Project Name - Olympics Data Analysis _ ML _ FA _ DA projects (Part 1)

Project Type - Data Analysis

Industry - Unified Mentor

Contribution - Individual

Member Name - Hare Krishana Mishra

Task - 1

Project Summary -

Project Description:

This project focuses on analyzing Summer Olympic data from 1976 to 2008, using data-driven approaches to extract meaningful insights, visualize patterns, and build a predictive model to identify athletes or events most likely to secure a medal. By cleaning and encoding the dataset, various trends such as medal distribution across countries, genders, sports, and years were examined. Additionally, a logistic regression model was developed to classify whether an athlete or event won a medal based on categorical features.

Objective:

- To perform exploratory data analysis (EDA) on historical Olympic data.
- To uncover key trends related to medal wins by country, gender, sport, and year.
- To preprocess and encode data for modeling.
- To train a logistic regression model that predicts medal wins based on encoded features.

Key Project Details:

Dataset: Summer Olympic Medals (1976–2008)

Total Records Analyzed: 15,433

Features Used: Country, Sport, Gender, Event_gender, Year, Medal

Techniques Applied:

Data Cleaning and Handling Missing Values

Label Encoding of Categorical Variables

Visualizations with Seaborn & Matplotlib

Binary Classification using Logistic Regression

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

Evaluation Metrics: Accuracy Score, Confusion Matrix, Classification Report

Tech Stack:

Python, Pandas, NumPy, Matplotlib, Seaborn, Scikit-learn

Key Insights:

Top-performing countries and sports identified

Medal distribution across genders analyzed

Participation and medal trends visualized by Olympic year

Let's Begin:-

Step 1: Data Preparation

In [ ]:

```
# Import necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [ ]:

```
# Load the dataset (assume CSV format
df = pd.read_csv('/content/Summer-Olympic-medals-1976-to-2008 (1).csv', encoding='latin1
```

In [ ]:

```
# Check the first few rows of the dataset
df.head()
```

Out[ ]:

	City	Year	Sport	Discipline	Event	Athlete	Gender	Country_Code	Country	Ev
0	Montreal	1976.0	Aquatics	Diving	3m springboard	KÖHLER, Christa	Women	GDR	East Germany	
1	Montreal	1976.0	Aquatics	Diving	3m springboard	KOSENKOV, Aleksandr	Men	URS	Soviet Union	
2	Montreal	1976.0	Aquatics	Diving	3m springboard	BOGGS, Philip George	Men	USA	United States	
3	Montreal	1976.0	Aquatics	Diving	3m springboard	CAGNOTTO, Giorgio Franco	Men	ITA	Italy	
4	Montreal	1976.0	Aquatics	Diving	10m platform	WILSON, Deborah Keplar	Women	USA	United States	

In [ ]:

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

```
# Summary of the dataset
print(df.info())
print(df.describe())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15433 entries, 0 to 15432
Data columns (total 11 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   City          15316 non-null  object
 1   Year          15316 non-null  float64
 2   Sport         15316 non-null  object
 3   Discipline    15316 non-null  object
 4   Event         15316 non-null  object
 5   Athlete       15316 non-null  object
 6   Gender        15316 non-null  object
 7   Country_Code  15316 non-null  object
 8   Country       15316 non-null  object
 9   Event_gender  15316 non-null  object
 10  Medal         15316 non-null  object
dtypes: float64(1), object(10)
memory usage: 1.3+ MB
None
               Year
count  15316.000000
mean    1993.620789
std       10.159851
min     1976.000000
25%     1984.000000
50%     1996.000000
75%     2004.000000
max     2008.000000
```

Step 2: Data Cleaning

In [ ]:

```
# Check for missing values
print(df.isnull().sum())
```

```
City            117
Year            117
Sport           117
Discipline      117
Event           117
Athlete         117
Gender          117
Country_Code    117
Country         117
Event_gender    117
Medal           117
dtype: int64
```

In [ ]:

```
df.shape
```

Out[ ]:

```
(15433, 11)
```

In [ ]:

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

```python
# Drop rows with missing values if any
df_cleaned = df.dropna()
```

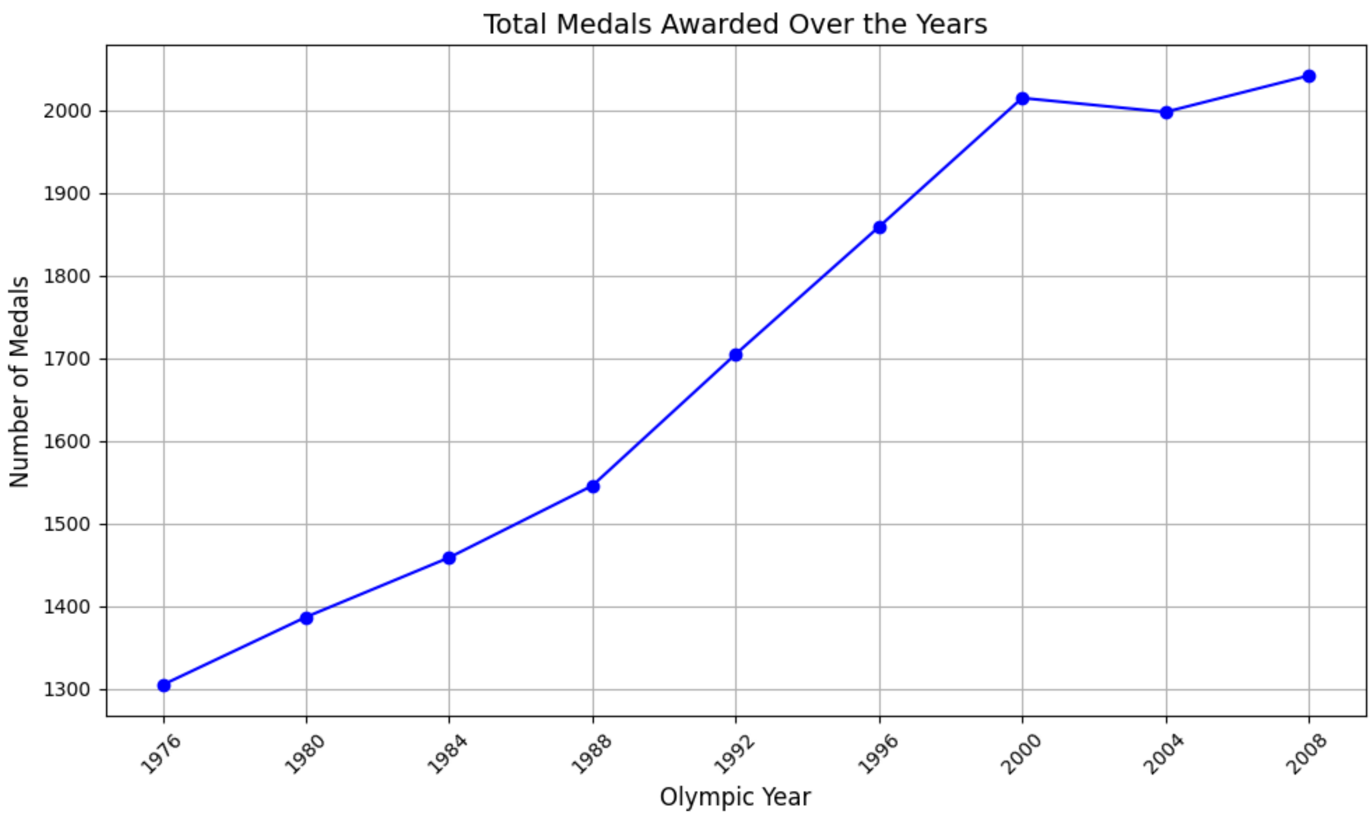
Step 3: Exploratory Data Analysis (EDA)

3.1 Medals Won Over the Years

In [ ]:

```python
# Grouping by Year and counting the medals won
medals_over_years = df_cleaned.groupby('Year')['Medal'].count().sort_index()

# Plotting the trend of medals won over the years
plt.figure(figsize=(10, 6))
plt.plot(medals_over_years.index.astype(int), medals_over_years.values, marker='o', line
plt.title("Total Medals Awarded Over the Years", fontsize=14)
plt.xlabel("Olympic Year", fontsize=12)
plt.ylabel("Number of Medals", fontsize=12)
plt.xticks(medals_over_years.index.astype(int), rotation=45)
plt.grid(True)
plt.tight_layout()
plt.show()
```



3.2 Total Medal Count by Country

In [ ]:

```python
# Total medals won by each country
medals_by_country = df_cleaned.groupby('Country')['Medal'].count().sort_values(ascending
medals_by_country
```

Out[ ]:

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

	Medal
Country	
United States	1992
Soviet Union	1021
Australia	798
Germany	691
China	679
...	...
Sri Lanka	1
Togo	1
United Arab Emirates	1
Uruguay	1
Virgin Islands*	1

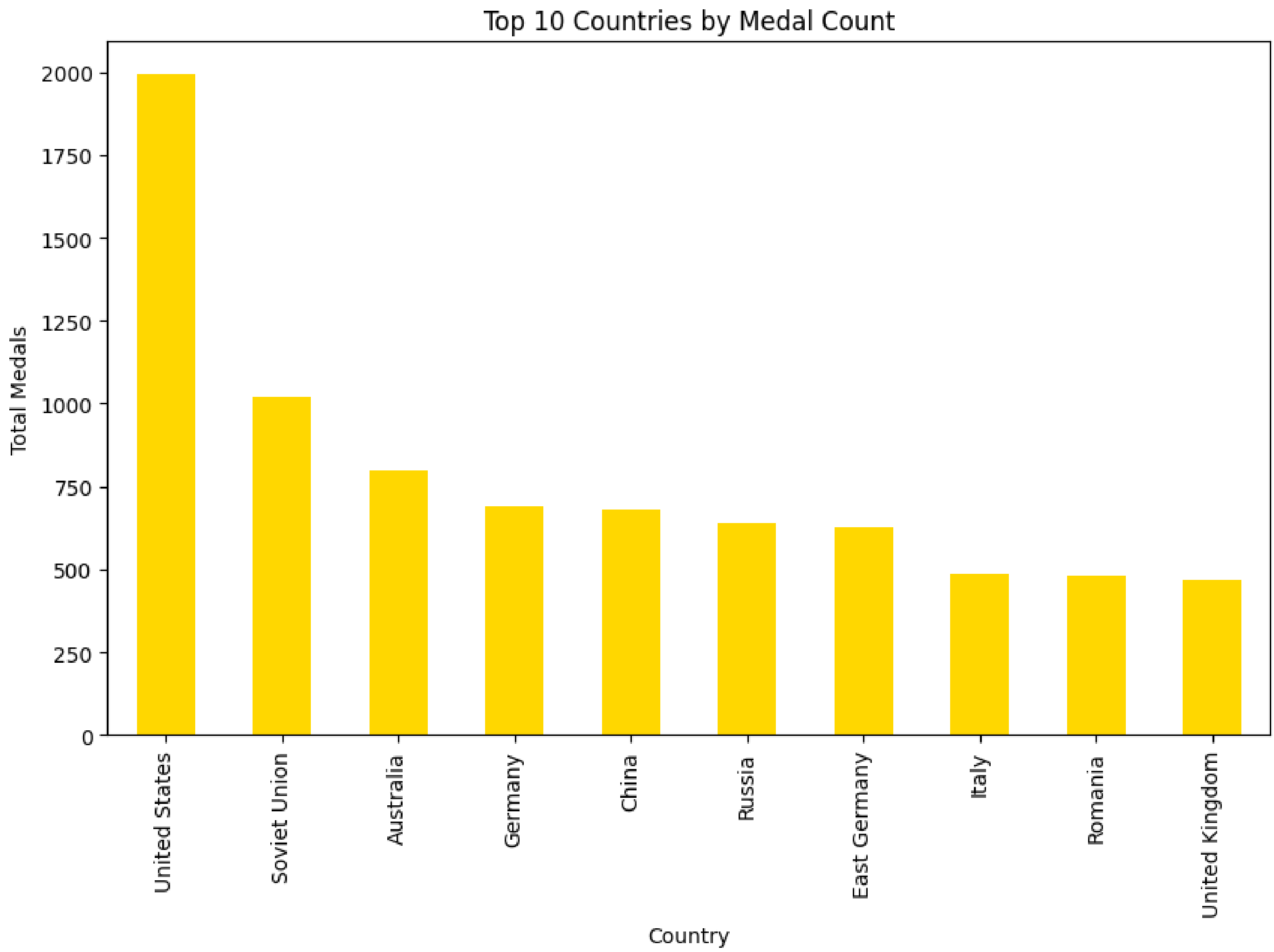
127 rows × 1 columns

dtype: int64

In [ ]:

```
# Plotting the top 10 countries by medals
plt.figure(figsize=(10, 6))
medals_by_country.head(10).plot(kind='bar', color='gold')
plt.title("Top 10 Countries by Medal Count")
plt.xlabel("Country")
plt.ylabel("Total Medals")
plt.show()
```

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF



3.3 Top Athletes with Most Medals

In [ ]:

```
# Group by Athlete and count the number of medals
athlete_medal_count = df_cleaned.groupby('Athlete')['Medal'].count().sort_values(ascendi
athlete_medal_count
```

Out[ ]:

Athlete	Medal
PHELPS, Michael	16
FISCHER, Birgit	12
ANDRIANOV, Nikolay	12
TORRES, Dara	12
THOMPSON, Jenny	12
...	...
ZVYAGINTSEV, Viktor	1
ZWEHL, Julia	1
ZWERING, Klaas-Erik	1

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

	Medal
Athlete	
ZUEVA, Natalia	1
ZUIJDWEG, Martijn	1

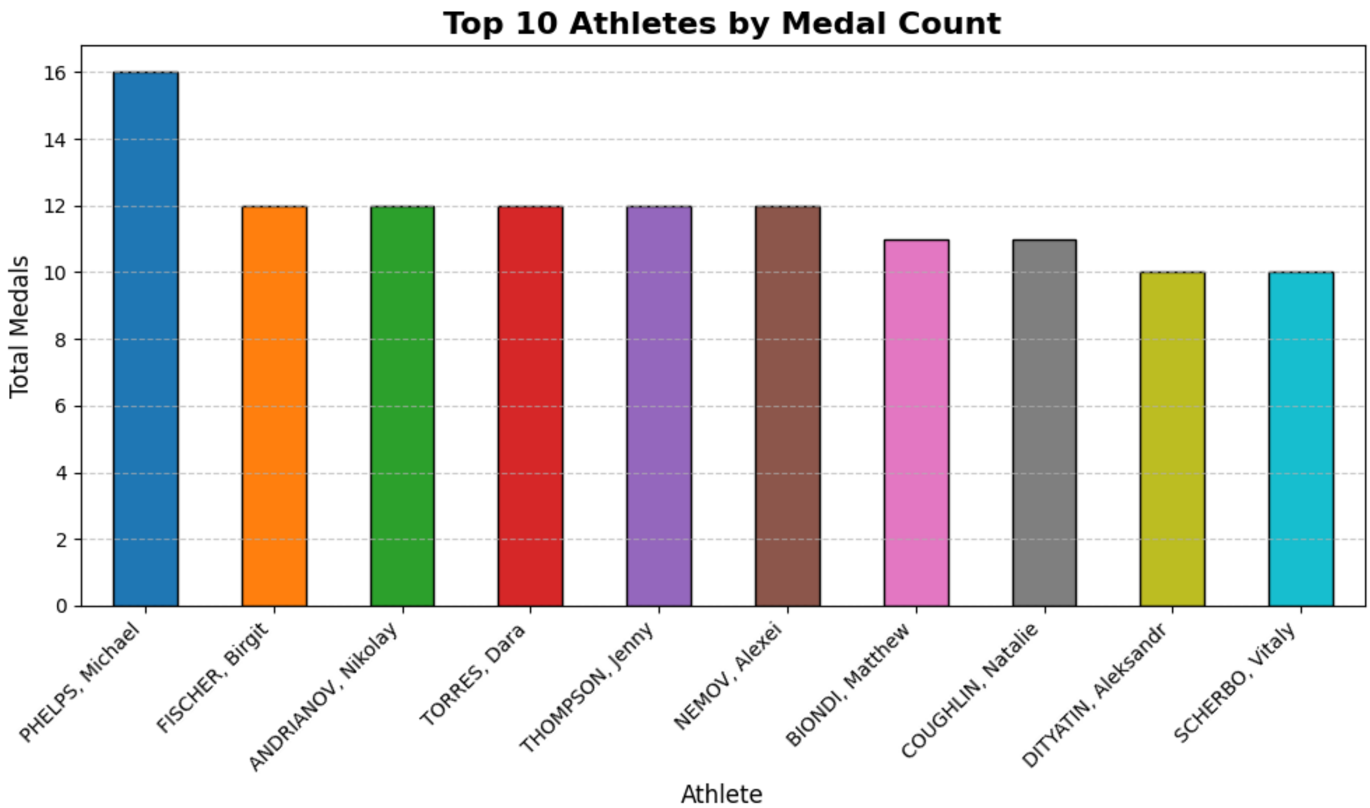
11337 rows × 1 columns

dtype: int64

In [ ]:

```
# Get a list of 10 different colors
colors = cm.tab10(np.linspace(0, 1, 10))

plt.figure(figsize=(10, 6))
athlete_medal_count.head(10).plot(
    kind='bar',
    color=colors,
    edgecolor='black'
)
plt.title("Top 10 Athletes by Medal Count", fontsize=16, fontweight='bold')
plt.xlabel("Athlete", fontsize=12)
plt.ylabel("Total Medals", fontsize=12)
plt.xticks(rotation=45, ha='right')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```

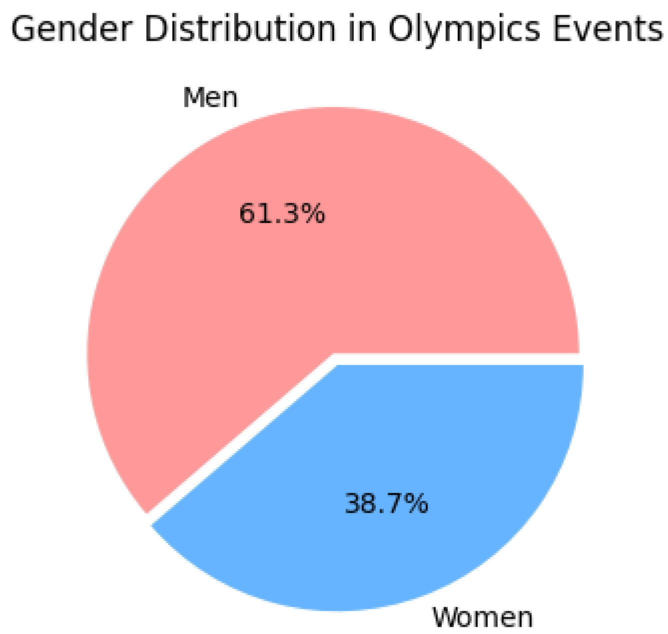


3.4 Gender Distribution in Events

In [ ]:

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

```python
# Gender distribution in events
gender_distribution = df_cleaned['Gender'].value_counts()
# Plotting gender distribution
plt.figure(figsize=(6, 4))
gender_distribution.plot(kind='pie', autopct='%1.1f%%',
colors=['#ff9999','#66b3ff'], explode=[0.05, 0])
plt.title("Gender Distribution in Olympics Events")
plt.ylabel('')
plt.show()
```

