (gold, silver, and bronze).
- To determine whether there is a statistically significant relationship between medal counts and the overall rank

3.Trend and Pattern Identification:

- To identify and analyze any noteworthy trends, patterns, or anomalies that emerged during the 2022 Commonwealth Games.
- To uncover insights into countries that may have achieved higher ranks due to a particular strength in a specific medal category.

4.Implications and Insights:

- To draw conclusions and insights from the analysis, providing a deeper understanding of the factors influencing medal standings.
- To assess the impact of gold, silver, and bronze medals on a country's overall rank and what this signifies for athletic performance.

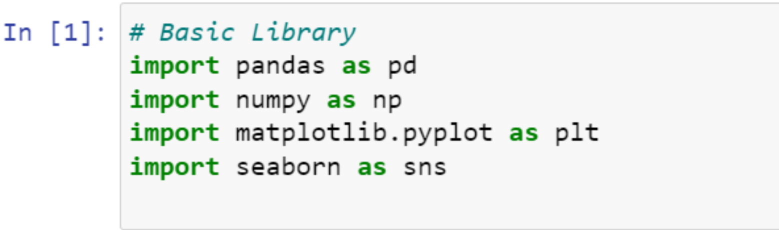
5.Recommendations and Applications:

- To offer recommendations or insights that may be applicable to sports enthusiasts, policymakers, and stakeholders in evaluating and appreciating the achievements of nations in international sporting events.
- To explore potential applications of the analysis findings, such as informing training programs, sports development policies, or future competition strategies.

Things To Cover

- Import Necessary Library
- Data Loading
- Data Preprocessing
- Exploratory Data Analysis
- Summary

Import Necessary Library



```
In [1]: # Basic Library
        import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        import seaborn as sns
```

1.pandas (imported as pd):

- Purpose: pandas is a powerful library for data manipulation and analysis. It provides data structures like DataFrames and Series, which allow you to easily load, clean, transform, and analyze your data.
- Functionality: You can use pandas to read data from various file formats (e.g., CSV, Excel), filter and select data, perform calculations, handle missing values, and much more.

2.numpy (imported as np):

- Purpose: numpy is a fundamental library for numerical operations in Python. It provides support for arrays and matrices, making it essential for mathematical and scientific computing.
- Functionality: You can use numpy for mathematical calculations, array manipulation, element-wise operations, and linear algebra operations, among other tasks.

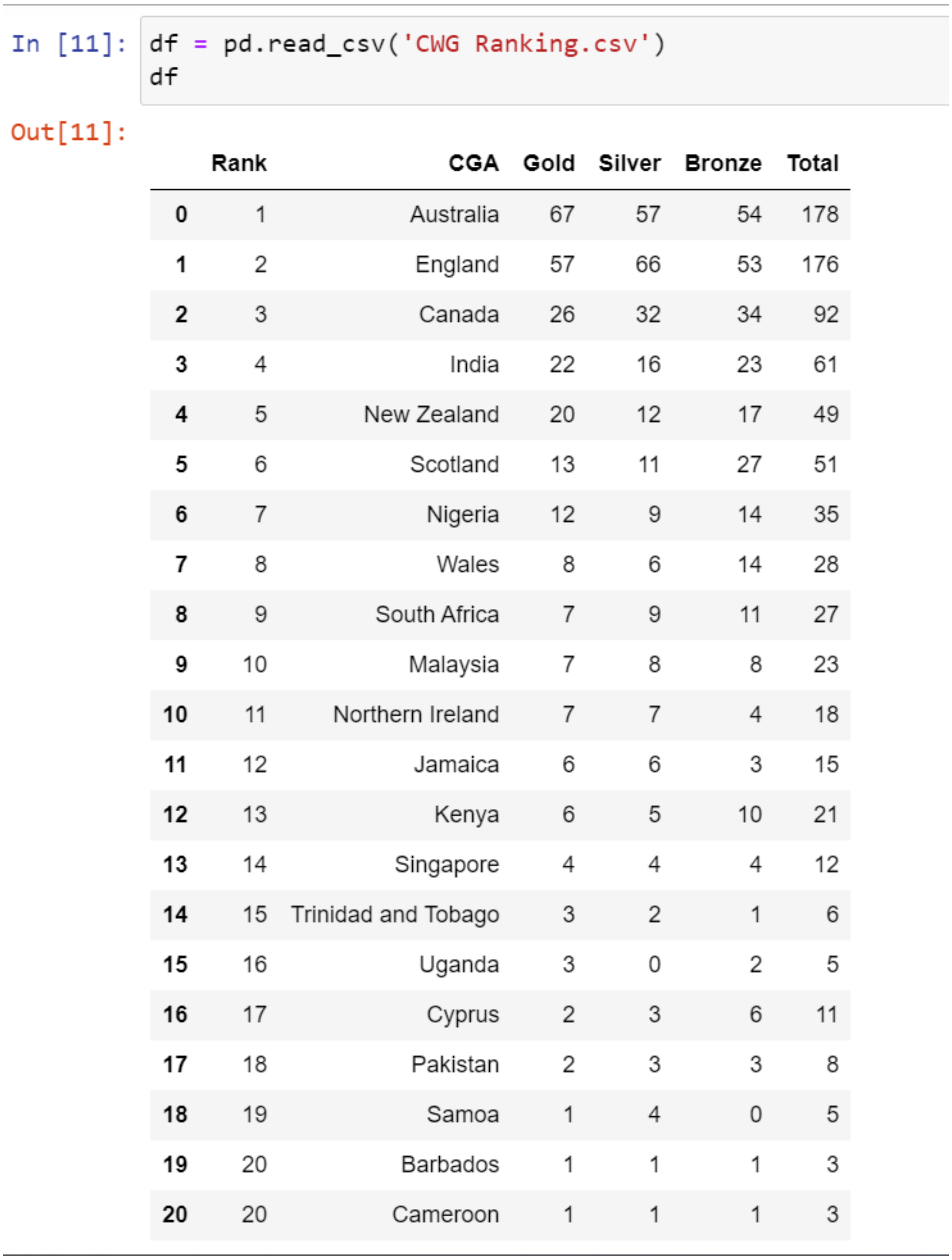
3.matplotlib.pyplot (imported as plt):

- Purpose: matplotlib is a widely used library for creating static, animated, and interactive visualizations in Python. pyplot is a module within matplotlib that provides a convenient interface for creating plots and charts.
- Functionality: With matplotlib.pyplot, you can create a wide range of visualizations, including line charts, scatter plots, bar charts, histograms, and more. It allows you to customize the appearance of your plots and add labels, titles, and legends.

4.seaborn (imported as sns):

- Purpose: seaborn is a data visualization library based on matplotlib. It provides a higher-level interface for creating aesthetically pleasing and informative statistical graphics.
- Functionality: seaborn simplifies the process of creating complex visualizations, including heatmaps, pair plots, violin plots, and regression plots. It also offers built-in themes and color palettes to enhance the visual appeal of your plots.

Data Loading



```
In [11]: df = pd.read_csv('CWG Ranking.csv')
         df
```

Out[11]:

	Rank	CGA	Gold	Silver	Bronze	Total
0	1	Australia	67	57	54	178
1	2	England	57	66	53	176
2	3	Canada	26	32	34	92
3	4	India	22	16	23	61
4	5	New Zealand	20	12	17	49
5	6	Scotland	13	11	27	51
6	7	Nigeria	12	9	14	35
7	8	Wales	8	6	14	28
8	9	South Africa	7	9	11	27
9	10	Malaysia	7	8	8	23
10	11	Northern Ireland	7	7	4	18
11	12	Jamaica	6	6	3	15
12	13	Kenya	6	5	10	21
13	14	Singapore	4	4	4	12
14	15	Trinidad and Tobago	3	2	1	6
15	16	Uganda	3	0	2	5
16	17	Cyprus	2	3	6	11
17	18	Pakistan	2	3	3	8
18	19	Samoa	1	4	0	5
19	20	Barbados	1	1	1	3
20	20	Cameroon	1	1	1	3

Steps to import csv file:

1.Importing the pandas Library:

- import pandas as pd: This line imports the pandas library and assigns it the alias pd. This is a common convention in Python to make it easier to reference the library throughout the code.

2.Loading the CSV File:

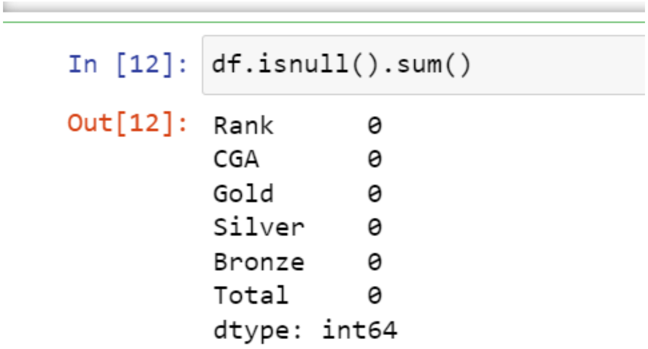
- df = pd.read_csv('your_file.csv'): This line of code reads the data from a CSV file and stores it in a pandas DataFrame named df. Here's what each part of this line does:
- pd.read_csv(): This is a pandas function specifically designed to read CSV files. It takes the file path of the CSV file as an argument.
- 'your_file.csv': Replace this with the actual file path to your CSV file. This should point to the location of the CSV file on your computer.
- df: This is the variable name assigned to the resulting DataFrame. You can choose any valid variable name you like.

3.Resulting DataFrame:

- After running the code, the data from the specified CSV file will be loaded into the df DataFrame. You can now work with this DataFrame to analyze and manipulate the data as needed.

Check Null Value

- **Check null value** that able to **impact the process** of analysis.

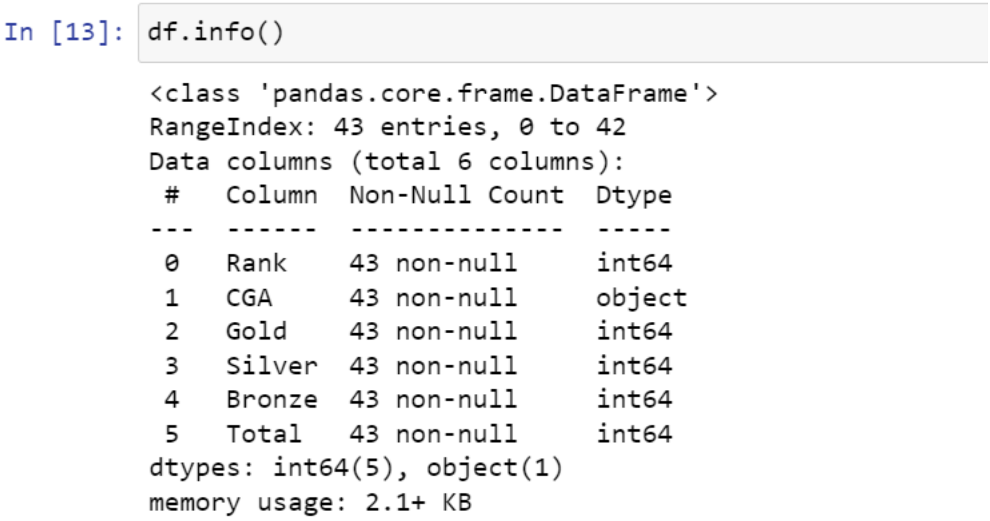


```
In [12]: df.isnull().sum()

Out[12]: Rank      0
         CGA       0
         Gold      0
         Silver    0
         Bronze    0
         Total     0
         dtype: int64
```

DATA PREPROCESSING

Check Columns Information

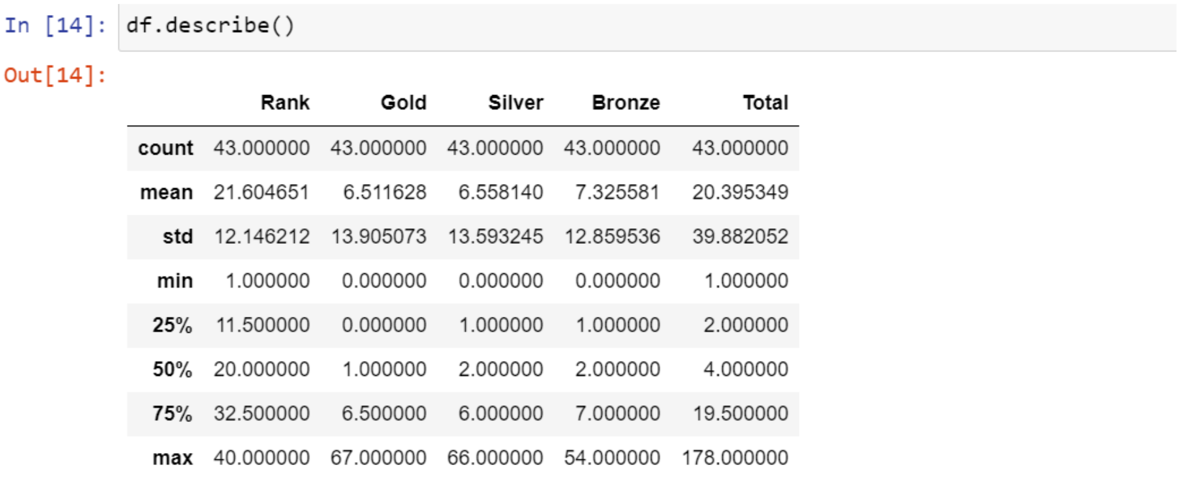
- To get general information of every columns in data frame.



```
In [13]: df.info()

         <class 'pandas.core.frame.DataFrame'>
         RangeIndex: 43 entries, 0 to 42
         Data columns (total 6 columns):
          #   Column  Non-Null Count  Dtype 
         ---  ------  --------------  ----- 
          0   Rank    43 non-null     int64 
          1   CGA     43 non-null     object
          2   Gold    43 non-null     int64 
          3   Silver  43 non-null     int64 
          4   Bronze  43 non-null     int64 
          5   Total   43 non-null     int64 
         dtypes: int64(5), object(1)
         memory usage: 2.1+ KB
```

EXPLORATORY DATA ANALYSIS

How is The Descriptive Analysis of The Data?



```
In [14]: df.describe()
```

```
Out[14]:
```

	Rank	Gold	Silver	Bronze	Total
count	43.000000	43.000000	43.000000	43.000000	43.000000
mean	21.604651	6.511628	6.558140	7.325581	20.395349
std	12.146212	13.905073	13.593245	12.859536	39.882052
min	1.000000	0.000000	0.000000	0.000000	1.000000
25%	11.500000	0.000000	1.000000	1.000000	2.000000
50%	20.000000	1.000000	2.000000	2.000000	4.000000
75%	32.500000	6.500000	6.000000	7.000000	19.500000
max	40.000000	67.000000	66.000000	54.000000	178.000000

Count: This row shows the number of non-null (non-missing) values for each numeric column. In your dataset, there are 43 countries represented.

Mean: The mean (average) value for each numeric column. For example, the mean rank is approximately 21.60, meaning that, on average, countries are ranked around the 21st position.

Std: The standard deviation measures the dispersion or spread of data around the mean. It indicates how much individual values deviate from the mean. For example, in the 'Gold' column, the standard deviation is approximately 13.91, suggesting that the number of gold medals varies significantly between countries.

Min: The minimum value observed in each column. For example, the minimum rank is 1, indicating that at least one country achieved the top rank.

25% (Q1): The 25th percentile, also known as the first quartile (Q1), is the value below which 25% of the data falls. For example, in the 'Gold' column, Q1 is 0, meaning that 25% of the countries have won no gold medals.

50% (Q2): The 50th percentile, also known as the median or second quartile (Q2), is the value that separates the higher half from the lower half of the data. For example, in the 'Silver' column, the median is 2, indicating that half of the countries won two or fewer silver medals.

75% (Q3): The 75th percentile, also known as the third quartile (Q3), is the value below which 75% of the data falls. For example, in the 'Bronze' column, Q3 is 7, meaning that 75% of the countries have won seven or fewer bronze medals.

Max: The maximum value observed in each column. For instance, the maximum number of gold medals won by a country is 67, and the maximum total medal count is 178.

What are 10 Countries that Have Highest Number of Total Gold Medal?

In [23]:

```
#What are 10 Countries that Have Highest Number of Total Gold Medal?
# Load the DataFrame from the CSV file
df = pd.read_csv('CWG Ranking.csv')

# Sort the DataFrame by 'Gold' column in descending order
df_sorted = df.sort_values(by='Gold', ascending=False)

# Select the top 10 rows with the highest number of gold medals
top_10_countries_gold = df_sorted.head(10)

# Display the top 10 countries with the highest number of gold medals
print(top_10_countries_gold[['CGA', 'Gold']])
```

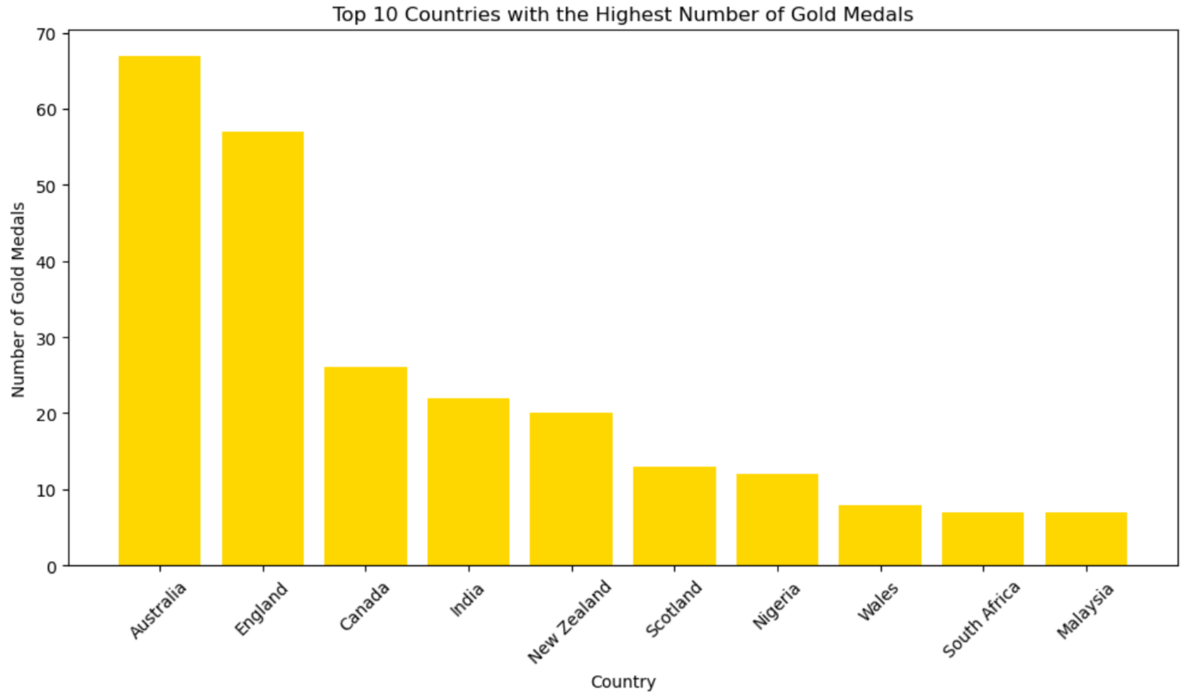
```
            CGA  Gold
0     Australia    67
1       England    57
2        Canada    26
3         India    22
4   New Zealand    20
5      Scotland    13
6       Nigeria    12
7         Wales     8
8  South Africa     7
9      Malaysia     7
```

In [7]:

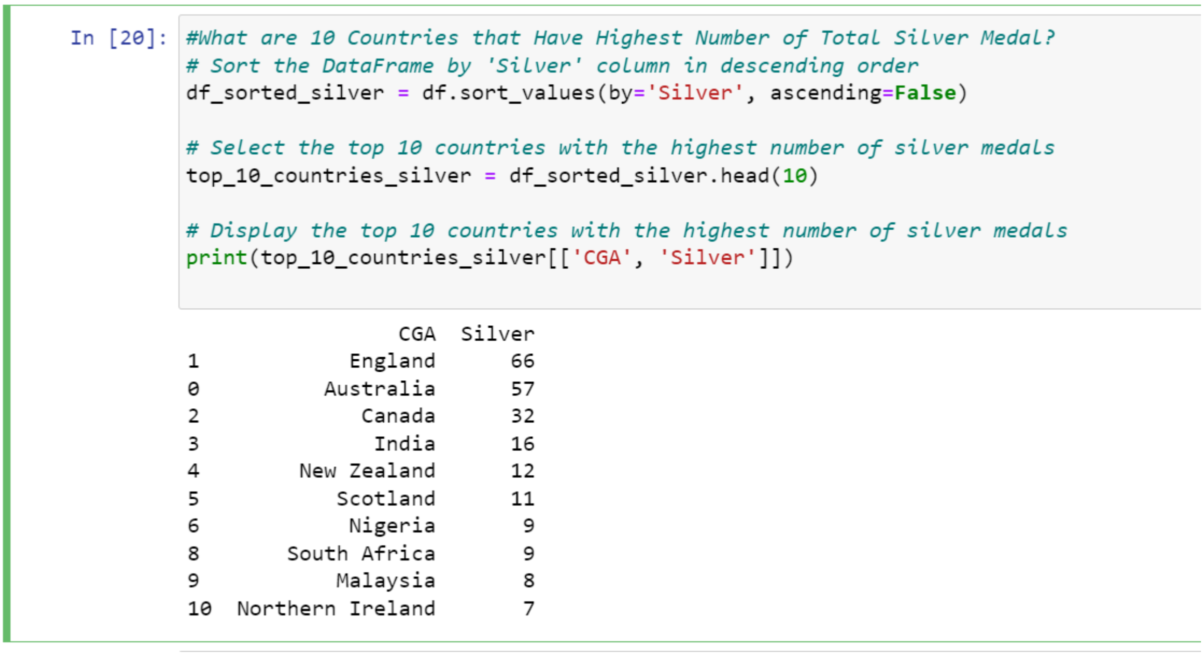
Bar chart

In [17]:

```
# Create a bar chart
plt.figure(figsize=(10, 6))
plt.bar(top_10_countries_gold['CGA'], top_10_countries_gold['Gold'], color='gold')
plt.xlabel('Country')
plt.ylabel('Number of Gold Medals')
plt.title('Top 10 Countries with the Highest Number of Gold Medals')

# Rotate the x-axis labels for better readability
plt.xticks(rotation=45)

# Display the chart
plt.tight_layout()
plt.show()
```



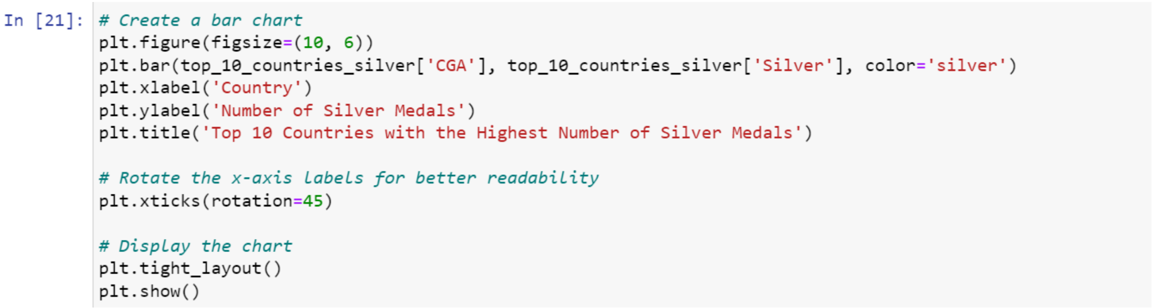
What are 10 Countries that Have Highest Number of Total Silver Medal?