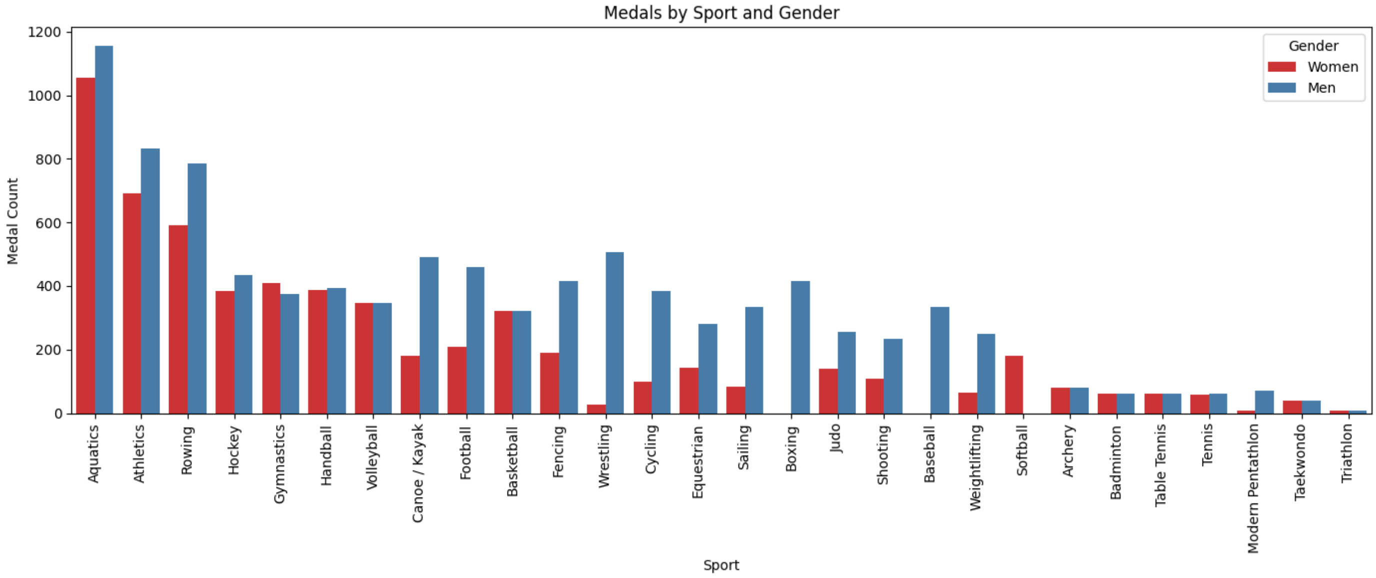


3.5 Medals by Sport and Gender

In [ ]:

```python
plt.figure(figsize=(14, 6))
sns.countplot(
    data=df_cleaned,
    x='Sport',
    hue='Gender',
    order=df_cleaned['Sport'].value_counts().index,
    palette='Set1'
)
plt.xticks(rotation=90)
plt.title("Medals by Sport and Gender")
plt.xlabel("Sport")
plt.ylabel("Medal Count")
plt.tight_layout()
plt.show()
```

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

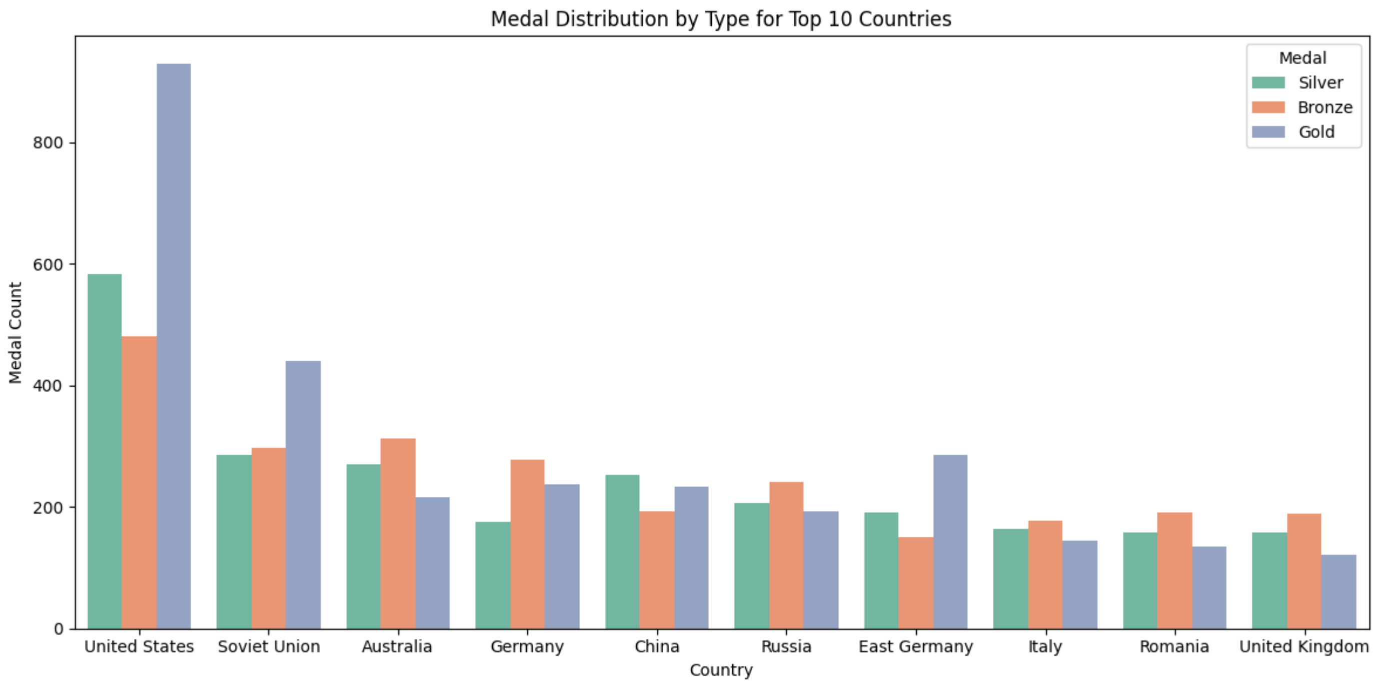


3.6 Medal Distribution by Type for Top 10 Countries

In [ ]:

```
top_10_countries = df_cleaned['Country'].value_counts().head(10).index

plt.figure(figsize=(12, 6))
sns.countplot(
    data=df_cleaned[df_cleaned['Country'].isin(top_10_countries)],
    x='Country',
    hue='Medal',
    order=top_10_countries,
    palette='Set2'
)
plt.title("Medal Distribution by Type for Top 10 Countries")
plt.xlabel("Country")
plt.ylabel("Medal Count")
plt.tight_layout()
plt.show()
```



Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

3.7 Top 20 Athletes by Number of Medals

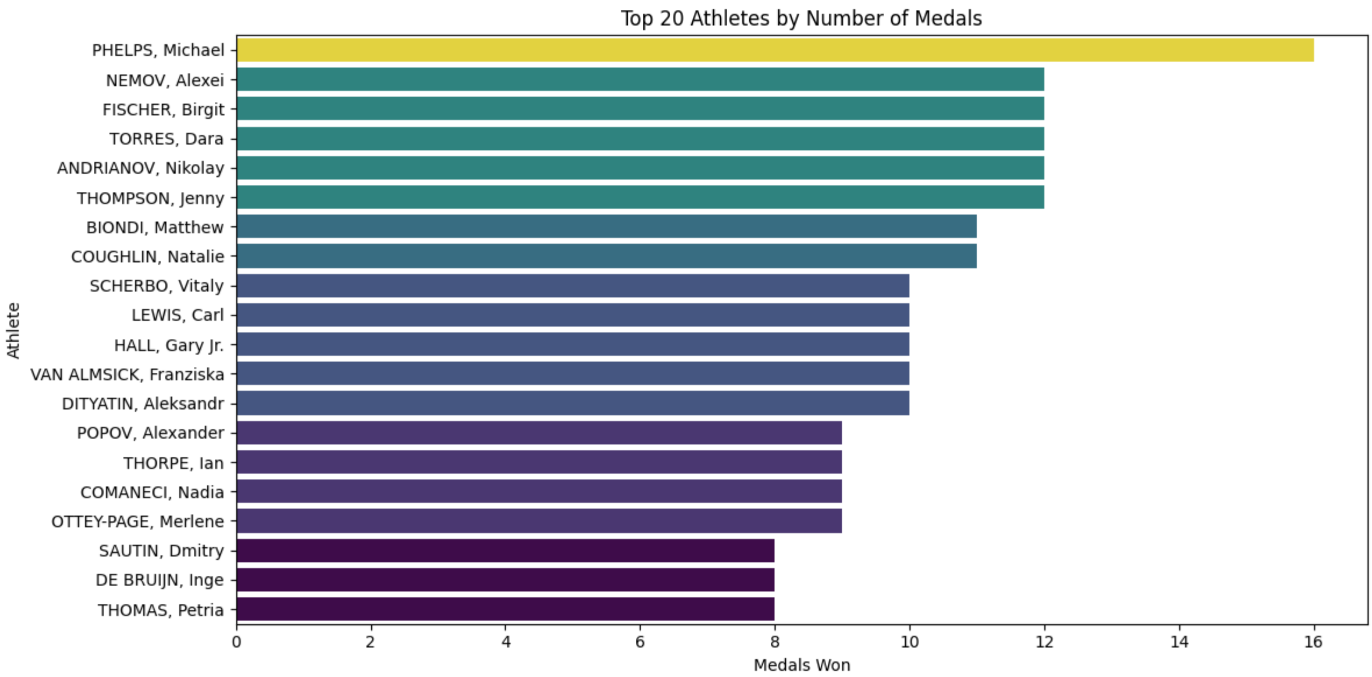
In [ ]:

```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Prepare the top 20 athlete data
athlete_medal_counts = df_cleaned['Athlete'].value_counts().head(20).reset_index()
athlete_medal_counts.columns = ['Athlete', 'Medal_Count']

# Add a fake hue column to enable palette usage
athlete_medal_counts['ColorGroup'] = athlete_medal_counts['Medal_Count']

# Plot with custom palette and no legend
plt.figure(figsize=(12, 6))
sns.barplot(
    data=athlete_medal_counts,
    x='Medal_Count',
    y='Athlete',
    hue='ColorGroup',
    palette='viridis',   # 🎨 You can try: 'coolwarm', 'rocket', 'cubehelix', 'mako', et
    dodge=False,
    legend=False
)
plt.title("Top 20 Athletes by Number of Medals")
plt.xlabel("Medals Won")
plt.ylabel("Athlete")
plt.tight_layout()
plt.show()
```



Step 4: Predictive Analysis (Machine Learning)

In [ ]:

```python
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
```

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

```python
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix,classification_report
```

In [ ]:

```python
# Encode categorical variables using LabelEncoder
le = LabelEncoder()
df_cleaned['Country_Code'] = le.fit_transform(df_cleaned['Country_Code'])
df_cleaned['Sport'] = le.fit_transform(df_cleaned['Sport'])
df_cleaned['Gender'] = le.fit_transform(df_cleaned['Gender'])
df_cleaned['Event_gender'] = le.fit_transform(df_cleaned['Event_gender'])
df_cleaned['Medal'] = df_cleaned['Medal'].map({'Gold': 1,'Silver': 1, 'Bronze': 1, np.na
```

```
/tmp/ipython-input-3408056398.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
  df_cleaned['Country_Code'] = le.fit_transform(df_cleaned['Country_Code'])
/tmp/ipython-input-3408056398.py:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
  df_cleaned['Sport'] = le.fit_transform(df_cleaned['Sport'])
/tmp/ipython-input-3408056398.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
  df_cleaned['Gender'] = le.fit_transform(df_cleaned['Gender'])
/tmp/ipython-input-3408056398.py:6: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
  df_cleaned['Event_gender'] = le.fit_transform(df_cleaned['Event_gender'])
/tmp/ipython-input-3408056398.py:7: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
  df_cleaned['Medal'] = df_cleaned['Medal'].map({'Gold': 1,'Silver': 1, 'Bronze': 1, np.
nan: 0})
```

In [ ]:

```python
# Features and target
X = df_cleaned[['Country_Code', 'Sport', 'Gender','Event_gender']]
y = df_cleaned['Medal']
```

In [ ]:

```python
# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y,test_size=0.3, random_state=42)
```

In [ ]:

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

```python
y = y.fillna(0)
```

In [ ]:

```python
y = y.astype(int)
```

In [ ]:

```python
model = LogisticRegression(max_iter=1000)
```

Step 5: Conclusion and Insights

● Top Performing Countries: We identified which countries won the most medals.

● Top Athletes: We identified athletes who won the most medals.

● Gender Participation: The gender distribution in different sports events was explored.

● Trend of Medals Over Years: We visualized the trend of medal wins over the years.

The logistic regression model allowed us to predict whether an athlete would win a medal based on various attributes like country, sport, and gender. This project can be extended by adding more sophisticated machine learning models (like decision trees or random forests), and further fine-tuning the models by including more features.

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

Project Name - Olympics Data Analysis _ ML _ FA _ DA projects (Part 2)

Project Type - Data Analysis

Industry - Unified Mentor

Contribution - Individual

Member Name - Hare Krishana Mishra

Task - 2

Project Summary -

Project Description:

The Olympics Data Analysis project uses a dataset of all Summer Olympics medal winners from 1976 (Montreal) to 2008 (Beijing). It covers details such as city, year, sport, discipline, event, athlete, gender, country, and medal type. The project involves cleaning and exploring the dataset, identifying patterns in medal distribution, studying gender participation, and analyzing country dominance in various sports. It also includes a basic machine learning model to predict whether an athlete will win a medal based on attributes like country, sport, and gender.

Objective:

- Analyze trends in medal distribution across countries, sports, and years.
- TIdentify top-performing countries and athletes.
- Examine gender distribution in events and medals.
- Visualize Olympic trends and insights using Python.
- Predict medal outcomes using machine learning techniques.

Key Project Details:

Domain: Data Analytics / Machine Learning

Tools: Python, Pandas, Matplotlib, Seaborn, Scikit-learn, SQL, Excel

Dataset: Summer Olympics medal data (1976–2008) with city, year, sport, event, athlete, gender, country, and medal type.

Analysis Highlights:

- Medal trends by country, year, and sport
- Gender participation patterns
- Top athletes and country dominance
- Visualize Olympic trends and insights using Python.
- Notable athletes who changed sports and still won medals

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

Let's Begin:-

In [ ]:

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g.pd.read_csv)
import matplotlib.pyplot as plt
```

In [ ]:

```
pd.options.display.max_rows = 4000
pd.options.display.max_columns= None
```

In [ ]:

```
data = pd.read_csv('/content/Summer-Olympic-medals-1976-to-2008.csv', encoding = 'latin1
data.head()
```

Out[ ]:

	City	Year	Sport	Discipline	Event	Athlete	Gender	Country_Code	Country	Ev
0	Montreal	1976.0	Aquatics	Diving	3m springboard	KÖHLER, Christa	Women	GDR	East Germany	
1	Montreal	1976.0	Aquatics	Diving	3m springboard	KOSENKOV, Aleksandr	Men	URS	Soviet Union	
2	Montreal	1976.0	Aquatics	Diving	3m springboard	BOGGS, Philip George	Men	USA	United States	
3	Montreal	1976.0	Aquatics	Diving	3m springboard	CAGNOTTO, Giorgio Franco	Men	ITA	Italy	
4	Montreal	1976.0	Aquatics	Diving	10m platform	WILSON, Deborah Keplar	Women	USA	United States	

In [ ]:

```
print(data.Gender.unique())
print(data.Event_gender.unique())
```

```
['Women' 'Men' nan]
['W' 'M' 'X' nan]
```

In [ ]:

```
data= data.drop('Event_gender', axis = 1)
data= data.drop('Country_Code', axis = 1)
data.head()
```

Out[ ]:

	City	Year	Sport	Discipline	Event	Athlete	Gender	Country	Medal
0	Montreal	1976.0	Aquatics	Diving	3m springboard	KÖHLER, Christa	Women	East Germany	Silver
1	Montreal	1976.0	Aquatics	Diving	3m springboard	KOSENKOV, Aleksandr	Men	Soviet Union	Bronze
2	Montreal	1976.0	Aquatics	Diving	3m	BOGGS, Philip	Men	United	Gold

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

	City	Year	Sport	Discipline	Event	Athlete	Gender	Country	Medal
					springboard	George		States	
3	Montreal	1976.0	Aquatics	Diving	3m springboard	CAGNOTTO, Giorgio Franco	Men	Italy	Silver
4	Montreal	1976.0	Aquatics	Diving	10m platform	WILSON, Deborah Keplar	Women	United States	Bronze

In [ ]:

```
print(data.isnull().sum())
data = data.dropna(how = 'all')
print(data.isnull().sum())
data = data.astype({'Year':'int'})
data.head()
```

```
City          117
Year          117
Sport         117
Discipline    117
Event         117
Athlete       117
Gender        117
Country       117
Medal         117
dtype: int64
City          0
Year          0
Sport         0
Discipline    0
Event         0
Athlete       0
Gender        0
Country       0
Medal         0
dtype: int64
```

Out[ ]:

	City	Year	Sport	Discipline	Event	Athlete	Gender	Country	Medal
0	Montreal	1976	Aquatics	Diving	3m springboard	KÖHLER, Christa	Women	East Germany	Silver
1	Montreal	1976	Aquatics	Diving	3m springboard	KOSENKOV, Aleksandr	Men	Soviet Union	Bronze
2	Montreal	1976	Aquatics	Diving	3m springboard	BOGGS, Philip George	Men	United States	Gold
3	Montreal	1976	Aquatics	Diving	3m springboard	CAGNOTTO, Giorgio Franco	Men	Italy	Silver
4	Montreal	1976	Aquatics	Diving	10m platform	WILSON, Deborah Keplar	Women	United States	Bronze

Q1. Which city hosted maximum number of olympics

Logic : Focus on City and Year. Get unique Year. Print the data.

Explore our developer-friendly HTML to PDF API   Printed using PDFCrowd   HTML to PDF

In [ ]:

```
q1_data = data[["City", 'Year']]
q1_data = q1_data.drop_duplicates('Year')
```

In [ ]:

```
q1_data
```

Out[ ]:

	City	Year
0	Montreal	1976
1422	Moscow	1980
2809	Los Angeles	1984
4268	Seoul	1988
5814	Barcelona	1992
7519	Atlanta	1996
9378	Sydney	2000
11393	Athens	2004
13391	Beijing	2008

Q2. Which city hosted most events.

logic: Focus on City.Find count of unique values.Print the count

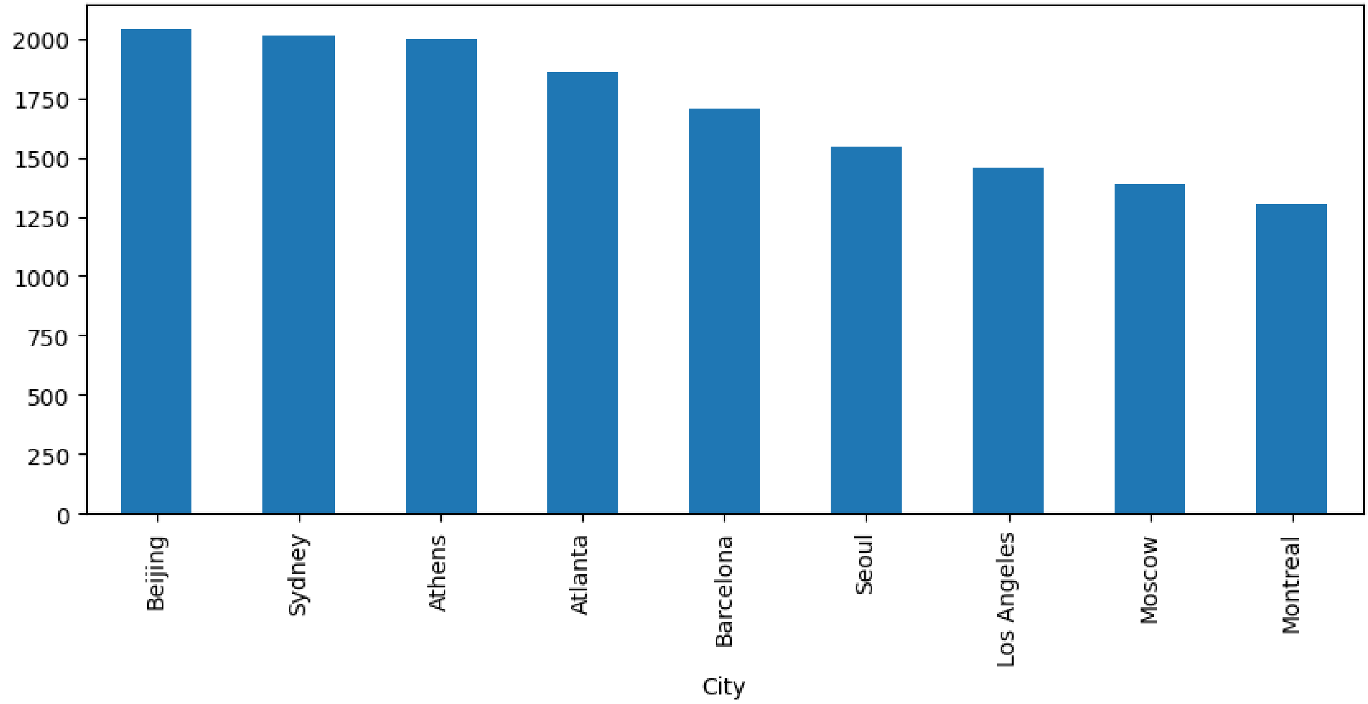
In [ ]:

```
q2_data = data['City'].value_counts()
q2_data.columns = ['City', 'Count']
plt.figure(figsize = (10,4))
q2_data.plot.bar(x = 'City', y = 'Count') # q2_data.plot(kind = 'bar', x= 'City', y = 'C
```

Out[ ]:

```
<Axes: xlabel='City'>
```

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF



Ans : Beijing has the hosted the biggest Olympics since 1976 till 2008. Followed by Sydney and Athens.

Q3. Understand the events themselves.

logic : Focus on Sport, Discipline and Event. Use groupby and see how many kinds and variations are there.

In [ ]:

```
q3_data = data[['Sport', 'Discipline','Event']].drop_duplicates()
print("Total number of unique events are held so far are :",len(q3_data))
q3_data
```

```
Total number of unique events are held so far are : 334
```

Out[ ]:

	Sport	Discipline	Event
0	Aquatics	Diving	3m springboard
4	Aquatics	Diving	10m platform
12	Aquatics	Swimming	4x100m freestyle relay
13	Aquatics	Swimming	400m freestyle
15	Aquatics	Swimming	1500m freestyle
16	Aquatics	Swimming	400m individual medley
17	Aquatics	Swimming	4x100m medley relay
18	Aquatics	Swimming	800m freestyle
21	Aquatics	Swimming	200m backstroke
25	Aquatics	Swimming	200m freestyle
26	Aquatics	Swimming	100m butterfly
27	Aquatics	Swimming	100m backstroke

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

	Sport	Discipline	Event
32	Aquatics	Swimming	4x200m freestyle relay
39	Aquatics	Swimming	200m breaststroke
46	Aquatics	Swimming	100m breaststroke
53	Aquatics	Swimming	200m butterfly
84	Aquatics	Swimming	100m freestyle
126	Aquatics	Water polo	water polo
159	Archery	Archery	individual FITA round
165	Athletics	Athletics	4x400m relay
166	Athletics	Athletics	4x100m relay
167	Athletics	Athletics	long jump
170	Athletics	Athletics	high jump
171	Athletics	Athletics	100m hurdles
172	Athletics	Athletics	400m
173	Athletics	Athletics	5000m
174	Athletics	Athletics	100m
176	Athletics	Athletics	pentathlon
178	Athletics	Athletics	200m
180	Athletics	Athletics	hammer throw
182	Athletics	Athletics	decathlon
184	Athletics	Athletics	800m
190	Athletics	Athletics	shot put
195	Athletics	Athletics	3000m steeplechase
196	Athletics	Athletics	1500m
198	Athletics	Athletics	triple jump
201	Athletics	Athletics	discus throw
202	Athletics	Athletics	javelin throw
213	Athletics	Athletics	marathon
219	Athletics	Athletics	110m hurdles
220	Athletics	Athletics	20km walk
235	Athletics	Athletics	400m hurdles
239	Athletics	Athletics	pole vault
253	Athletics	Athletics	10000m
312	Basketball	Basketball	basketball
384	Boxing	Boxing	63.5 - 67kg (welterweight)
385	Boxing	Boxing	67 - 71kg (light-middleweight)

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

	Sport	Discipline	Event
386	Boxing	Boxing	60 - 63.5kg (light-welterweight)
388	Boxing	Boxing	71-75kg
389	Boxing	Boxing	57 - 60kg (lightweight)
390	Boxing	Boxing	75 - 81kg (light-heavyweight)
391	Boxing	Boxing	+ 81kg (heavyweight)
392	Boxing	Boxing	54 - 57kg (featherweight)
393	Boxing	Boxing	48 - 51kg (flyweight)
395	Boxing	Boxing	51 - 54kg (bantamweight)
398	Boxing	Boxing	- 48kg (light-flyweight)
428	Canoe / Kayak	Canoe / Kayak F	K-1 500m (kayak single)
429	Canoe / Kayak	Canoe / Kayak F	C-1 500m (canoe single)
430	Canoe / Kayak	Canoe / Kayak F	C-2 500m (canoe double)
431	Canoe / Kayak	Canoe / Kayak F	K-4 1000m (kayak four)
432	Canoe / Kayak	Canoe / Kayak F	K-1 1000m (kayak single)
433	Canoe / Kayak	Canoe / Kayak F	K-2 500m (kayak double)
437	Canoe / Kayak	Canoe / Kayak F	C-1 1000m (canoe single)
440	Canoe / Kayak	Canoe / Kayak F	K-2 1000m (kayak double)
445	Canoe / Kayak	Canoe / Kayak F	C-2 1000m (canoe double)
485	Cycling	Cycling Road	individual road race
486	Cycling	Cycling Road	team time trial
500	Cycling	Cycling Track	Team Pursuit (4000m)
501	Cycling	Cycling Track	Sprint indivual
504	Cycling	Cycling Track	1km time trial
505	Cycling	Cycling Track	Individual Pursuit
521	Equestrian	Dressage	team
523	Equestrian	Dressage	individual
533	Equestrian	Eventing	team
536	Equestrian	Eventing	individual
548	Equestrian	Jumping	team
549	Equestrian	Jumping	individual
563	Fencing	Fencing	épée team
564	Fencing	Fencing	sabre team
565	Fencing	Fencing	foil team
573	Fencing	Fencing	foil individual
586	Fencing	Fencing	sabre individual

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

	Sport	Discipline	Event
591	Fencing	Fencing	épée individual
634	Football	Football	football
685	Gymnastics	Artistic G.	team competition
688	Gymnastics	Artistic G.	individual all-round
689	Gymnastics	Artistic G.	parallel bars
691	Gymnastics	Artistic G.	vault
695	Gymnastics	Artistic G.	pommel horse
702	Gymnastics	Artistic G.	horizontal bar
704	Gymnastics	Artistic G.	uneven bars
707	Gymnastics	Artistic G.	balance beam
711	Gymnastics	Artistic G.	floor exercises
716	Gymnastics	Artistic G.	rings
759	Handball	Handball	handball
960	Hockey	Hockey	hockey
1008	Judo	Judo	70 - 80kg (middleweight)
1009	Judo	Judo	+ 93kg (heavyweight)
1010	Judo	Judo	open category
1011	Judo	Judo	63 - 70kg (half-middleweight)
1012	Judo	Judo	80 - 93kg (half-heavyweight)
1016	Judo	Judo	- 63kg (lightweight)
1032	Modern Pentathlon	Modern Pentath.	Team competition
1033	Modern Pentathlon	Modern Pentath.	Individual competition
1044	Rowing	Rowing	eight with coxswain (8+)
1046	Rowing	Rowing	quadruple sculls without coxswain (4x)
1047	Rowing	Rowing	single sculls (1x)
1048	Rowing	Rowing	pair without coxswain (2-)
1049	Rowing	Rowing	four-oared shell with coxswain (4-)
1052	Rowing	Rowing	quadruple sculls with coxswain (4x)
1058	Rowing	Rowing	four without coxswain (4-)
1065	Rowing	Rowing	pair-oared shell with coxswain (2+)
1076	Rowing	Rowing	coxless pair (2-)
1090	Rowing	Rowing	double sculls (2x)
1206	Sailing	Sailing	fleet/match race keelboat open (Soling)
1207	Sailing	Sailing	tempest
1208	Sailing	Sailing	flying dutchman

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

	Sport	Discipline	Event
1209	Sailing	Sailing	470 - Two Person Dinghy
1219	Sailing	Sailing	Tornado - Multihull
1221	Sailing	Sailing	single-handed dinghy (Finn)
1242	Shooting	Shooting	50m running target (30+30 shots)
1243	Shooting	Shooting	trap (125 targets)
1244	Shooting	Shooting	50m rifle 3 positions (3x40 shots)
1245	Shooting	Shooting	skeet (125 targets)
1246	Shooting	Shooting	50m pistol (60 shots)
1251	Shooting	Shooting	50m rifle prone (60 shots)
1257	Shooting	Shooting	25m rapid fire pistol (60 shots)
1263	Volleyball	Volleyball	volleyball
1335	Weightlifting	Weightlifting	82.5 - 90kg, total (middle-heavyweight)
1336	Weightlifting	Weightlifting	+ 110kg, total (super heavyweight)
1338	Weightlifting	Weightlifting	75 - 82.5kg, total (light-heavyweight)
1341	Weightlifting	Weightlifting	- 56kg, total (bantamweight)
1342	Weightlifting	Weightlifting	60 - 67.5kg, total (lightweight)
1343	Weightlifting	Weightlifting	67.5 - 75kg, total (middleweight)
1345	Weightlifting	Weightlifting	- 52kg, total (flyweight)
1347	Weightlifting	Weightlifting	56 - 60kg, total (featherweight)
1349	Weightlifting	Weightlifting	91 - 110kg, total (heavyweight)
1362	Wrestling	Wrestling Free.	57 - 62kg (featherweight)
1363	Wrestling	Wrestling Free.	62 - 68kg (lightweight)
1364	Wrestling	Wrestling Free.	74 - 82kg (middleweight)
1365	Wrestling	Wrestling Free.	+ 100kg (super heavyweight)
1367	Wrestling	Wrestling Free.	48 - 52kg (flyweight)
1371	Wrestling	Wrestling Free.	82 - 90kg (light-heavyweight)
1374	Wrestling	Wrestling Free.	90 - 100kg (heavyweight)
1375	Wrestling	Wrestling Free.	68 - 74kg (welterweight)
1376	Wrestling	Wrestling Free.	- 48kg (light-flyweight)
1377	Wrestling	Wrestling Free.	52 - 57kg (bantamweight)
1392	Wrestling	Wrestling Gre-R	62 - 68kg (lightweight)
1393	Wrestling	Wrestling Gre-R	68 - 74kg (welterweight)
1394	Wrestling	Wrestling Gre-R	- 48kg (light-flyweight)
1396	Wrestling	Wrestling Gre-R	52 - 57kg (bantamweight)
1397	Wrestling	Wrestling Gre-R	+ 100kg (super heavyweight)

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

	Sport	Discipline	Event
1398	Wrestling	Wrestling Gre-R	90 - 100kg (heavyweight)
1401	Wrestling	Wrestling Gre-R	48 - 52kg (flyweight)
1402	Wrestling	Wrestling Gre-R	82 - 90kg (light-heavyweight)
1405	Wrestling	Wrestling Gre-R	57 - 62kg (featherweight)
1406	Wrestling	Wrestling Gre-R	74 - 82kg (middleweight)
1609	Athletics	Athletics	50km walk
2385	Judo	Judo	- 60 kg
2386	Judo	Judo	86 - 95kg (half-heavyweight)
2387	Judo	Judo	71 - 78kg (half-middleweight)
2389	Judo	Judo	65 - 71kg (lightweight)
2392	Judo	Judo	+ 95kg (heavyweight)
2393	Judo	Judo	78 - 86kg (middleweight)
2397	Judo	Judo	60 - 65kg (half-lightweight)
2591	Sailing	Sailing	two-person keelboat open (Star)
2727	Weightlifting	Weightlifting	90 - 100kg, total (first-heavyweight)
2732	Weightlifting	Weightlifting	100 - 110kg, total (heavyweight)
2857	Aquatics	Swimming	200m individual medley
2953	Aquatics	Synchronized S.	solo
2955	Aquatics	Synchronized S.	duet
3061	Athletics	Athletics	3000m
3085	Athletics	Athletics	heptathlon
3240	Boxing	Boxing	81 - 91kg (heavyweight)
3253	Boxing	Boxing	+ 91kg (super heavyweight)
3298	Canoe / Kayak	Canoe / Kayak F	K-4 500m (kayak four)
3389	Cycling	Cycling Track	Points Race
3644	Gymnastics	Rhythmic G.	individual all-round
4041	Sailing	Sailing	board (windglider)
4073	Shooting	Shooting	25m pistol (30+30 shots)
4077	Shooting	Shooting	10m air rifle (60 shots)
4080	Shooting	Shooting	50m rifle 3 positions (3x20 shots)
4088	Shooting	Shooting	10m air rifle (40 shots)
4310	Aquatics	Swimming	50m freestyle
4470	Archery	Archery	teams FITA round
4878	Cycling	Cycling Track	sprint
5535	Sailing	Sailing	board (division II)

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

	Sport	Discipline	Event
5574	Shooting	Shooting	10m air pistol (60 shots)
5584	Shooting	Shooting	10m air pistol (40 shots)
5610	Table Tennis	Table Tennis	doubles
5612	Table Tennis	Table Tennis	singles
5628	Tennis	Tennis	doubles
5630	Tennis	Tennis	singles
5764	Wrestling	Wrestling Free.	100 - 130kg (super heavyweight)
5790	Wrestling	Wrestling Gre-R	100 - 130kg (super heavyweight)
6042	Archery	Archery	team (FITA Olympic round - 70m)
6043	Archery	Archery	individual (FITA Olympic round - 70m)
6076	Athletics	Athletics	10000m walk
6244	Badminton	Badminton	singles
6245	Badminton	Badminton	doubles
6268	Baseball	Baseball	baseball
6517	Canoe / Kayak	Canoe / Kayak S	C-2 (canoe double)
6518	Canoe / Kayak	Canoe / Kayak S	C-1 (canoe single)
6519	Canoe / Kayak	Canoe / Kayak S	K-1 (kayak single)
6550	Cycling	Cycling Track	individual pursuit
6997	Judo	Judo	61 - 66kg (middleweight)
6998	Judo	Judo	52 - 56kg (lightweight)
6999	Judo	Judo	56 - 61kg (half-middleweight)
7000	Judo	Judo	+ 72kg (heavyweight)
7003	Judo	Judo	66 - 72kg (half-heavyweight)
7010	Judo	Judo	- 48kg (extra-lightweight)
7011	Judo	Judo	48 - 52kg (half-lightweight)
7075	Rowing	Rowing	coxless four (4-)
7224	Sailing	Sailing	board (lechner)
7229	Sailing	Sailing	single-handed dinghy (Europe)
7712	Aquatics	Synchronized S.	team
8275	Cycling	Cycling Road	individual time trial
8285	Cycling	Cycling Track	points race
8319	Cycling	Mountain Bike	cross-country
8614	Gymnastics	Rhythmic G.	group competition
8830	Judo	Judo	90 - 100kg (half-heavyweight)
8835	Judo	Judo	73 - 81kg (half-middleweight)

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

	Sport	Discipline	Event
8837	Judo	Judo	60 - 66kg (half-lightweight)
8839	Judo	Judo	66 - 73kg (lightweight)
8845	Judo	Judo	+ 100kg (heavyweight)
8854	Judo	Judo	81 - 90kg (middleweight)
8887	Rowing	Rowing	lightweight double sculls (2x)
8897	Rowing	Rowing	lightweight coxless four (4-)
9032	Sailing	Sailing	board (Mistral)
9055	Sailing	Sailing	single-handed dinghy open (Laser)
9079	Shooting	Shooting	double trap (120 targets)
9106	Shooting	Shooting	double trap (150 targets)
9123	Softball	Softball	softball
9204	Volleyball	Beach volley.	beach volleyball
9288	Weightlifting	Weightlifting	76 - 83kg, total (light-heavyweight)
9289	Weightlifting	Weightlifting	64 - 70kg, total (lightweight)
9290	Weightlifting	Weightlifting	99 - 108kg, total (heavyweight)
9292	Weightlifting	Weightlifting	70 - 76kg, total (middleweight)
9294	Weightlifting	Weightlifting	- 54kg, total (flyweight)
9295	Weightlifting	Weightlifting	59 - 64kg, total (featherweight)
9296	Weightlifting	Weightlifting	91 - 99kg, total (first-heavyweight)
9297	Weightlifting	Weightlifting	83 - 91kg, total (middle-heavyweight)
9298	Weightlifting	Weightlifting	+ 108kg, total (super heavyweight)
9309	Weightlifting	Weightlifting	54 - 59kg, total (bantamweight)
9379	Aquatics	Diving	synchronized diving 10m platform
9380	Aquatics	Diving	synchronized diving 3m springboard
9762	Athletics	Athletics	20km race walk
10232	Cycling	Cycling Track	Olympic Sprint
10236	Cycling	Cycling Track	500m time trial
10238	Cycling	Cycling Track	Madison
10239	Cycling	Cycling Track	Keirin
10602	Gymnastics	Trampoline	individual
10796	Judo	Judo	+ 78kg (heavyweight)
10797	Judo	Judo	57 - 63kg (half-middleweight)
10800	Judo	Judo	63 - 70kg (middleweight)
10802	Judo	Judo	52 - 57kg (lightweight)
10818	Judo	Judo	70 - 78kg (half-heavyweight)

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

	Sport	Discipline	Event
11006	Sailing	Sailing	49er - Skiff
11068	Shooting	Shooting	10m running target (30+30 shots)
11071	Shooting	Shooting	trap (75 targets)
11085	Shooting	Shooting	skeet (75 targets)
11168	Taekwondo	Taekwondo	58 - 68 kg
11169	Taekwondo	Taekwondo	+ 67 kg
11170	Taekwondo	Taekwondo	+ 80 kg
11171	Taekwondo	Taekwondo	57 - 67 kg
11172	Taekwondo	Taekwondo	- 49 kg
11177	Taekwondo	Taekwondo	68 - 80 kg
11179	Taekwondo	Taekwondo	49 - 57 kg
11180	Taekwondo	Taekwondo	- 58 kg
11210	Triathlon	Triathlon	Individual
11300	Weightlifting	Weightlifting	48kg
11301	Weightlifting	Weightlifting	75kg
11302	Weightlifting	Weightlifting	63kg
11305	Weightlifting	Weightlifting	+ 105kg
11306	Weightlifting	Weightlifting	+ 75kg
11309	Weightlifting	Weightlifting	58kg
11310	Weightlifting	Weightlifting	53kg
11313	Weightlifting	Weightlifting	77kg
11315	Weightlifting	Weightlifting	62kg
11316	Weightlifting	Weightlifting	94kg
11319	Weightlifting	Weightlifting	69kg
11321	Weightlifting	Weightlifting	56kg
11322	Weightlifting	Weightlifting	85kg
11323	Weightlifting	Weightlifting	105kg
11345	Wrestling	Wrestling Free.	54 - 58kg
11346	Wrestling	Wrestling Free.	97 - 130kg
11348	Wrestling	Wrestling Free.	58 - 63kg
11349	Wrestling	Wrestling Free.	69 - 76kg
11350	Wrestling	Wrestling Free.	76 - 85kg
11351	Wrestling	Wrestling Free.	85 - 97kg
11353	Wrestling	Wrestling Free.	48 - 54kg
11354	Wrestling	Wrestling Free.	63 - 69kg

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

	Sport	Discipline	Event
11369	Wrestling	Wrestling Gre-R	63 - 69kg
11370	Wrestling	Wrestling Gre-R	97 - 130kg
11371	Wrestling	Wrestling Gre-R	85 - 97kg
11372	Wrestling	Wrestling Gre-R	48 - 54kg
11374	Wrestling	Wrestling Gre-R	76 - 85kg
11375	Wrestling	Wrestling Gre-R	69 - 76kg
11379	Wrestling	Wrestling Gre-R	58 - 63kg
11381	Wrestling	Wrestling Gre-R	54 - 58kg
12099	Boxing	Boxing	60 - 64 kg
12101	Boxing	Boxing	69 - 75 kg
12106	Boxing	Boxing	48kg (light flywieght)
12113	Boxing	Boxing	64 - 69 kg
12244	Cycling	Cycling Track	Team Sprint
12799	Judo	Judo	- 48 kg
12995	Sailing	Sailing	Yngling - Keelboat
13002	Sailing	Sailing	Star - Keelboat
13337	Wrestling	Wrestling Free.	60 - 66kg
13338	Wrestling	Wrestling Free.	48 - 55kg
13339	Wrestling	Wrestling Free.	96 - 120kg
13340	Wrestling	Wrestling Free.	- 55kg
13341	Wrestling	Wrestling Free.	- 48kg
13342	Wrestling	Wrestling Free.	55 - 60kg
13343	Wrestling	Wrestling Free.	63 - 72kg
13344	Wrestling	Wrestling Free.	55 - 63kg
13351	Wrestling	Wrestling Free.	74 - 84kg
13353	Wrestling	Wrestling Free.	84 - 96kg
13357	Wrestling	Wrestling Free.	66 - 74kg
13370	Wrestling	Wrestling Gre-R	55 - 60kg
13372	Wrestling	Wrestling Gre-R	74 - 84kg
13373	Wrestling	Wrestling Gre-R	96 - 120kg
13374	Wrestling	Wrestling Gre-R	60 - 66kg
13377	Wrestling	Wrestling Gre-R	84 - 96kg
13378	Wrestling	Wrestling Gre-R	- 55kg
13380	Wrestling	Wrestling Gre-R	66 - 74kg
13433	Aquatics	Swimming	marathon 10km

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

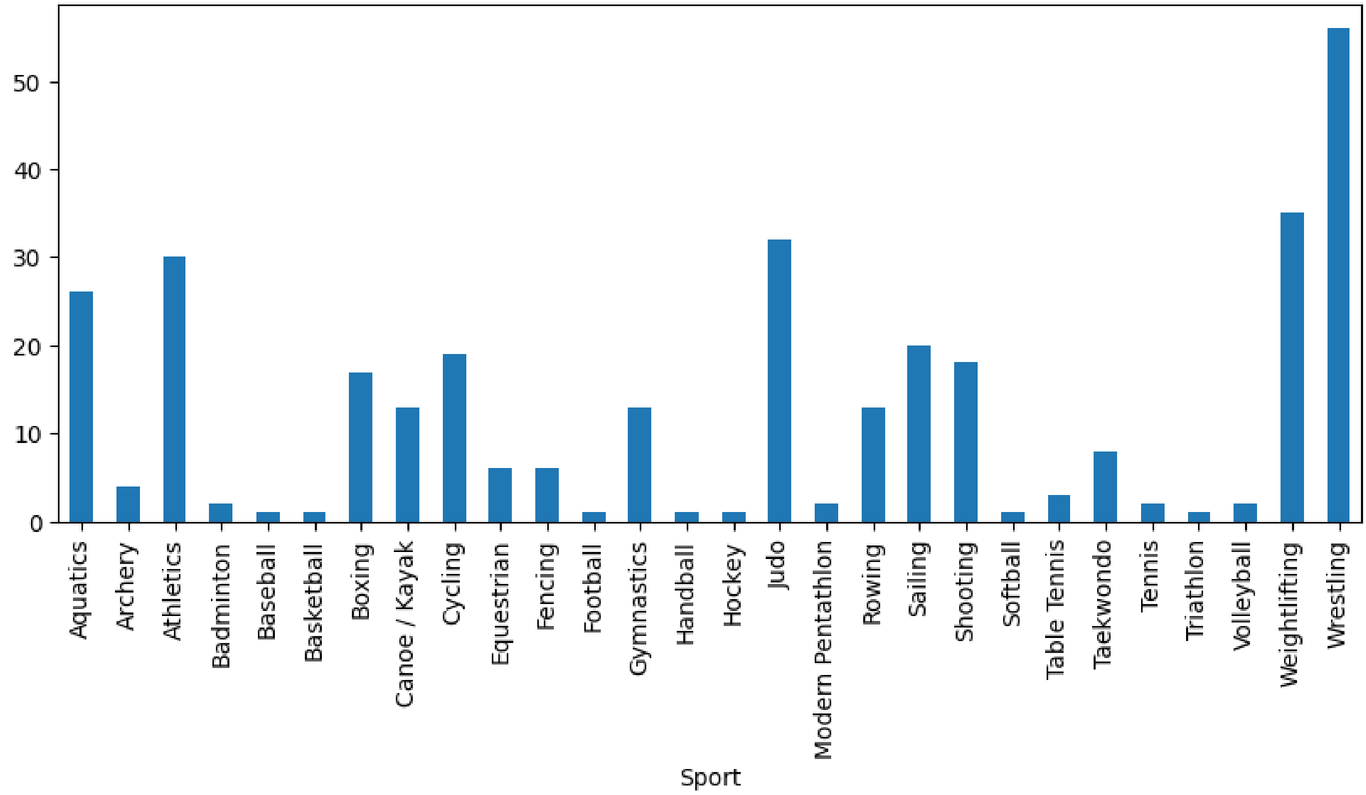
	Sport	Discipline	Event
14235	Cycling	BMX	individual
14238	Cycling	BMX	Individual
15016	Sailing	Sailing	Laser Radial - One Person Dinghy
15020	Sailing	Sailing	Finn - Heavyweight Dinghy
15026	Sailing	Sailing	Laser - One Person Dinghy
15029	Sailing	Sailing	RS:X - Windsurfer
15153	Table Tennis	Table Tennis	team

In [ ]:

```
q3_data = q3_data.groupby(['Sport'])['Sport'].size()
plt.figure(figsize = (10,4))
q3_data.plot.bar(x = 'Sport', y = 'Count')
```

Out[ ]:

```
<Axes: xlabel='Sport'>
```



Ans. Sports with most events are Wrestling, Weightlifting and Judo. Total number of unique events are held: 334

Q4. Put some light on gender ratio in winning teams?

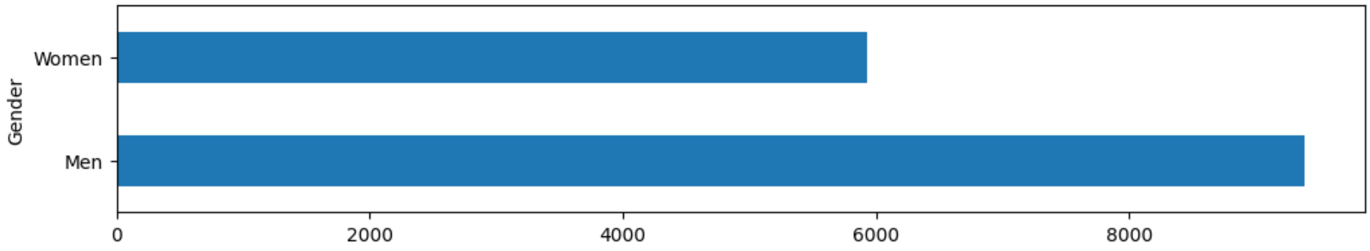
In [ ]:

```
q5_data = data.groupby(['Gender'])['Gender'].count()
plt.figure(figsize = (12,2))
q5_data.plot.barh(x = 'Athlete', y = 'Count')
```

Out[ ]:

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

```
<Axes: ylabel='Gender'>
```



It seems that there are some events which are made only for male

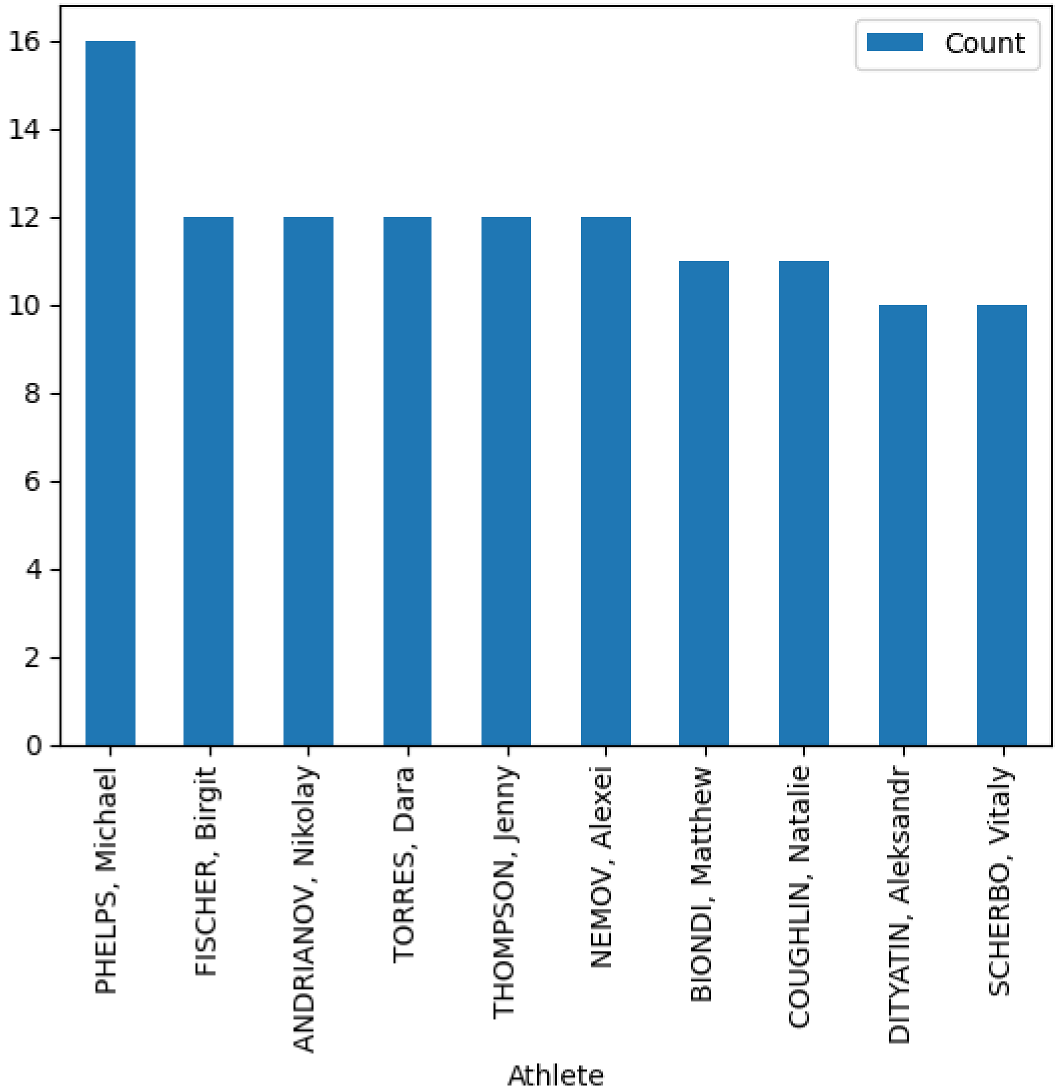
Q5. Which Athlete has win most medal from given period?

In [ ]:

```
q4_data = data.groupby(['Athlete'])['Athlete'].count().reset_index(name ='Count').sort_v
q4_data = q4_data[:10]
q4_data.plot.bar(x = 'Athlete', y = 'Count')
```

Out[ ]:

```
<Axes: xlabel='Athlete'>
```



Ans. So Michael Phelps won 16 mdeal durin 1976 to 2008. Clearly mindblowing record !!

In [ ]:

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