In [20]:

```
#What are 10 Countries that Have Highest Number of Total Silver Medal?
# Sort the DataFrame by 'Silver' column in descending order
df_sorted_silver = df.sort_values(by='Silver', ascending=False)

# Select the top 10 countries with the highest number of silver medals
top_10_countries_silver = df_sorted_silver.head(10)

# Display the top 10 countries with the highest number of silver medals
print(top_10_countries_silver[['CGA', 'Silver']])
```

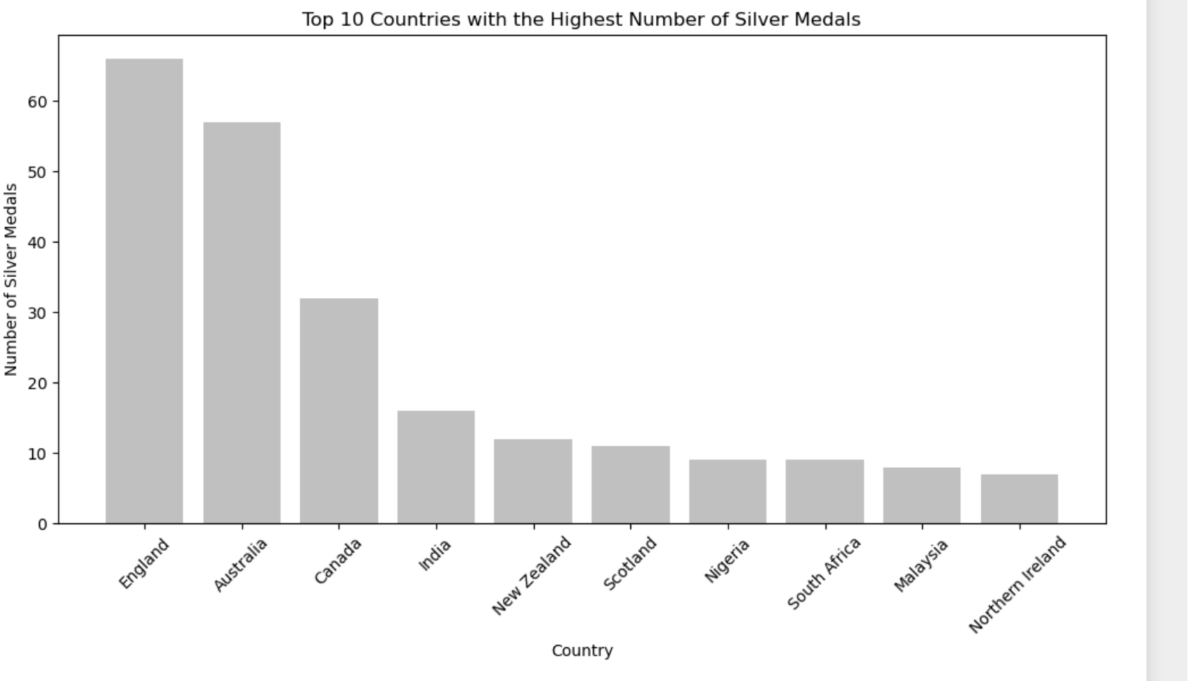
```
                 CGA  Silver
1            England      66
0          Australia      57
2             Canada      32
3              India      16
4        New Zealand      12
5           Scotland      11
6            Nigeria       9
8       South Africa       9
9           Malaysia       8
10  Northern Ireland       7
```

Bar chart

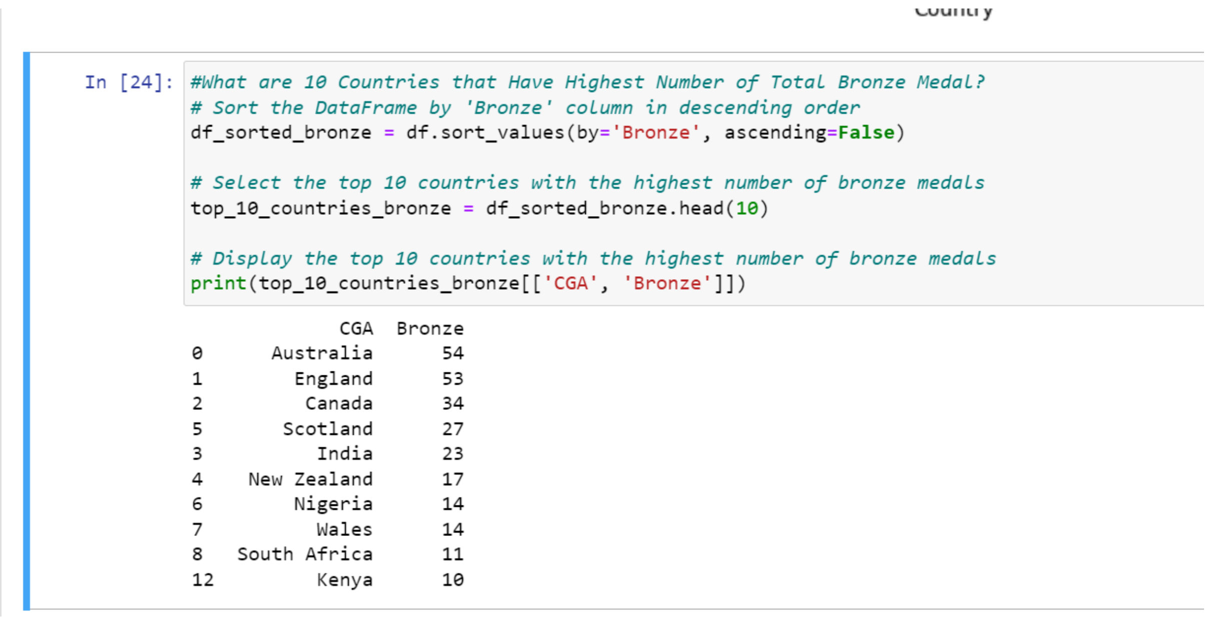


In [21]:

```
# Create a bar chart
plt.figure(figsize=(10, 6))
plt.bar(top_10_countries_silver['CGA'], top_10_countries_silver['Silver'], color='silver')
plt.xlabel('Country')
plt.ylabel('Number of Silver Medals')
plt.title('Top 10 Countries with the Highest Number of Silver Medals')

# Rotate the x-axis labels for better readability
plt.xticks(rotation=45)

# Display the chart
plt.tight_layout()
plt.show()
```



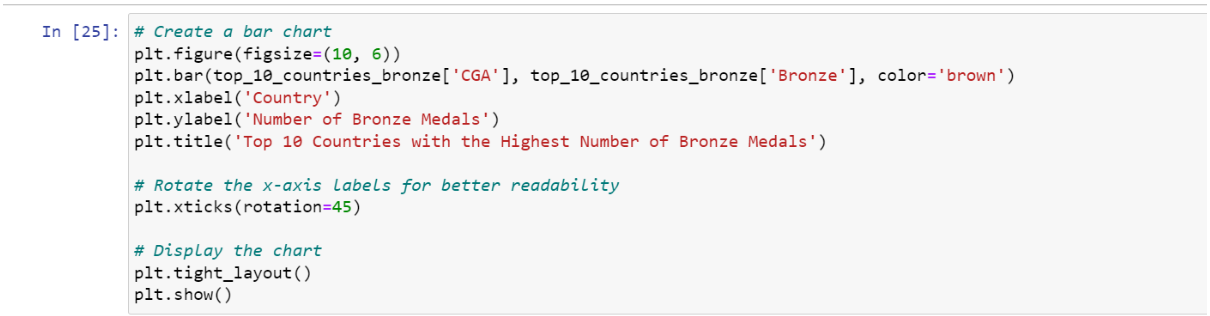
What are 10 Countries that Have Highest Number of Total Bronze Medal?



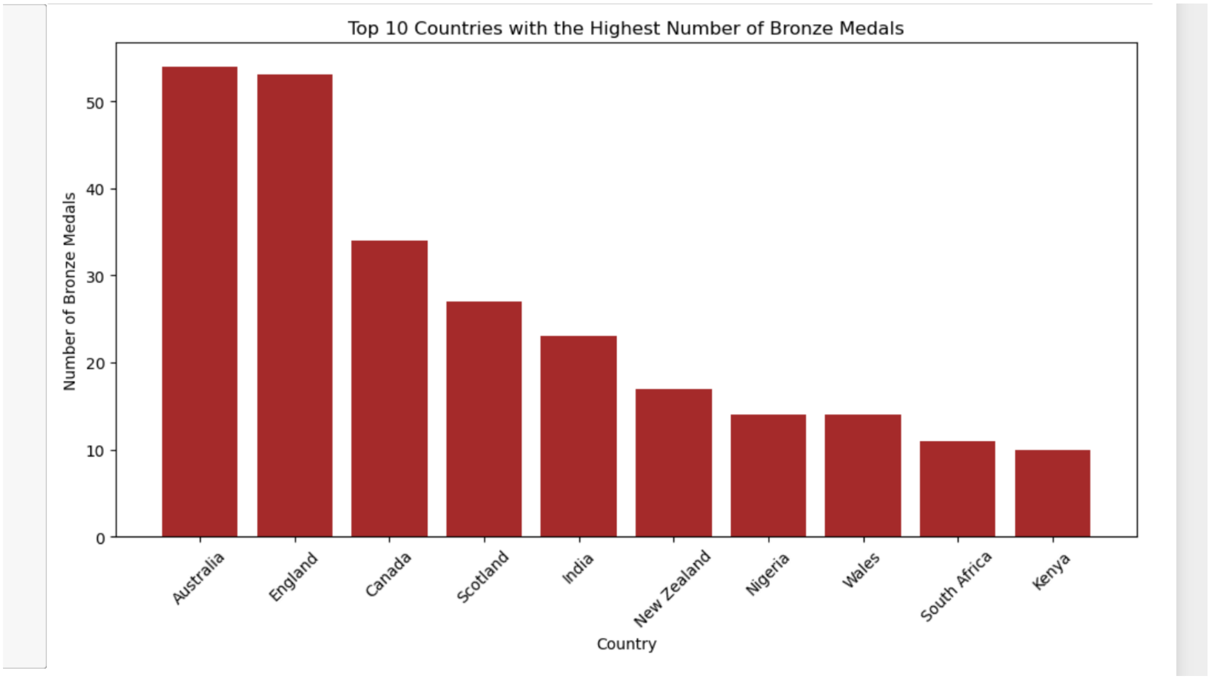
In [24]:
```python
#What are 10 Countries that Have Highest Number of Total Bronze Medal?
# Sort the DataFrame by 'Bronze' column in descending order
df_sorted_bronze = df.sort_values(by='Bronze', ascending=False)

# Select the top 10 countries with the highest number of bronze medals
top_10_countries_bronze = df_sorted_bronze.head(10)

# Display the top 10 countries with the highest number of bronze medals
print(top_10_countries_bronze[['CGA', 'Bronze']])
```

```
             CGA  Bronze
0      Australia      54
1        England      53
2         Canada      34
5       Scotland      27
3          India      23
4    New Zealand      17
6        Nigeria      14
7          Wales      14
8   South Africa      11
12         Kenya      10
```

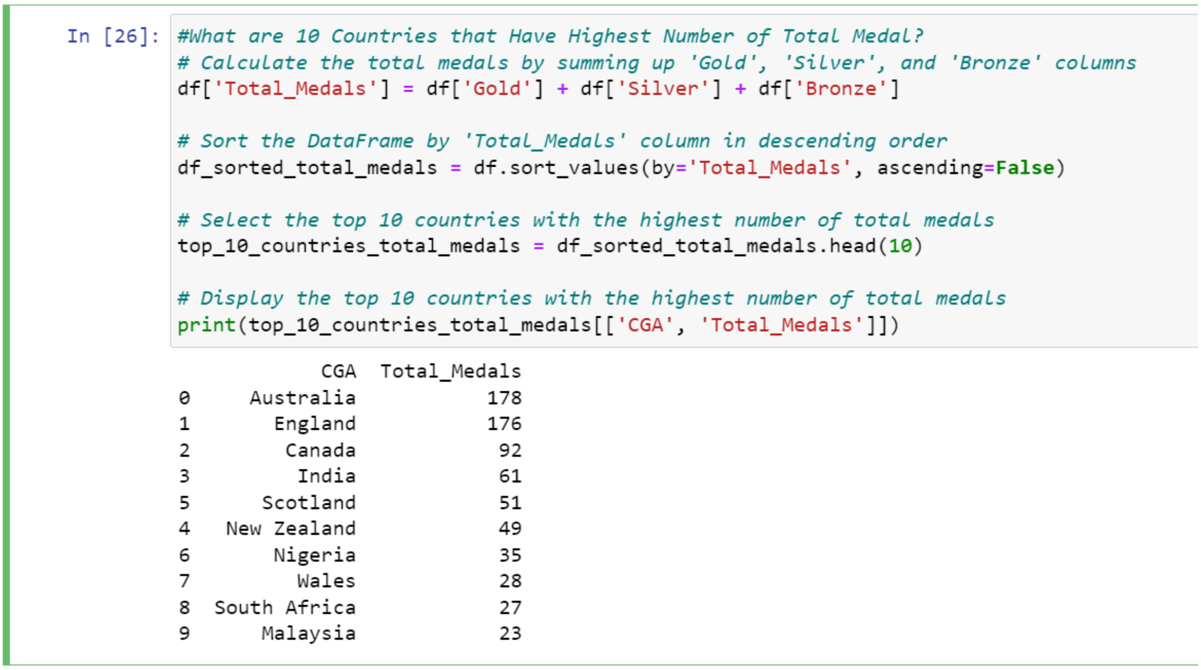
Bar chart



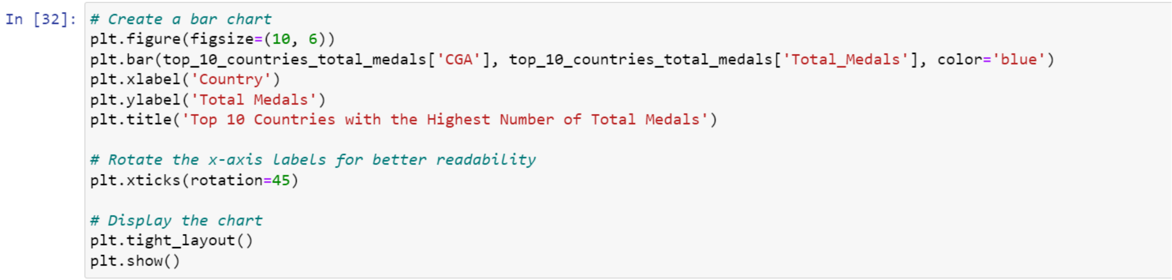
In [25]:
```python
# Create a bar chart
plt.figure(figsize=(10, 6))
plt.bar(top_10_countries_bronze['CGA'], top_10_countries_bronze['Bronze'], color='brown')
plt.xlabel('Country')
plt.ylabel('Number of Bronze Medals')
plt.title('Top 10 Countries with the Highest Number of Bronze Medals')

# Rotate the x-axis labels for better readability
plt.xticks(rotation=45)

# Display the chart
plt.tight_layout()
plt.show()
```



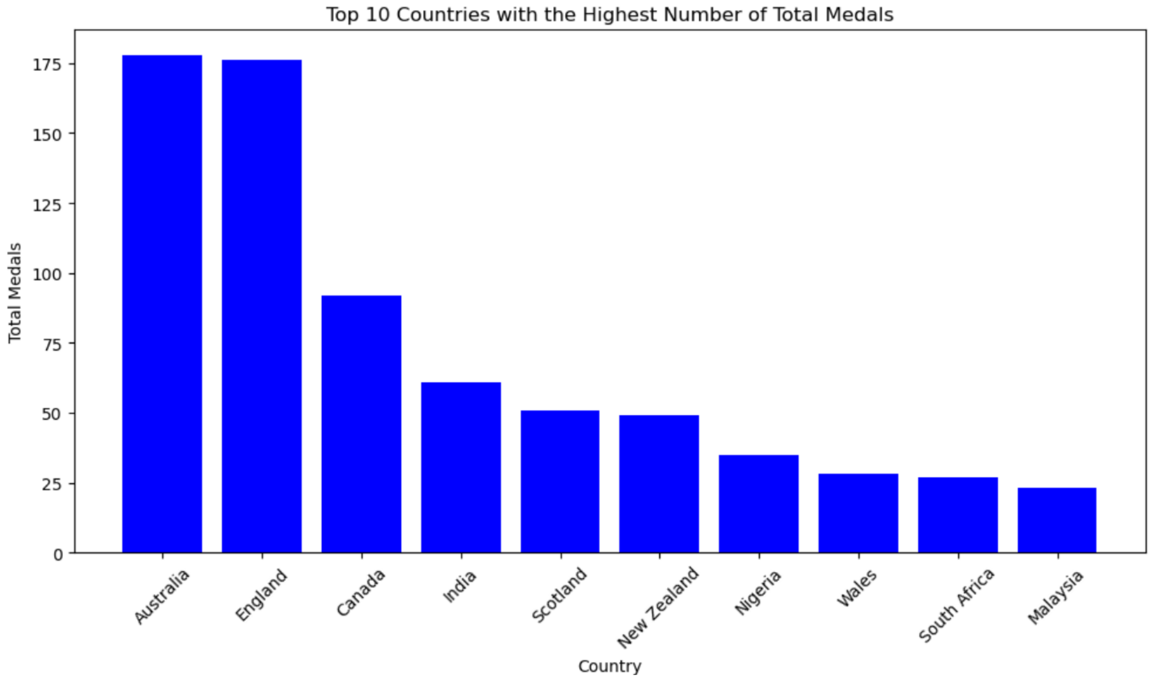
What are 10 Countries that Have Highest Number of Total Medal?



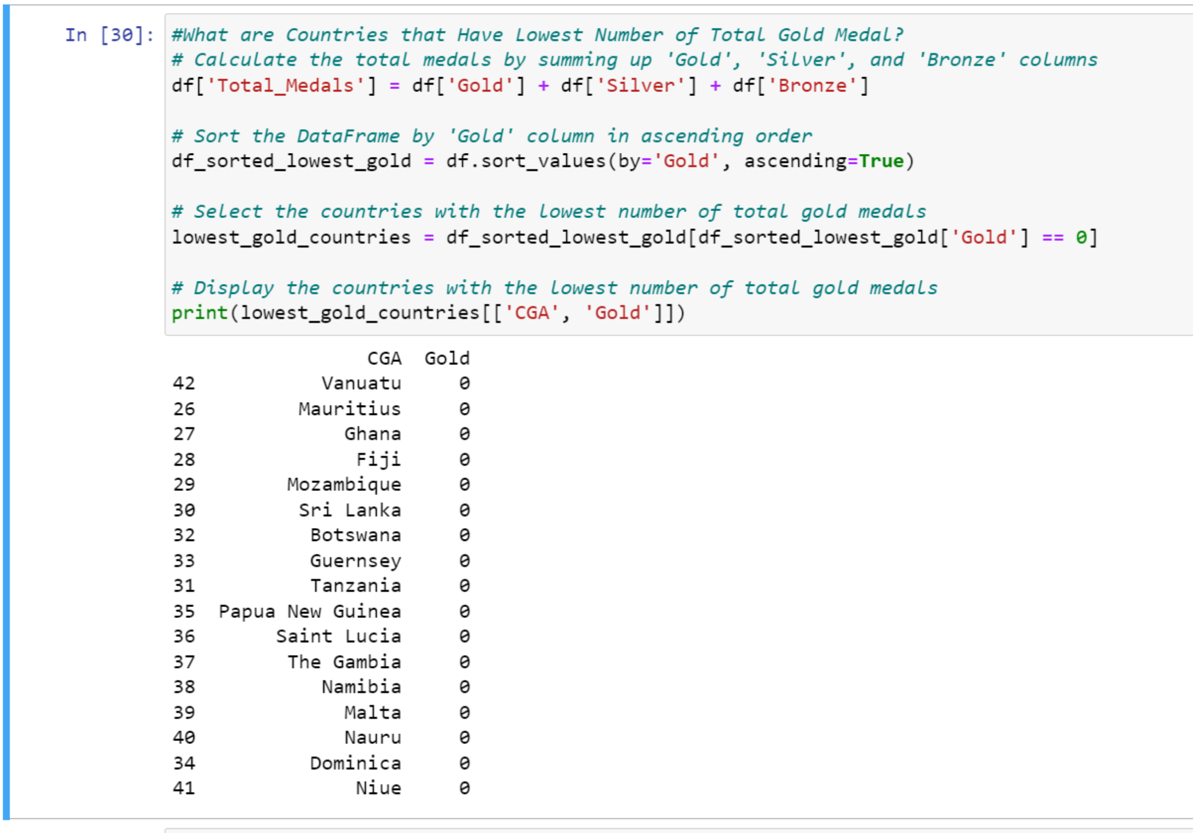
```
In [26]: #What are 10 Countries that Have Highest Number of Total Medal?
         # Calculate the total medals by summing up 'Gold', 'Silver', and 'Bronze' columns
         df['Total_Medals'] = df['Gold'] + df['Silver'] + df['Bronze']

         # Sort the DataFrame by 'Total_Medals' column in descending order
         df_sorted_total_medals = df.sort_values(by='Total_Medals', ascending=False)

         # Select the top 10 countries with the highest number of total medals
         top_10_countries_total_medals = df_sorted_total_medals.head(10)

         # Display the top 10 countries with the highest number of total medals
         print(top_10_countries_total_medals[['CGA', 'Total_Medals']])

                     CGA  Total_Medals
         0     Australia           178
         1       England           176
         2        Canada            92
         3         India            61
         5      Scotland            51
         4   New Zealand            49
         6       Nigeria            35
         7         Wales            28
         8  South Africa            27
         9      Malaysia            23
```



```
In [32]: # Create a bar chart
         plt.figure(figsize=(10, 6))
         plt.bar(top_10_countries_total_medals['CGA'], top_10_countries_total_medals['Total_Medals'], color='blue')
         plt.xlabel('Country')
         plt.ylabel('Total Medals')
         plt.title('Top 10 Countries with the Highest Number of Total Medals')

         # Rotate the x-axis labels for better readability
         plt.xticks(rotation=45)

         # Display the chart
         plt.tight_layout()
         plt.show()
```



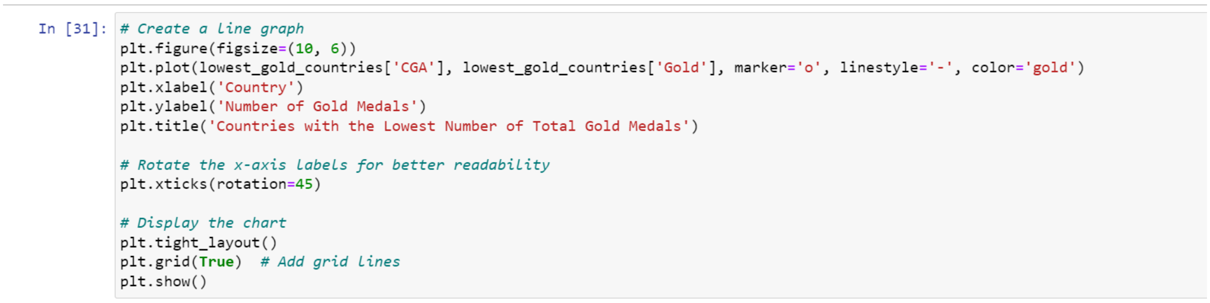
What are Countries that Have Lowest Number of Total Gold Medal?



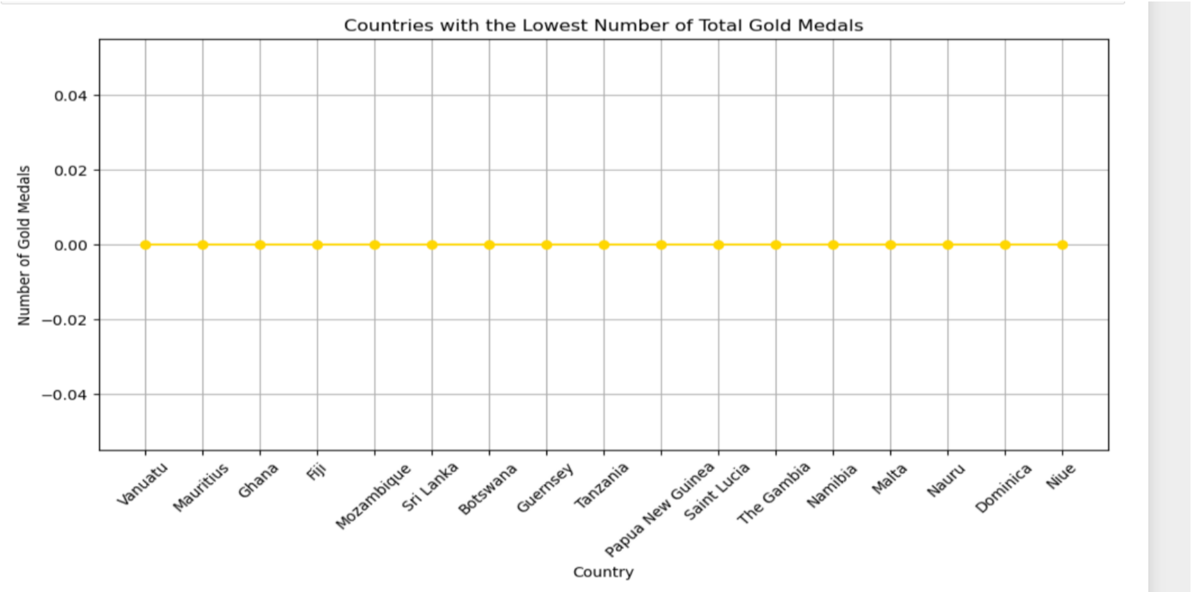
```
In [30]: #What are Countries that Have Lowest Number of Total Gold Medal?
         # Calculate the total medals by summing up 'Gold', 'Silver', and 'Bronze' columns
         df['Total_Medals'] = df['Gold'] + df['Silver'] + df['Bronze']

         # Sort the DataFrame by 'Gold' column in ascending order
         df_sorted_lowest_gold = df.sort_values(by='Gold', ascending=True)

         # Select the countries with the lowest number of total gold medals
         lowest_gold_countries = df_sorted_lowest_gold[df_sorted_lowest_gold['Gold'] == 0]

         # Display the countries with the lowest number of total gold medals
         print(lowest_gold_countries[['CGA', 'Gold']])
```

```
                 CGA  Gold
42           Vanuatu     0
26         Mauritius     0
27             Ghana     0
28              Fiji     0
29        Mozambique     0
30         Sri Lanka     0
32          Botswana     0
33          Guernsey     0
31          Tanzania     0
35  Papua New Guinea     0
36       Saint Lucia     0
37        The Gambia     0
38           Namibia     0
39             Malta     0
40             Nauru     0
34          Dominica     0
41              Niue     0
```