```
q5_data = data[['Event', 'Gender']]
q5_data = q5_data.groupby(['Event','Gender'])['Gender'].count()
q5_data
```

Out[ ]:

		Gender
Event	**Gender**	
+ 100kg (heavyweight)	**Men**	16
+ 100kg (super heavyweight)	**Men**	18
+ 105kg	**Men**	9
+ 108kg, total (super heavyweight)	**Men**	3
+ 110kg, total (super heavyweight)	**Men**	15
+ 67 kg	**Women**	10
+ 72kg (heavyweight)	**Women**	8
+ 75kg	**Women**	9
+ 78kg (heavyweight)	**Women**	12
+ 80 kg	**Men**	10
+ 81kg (heavyweight)	**Men**	8
+ 91kg (super heavyweight)	**Men**	28
+ 93kg (heavyweight)	**Men**	4
+ 95kg (heavyweight)	**Men**	16
- 48 kg	**Women**	8
- 48kg	**Women**	7
- 48kg (extra-lightweight)	**Women**	12
- 48kg (light-flyweight)	**Men**	64
- 49 kg	**Women**	10
- 52kg, total (flyweight)	**Men**	15
- 54kg, total (flyweight)	**Men**	3
- 55kg	**Men**	14
- 56kg, total (bantamweight)	**Men**	21
- 58 kg	**Men**	10
- 60 kg	**Men**	32
- 63kg (lightweight)	**Men**	4
100 - 110kg, total (heavyweight)	**Men**	12
100 - 130kg (super heavyweight)	**Men**	18
10000m	**Men**	27
	Women	18

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

		Gender
Event	**Gender**	
10000m walk	**Women**	6
100m	**Men**	27
	Women	27
100m backstroke	**Men**	28
	Women	27
100m breaststroke	**Men**	27
	Women	27
100m butterfly	**Men**	27
	Women	27
100m freestyle	**Men**	28
	Women	28
100m hurdles	**Women**	28
105kg	**Men**	9
10m air pistol (40 shots)	**Women**	18
10m air pistol (60 shots)	**Men**	18
10m air rifle (40 shots)	**Women**	21
10m air rifle (60 shots)	**Men**	21
10m platform	**Men**	27
	Women	27
10m running target (30+30 shots)	**Men**	6
110m hurdles	**Men**	27
1500m	**Men**	27
	Women	27
1500m freestyle	**Men**	27
1km time trial	**Men**	24
200m	**Men**	27
	Women	27
200m backstroke	**Men**	27
	Women	28
200m breaststroke	**Men**	27
	Women	27
200m butterfly	**Men**	27
	Women	27
200m freestyle	**Men**	27

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

Event	Gender	Gender
	Women	27
200m individual medley	Men	21
	Women	21
20km race walk	Women	9
20km walk	Men	27
25m pistol (30+30 shots)	Women	21
25m rapid fire pistol (60 shots)	Men	27
3000m	Women	9
3000m steeplechase	Men	27
	Women	3
3m springboard	Men	27
	Women	27
400m	Men	27
	Women	27
400m freestyle	Men	27
	Women	27
400m hurdles	Men	27
	Women	21
400m individual medley	Men	27
	Women	27
470 - Two Person Dinghy	Men	54
	Women	36
48 - 51kg (flyweight)	Men	36
48 - 52kg (flyweight)	Men	36
48 - 52kg (half-lightweight)	Women	20
48 - 54kg	Men	6
48 - 55kg	Women	7
48kg	Women	9
48kg (light flywieght)	Men	8
49 - 57 kg	Women	10
49er - Skiff	Men	18
4x100m freestyle relay	Men	110
	Women	131
4x100m medley relay	Men	144

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

		Gender
Event	Gender	
	Women	147
4x100m relay	Men	116
	Women	116
4x200m freestyle relay	Men	133
	Women	74
4x400m relay	Men	119
	Women	122
5000m	Men	27
	Women	12
500m time trial	Women	6
50km walk	Men	24
50m freestyle	Men	18
	Women	19
50m pistol (60 shots)	Men	27
50m rifle 3 positions (3x20 shots)	Women	21
50m rifle 3 positions (3x40 shots)	Men	26
	Women	1
50m rifle prone (60 shots)	Men	27
50m running target (30+30 shots)	Men	18
51 - 54kg (bantamweight)	Men	36
52 - 56kg (lightweight)	Women	8
52 - 57kg (bantamweight)	Men	36
52 - 57kg (lightweight)	Women	12
53kg	Women	9
54 - 57kg (featherweight)	Men	36
54 - 58kg	Men	6
54 - 59kg, total (bantamweight)	Men	3
55 - 60kg	Men	14
55 - 63kg	Women	7
56 - 60kg, total (featherweight)	Men	15
56 - 61kg (half-middleweight)	Women	8
56kg	Men	3
57 - 60kg (lightweight)	Men	36
57 - 62kg (featherweight)	Men	36

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

		Gender
Event	Gender	
57 - 63kg (half-middleweight)	Women	12
57 - 67 kg	Women	10
58 - 63kg	Men	6
58 - 68 kg	Men	10
58kg	Women	9
59 - 64kg, total (featherweight)	Men	3
60 - 63.5kg (light-welterweight)	Men	28
60 - 64 kg	Men	8
60 - 65kg (half-lightweight)	Men	16
60 - 66kg	Men	14
60 - 66kg (half-lightweight)	Men	16
60 - 67.5kg, total (lightweight)	Men	15
61 - 66kg (middleweight)	Women	8
62 - 68kg (lightweight)	Men	36
62kg	Men	9
63 - 69kg	Men	6
63 - 70kg (half-middleweight)	Men	4
63 - 70kg (middleweight)	Women	12
63 - 72kg	Women	7
63.5 - 67kg (welterweight)	Men	28
63kg	Women	9
64 - 69 kg	Men	8
64 - 70kg, total (lightweight)	Men	3
65 - 71kg (lightweight)	Men	16
66 - 72kg (half-heavyweight)	Women	8
66 - 73kg (lightweight)	Men	16
66 - 74kg	Men	14
67 - 71kg (light-middleweight)	Men	28
67.5 - 75kg, total (middleweight)	Men	15
68 - 74kg (welterweight)	Men	36
68 - 80 kg	Men	10
69 - 75 kg	Men	8
69 - 76kg	Men	6
69kg	Men	9

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

		Gender
Event	**Gender**	
	Women	9
70 - 76kg, total (middleweight)	**Men**	3
70 - 78kg (half-heavyweight)	**Women**	12
70 - 80kg (middleweight)	**Men**	4
71 - 78kg (half-middleweight)	**Men**	16
71-75kg	**Men**	28
73 - 81kg (half-middleweight)	**Men**	16
74 - 82kg (middleweight)	**Men**	36
74 - 84kg	**Men**	13
75 - 81kg (light-heavyweight)	**Men**	36
75 - 82.5kg, total (light-heavyweight)	**Men**	14
75kg	**Women**	9
76 - 83kg, total (light-heavyweight)	**Men**	3
76 - 85kg	**Men**	6
77kg	**Men**	9
78 - 86kg (middleweight)	**Men**	16
80 - 93kg (half-heavyweight)	**Men**	4
800m	**Men**	27
	Women	27
800m freestyle	**Women**	27
81 - 90kg (middleweight)	**Men**	16
81 - 91kg (heavyweight)	**Men**	28
82 - 90kg (light-heavyweight)	**Men**	36
82.5 - 90kg, total (middle-heavyweight)	**Men**	15
83 - 91kg, total (middle-heavyweight)	**Men**	3
84 - 96kg	**Men**	14
85 - 97kg	**Men**	6
85kg	**Men**	9
86 - 95kg (half-heavyweight)	**Men**	16
90 - 100kg (half-heavyweight)	**Men**	16
90 - 100kg (heavyweight)	**Men**	36
90 - 100kg, total (first-heavyweight)	**Men**	12
91 - 110kg, total (heavyweight)	**Men**	3
91 - 99kg, total (first-heavyweight)	**Men**	3

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

		Gender
Event	Gender	
94kg	Men	9
96 - 120kg	Men	14
97 - 130kg	Men	6
99 - 108kg, total (heavyweight)	Men	3
C-1 (canoe single)	Men	15
C-1 1000m (canoe single)	Men	27
C-1 500m (canoe single)	Men	27
C-2 (canoe double)	Men	30
C-2 1000m (canoe double)	Men	54
C-2 500m (canoe double)	Men	54
Finn - Heavyweight Dinghy	Men	3
Individual	Men	12
	Women	9
Individual Pursuit	Men	27
Individual competition	Men	27
	Women	9
K-1 (kayak single)	Men	15
	Women	15
K-1 1000m (kayak single)	Men	27
K-1 500m (kayak single)	Men	27
	Women	27
K-2 1000m (kayak double)	Men	54
K-2 500m (kayak double)	Men	54
	Women	54
K-4 1000m (kayak four)	Men	108
K-4 500m (kayak four)	Women	84
Keirin	Men	9
Laser - One Person Dinghy	Men	3
Laser Radial - One Person Dinghy	Women	3
Madison	Men	18
Olympic Sprint	Men	9
Points Race	Men	21
RS:X - Windsurfer	Men	3
	Women	3

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

		Gender
Event	**Gender**	
Sprint indivual	**Men**	27
Star - Keelboat	**Men**	12
Team Pursuit (4000m)	**Men**	118
Team Sprint	**Men**	18
Team competition	**Men**	45
Tornado - Multihull	**Men**	54
Yngling - Keelboat	**Women**	18
balance beam	**Women**	28
baseball	**Men**	335
basketball	**Men**	323
	Women	323
beach volleyball	**Men**	24
	Women	24
board (Mistral)	**Men**	9
	Women	9
board (division II)	**Men**	3
board (lechner)	**Men**	3
	Women	3
board (windglider)	**Men**	3
coxless four (4-)	**Women**	12
coxless pair (2-)	**Men**	54
cross-country	**Men**	12
	Women	12
decathlon	**Men**	27
discus throw	**Men**	27
	Women	27
double sculls (2x)	**Men**	54
	Women	54
double trap (120 targets)	**Women**	9
double trap (150 targets)	**Men**	12
doubles	**Men**	116
	Women	114
duet	**Women**	36
eight with coxswain (8+)	**Men**	243

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

		Gender
Event	**Gender**	
	Women	243
fleet/match race keelboat open (Soling)	**Men**	63
floor exercises	**Men**	29
	Women	30
flying dutchman	**Men**	30
foil individual	**Men**	27
	Women	27
foil team	**Men**	105
	Women	105
football	**Men**	461
	Women	208
four without coxswain (4-)	**Men**	108
four-oared shell with coxswain (4-)	**Men**	75
	Women	60
group competition	**Women**	72
hammer throw	**Men**	27
	Women	9
handball	**Men**	393
	Women	387
heptathlon	**Women**	21
high jump	**Men**	30
	Women	28
hockey	**Men**	434
	Women	383
horizontal bar	**Men**	31
individual	**Men**	66
	Women	45
individual (FITA Olympic round - 70m)	**Men**	15
	Women	15
individual FITA round	**Men**	12
	Women	12
individual all-round	**Men**	27
	Women	49
individual pursuit	**Women**	15

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

		Gender
Event	Gender	
individual road race	Men	26
	Women	21
individual time trial	Men	12
	Women	12
javelin throw	Men	27
	Women	27
lightweight coxless four (4-)	Men	48
lightweight double sculls (2x)	Men	24
	Women	24
long jump	Men	27
	Women	27
marathon	Men	28
	Women	20
marathon 10km	Men	3
	Women	3
open category	Men	12
pair without coxswain (2-)	Women	54
pair-oared shell with coxswain (2+)	Men	45
parallel bars	Men	29
pentathlon	Women	6
points race	Women	12
pole vault	Men	28
	Women	9
pommel horse	Men	28
quadruple sculls with coxswain (4x)	Women	45
quadruple sculls without coxswain (4x)	Men	108
	Women	72
rings	Men	28
sabre individual	Men	27
	Women	6
sabre team	Men	115
	Women	11
shot put	Men	27
	Women	27

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

		Gender
Event	**Gender**	
single sculls (1x)	**Men**	27
	Women	27
single-handed dinghy (Europe)	**Women**	12
single-handed dinghy (Finn)	**Men**	24
single-handed dinghy open (Laser)	**Men**	9
singles	**Men**	55
	Women	55
skeet (125 targets)	**Men**	26
	Women	1
skeet (75 targets)	**Women**	9
softball	**Women**	180
solo	**Women**	9
sprint	**Women**	18
synchronized diving 10m platform	**Men**	18
	Women	18
synchronized diving 3m springboard	**Men**	18
	Women	18
team	**Men**	233
	Women	226
team (FITA Olympic round - 70m)	**Men**	45
	Women	45
team competition	**Men**	165
	Women	164
team time trial	**Men**	60
teams FITA round	**Men**	9
	Women	9
tempest	**Men**	6
trap (125 targets)	**Men**	27
trap (75 targets)	**Women**	9
triple jump	**Men**	27
	Women	12
two-person keelboat open (Star)	**Men**	36
uneven bars	**Women**	29
vault	**Men**	29

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

		Gender
Event	**Gender**	
	Women	27
volleyball	**Men**	323
	Women	324
water polo	**Men**	338
	Women	117
épée individual	**Men**	27
	Women	12
épée team	**Men**	114
	Women	30

dtype: int64

Ans. So there is a huge difference in number of male winners and female winners implying number of sporting event for male are way more than for female¶ (This bust the myth of someone like me who thought that every sport has both male and female version. But thats not true. Some are reserved for male and some are for female at various year.)

Q6. Which country has win most medal and how many in each year?

In [ ]:

```
q6_data = data[['Year', 'Country', 'Medal']]
q6_data = q6_data.groupby(['Year', 'Country','Medal'])['Country'].count().reset_index(na
q6_data['Medal'] = pd.Categorical(q6_data['Medal'],categories=['Gold', 'Silver', 'Bronze
q6_data = q6_data.sort_values(ascending = [True, True, True],by = ['Year', 'Country','Me
q6_data = q6_data.pivot( index = ['Year','Country'], columns =['Medal'], values = ['Coun
q6_data = q6_data.replace(np.nan, 0)
q6_data['Sum'] = q6_data['Count', 'Bronze'] + q6_data['Count','Gold'] + q6_data['Count',
q6_data = q6_data.sort_values(ascending = [True, False],by =['Year','Sum'])
q6_data.columns = q6_data.columns.droplevel(0)
q6_data.columns = ['Year', 'Country', 'Gold', 'Silver','Bronze', 'Sum']
print(q6_data.Country.unique())
q6_data
```

```
['Soviet Union' 'East Germany' 'United States' 'West Germany' 'Poland'
 'Hungary' 'Romania' 'Japan' 'Bulgaria' 'United Kingdom' 'Italy'
 'New Zealand' 'Australia' 'Cuba' 'Canada' 'France' 'Yugoslavia'
 'Korea, South' 'Pakistan' 'Czechoslovakia' 'Netherlands' 'Sweden'
 'Switzerland' 'Belgium' 'Denmark' 'Finland' 'Norway' 'Spain' 'Brazil'
 'Iran' 'Jamaica' 'Korea, North' 'Mexico' 'Portugal' 'Austria' 'Bermuda*'
 'Mongolia' 'Puerto Rico*' 'Thailand' 'Trinidad and Tobago' 'Venezuela'
 'India' 'Zimbabwe' 'Greece' 'Ethiopia' 'Ireland' 'Tanzania' 'Guyana'
 'Lebanon' 'Uganda' 'China' 'Nigeria' 'Kenya' 'Turkey' 'Algeria' 'Morocco'
 'Cameroon' 'Colombia' "Cote d'Ivoire" 'Dominican Republic' 'Egypt'
 'Iceland' 'Peru' 'Syria' 'Taiwan' 'Zambia' 'Argentina' 'Indonesia'
 'Chile' 'Costa Rica' 'Djibouti' 'Netherlands Antilles*' 'Philippines'
 'Senegal' 'Suriname' 'Virgin Islands*' 'Unified team' 'Germany' 'Croatia'
 'Ghana' 'Lithuania' 'Slovenia' 'Estonia'
 'Independent Olympic Participants (1992)' 'Latvia' 'South Africa'
```

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