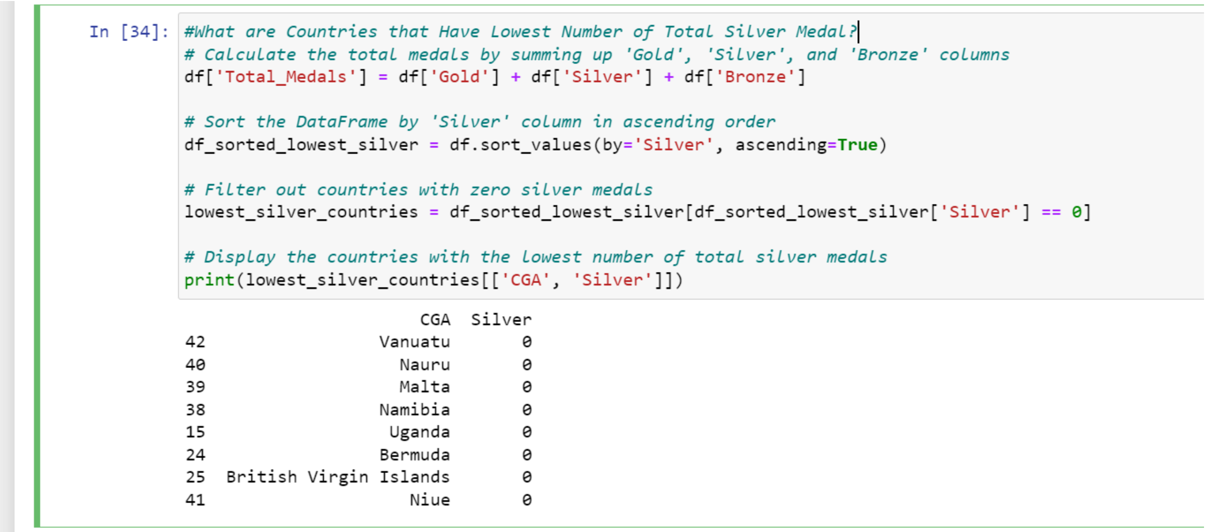
Line Graph



```
In [31]: # Create a line graph
         plt.figure(figsize=(10, 6))
         plt.plot(lowest_gold_countries['CGA'], lowest_gold_countries['Gold'], marker='o', linestyle='-', color='gold')
         plt.xlabel('Country')
         plt.ylabel('Number of Gold Medals')
         plt.title('Countries with the Lowest Number of Total Gold Medals')

         # Rotate the x-axis labels for better readability
         plt.xticks(rotation=45)

         # Display the chart
         plt.tight_layout()
         plt.grid(True)  # Add grid lines
         plt.show()
```



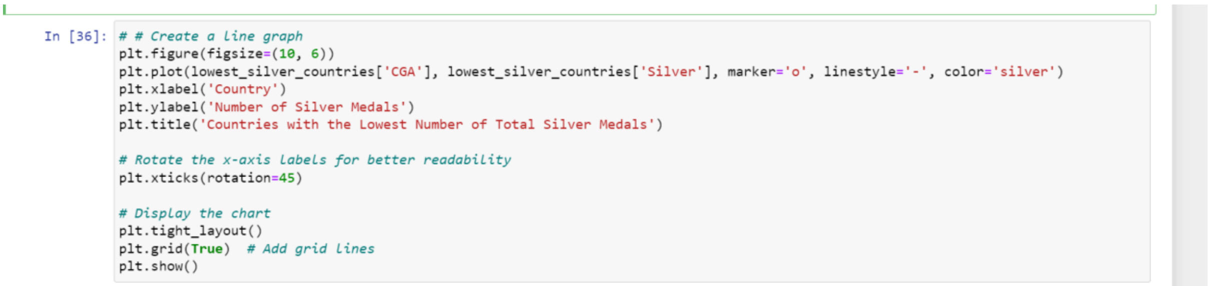
What are Countries that Have Lowest Number of Total Silver Medal?



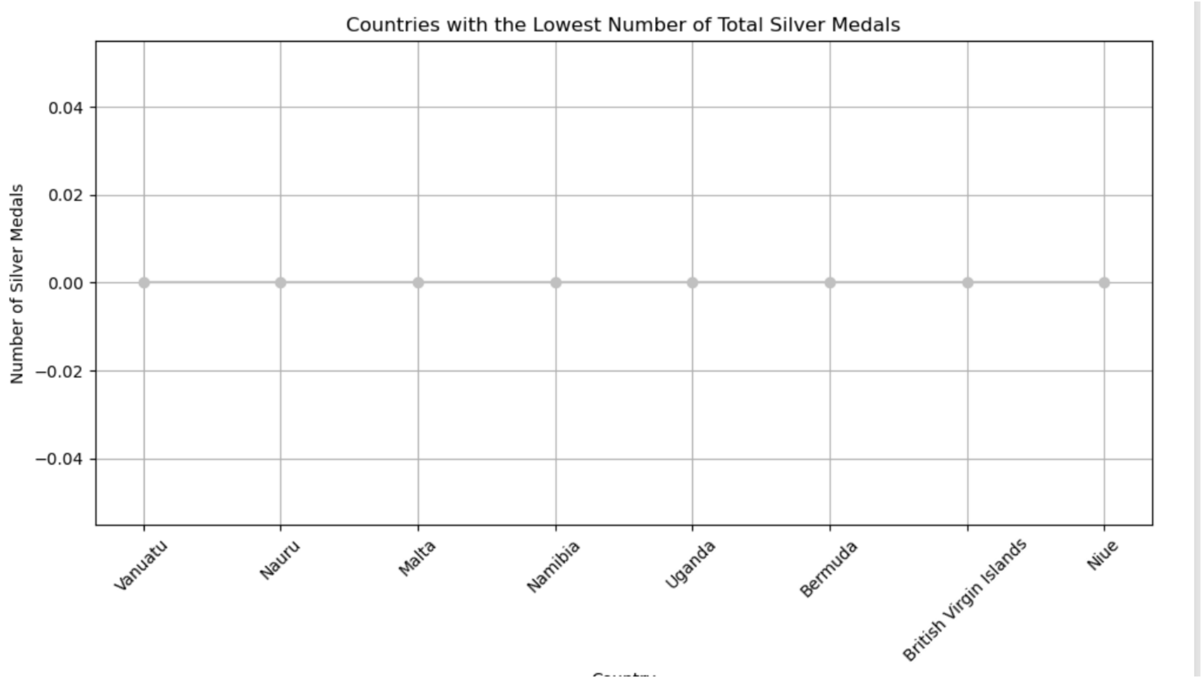
```
In [34]: #What are Countries that Have Lowest Number of Total Silver Medal?
         # Calculate the total medals by summing up 'Gold', 'Silver', and 'Bronze' columns
         df['Total_Medals'] = df['Gold'] + df['Silver'] + df['Bronze']

         # Sort the DataFrame by 'Silver' column in ascending order
         df_sorted_lowest_silver = df.sort_values(by='Silver', ascending=True)

         # Filter out countries with zero silver medals
         lowest_silver_countries = df_sorted_lowest_silver[df_sorted_lowest_silver['Silver'] == 0]

         # Display the countries with the lowest number of total silver medals
         print(lowest_silver_countries[['CGA', 'Silver']])
```

```
                       CGA  Silver
42                 Vanuatu       0
40                   Nauru       0
39                   Malta       0
38                 Namibia       0
15                  Uganda       0
24                 Bermuda       0
25  British Virgin Islands       0
41                    Niue       0
```

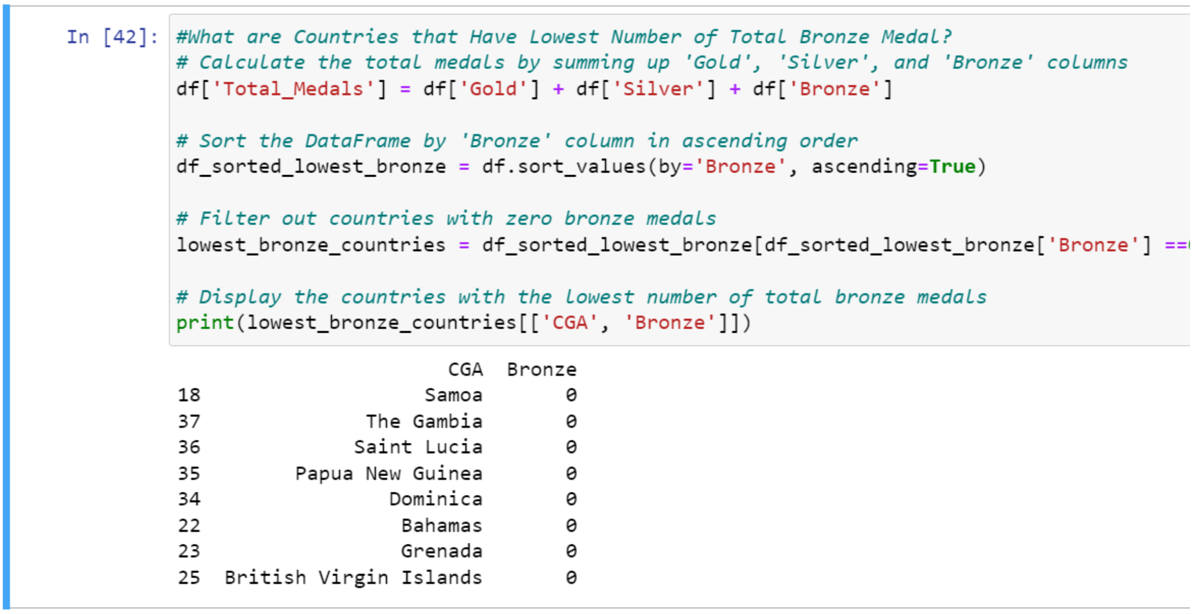
Line Graph



```
In [36]: # # Create a line graph
         plt.figure(figsize=(10, 6))
         plt.plot(lowest_silver_countries['CGA'], lowest_silver_countries['Silver'], marker='o', linestyle='-', color='silver')
         plt.xlabel('Country')
         plt.ylabel('Number of Silver Medals')
         plt.title('Countries with the Lowest Number of Total Silver Medals')

         # Rotate the x-axis labels for better readability
         plt.xticks(rotation=45)

         # Display the chart
         plt.tight_layout()
         plt.grid(True)  # Add grid lines
         plt.show()
```



What are Countries that Have Lowest Number of Total Bronze Medal?

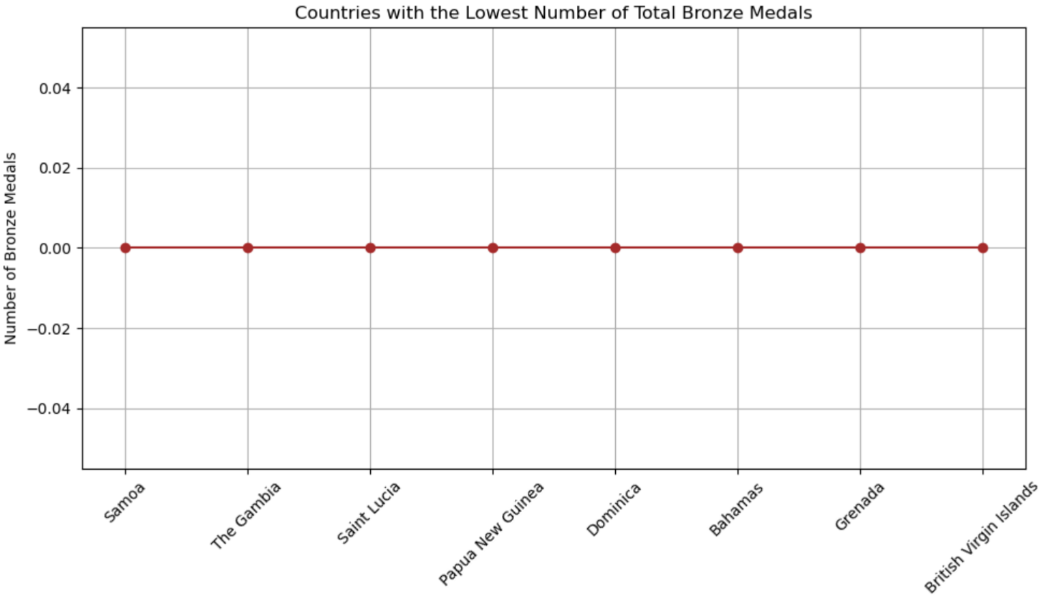


```
In [42]: #What are Countries that Have Lowest Number of Total Bronze Medal?
         # Calculate the total medals by summing up 'Gold', 'Silver', and 'Bronze' columns
         df['Total_Medals'] = df['Gold'] + df['Silver'] + df['Bronze']

         # Sort the DataFrame by 'Bronze' column in ascending order
         df_sorted_lowest_bronze = df.sort_values(by='Bronze', ascending=True)

         # Filter out countries with zero bronze medals
         lowest_bronze_countries = df_sorted_lowest_bronze[df_sorted_lowest_bronze['Bronze'] ==

         # Display the countries with the lowest number of total bronze medals
         print(lowest_bronze_countries[['CGA', 'Bronze']])
```

```
                       CGA  Bronze
18                   Samoa       0
37              The Gambia       0
36             Saint Lucia       0
35        Papua New Guinea       0
34                Dominica       0
22                 Bahamas       0
23                 Grenada       0
25  British Virgin Islands       0
```

Line Graph

In [43]:
```
# Create a line graph
plt.figure(figsize=(10, 6))
plt.plot(lowest_bronze_countries['CGA'], lowest_bronze_countries['Bronze'], marker='o', linestyle='-', color='brown')
plt.xlabel('Country')
plt.ylabel('Number of Bronze Medals')
plt.title('Countries with the Lowest Number of Total Bronze Medals')

# Rotate the x-axis labels for better readability
plt.xticks(rotation=45)

# Display the chart
plt.tight_layout()
plt.grid(True)  # Add grid lines
plt.show()
```



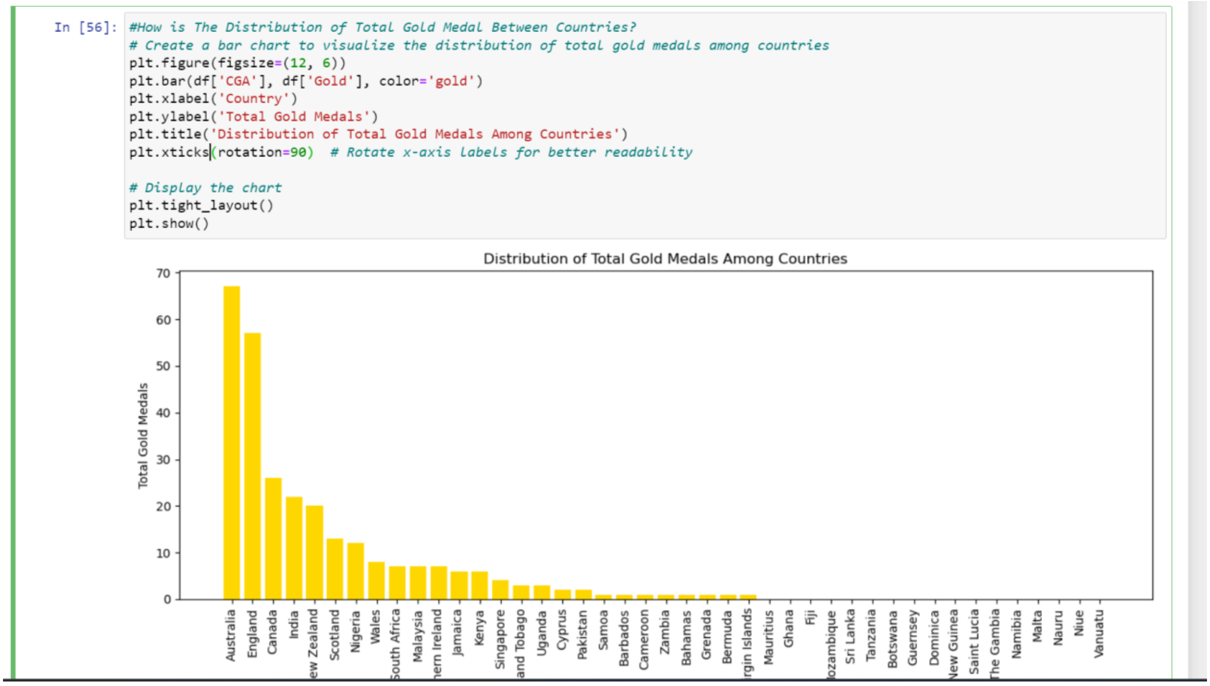
What are Countries that Have Lowest Number of Total Medal?

In [51]:
```
#What are Countries that Have Lowest Number of Total Medal?
# Calculate the total medals by summing up 'Gold', 'Silver', and 'Bronze' columns
df['Total_Medals'] = df['Gold'] + df['Silver'] + df['Bronze']

# Sort the DataFrame by 'Total_Medals' column in ascending order
df_sorted_lowest_total_medals = df.sort_values(by='Total_Medals', ascending=True)

# Filter out countries with zero total medals
lowest_total_medals_countries =df_sorted_lowest_total_medals[df_sorted_lowest_total_medals['Total_Medals'] ==1]

# Display the countries with the lowest number of total medals
print(lowest_total_medals_countries[['CGA', 'Total_Medals']])
```

```
                       CGA  Total_Medals
42                 Vanuatu             1
40                   Nauru             1
39                   Malta             1
37              The Gambia             1
36             Saint Lucia             1
35        Papua New Guinea             1
34                Dominica             1
41                    Niue             1
25  British Virgin Islands             1
```

How is The Distribution of Total Gold Medal Between Countries?

This figure is sorted by their ranking on standings. Based on figure "The Distribution of Total Gold Medal Between Countries", this figure clearly shows that higher the total medal**,** the higher the country rank**.**

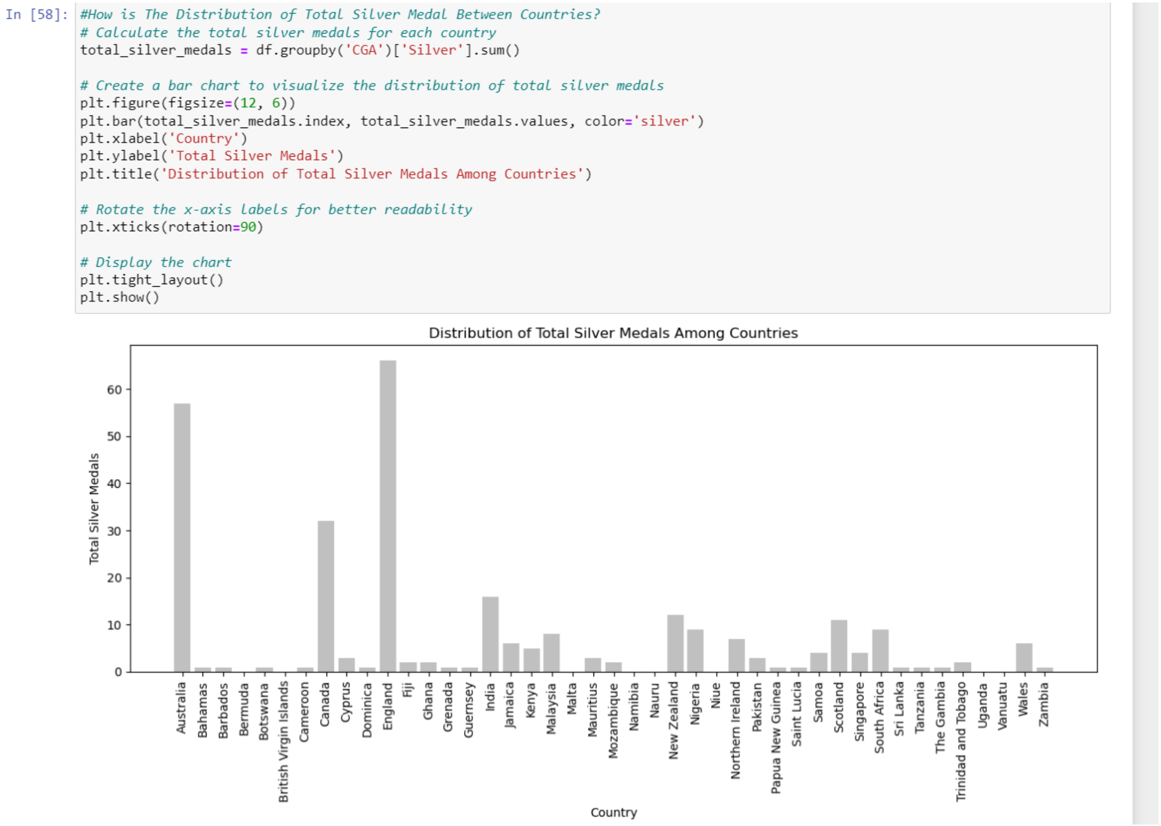


In [56]:

```
#How is The Distribution of Total Gold Medal Between Countries?
# Create a bar chart to visualize the distribution of total gold medals among countries
plt.figure(figsize=(12, 6))
plt.bar(df['CGA'], df['Gold'], color='gold')
plt.xlabel('Country')
plt.ylabel('Total Gold Medals')
plt.title('Distribution of Total Gold Medals Among Countries')
plt.xticks(rotation=90)  # Rotate x-axis labels for better readability

# Display the chart
plt.tight_layout()
plt.show()
```

How is The Distribution of Total Silver Medal Between Countries?

This figure is sorted by their ranking on standings. Based on figure "The Distribution of Total Silver Medal Between Countries", even though this figure not clearly disrtibuted decreasing figure, this also shows that higher the total medal, the higher the country rank.



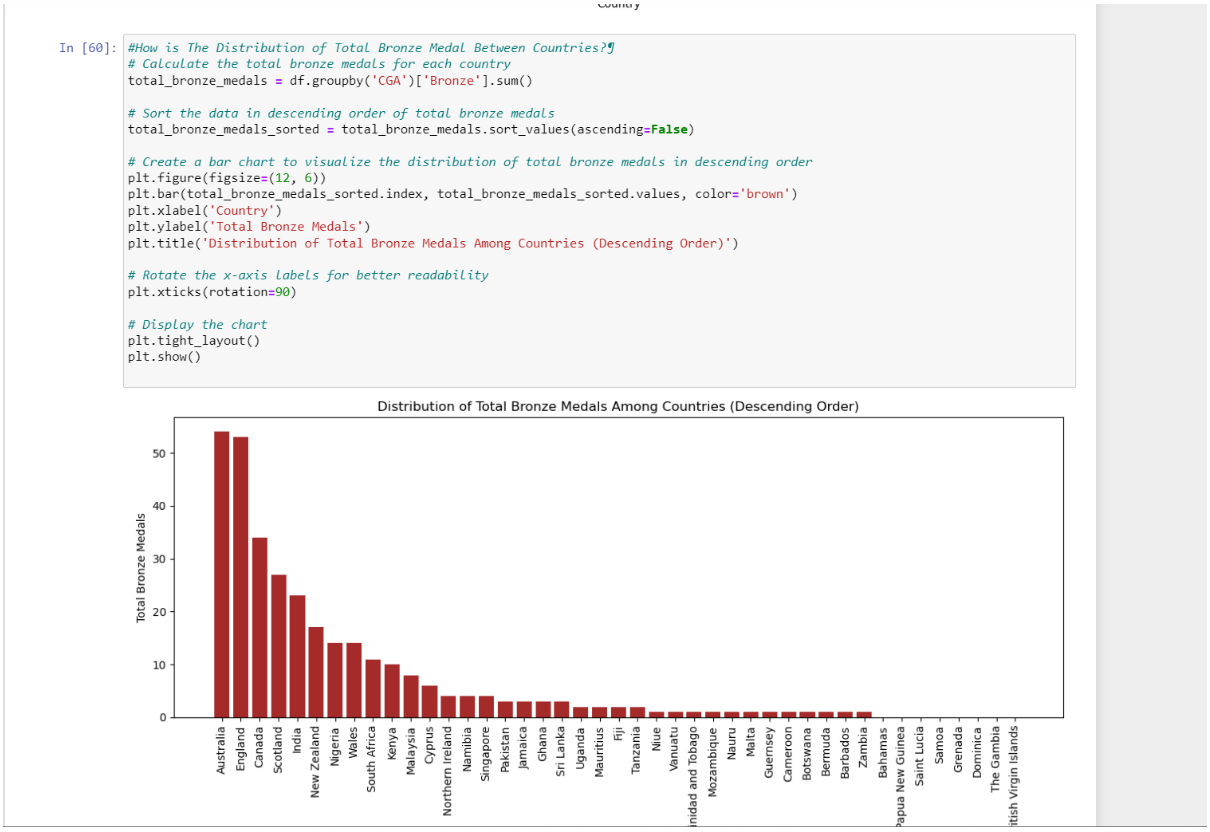
In [58]:

```
#How is The Distribution of Total Silver Medal Between Countries?
# Calculate the total silver medals for each country
total_silver_medals = df.groupby('CGA')['Silver'].sum()

# Create a bar chart to visualize the distribution of total silver medals
plt.figure(figsize=(12, 6))
plt.bar(total_silver_medals.index, total_silver_medals.values, color='silver')
plt.xlabel('Country')
plt.ylabel('Total Silver Medals')
plt.title('Distribution of Total Silver Medals Among Countries')

# Rotate the x-axis labels for better readability
plt.xticks(rotation=90)

# Display the chart
plt.tight_layout()
plt.show()
```

]

How is The Distribution of Total Bronze Medal Between Countries?