```
 'Israel' 'Malaysia' 'Namibia' 'Bahamas' 'Qatar' 'Russia' 'Ukraine'
 'Belarus' 'Czech Republic' 'Kazakhstan' 'Moldova' 'Slovakia' 'Armenia'
 'Georgia' 'Uzbekistan' 'Azerbaijan' 'Burundi' 'Ecuador' 'Hong Kong*'
 'Mozambique' 'Tonga' 'Tunisia' 'Saudi Arabia' 'Barbados' 'Kuwait'
 'Kyrgyzstan' 'Macedonia' 'Sri Lanka' 'Uruguay' 'Vietnam' 'Paraguay'
 'Serbia' 'Eritrea' 'United Arab Emirates' 'Singapore' 'Tajikistan'
 'Afghanistan' 'Mauritius' 'Panama' 'Sudan' 'Togo']
```

Out[ ]:

	Year	Country	Gold	Silver	Bronze	Sum
30	1976	Soviet Union	113.0	93.0	79.0	285.0
10	1976	East Germany	99.0	51.0	42.0	192.0
37	1976	United States	63.0	56.0	36.0	155.0
39	1976	West Germany	21.0	24.0	30.0	75.0
26	1976	Poland	18.0	29.0	26.0	73.0
13	1976	Hungary	14.0	6.0	35.0	55.0
29	1976	Romania	4.0	28.0	23.0	55.0
17	1976	Japan	25.0	6.0	10.0	41.0
5	1976	Bulgaria	8.0	13.0	18.0	39.0
36	1976	United Kingdom	6.0	15.0	11.0	32.0
15	1976	Italy	2.0	25.0	4.0	31.0
23	1976	New Zealand	17.0	1.0	9.0	27.0
0	1976	Australia	0.0	16.0	8.0	24.0
7	1976	Cuba	6.0	4.0	14.0	24.0
6	1976	Canada	0.0	8.0	12.0	20.0
12	1976	France	5.0	7.0	8.0	20.0
40	1976	Yugoslavia	2.0	14.0	3.0	19.0
19	1976	Korea, South	1.0	1.0	15.0	17.0
25	1976	Pakistan	0.0	0.0	16.0	16.0
8	1976	Czechoslovakia	2.0	4.0	9.0	15.0
22	1976	Netherlands	0.0	2.0	13.0	15.0
32	1976	Sweden	9.0	1.0	0.0	10.0
33	1976	Switzerland	1.0	3.0	6.0	10.0
2	1976	Belgium	0.0	3.0	6.0	9.0
9	1976	Denmark	3.0	0.0	5.0	8.0
11	1976	Finland	4.0	2.0	0.0	6.0
24	1976	Norway	2.0	4.0	0.0	6.0
31	1976	Spain	0.0	6.0	0.0	6.0
4	1976	Brazil	0.0	0.0	3.0	3.0
14	1976	Iran	0.0	1.0	1.0	2.0

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

	Year	Country	Gold	Silver	Bronze	Sum
16	1976	Jamaica	1.0	1.0	0.0	2.0
18	1976	Korea, North	1.0	1.0	0.0	2.0
20	1976	Mexico	1.0	0.0	1.0	2.0
27	1976	Portugal	0.0	2.0	0.0	2.0
1	1976	Austria	0.0	0.0	1.0	1.0
3	1976	Bermuda*	0.0	0.0	1.0	1.0
21	1976	Mongolia	0.0	1.0	0.0	1.0
28	1976	Puerto Rico*	0.0	0.0	1.0	1.0
34	1976	Thailand	0.0	0.0	1.0	1.0
35	1976	Trinidad and Tobago	1.0	0.0	0.0	1.0
38	1976	Venezuela	0.0	1.0	0.0	1.0
67	1980	Soviet Union	192.0	127.0	123.0	442.0
49	1980	East Germany	112.0	87.0	61.0	260.0
45	1980	Bulgaria	8.0	46.0	40.0	94.0
66	1980	Romania	7.0	15.0	50.0	72.0
55	1980	Hungary	8.0	13.0	40.0	61.0
75	1980	Yugoslavia	13.0	27.0	17.0	57.0
47	1980	Czechoslovakia	18.0	18.0	16.0	52.0
65	1980	Poland	3.0	25.0	22.0	50.0
73	1980	United Kingdom	5.0	18.0	22.0	45.0
58	1980	Italy	8.0	21.0	8.0	37.0
52	1980	France	18.0	5.0	6.0	29.0
68	1980	Spain	2.0	19.0	3.0	24.0
46	1980	Cuba	8.0	7.0	5.0	20.0
69	1980	Sweden	3.0	6.0	9.0	18.0
56	1980	India	16.0	0.0	0.0	16.0
76	1980	Zimbabwe	16.0	0.0	0.0	16.0
41	1980	Australia	5.0	2.0	5.0	12.0
62	1980	Mexico	0.0	1.0	11.0	12.0
44	1980	Brazil	4.0	0.0	5.0	9.0
51	1980	Finland	3.0	1.0	5.0	9.0
48	1980	Denmark	4.0	2.0	2.0	8.0
64	1980	Netherlands	0.0	1.0	5.0	6.0
42	1980	Austria	1.0	3.0	1.0	5.0
53	1980	Greece	1.0	0.0	4.0	5.0

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

	Year	Country	Gold	Silver	Bronze	Sum
60	1980	Korea, North	0.0	3.0	2.0	5.0
50	1980	Ethiopia	2.0	0.0	2.0	4.0
63	1980	Mongolia	0.0	2.0	2.0	4.0
57	1980	Ireland	0.0	2.0	1.0	3.0
59	1980	Jamaica	0.0	0.0	3.0	3.0
70	1980	Switzerland	2.0	0.0	0.0	2.0
71	1980	Tanzania	0.0	2.0	0.0	2.0
43	1980	Belgium	1.0	0.0	0.0	1.0
54	1980	Guyana	0.0	0.0	1.0	1.0
61	1980	Lebanon	0.0	0.0	1.0	1.0
72	1980	Uganda	0.0	1.0	0.0	1.0
74	1980	Venezuela	0.0	1.0	0.0	1.0
119	1984	United States	168.0	115.0	50.0	333.0
121	1984	West Germany	31.0	74.0	52.0	157.0
110	1984	Romania	40.0	31.0	35.0	106.0
122	1984	Yugoslavia	47.0	5.0	35.0	87.0
83	1984	Canada	19.0	35.0	32.0	86.0
84	1984	China	26.0	13.0	37.0	76.0
118	1984	United Kingdom	9.0	20.0	43.0	72.0
91	1984	France	21.0	16.0	31.0	68.0
95	1984	Italy	29.0	6.0	28.0	63.0
78	1984	Australia	7.0	14.0	29.0	50.0
97	1984	Japan	10.0	8.0	31.0	49.0
99	1984	Korea, South	6.0	29.0	7.0	42.0
102	1984	Netherlands	20.0	6.0	14.0	40.0
81	1984	Brazil	1.0	34.0	2.0	37.0
112	1984	Sweden	3.0	18.0	11.0	32.0
103	1984	New Zealand	16.0	1.0	6.0	23.0
111	1984	Spain	2.0	14.0	3.0	19.0
106	1984	Pakistan	16.0	0.0	0.0	16.0
87	1984	Denmark	0.0	5.0	10.0	15.0
113	1984	Switzerland	0.0	9.0	6.0	15.0
90	1984	Finland	4.0	2.0	6.0	12.0
96	1984	Jamaica	0.0	4.0	2.0	6.0
100	1984	Mexico	2.0	3.0	1.0	6.0

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

	Year	Country	Gold	Silver	Bronze	Sum
80	1984	Belgium	1.0	2.0	2.0	5.0
104	1984	Nigeria	0.0	1.0	4.0	5.0
105	1984	Norway	0.0	1.0	3.0	4.0
79	1984	Austria	1.0	1.0	1.0	3.0
98	1984	Kenya	1.0	0.0	2.0	3.0
108	1984	Portugal	1.0	0.0	2.0	3.0
117	1984	Turkey	0.0	0.0	3.0	3.0
120	1984	Venezuela	0.0	0.0	3.0	3.0
77	1984	Algeria	0.0	0.0	2.0	2.0
92	1984	Greece	0.0	1.0	1.0	2.0
101	1984	Morocco	2.0	0.0	0.0	2.0
109	1984	Puerto Rico*	0.0	1.0	1.0	2.0
82	1984	Cameroon	0.0	0.0	1.0	1.0
85	1984	Colombia	0.0	1.0	0.0	1.0
86	1984	Cote d'Ivoire	0.0	1.0	0.0	1.0
88	1984	Dominican Republic	0.0	0.0	1.0	1.0
89	1984	Egypt	0.0	1.0	0.0	1.0
93	1984	Iceland	0.0	0.0	1.0	1.0
94	1984	Ireland	0.0	1.0	0.0	1.0
107	1984	Peru	0.0	1.0	0.0	1.0
114	1984	Syria	0.0	1.0	0.0	1.0
115	1984	Taiwan	0.0	0.0	1.0	1.0
116	1984	Thailand	0.0	1.0	0.0	1.0
123	1984	Zambia	0.0	0.0	1.0	1.0
164	1988	Soviet Union	134.0	65.0	95.0	294.0
172	1988	United States	77.0	64.0	52.0	193.0
138	1988	East Germany	75.0	52.0	47.0	174.0
174	1988	West Germany	32.0	37.0	44.0	113.0
149	1988	Korea, South	28.0	37.0	12.0	77.0
175	1988	Yugoslavia	15.0	27.0	21.0	63.0
132	1988	China	6.0	16.0	31.0	53.0
171	1988	United Kingdom	22.0	16.0	15.0	53.0
162	1988	Romania	8.0	30.0	13.0	51.0
142	1988	Hungary	20.0	9.0	15.0	44.0
153	1988	Netherlands	3.0	5.0	36.0	44.0

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

	Year	Country	Gold	Silver	Bronze	Sum
129	1988	Bulgaria	10.0	14.0	17.0	41.0
125	1988	Australia	18.0	6.0	10.0	34.0
140	1988	France	12.0	4.0	13.0	29.0
145	1988	Italy	11.0	10.0	8.0	29.0
128	1988	Brazil	1.0	22.0	5.0	28.0
155	1988	New Zealand	4.0	4.0	16.0	24.0
156	1988	Norway	2.0	21.0	0.0	23.0
130	1988	Canada	4.0	5.0	12.0	21.0
160	1988	Poland	2.0	9.0	10.0	21.0
147	1988	Japan	4.0	3.0	13.0	20.0
167	1988	Sweden	0.0	5.0	11.0	16.0
124	1988	Argentina	0.0	1.0	12.0	13.0
158	1988	Peru	0.0	12.0	0.0	12.0
135	1988	Czechoslovakia	3.0	4.0	3.0	10.0
148	1988	Kenya	5.0	2.0	2.0	9.0
168	1988	Switzerland	0.0	6.0	2.0	8.0
136	1988	Denmark	3.0	1.0	3.0	7.0
146	1988	Jamaica	0.0	5.0	0.0	5.0
165	1988	Spain	1.0	2.0	2.0	5.0
139	1988	Finland	1.0	1.0	2.0	4.0
143	1988	Indonesia	0.0	3.0	0.0	3.0
152	1988	Morocco	1.0	0.0	2.0	3.0
127	1988	Belgium	0.0	0.0	2.0	2.0
150	1988	Mexico	0.0	0.0	2.0	2.0
170	1988	Turkey	1.0	1.0	0.0	2.0
126	1988	Austria	1.0	0.0	0.0	1.0
131	1988	Chile	0.0	1.0	0.0	1.0
133	1988	Colombia	0.0	0.0	1.0	1.0
134	1988	Costa Rica	0.0	1.0	0.0	1.0
137	1988	Djibouti	0.0	0.0	1.0	1.0
141	1988	Greece	0.0	0.0	1.0	1.0
144	1988	Iran	0.0	1.0	0.0	1.0
151	1988	Mongolia	0.0	0.0	1.0	1.0
154	1988	Netherlands Antilles*	0.0	1.0	0.0	1.0
157	1988	Pakistan	0.0	0.0	1.0	1.0

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

	Year	Country	Gold	Silver	Bronze	Sum
159	1988	Philippines	0.0	0.0	1.0	1.0
161	1988	Portugal	1.0	0.0	0.0	1.0
163	1988	Senegal	0.0	1.0	0.0	1.0
166	1988	Suriname	1.0	0.0	0.0	1.0
169	1988	Thailand	0.0	0.0	1.0	1.0
173	1988	Virgin Islands*	0.0	1.0	0.0	1.0
239	1992	United States	89.0	50.0	85.0	224.0
237	1992	Unified team	92.0	65.0	66.0	223.0
195	1992	Germany	81.0	57.0	60.0	198.0
185	1992	China	18.0	46.0	19.0	83.0
188	1992	Cuba	44.0	13.0	14.0	71.0
230	1992	Spain	44.0	20.0	2.0	66.0
178	1992	Australia	14.0	27.0	16.0	57.0
194	1992	France	9.0	5.0	43.0	57.0
227	1992	Romania	8.0	31.0	14.0	53.0
238	1992	United Kingdom	8.0	3.0	39.0	50.0
209	1992	Korea, South	28.0	5.0	16.0	49.0
206	1992	Japan	3.0	8.0	36.0	47.0
204	1992	Italy	22.0	10.0	14.0	46.0
198	1992	Hungary	14.0	23.0	8.0	45.0
184	1992	Canada	27.0	5.0	12.0	44.0
224	1992	Poland	5.0	18.0	19.0	42.0
232	1992	Sweden	1.0	27.0	7.0	35.0
217	1992	Netherlands	5.0	20.0	8.0	33.0
220	1992	Norway	2.0	20.0	1.0	23.0
234	1992	Taiwan	0.0	20.0	0.0	20.0
183	1992	Bulgaria	3.0	7.0	7.0	17.0
221	1992	Pakistan	0.0	0.0	16.0	16.0
187	1992	Croatia	0.0	12.0	3.0	15.0
218	1992	New Zealand	1.0	9.0	5.0	15.0
182	1992	Brazil	13.0	1.0	0.0	14.0
190	1992	Denmark	3.0	2.0	9.0	14.0
196	1992	Ghana	0.0	0.0	13.0	13.0
211	1992	Lithuania	1.0	0.0	12.0	13.0
219	1992	Nigeria	0.0	7.0	4.0	11.0

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

	Year	Country	Gold	Silver	Bronze	Sum
208	1992	Korea, North	4.0	0.0	6.0	10.0
189	1992	Czechoslovakia	4.0	3.0	1.0	8.0
207	1992	Kenya	2.0	4.0	2.0	8.0
193	1992	Finland	1.0	4.0	2.0	7.0
179	1992	Austria	0.0	6.0	0.0	6.0
200	1992	Indonesia	2.0	3.0	1.0	6.0
228	1992	Slovenia	0.0	0.0	6.0	6.0
236	1992	Turkey	2.0	2.0	2.0	6.0
205	1992	Jamaica	0.0	3.0	1.0	4.0
181	1992	Belgium	0.0	1.0	2.0	3.0
191	1992	Estonia	1.0	0.0	2.0	3.0
192	1992	Ethiopia	1.0	0.0	2.0	3.0
199	1992	Independent Olympic Participants (1992)	0.0	1.0	2.0	3.0
201	1992	Iran	0.0	1.0	2.0	3.0
210	1992	Latvia	0.0	2.0	1.0	3.0
215	1992	Morocco	1.0	1.0	1.0	3.0
229	1992	South Africa	0.0	3.0	0.0	3.0
176	1992	Algeria	1.0	0.0	1.0	2.0
177	1992	Argentina	0.0	0.0	2.0	2.0
197	1992	Greece	2.0	0.0	0.0	2.0
202	1992	Ireland	1.0	1.0	0.0	2.0
203	1992	Israel	0.0	1.0	1.0	2.0
212	1992	Malaysia	0.0	0.0	2.0	2.0
214	1992	Mongolia	0.0	0.0	2.0	2.0
216	1992	Namibia	0.0	2.0	0.0	2.0
180	1992	Bahamas	0.0	0.0	1.0	1.0
186	1992	Colombia	0.0	0.0	1.0	1.0
213	1992	Mexico	0.0	1.0	0.0	1.0
222	1992	Peru	0.0	1.0	0.0	1.0
223	1992	Philippines	0.0	0.0	1.0	1.0
225	1992	Puerto Rico*	0.0	0.0	1.0	1.0
226	1992	Qatar	0.0	0.0	1.0	1.0
231	1992	Suriname	0.0	0.0	1.0	1.0
233	1992	Switzerland	1.0	0.0	0.0	1.0
235	1992	Thailand	0.0	0.0	1.0	1.0

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

	Year	Country	Gold	Silver	Bronze	Sum
315	1996	United States	160.0	48.0	52.0	260.0
243	1996	Australia	32.0	16.0	84.0	132.0
264	1996	Germany	42.0	35.0	47.0	124.0
298	1996	Russia	36.0	45.0	34.0	115.0
253	1996	China	19.0	74.0	17.0	110.0
289	1996	Netherlands	38.0	9.0	26.0	73.0
273	1996	Italy	19.0	24.0	28.0	71.0
302	1996	Spain	24.0	21.0	22.0	67.0
279	1996	Korea, South	10.0	49.0	7.0	66.0
249	1996	Brazil	5.0	15.0	44.0	64.0
256	1996	Cuba	39.0	8.0	10.0	57.0
252	1996	Canada	8.0	31.0	12.0	51.0
262	1996	France	21.0	10.0	20.0	51.0
267	1996	Hungary	8.0	9.0	26.0	43.0
275	1996	Japan	3.0	26.0	14.0	43.0
297	1996	Romania	13.0	10.0	15.0	38.0
313	1996	Ukraine	10.0	5.0	19.0	34.0
303	1996	Sweden	3.0	23.0	5.0	31.0
255	1996	Croatia	16.0	13.0	0.0	29.0
291	1996	Nigeria	19.0	4.0	3.0	26.0
314	1996	United Kingdom	2.0	15.0	9.0	26.0
317	1996	Yugoslavia	1.0	12.0	13.0	26.0
292	1996	Norway	2.0	3.0	20.0	25.0
258	1996	Denmark	22.0	1.0	1.0	24.0
247	1996	Belarus	1.0	6.0	16.0	23.0
250	1996	Bulgaria	3.0	12.0	6.0	21.0
294	1996	Poland	7.0	7.0	7.0	21.0
241	1996	Argentina	0.0	19.0	1.0	20.0
274	1996	Jamaica	1.0	3.0	12.0	16.0
257	1996	Czech Republic	4.0	5.0	4.0	13.0
281	1996	Lithuania	0.0	0.0	12.0	12.0
276	1996	Kazakhstan	3.0	4.0	4.0	11.0
304	1996	Switzerland	5.0	6.0	0.0	11.0
290	1996	New Zealand	3.0	2.0	4.0	9.0
265	1996	Greece	4.0	4.0	0.0	8.0

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

	Year	Country	Gold	Silver	Bronze	Sum
277	1996	Kenya	1.0	4.0	3.0	8.0
248	1996	Belgium	2.0	2.0	2.0	6.0
269	1996	Indonesia	2.0	1.0	3.0	6.0
311	1996	Turkey	4.0	1.0	1.0	6.0
246	1996	Bahamas	0.0	5.0	0.0	5.0
278	1996	Korea, North	2.0	1.0	2.0	5.0
301	1996	South Africa	3.0	1.0	1.0	5.0
261	1996	Finland	1.0	2.0	1.0	4.0
271	1996	Ireland	3.0	0.0	1.0	4.0
240	1996	Algeria	2.0	0.0	1.0	3.0
244	1996	Austria	0.0	1.0	2.0	3.0
260	1996	Ethiopia	2.0	0.0	1.0	3.0
270	1996	Iran	1.0	1.0	1.0	3.0
282	1996	Malaysia	0.0	2.0	1.0	3.0
284	1996	Moldova	0.0	2.0	1.0	3.0
295	1996	Portugal	1.0	0.0	2.0	3.0
299	1996	Slovakia	1.0	1.0	1.0	3.0
242	1996	Armenia	1.0	1.0	0.0	2.0
263	1996	Georgia	0.0	0.0	2.0	2.0
286	1996	Morocco	0.0	0.0	2.0	2.0
288	1996	Namibia	0.0	2.0	0.0	2.0
300	1996	Slovenia	0.0	2.0	0.0	2.0
307	1996	Thailand	1.0	0.0	1.0	2.0
309	1996	Trinidad and Tobago	0.0	0.0	2.0	2.0
316	1996	Uzbekistan	0.0	1.0	1.0	2.0
245	1996	Azerbaijan	0.0	1.0	0.0	1.0
251	1996	Burundi	1.0	0.0	0.0	1.0
254	1996	Costa Rica	1.0	0.0	0.0	1.0
259	1996	Ecuador	1.0	0.0	0.0	1.0
266	1996	Hong Kong*	1.0	0.0	0.0	1.0
268	1996	India	0.0	0.0	1.0	1.0
272	1996	Israel	0.0	0.0	1.0	1.0
280	1996	Latvia	0.0	1.0	0.0	1.0
283	1996	Mexico	0.0	0.0	1.0	1.0
285	1996	Mongolia	0.0	0.0	1.0	1.0

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

	Year	Country	Gold	Silver	Bronze	Sum
287	1996	Mozambique	0.0	0.0	1.0	1.0
293	1996	Philippines	0.0	1.0	0.0	1.0
296	1996	Puerto Rico*	0.0	0.0	1.0	1.0
305	1996	Syria	1.0	0.0	0.0	1.0
306	1996	Taiwan	0.0	1.0	0.0	1.0
308	1996	Tonga	0.0	1.0	0.0	1.0
310	1996	Tunisia	0.0	0.0	1.0	1.0
312	1996	Uganda	0.0	0.0	1.0	1.0
318	1996	Zambia	0.0	1.0	0.0	1.0
394	2000	United States	130.0	66.0	52.0	248.0
379	2000	Russia	66.0	68.0	54.0	188.0
322	2000	Australia	60.0	69.0	54.0	183.0
346	2000	Germany	31.0	23.0	65.0	119.0
334	2000	China	39.0	23.0	17.0	79.0
371	2000	Netherlands	27.0	29.0	23.0	79.0
361	2000	Korea, South	12.0	26.0	35.0	73.0
338	2000	Cuba	22.0	35.0	12.0	69.0
344	2000	France	22.0	30.0	14.0	66.0
355	2000	Italy	22.0	14.0	29.0	65.0
393	2000	United Kingdom	22.0	20.0	13.0	55.0
348	2000	Hungary	25.0	24.0	4.0	53.0
329	2000	Brazil	0.0	12.0	36.0	48.0
378	2000	Romania	27.0	6.0	13.0	46.0
374	2000	Norway	21.0	4.0	19.0	44.0
357	2000	Japan	5.0	30.0	8.0	43.0
384	2000	Spain	3.0	20.0	20.0	43.0
392	2000	Ukraine	3.0	20.0	12.0	35.0
386	2000	Sweden	4.0	20.0	8.0	32.0
332	2000	Canada	4.0	4.0	23.0	31.0
398	2000	Yugoslavia	12.0	1.0	13.0	26.0
340	2000	Denmark	18.0	3.0	4.0	25.0
375	2000	Poland	7.0	10.0	7.0	24.0
356	2000	Jamaica	0.0	14.0	9.0	23.0
327	2000	Belarus	3.0	8.0	11.0	22.0
320	2000	Argentina	0.0	17.0	3.0	20.0

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

	Year	Country	Gold	Silver	Bronze	Sum
331	2000	Cameroon	18.0	0.0	0.0	18.0
333	2000	Chile	0.0	0.0	18.0	18.0
347	2000	Greece	4.0	6.0	8.0	18.0
365	2000	Lithuania	2.0	0.0	15.0	17.0
387	2000	Switzerland	1.0	11.0	2.0	14.0
330	2000	Bulgaria	5.0	6.0	2.0	13.0
337	2000	Croatia	1.0	0.0	9.0	10.0
339	2000	Czech Republic	2.0	3.0	4.0	9.0
342	2000	Ethiopia	4.0	1.0	3.0	8.0
351	2000	Indonesia	2.0	4.0	2.0	8.0
373	2000	Nigeria	0.0	8.0	0.0	8.0
328	2000	Belgium	0.0	3.0	4.0	7.0
358	2000	Kazakhstan	3.0	4.0	0.0	7.0
359	2000	Kenya	2.0	3.0	2.0	7.0
325	2000	Bahamas	6.0	0.0	0.0	6.0
345	2000	Georgia	0.0	0.0	6.0	6.0
367	2000	Mexico	1.0	2.0	3.0	6.0
381	2000	Slovakia	2.0	3.0	1.0	6.0
319	2000	Algeria	1.0	1.0	3.0	5.0
343	2000	Finland	3.0	1.0	1.0	5.0
369	2000	Morocco	0.0	1.0	4.0	5.0
383	2000	South Africa	0.0	2.0	3.0	5.0
388	2000	Taiwan	0.0	1.0	4.0	5.0
391	2000	Turkey	3.0	0.0	2.0	5.0
323	2000	Austria	3.0	1.0	0.0	4.0
352	2000	Iran	3.0	0.0	1.0	4.0
360	2000	Korea, North	0.0	1.0	3.0	4.0
372	2000	New Zealand	1.0	0.0	3.0	4.0
396	2000	Uzbekistan	1.0	1.0	2.0	4.0
324	2000	Azerbaijan	2.0	0.0	1.0	3.0
341	2000	Estonia	1.0	0.0	2.0	3.0
364	2000	Latvia	1.0	1.0	1.0	3.0
382	2000	Slovenia	3.0	0.0	0.0	3.0
389	2000	Thailand	1.0	0.0	2.0	3.0
336	2000	Costa Rica	0.0	0.0	2.0	2.0

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

	Year	Country	Gold	Silver	Bronze	Sum
368	2000	Moldova	0.0	1.0	1.0	2.0
376	2000	Portugal	0.0	0.0	2.0	2.0
380	2000	Saudi Arabia	0.0	1.0	1.0	2.0
390	2000	Trinidad and Tobago	0.0	1.0	1.0	2.0
321	2000	Armenia	0.0	0.0	1.0	1.0
326	2000	Barbados	0.0	0.0	1.0	1.0
335	2000	Colombia	1.0	0.0	0.0	1.0
349	2000	Iceland	0.0	0.0	1.0	1.0
350	2000	India	0.0	0.0	1.0	1.0
353	2000	Ireland	0.0	1.0	0.0	1.0
354	2000	Israel	0.0	0.0	1.0	1.0
362	2000	Kuwait	0.0	0.0	1.0	1.0
363	2000	Kyrgyzstan	0.0	0.0	1.0	1.0
366	2000	Macedonia	0.0	0.0	1.0	1.0
370	2000	Mozambique	1.0	0.0	0.0	1.0
377	2000	Qatar	0.0	0.0	1.0	1.0
385	2000	Sri Lanka	0.0	1.0	0.0	1.0
395	2000	Uruguay	0.0	1.0	0.0	1.0
397	2000	Vietnam	0.0	1.0	0.0	1.0
469	2004	United States	116.0	75.0	73.0	264.0
453	2004	Russia	47.0	49.0	96.0	192.0
400	2004	Australia	49.0	78.0	30.0	157.0
425	2004	Germany	41.0	45.0	63.0	149.0
433	2004	Italy	24.0	39.0	39.0	102.0
411	2004	China	52.0	27.0	15.0	94.0
435	2004	Japan	21.0	20.0	53.0	94.0
445	2004	Netherlands	4.0	50.0	22.0	76.0
414	2004	Cuba	31.0	8.0	22.0	61.0
468	2004	United Kingdom	17.0	25.0	15.0	57.0
423	2004	France	21.0	10.0	22.0	53.0
439	2004	Korea, South	14.0	28.0	10.0	52.0
466	2004	Ukraine	9.0	8.0	31.0	48.0
399	2004	Argentina	26.0	0.0	21.0	47.0
406	2004	Brazil	18.0	19.0	3.0	40.0
428	2004	Hungary	24.0	12.0	4.0	40.0

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

	Year	Country	Gold	Silver	Bronze	Sum
452	2004	Romania	23.0	5.0	11.0	39.0
426	2004	Greece	8.0	18.0	5.0	31.0
416	2004	Denmark	19.0	0.0	10.0	29.0
458	2004	Spain	4.0	15.0	8.0	27.0
413	2004	Croatia	14.0	3.0	3.0	20.0
404	2004	Belarus	2.0	6.0	9.0	17.0
407	2004	Bulgaria	2.0	1.0	14.0	17.0
409	2004	Canada	3.0	10.0	4.0	17.0
449	2004	Paraguay	0.0	17.0	0.0	17.0
454	2004	Serbia	0.0	14.0	0.0	14.0
434	2004	Jamaica	6.0	1.0	6.0	13.0
415	2004	Czech Republic	1.0	6.0	5.0	12.0
450	2004	Poland	4.0	2.0	6.0	12.0
459	2004	Sweden	5.0	5.0	2.0	12.0
455	2004	Slovakia	3.0	2.0	5.0	10.0
457	2004	South Africa	4.0	3.0	3.0	10.0
465	2004	Turkey	3.0	3.0	4.0	10.0
462	2004	Taiwan	2.0	4.0	3.0	9.0
401	2004	Austria	3.0	4.0	1.0	8.0
436	2004	Kazakhstan	1.0	4.0	3.0	8.0
447	2004	Nigeria	0.0	0.0	8.0	8.0
463	2004	Thailand	3.0	1.0	4.0	8.0
421	2004	Ethiopia	2.0	3.0	2.0	7.0
437	2004	Kenya	1.0	4.0	2.0	7.0
448	2004	Norway	5.0	0.0	2.0	7.0
460	2004	Switzerland	1.0	2.0	4.0	7.0
431	2004	Iran	2.0	2.0	2.0	6.0
446	2004	New Zealand	4.0	2.0	0.0	6.0
402	2004	Azerbaijan	1.0	0.0	4.0	5.0
418	2004	Egypt	1.0	1.0	3.0	5.0
430	2004	Indonesia	1.0	1.0	3.0	5.0
438	2004	Korea, North	0.0	4.0	1.0	5.0
456	2004	Slovenia	0.0	2.0	3.0	5.0
470	2004	Uzbekistan	2.0	1.0	2.0	5.0
410	2004	Chile	3.0	0.0	1.0	4.0

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

	Year	Country	Gold	Silver	Bronze	Sum
424	2004	Georgia	2.0	2.0	0.0	4.0
440	2004	Latvia	0.0	4.0	0.0	4.0
442	2004	Mexico	0.0	3.0	1.0	4.0
405	2004	Belgium	1.0	0.0	2.0	3.0
420	2004	Estonia	0.0	1.0	2.0	3.0
441	2004	Lithuania	1.0	2.0	0.0	3.0
444	2004	Morocco	2.0	1.0	0.0	3.0
451	2004	Portugal	0.0	2.0	1.0	3.0
472	2004	Zimbabwe	1.0	1.0	1.0	3.0
403	2004	Bahamas	1.0	0.0	1.0	2.0
412	2004	Colombia	0.0	0.0	2.0	2.0
422	2004	Finland	0.0	2.0	0.0	2.0
427	2004	Hong Kong*	0.0	2.0	0.0	2.0
432	2004	Israel	1.0	0.0	1.0	2.0
471	2004	Venezuela	0.0	0.0	2.0	2.0
408	2004	Cameroon	1.0	0.0	0.0	1.0
417	2004	Dominican Republic	1.0	0.0	0.0	1.0
419	2004	Eritrea	0.0	0.0	1.0	1.0
429	2004	India	0.0	1.0	0.0	1.0
443	2004	Mongolia	0.0	0.0	1.0	1.0
461	2004	Syria	0.0	0.0	1.0	1.0
464	2004	Trinidad and Tobago	0.0	0.0	1.0	1.0
467	2004	United Arab Emirates	1.0	0.0	0.0	1.0
554	2008	United States	125.0	109.0	81.0	315.0
488	2008	China	74.0	53.0	57.0	184.0
477	2008	Australia	31.0	42.0	76.0	149.0
535	2008	Russia	43.0	44.0	56.0	143.0
502	2008	Germany	42.0	16.0	43.0	101.0
517	2008	Korea, South	41.0	11.0	26.0	78.0
553	2008	United Kingdom	31.0	25.0	21.0	77.0
500	2008	France	25.0	23.0	28.0	76.0
483	2008	Brazil	14.0	34.0	27.0	75.0
541	2008	Spain	7.0	48.0	16.0	71.0
527	2008	Netherlands	40.0	18.0	4.0	62.0
475	2008	Argentina	20.0	0.0	31.0	51.0

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

	Year	Country	Gold	Silver	Bronze	Sum
513	2008	Japan	23.0	11.0	17.0	51.0
491	2008	Cuba	2.0	34.0	11.0	47.0
511	2008	Italy	8.0	14.0	20.0	42.0
486	2008	Canada	11.0	13.0	10.0	34.0
552	2008	Ukraine	10.0	5.0	16.0	31.0
481	2008	Belarus	8.0	5.0	17.0	30.0
504	2008	Hungary	16.0	8.0	3.0	27.0
529	2008	Nigeria	0.0	18.0	6.0	24.0
530	2008	Norway	16.0	5.0	1.0	22.0
534	2008	Romania	5.0	1.0	16.0	22.0
532	2008	Poland	6.0	13.0	1.0	20.0
493	2008	Denmark	6.0	6.0	6.0	18.0
512	2008	Jamaica	9.0	3.0	5.0	17.0
536	2008	Serbia	0.0	1.0	14.0	15.0
505	2008	Iceland	0.0	14.0	0.0	14.0
515	2008	Kenya	6.0	4.0	4.0	14.0
528	2008	New Zealand	4.0	2.0	8.0	14.0
514	2008	Kazakhstan	2.0	4.0	7.0	13.0
544	2008	Switzerland	3.0	0.0	8.0	11.0
538	2008	Slovakia	4.0	5.0	1.0	10.0
551	2008	Turkey	1.0	4.0	3.0	8.0
479	2008	Azerbaijan	1.0	2.0	4.0	7.0
492	2008	Czech Republic	3.0	4.0	0.0	7.0
498	2008	Ethiopia	4.0	1.0	2.0	7.0
503	2008	Greece	0.0	3.0	4.0	7.0
507	2008	Indonesia	2.0	2.0	3.0	7.0
543	2008	Sweden	0.0	5.0	2.0	7.0
476	2008	Armenia	0.0	0.0	6.0	6.0
501	2008	Georgia	3.0	0.0	3.0	6.0
516	2008	Korea, North	2.0	1.0	3.0	6.0
555	2008	Uzbekistan	1.0	2.0	3.0	6.0
480	2008	Bahamas	0.0	4.0	1.0	5.0
482	2008	Belgium	1.0	4.0	0.0	5.0
484	2008	Bulgaria	1.0	1.0	3.0	5.0
490	2008	Croatia	0.0	2.0	3.0	5.0

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

	Year	Country	Gold	Silver	Bronze	Sum
499	2008	Finland	1.0	2.0	2.0	5.0
520	2008	Lithuania	0.0	2.0	3.0	5.0
539	2008	Slovenia	1.0	2.0	2.0	5.0
549	2008	Trinidad and Tobago	0.0	5.0	0.0	5.0
523	2008	Mexico	2.0	0.0	2.0	4.0
525	2008	Mongolia	2.0	2.0	0.0	4.0
545	2008	Taiwan	0.0	0.0	4.0	4.0
547	2008	Thailand	2.0	2.0	0.0	4.0
558	2008	Zimbabwe	1.0	3.0	0.0	4.0
478	2008	Austria	0.0	1.0	2.0	3.0
497	2008	Estonia	1.0	2.0	0.0	3.0
506	2008	India	1.0	0.0	2.0	3.0
509	2008	Ireland	0.0	1.0	2.0	3.0
519	2008	Latvia	1.0	1.0	1.0	3.0
537	2008	Singapore	0.0	3.0	0.0	3.0
474	2008	Algeria	0.0	1.0	1.0	2.0
489	2008	Colombia	0.0	1.0	1.0	2.0
494	2008	Dominican Republic	1.0	1.0	0.0	2.0
508	2008	Iran	1.0	0.0	1.0	2.0
518	2008	Kyrgyzstan	0.0	1.0	1.0	2.0
526	2008	Morocco	0.0	1.0	1.0	2.0
533	2008	Portugal	1.0	1.0	0.0	2.0
546	2008	Tajikistan	0.0	1.0	1.0	2.0
473	2008	Afghanistan	0.0	0.0	1.0	1.0
485	2008	Cameroon	1.0	0.0	0.0	1.0
487	2008	Chile	0.0	1.0	0.0	1.0
495	2008	Ecuador	0.0	1.0	0.0	1.0
496	2008	Egypt	0.0	0.0	1.0	1.0
510	2008	Israel	0.0	0.0	1.0	1.0
521	2008	Malaysia	0.0	1.0	0.0	1.0
522	2008	Mauritius	0.0	0.0	1.0	1.0
524	2008	Moldova	0.0	0.0	1.0	1.0
531	2008	Panama	1.0	0.0	0.0	1.0
540	2008	South Africa	0.0	1.0	0.0	1.0
542	2008	Sudan	0.0	1.0	0.0	1.0

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

	Year	Country	Gold	Silver	Bronze	Sum
548	2008	Togo	0.0	0.0	1.0	1.0
550	2008	Tunisia	1.0	0.0	0.0	1.0
556	2008	Venezuela	0.0	0.0	1.0	1.0
557	2008	Vietnam	0.0	1.0	0.0	1.0

Ans. So I created an interactive solution here. Input the country name from above list. And check its performance over year.¶ Note : This may not resemble actual table tally because for eg., a gold in hockey is just one gold in table but here it is 16 gold because sixteen people got it. So it is more like how many people got a medal instead of how many gold medal a country got.

Q7. Can you tell me which country has dominated any particular sport?

In [ ]:

```
q7_data = data.groupby(['Sport','Country'])['Country'].count().reset_index(name ='Count'
q7_data.Sport.unique()
```

Out[ ]:

```
array(['Aquatics', 'Archery', 'Athletics', 'Badminton', 'Baseball',
       'Basketball', 'Boxing', 'Canoe / Kayak', 'Cycling', 'Equestrian',
       'Fencing', 'Football', 'Gymnastics', 'Handball', 'Hockey', 'Judo',
       'Modern Pentathlon', 'Rowing', 'Sailing', 'Shooting', 'Softball',
       'Table Tennis', 'Taekwondo', 'Tennis', 'Triathlon', 'Volleyball',
       'Weightlifting', 'Wrestling'], dtype=object)
```

In [ ]:

```
inp = 'Archery'
try:
  inp = input("Select a Sport from above list")
except:
  print("Input is interrupted")
temp = q7_data[q7_data['Sport'] == inp].head(3)
print(temp)
```

```
Select a Sport from above listArcehery
Empty DataFrame
Columns: [Sport, Country, Count]
Index: []
```

Ans. So Here we have an interactive way to see which country has dominated which sport. For e.g., Netherland and Australia dominated Hockey in the given period. Note : This may not resemble actual table tally because for eg., a gold in hockey is just one gold in table but here it is 16 gold because sixteen people got it. So it is more like how many people got a medal instead of how many gold medal a country got

Q8. Has any athlete changed his or her Event or Discipline or sport and still win the medal?

In [ ]:

```
temp = data[['Athlete','Sport']].drop_duplicates()
temp = temp.groupby(['Athlete'])
for k,v in temp:
    if len(v['Sport'].tolist()) >1:
        print(k,v['Sport'].tolist())
```

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

```
('BELOVA, Irina',) ['Athletics', 'Gymnastics']
('CHEN, Jing',) ['Table Tennis', 'Volleyball']
('DIMITROV, Stefan',) ['Volleyball', 'Weightlifting']
('GAVRILOV, Yuri',) ['Football', 'Handball']
('GONZALEZ, Raul',) ['Athletics', 'Handball']
('KOLESNIKOV, Nikolai',) ['Athletics', 'Weightlifting']
('KOVACS, Istvan',) ['Wrestling', 'Boxing']
('KOVALENKO, Alexandre',) ['Athletics', 'Aquatics']
('KUZNETSOV, Mikhail',) ['Rowing', 'Canoe / Kayak']
('KUZNETSOV, Nikolai',) ['Rowing', 'Cycling']
('LEE, Eun Kyung',) ['Archery', 'Hockey']
('LI, Na',) ['Aquatics', 'Fencing']
('LI, Ting',) ['Aquatics', 'Tennis']
('OVCHINNIKOVA, Elena',) ['Volleyball', 'Aquatics']
('ROMERO, Rebecca',) ['Rowing', 'Cycling']
('THOMPSON, Richard',) ['Baseball', 'Athletics']
('TOMA, Sanda',) ['Rowing', 'Canoe / Kayak']
('WANG, Liping',) ['Football', 'Athletics']
('WELLS, Matthew',) ['Hockey', 'Rowing']
('YANG, Wei',) ['Badminton', 'Gymnastics']
('YOUNG, Tim',) ['Rowing', 'Baseball']
```

Ans. So there has been quite a few player who has changed the sport and still won a medal. Kudos to them !! Note : Here two different person had same name. for eg., Yang Wei from Gymanstic and from Badminton are different player. From thr given data we cannot distinguish between them. So take it with a pinch of salt

Q9. (Follow up of Q6) Elaborate the result and dive into detials.(Pick any 5 country for this

In [ ]:

```python
q9_data = q6_data[['Year', 'Country','Sum']].groupby(['Year']).apply(lambda x : x.nlarge
q9_data = q9_data.pivot( index = ['Year'], columns =['Country'], values = ['Sum']).reset
q9_data.columns = q9_data.columns.droplevel(0)
# q9_data.columns = ['Year', 'Country', 'Gold', 'Silver','Bronze', 'Sum']
q9_data = q9_data.rename(columns={ q9_data.columns[0]: "Year"})
q9_data
# temp =
q6_data.where(q6_data.Country.isin(q9_data.columns)).dropna()[["Year", "Country", "Sum"]
```

```
/tmp/ipython-input-2847804065.py:1: DeprecationWarning: DataFrameGroupBy.apply operated
on the grouping columns. This behavior is deprecated, and in a future version of pandas
the grouping columns will be excluded from the operation. Either pass `include_groups=Fa
lse` to exclude the groupings or explicitly select the grouping columns after groupby to
silence this warning.
  q9_data = q6_data[['Year', 'Country','Sum']].groupby(['Year']).apply(lambda x : x.nlar
gest(5,'Sum'))
```

Out[ ]:

	Year	Country	Sum
30	1976.0	Soviet Union	285.0
10	1976.0	East Germany	192.0
37	1976.0	United States	155.0
39	1976.0	West Germany	75.0
26	1976.0	Poland	73.0
13	1976.0	Hungary	55.0

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

	Year	Country	Sum
29	1976.0	Romania	55.0
5	1976.0	Bulgaria	39.0
15	1976.0	Italy	31.0
0	1976.0	Australia	24.0
7	1976.0	Cuba	24.0
6	1976.0	Canada	20.0
40	1976.0	Yugoslavia	19.0
19	1976.0	Korea, South	17.0
67	1980.0	Soviet Union	442.0
49	1980.0	East Germany	260.0
45	1980.0	Bulgaria	94.0
66	1980.0	Romania	72.0
55	1980.0	Hungary	61.0
75	1980.0	Yugoslavia	57.0
65	1980.0	Poland	50.0
58	1980.0	Italy	37.0
46	1980.0	Cuba	20.0
41	1980.0	Australia	12.0
119	1984.0	United States	333.0
121	1984.0	West Germany	157.0
110	1984.0	Romania	106.0
122	1984.0	Yugoslavia	87.0
83	1984.0	Canada	86.0
84	1984.0	China	76.0
95	1984.0	Italy	63.0
78	1984.0	Australia	50.0
99	1984.0	Korea, South	42.0
164	1988.0	Soviet Union	294.0
172	1988.0	United States	193.0
138	1988.0	East Germany	174.0
174	1988.0	West Germany	113.0
149	1988.0	Korea, South	77.0
175	1988.0	Yugoslavia	63.0
132	1988.0	China	53.0
162	1988.0	Romania	51.0

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

	Year	Country	Sum
142	1988.0	Hungary	44.0
129	1988.0	Bulgaria	41.0
125	1988.0	Australia	34.0
145	1988.0	Italy	29.0
130	1988.0	Canada	21.0
160	1988.0	Poland	21.0
239	1992.0	United States	224.0
237	1992.0	Unified team	223.0
195	1992.0	Germany	198.0
185	1992.0	China	83.0
188	1992.0	Cuba	71.0
178	1992.0	Australia	57.0
227	1992.0	Romania	53.0
209	1992.0	Korea, South	49.0
204	1992.0	Italy	46.0
198	1992.0	Hungary	45.0
184	1992.0	Canada	44.0
224	1992.0	Poland	42.0
183	1992.0	Bulgaria	17.0
315	1996.0	United States	260.0
243	1996.0	Australia	132.0
264	1996.0	Germany	124.0
298	1996.0	Russia	115.0
253	1996.0	China	110.0
273	1996.0	Italy	71.0
279	1996.0	Korea, South	66.0
256	1996.0	Cuba	57.0
252	1996.0	Canada	51.0
267	1996.0	Hungary	43.0
297	1996.0	Romania	38.0
317	1996.0	Yugoslavia	26.0
250	1996.0	Bulgaria	21.0
294	1996.0	Poland	21.0
394	2000.0	United States	248.0
379	2000.0	Russia	188.0

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

	Year	Country	Sum
322	2000.0	Australia	183.0
346	2000.0	Germany	119.0
334	2000.0	China	79.0
361	2000.0	Korea, South	73.0
338	2000.0	Cuba	69.0
355	2000.0	Italy	65.0
348	2000.0	Hungary	53.0
378	2000.0	Romania	46.0
332	2000.0	Canada	31.0
398	2000.0	Yugoslavia	26.0
375	2000.0	Poland	24.0
330	2000.0	Bulgaria	13.0
469	2004.0	United States	264.0
453	2004.0	Russia	192.0
400	2004.0	Australia	157.0
425	2004.0	Germany	149.0
433	2004.0	Italy	102.0
411	2004.0	China	94.0
414	2004.0	Cuba	61.0
439	2004.0	Korea, South	52.0
428	2004.0	Hungary	40.0
452	2004.0	Romania	39.0
407	2004.0	Bulgaria	17.0
409	2004.0	Canada	17.0
450	2004.0	Poland	12.0
554	2008.0	United States	315.0
488	2008.0	China	184.0
477	2008.0	Australia	149.0
535	2008.0	Russia	143.0
502	2008.0	Germany	101.0
517	2008.0	Korea, South	78.0
491	2008.0	Cuba	47.0
511	2008.0	Italy	42.0
486	2008.0	Canada	34.0
504	2008.0	Hungary	27.0

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

	Year	Country	Sum
534	2008.0	Romania	22.0
532	2008.0	Poland	20.0
484	2008.0	Bulgaria	5.0

In [ ]:

```
# Get top 5 countries for each year based on Sum
top5_countries = (
    q6_data[['Year', 'Country', 'Sum']]
    .groupby('Year')
    .apply(lambda x: x.nlargest(5, 'Sum'))
    .reset_index(drop=True)
)

# Filter q6_data for only these top countries
temp = q6_data[q6_data['Country'].isin(top5_countries['Country'])][["Year", "Country", "

# Pivot table so years are rows and countries are columns
temp = temp.pivot(index='Year', columns='Country', values='Sum').reset_index()

# Clean column names
temp.columns.name = None  # remove pivot name
temp = temp.rename(columns={temp.columns[0]: "Year"})

# Replace NaN with 0
q9_data = temp.fillna(0)

print(q9_data)
```

```
   Year  Australia  Bulgaria  Canada  China  Cuba  East Germany  Germany  \
0  1976       24.0      39.0    20.0    0.0  24.0         192.0      0.0
1  1980       12.0      94.0     0.0    0.0  20.0         260.0      0.0
2  1984       50.0       0.0    86.0   76.0   0.0           0.0      0.0
3  1988       34.0      41.0    21.0   53.0   0.0         174.0      0.0
4  1992       57.0      17.0    44.0   83.0  71.0           0.0    198.0
5  1996      132.0      21.0    51.0  110.0  57.0           0.0    124.0
6  2000      183.0      13.0    31.0   79.0  69.0           0.0    119.0
7  2004      157.0      17.0    17.0   94.0  61.0           0.0    149.0
8  2008      149.0       5.0    34.0  184.0  47.0           0.0    101.0