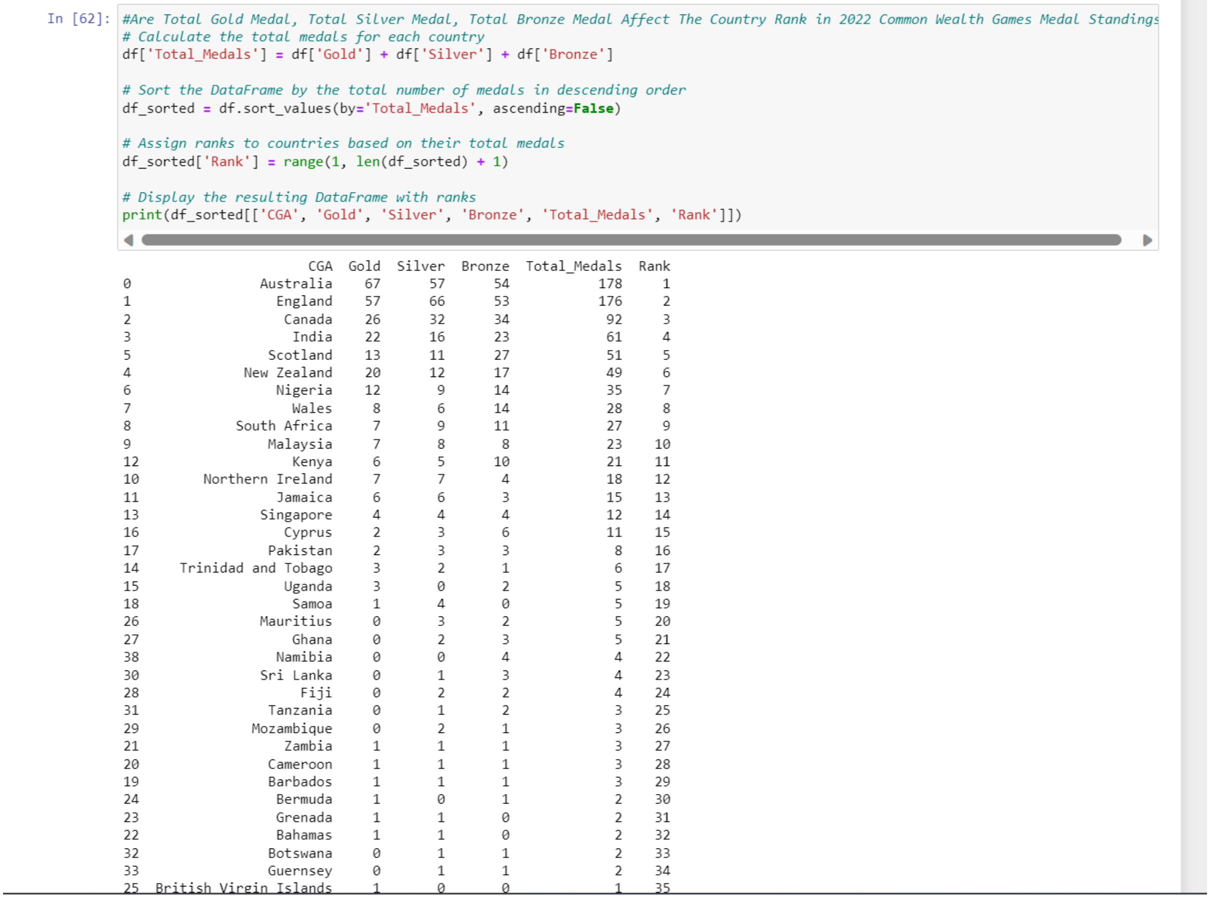
This figure is sorted by their ranking on standings. Based on figure "The Distribution of Total Bronze Medal Between Countries", eventhough this figure not clearly disrtibuted decreasing figure**,** this also shows that higher the total medal**,** the higher the country rank**.**



In [60]:

```
#How is The Distribution of Total Bronze Medal Between Countries?¶
# Calculate the total bronze medals for each country
total_bronze_medals = df.groupby('CGA')['Bronze'].sum()

# Sort the data in descending order of total bronze medals
total_bronze_medals_sorted = total_bronze_medals.sort_values(ascending=False)

# Create a bar chart to visualize the distribution of total bronze medals in descending order
plt.figure(figsize=(12, 6))
plt.bar(total_bronze_medals_sorted.index, total_bronze_medals_sorted.values, color='brown')
plt.xlabel('Country')
plt.ylabel('Total Bronze Medals')
plt.title('Distribution of Total Bronze Medals Among Countries (Descending Order)')

# Rotate the x-axis labels for better readability
plt.xticks(rotation=90)

# Display the chart
plt.tight_layout()
plt.show()
```

Are Total Gold Medal, Total Silver Medal, Total Bronze Medal Affect The Country Rank in 2022 Common Wealth Games Medal Standings?



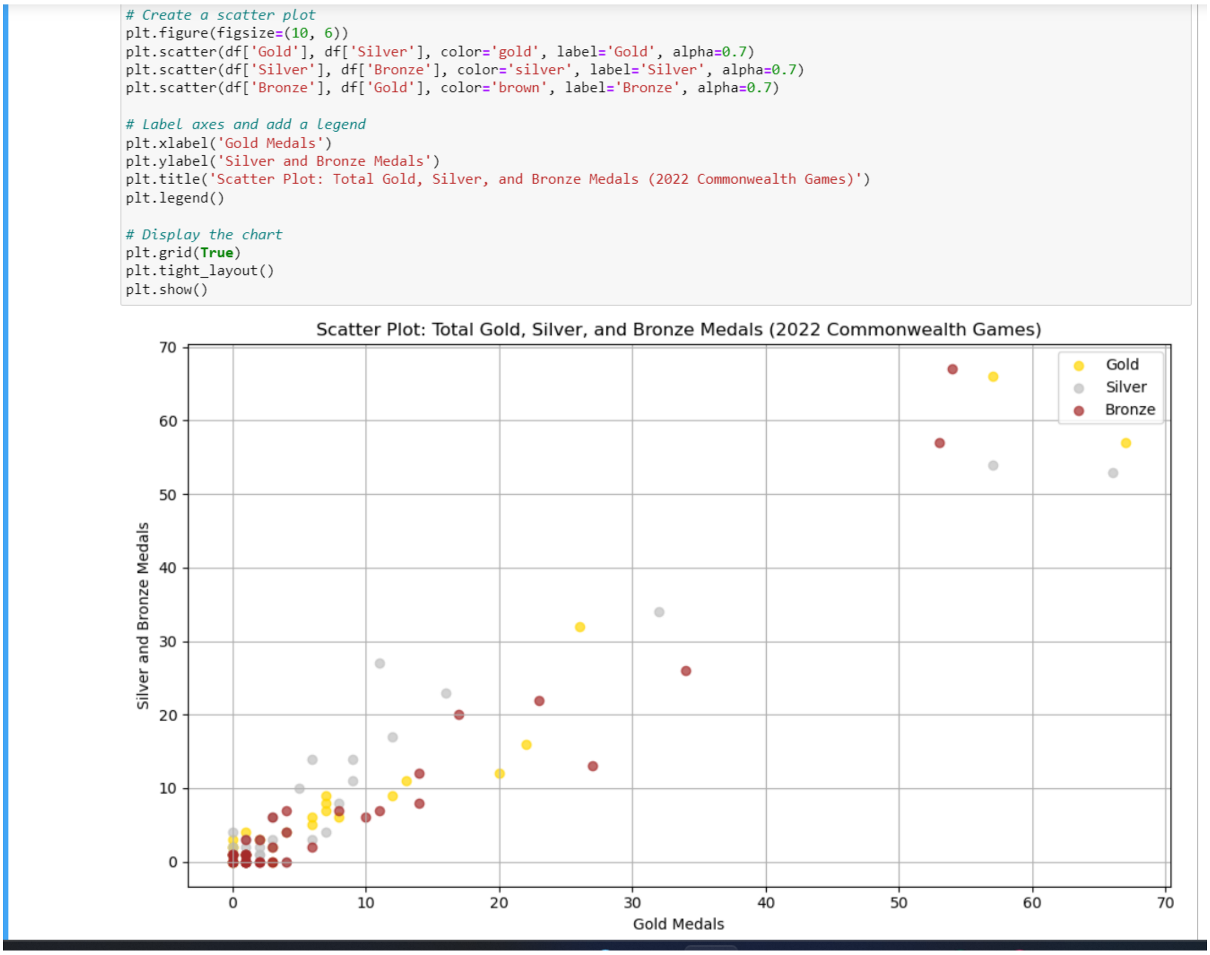
In [62]:

```
#Are Total Gold Medal, Total Silver Medal, Total Bronze Medal Affect The Country Rank in 2022 Common Wealth Games Medal Standings
# Calculate the total medals for each country
df['Total_Medals'] = df['Gold'] + df['Silver'] + df['Bronze']

# Sort the DataFrame by the total number of medals in descending order
df_sorted = df.sort_values(by='Total_Medals', ascending=False)

# Assign ranks to countries based on their total medals
df_sorted['Rank'] = range(1, len(df_sorted) + 1)

# Display the resulting DataFrame with ranks
print(df_sorted[['CGA', 'Gold', 'Silver', 'Bronze', 'Total_Medals', 'Rank']])
```

```
                       CGA  Gold  Silver  Bronze  Total_Medals  Rank
0                Australia    67      57      54           178     1
1                  England    57      66      53           176     2
2                   Canada    26      32      34            92     3
3                    India    22      16      23            61     4
5                 Scotland    13      11      27            51     5
4              New Zealand    20      12      17            49     6
6                  Nigeria    12       9      14            35     7
7                    Wales     8       6      14            28     8
8             South Africa     7       9      11            27     9
9                 Malaysia     7       8       8            23    10
12                   Kenya     6       5      10            21    11
10        Northern Ireland     7       7       4            18    12
11                 Jamaica     6       6       3            15    13
13               Singapore     4       4       4            12    14
16                  Cyprus     2       3       6            11    15
17                Pakistan     2       3       3             8    16
14     Trinidad and Tobago     3       2       1             6    17
15                  Uganda     3       0       2             5    18
18                   Samoa     1       4       0             5    19
26               Mauritius     0       3       2             5    20
27                   Ghana     0       2       3             5    21
38                 Namibia     0       0       4             4    22
30               Sri Lanka     0       1       3             4    23
28                    Fiji     0       2       2             4    24
31                Tanzania     0       1       2             3    25
29              Mozambique     0       2       1             3    26
21                  Zambia     1       1       1             3    27
20                Cameroon     1       1       1             3    28
19                Barbados     1       1       1             3    29
24                 Bermuda     1       0       1             2    30
23                 Grenada     1       1       0             2    31
22                 Bahamas     1       1       0             2    32
32                Botswana     0       1       1             2    33
33                Guernsey     0       1       1             2    34
25  British Virgin Islands     1       0       0             1    35
```

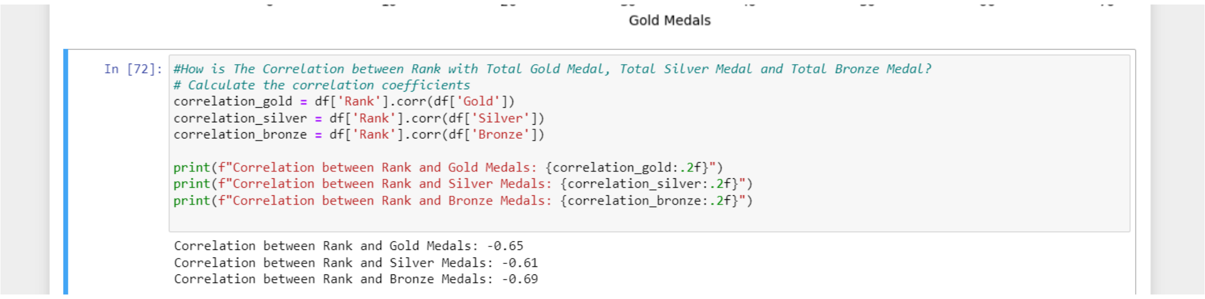
Scatter plot

"Scatter Plot Distribution of Total Gold, Silver and Bronze Medal", this figure clearly shows that total gold medal, total silver medal and total bronze medal affects the country rank in 2022 Common Wealth Games Medal Standings. This indicates that, if country has high total medal, this country may have high ranking in 2022 Common Wealth Games Medal Standings.



```
# Create a scatter plot
plt.figure(figsize=(10, 6))
plt.scatter(df['Gold'], df['Silver'], color='gold', label='Gold', alpha=0.7)
plt.scatter(df['Silver'], df['Bronze'], color='silver', label='Silver', alpha=0.7)
plt.scatter(df['Bronze'], df['Gold'], color='brown', label='Bronze', alpha=0.7)

# Label axes and add a legend
plt.xlabel('Gold Medals')
plt.ylabel('Silver and Bronze Medals')
plt.title('Scatter Plot: Total Gold, Silver, and Bronze Medals (2022 Commonwealth Games)')
plt.legend()

# Display the chart
plt.grid(True)
plt.tight_layout()
plt.show()
```

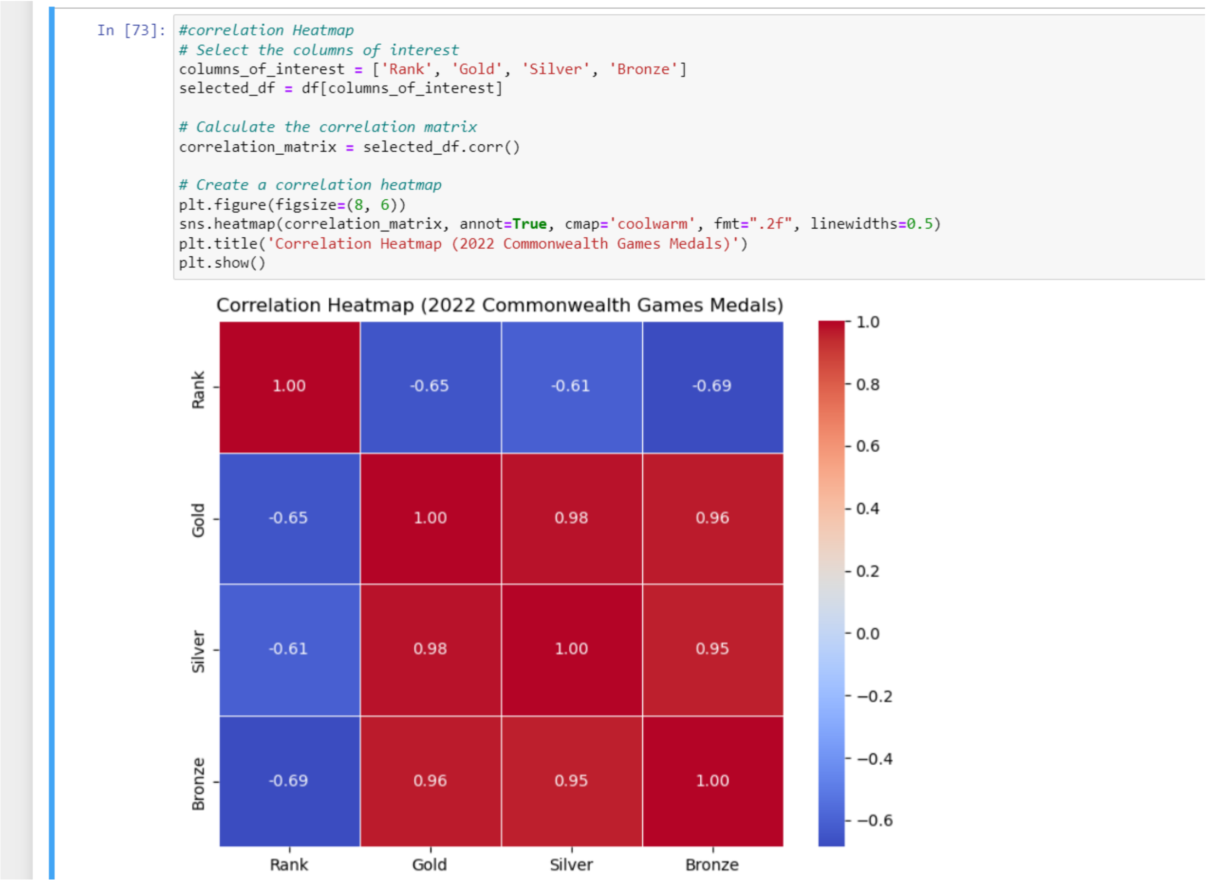
How is The Correlation between Rank with Total Gold Medal, Total Silver Medal and Total Bronze Medal?



```
#How is The Correlation between Rank with Total Gold Medal, Total Silver Medal and Total Bronze Medal?
# Calculate the correlation coefficients
correlation_gold = df['Rank'].corr(df['Gold'])
correlation_silver = df['Rank'].corr(df['Silver'])
correlation_bronze = df['Rank'].corr(df['Bronze'])

print(f"Correlation between Rank and Gold Medals: {correlation_gold:.2f}")
print(f"Correlation between Rank and Silver Medals: {correlation_silver:.2f}")
print(f"Correlation between Rank and Bronze Medals: {correlation_bronze:.2f}")
```

```
Correlation between Rank and Gold Medals: -0.65
Correlation between Rank and Silver Medals: -0.61
Correlation between Rank and Bronze Medals: -0.69
```

Correlation Heatmap:



```
#correlation Heatmap
# Select the columns of interest
columns_of_interest = ['Rank', 'Gold', 'Silver', 'Bronze']
selected_df = df[columns_of_interest]

# Calculate the correlation matrix
correlation_matrix = selected_df.corr()

# Create a correlation heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=0.5)
plt.title('Correlation Heatmap (2022 Commonwealth Games Medals)')
plt.show()
```

Explanation:

1.Rank has high negative correlation with totalMedal_Gold (-0.65), totalMedal_Silver (-0.61), and totalMedal_Bronze (-0.69):

- These negative correlations suggest that as the country's rank improves (i.e., it achieves a lower rank), it tends to win more gold, silver, and bronze medals, and consequently, more total medals. In other words, countries with lower ranks tend to have higher medal counts.

2.totalMedal_Gold has high positive correlation with totalMedal_Bronze (0.96) and totalMedal_Silver (0.98):

- These strong positive correlations indicate that there is a strong relationship between the number of gold medals a country wins and the number of silver and bronze medals they win. When a country excels in winning gold medals, it also tends to win more silver and bronze medals.

3.totalMedal_Gold has high negative correlation with Rank (-0.65):

- This negative correlation suggests that as a country wins more gold medals, its rank tends to improve (i.e., it achieves a lower rank). Countries with higher gold medal counts tend to have better ranks.

4.totalMedal_Silver has high positive correlation with totalMedal_Bronze (0.95) and totalMedal_Gold (0.98):

- Similar to totalMedal_Gold, totalMedal_Silver exhibits strong positive correlations with both totalMedal_Bronze and totalMedal_Gold. This implies that countries that excel in winning silver medals also tend to win more gold and bronze medals.

5.totalMedal_Silver has high negative correlation with Rank (-0.61):

- This negative correlation suggests that as a country wins more silver medals, its rank tends to improve (i.e., it achieves a lower rank). Countries with higher silver medal counts tend to have better ranks.

6.totalMedal_Bronze has high positive correlation with totalMedal_Silver (0.95) and totalMedal_Gold (0.96):

- Similar to the other two medal categories, totalMedal_Bronze exhibits strong positive correlations with both totalMedal_Silver and totalMedal_Gold. This means that countries that excel in winning bronze medals also tend to win more silver and gold medals.

7.totalMedal_Bronze has high negative correlation with Rank (-0.69):

- This negative correlation suggests that as a country wins more bronze medals, its rank tends to improve (i.e., it achieves a lower rank). Countries with higher bronze medal counts tend to have better ranks.

Key findings :

1.Medal Distribution Analysis:

- Australia emerged as the top-performing nation, securing the highest number of gold, silver, and bronze medals, resulting in an overall top rank.
- England closely followed, showcasing a well-rounded performance with a substantial medal count in all categories.
- Canada, India, and New Zealand secured their positions in the top five, consistently winning medals across all categories.
- The top-performing nations displayed a dominance in specific medal categories, with Australia excelling in gold medals, while England and Canada displayed strength in silver and bronze, respectively.

2. Correlation Investigation:

- There exists a strong negative correlation between a country's rank and its gold, silver, and bronze medal counts.
- Countries with lower ranks tend to win more medals, indicating a significant impact of medal count on rank.
- Gold, silver, and bronze medals exhibit strong positive correlations among themselves. Nations excelling in one medal category tend to perform well in others as well.
- The correlation analysis underscores the importance of medal count as a determinant of a country's rank in the standings.

3.Trend and Pattern Identification:

- Trends revealed that countries achieving higher ranks often had a strategic focus on excelling in specific medal categories.
- This strategic approach contributed significantly to their overall rankings. Some countries achieved higher rankings due to their exceptional performance in one particular medal category, compensating for lower counts in others.

4.Implications and Insights:

- The analysis highlights that securing a high number of medals, irrespective of category, is pivotal for achieving a top rank in the Commonwealth Games standings.
- Nations may benefit from strategic allocation of resources and training programs to excel in specific medal categories, thereby enhancing their overall ranking.
- The importance of fostering excellence across all medal categories is emphasized, as it contributes to a well-rounded and competitive performance on the global stage.

5.Recommendations and Applications:

- The findings of this analysis offer valuable insights for sports enthusiasts, policymakers, and stakeholders, aiding them in evaluating and appreciating the achievements of nations in international sporting events.
- The implications of this analysis extend to sports development policies, training programs, and future competition strategies, where a balanced approach to medal acquisition is advised.

Managerial Insights

1.Strategic Medal Allocation: Nations that strategically allocate resources and training efforts towards excelling in specific medal categories, such as gold, silver, or bronze, can significantly impact their overall rankings. Managers and sports authorities should consider aligning resources with areas of strength to optimize medal acquisition.

2. Data-Driven Decision-Making: The analysis highlights the importance of data-driven decision-making in sports management. Utilizing data analytics to assess medal counts, strengths, and weaknesses can help countries make informed decisions about where to focus their efforts and resources.

3.Balanced Approach: While strategic specialization in medal categories can be effective, it's essential to maintain a balanced approach. Excelling in one category should not come at the cost of neglecting others. A balanced strategy can help nations achieve better overall rankings.

4.Global Competition: The Commonwealth Games serve as a platform for countries to assess their athletic prowess on a global scale. Understanding the competition and the strengths of other participating nations can inform training and preparation strategies.

Implications for Sports Management

Training and Development: Sports authorities and national teams should design training and development programs that align with the country's strengths and weaknesses in specific medal categories. Tailored training can lead to improved performance and medal acquisition.

Resource Allocation: Sports governing bodies should allocate resources strategically to support athletes and teams that have the potential to excel in medal-winning events. This includes funding, coaching, equipment, and facilities.

Performance Metrics: Establishing performance metrics that track progress in medal categories can help countries monitor their success and make real-time adjustments to their strategies

Long-Term Planning: Success in international competitions often requires long-term planning. Sports authorities should invest in grassroots development programs to nurture young talent and ensure a sustainable pipeline of skilled athletes.

Coaching and Mentorship: Access to experienced coaches and mentors can make a significant difference in athlete performance. Identifying and retaining top coaching talent is essential for achieving medal-winning results.

Promotion and Sponsorship: Success in the Commonwealth Games can attract sponsors and support for sports programs. Countries should leverage their medal achievements to secure funding for sports development initiatives.

In conclusion, the analysis of the 2022 Commonwealth Games medal standings provides valuable insights and actionable recommendations for sports management and authorities. By embracing data-driven strategies, optimizing resource allocation, and fostering a culture of excellence, nations can enhance their performance on the global sports stage and achieve higher rankings in future competitions.