   Hungary  Italy  Korea, South  Poland  Romania  Russia  Soviet Union  \
0     55.0   31.0          17.0    73.0     55.0     0.0         285.0
1     61.0   37.0           0.0    50.0     72.0     0.0         442.0
2      0.0   63.0          42.0     0.0    106.0     0.0           0.0
3     44.0   29.0          77.0    21.0     51.0     0.0         294.0
4     45.0   46.0          49.0    42.0     53.0     0.0           0.0
5     43.0   71.0          66.0    21.0     38.0   115.0           0.0
6     53.0   65.0          73.0    24.0     46.0   188.0           0.0
7     40.0  102.0          52.0    12.0     39.0   192.0           0.0
8     27.0   42.0          78.0    20.0     22.0   143.0           0.0

   Unified team  United States  West Germany  Yugoslavia
0           0.0          155.0          75.0        19.0
1           0.0            0.0           0.0        57.0
2           0.0          333.0         157.0        87.0
3           0.0          193.0         113.0        63.0
```

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

```
4        223.0          224.0          0.0         0.0
5          0.0          260.0          0.0        26.0
6          0.0          248.0          0.0        26.0
7          0.0          264.0          0.0         0.0
8          0.0          315.0          0.0         0.0
/tmp/ipython-input-3295503498.py:5: DeprecationWarning: DataFrameGroupBy.apply operated on the grouping columns. This behavior is deprecated, and in a future version of pandas the grouping columns will be excluded from the operation. Either pass `include_groups=False` to exclude the groupings or explicitly select the grouping columns after groupby to silence this warning.
  .apply(lambda x: x.nlargest(5, 'Sum'))
```

So these are the top 5 countries in each olympic game. Lets Combine Soviet Union + Unified Team +Russia and East Germany + West Germany + Germany. Also lets drop Yugoslavia, Poland, South Korea, Italy, Hungary, Cuba, Canada, Bulgaria as they are only shown up once in top 5

In [ ]:

```
# Merge Germany related columns
q9_data['Germany'] = (
    q9_data.get('Germany', 0).fillna(0) +
    q9_data.get('East Germany', 0).fillna(0) +
    q9_data.get('West Germany', 0).fillna(0)
)

# Merge Russia related columns
q9_data['Russia'] = (
    q9_data.get('Soviet Union', 0).fillna(0) +
    q9_data.get('Russia', 0).fillna(0) +
    q9_data.get('Unified team', 0).fillna(0)
)

# Drop unnecessary countries
drop_countries = [
    'Yugoslavia', 'Poland', 'Korea, South', 'Italy', 'Hungary',
    'Cuba', 'Canada', 'Bulgaria',
    'East Germany', 'West Germany', 'Soviet Union', 'Unified team'
]

q9_data = q9_data.drop(columns=[c for c in drop_countries if c in q9_data.columns])

# Set 'Year' as index
q9_data = q9_data.set_index('Year')

print(q9_data)
```

```
      Australia  China  Germany  Romania  Russia  United States
Year
1976       24.0    0.0    267.0     55.0   285.0          155.0
1980       12.0    0.0    260.0     72.0   442.0            0.0
1984       50.0   76.0    157.0    106.0     0.0          333.0
1988       34.0   53.0    287.0     51.0   294.0          193.0
1992       57.0   83.0    198.0     53.0   223.0          224.0
1996      132.0  110.0    124.0     38.0   115.0          260.0
2000      183.0   79.0    119.0     46.0   188.0          248.0
2004      157.0   94.0    149.0     39.0   192.0          264.0
2008      149.0  184.0    101.0     22.0   143.0          315.0
```

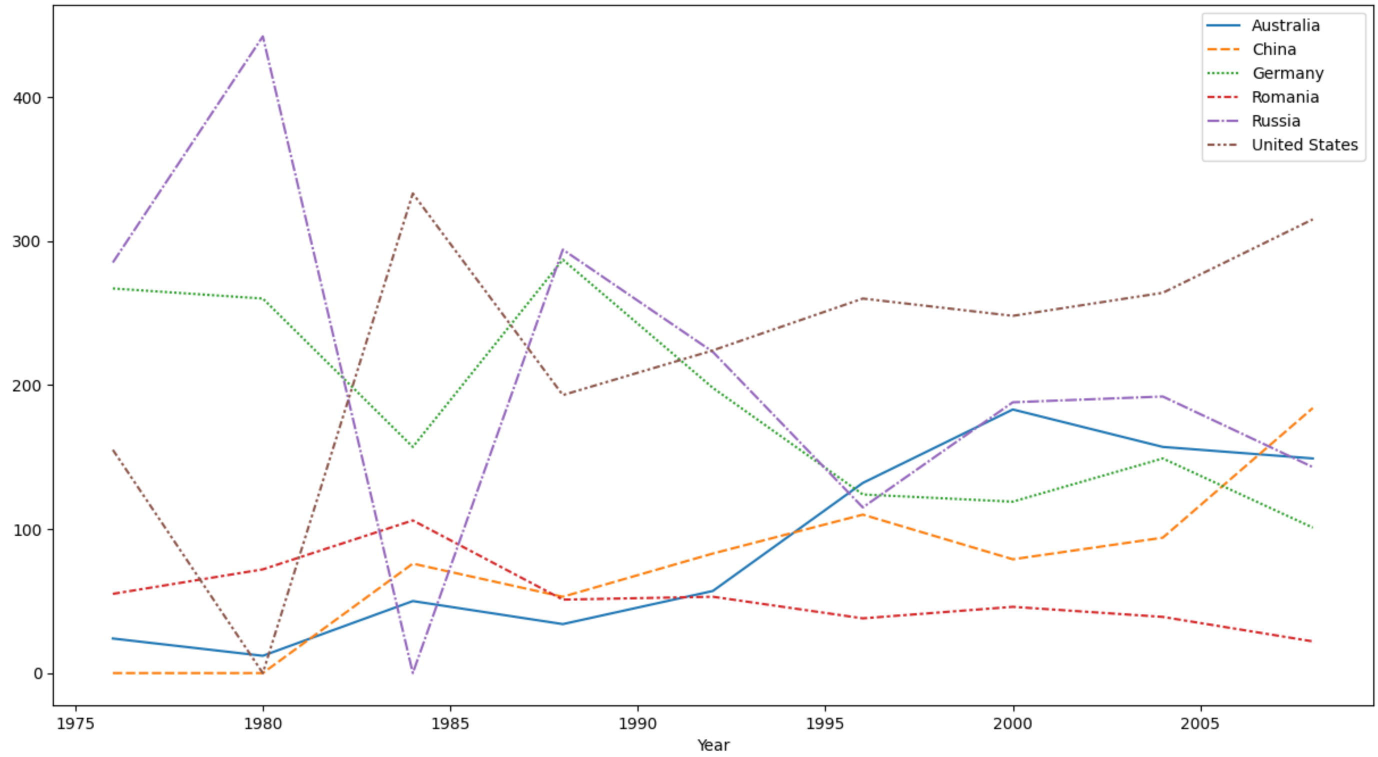
In [ ]:

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

```
# q9_data.plot(x = 'Year', y= q9_data.columns[1:])
import seaborn as sns
plt.figure(figsize=(15,8))
sns.lineplot(data = q9_data)
```

Out[ ]:

<Axes: xlabel='Year'>



Ans. We can clearly see some pattern here.

● Soviet Union(Russia here) dominated Olypics with decline over time except 1982 where it boycotted entire olypics.

● US after boycotting 1980 olypics, rose up to be the dominating player here.

● Germany as a whole country inlcuding (west and east), saw continous decline over period of time.

● China and Australia has witnessed steady rise in their medal tally

● Romania has been same overpeiod with little decline.

Note: The number do not represent number of medal but the total people who won it. E.g., Winner in hockey gets one gold, but 16 people are given the medal. So here we are counting 16.

Medal Distribution by Sport (Top 10 Sports)

In [ ]:

```
# Group by Sport and count medals
sport_medals = data.groupby('Sport')['Medal'].count().sort_values(ascending=False).head(

# Plot
plt.figure(figsize=(10,6))
sns.barplot(x=sport_medals.values, y=sport_medals.index, palette="viridis")
```

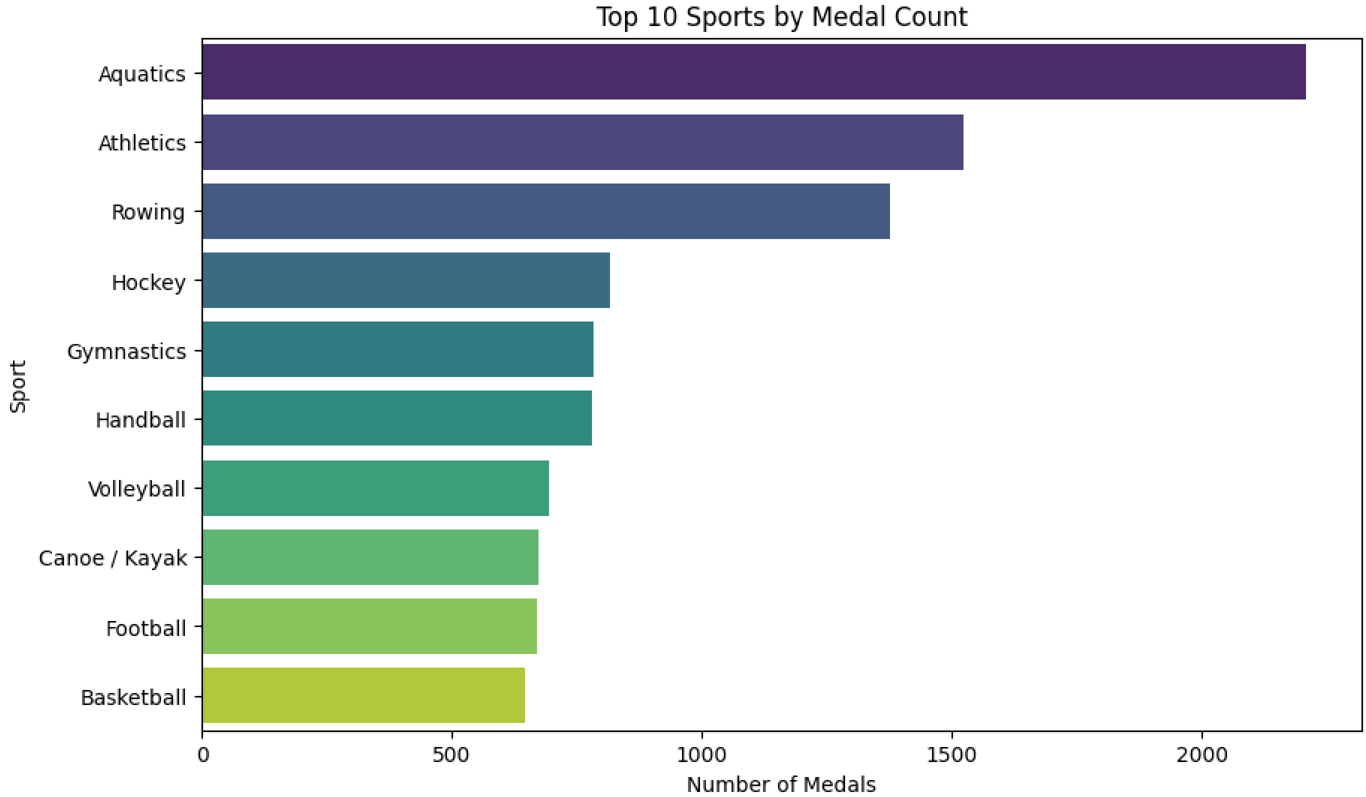
Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

```
plt.title("Top 10 Sports by Medal Count")
plt.xlabel("Number of Medals")
plt.ylabel("Sport")
plt.show()
```

```
/tmp/ipython-input-624454369.py:6: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0.
Assign the `y` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(x=sport_medals.values, y=sport_medals.index, palette="viridis")
```



Medal Tally Heatmap by Country and Year

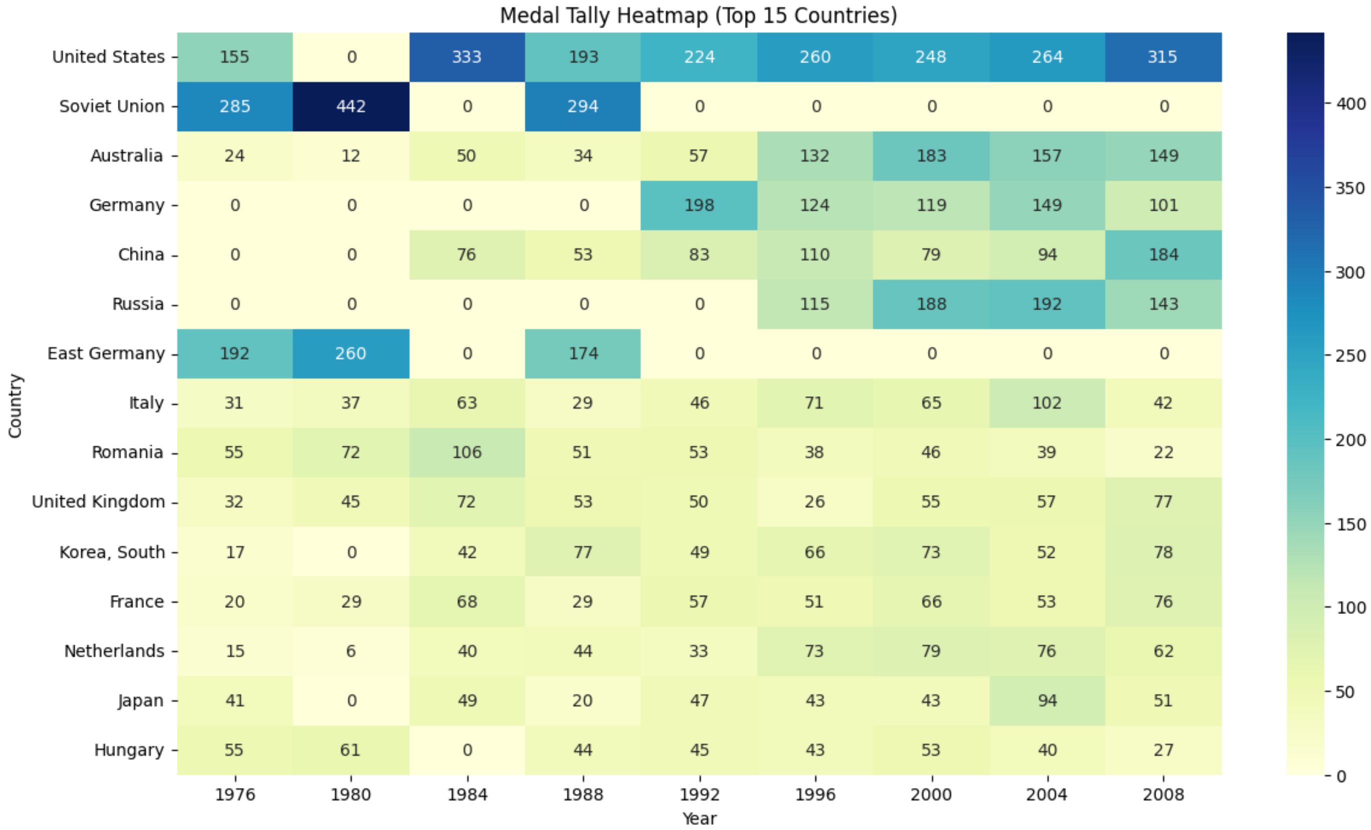
In [ ]:

```
# Create pivot table for heatmap
heatmap_data = data.pivot_table(index='Country', columns='Year', values='Medal', aggfunc

# Select top 15 countries by total medals
top_countries = heatmap_data.sum(axis=1).sort_values(ascending=False).head(15)
heatmap_data = heatmap_data.loc[top_countries.index]

plt.figure(figsize=(14,8))
sns.heatmap(heatmap_data, cmap="YlGnBu", annot=True, fmt=".0f")
plt.title("Medal Tally Heatmap (Top 15 Countries)")
plt.show()
```

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF



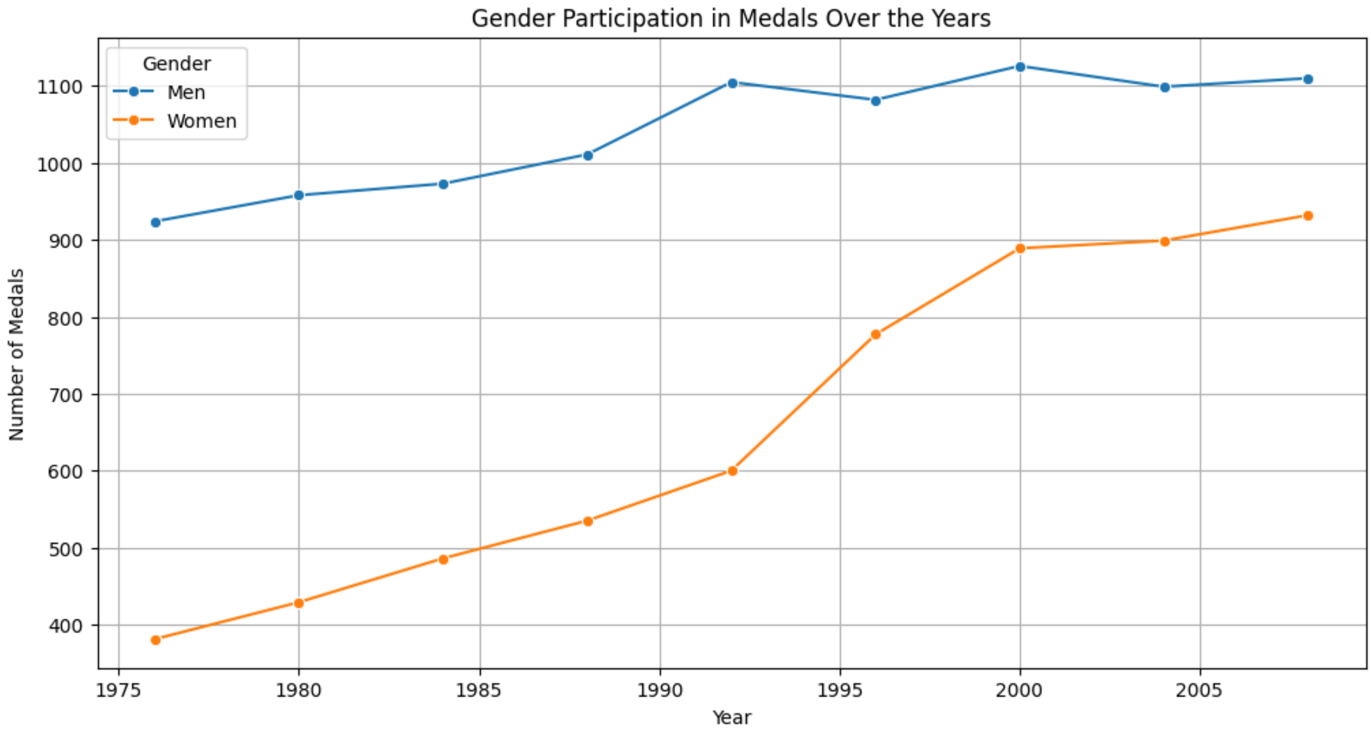
Gender Ratio Trend Over the Years

In [ ]:

```
# Count medals by gender per year
gender_year = data.groupby(['Year', 'Gender'])['Medal'].count().reset_index()

plt.figure(figsize=(12,6))
sns.lineplot(data=gender_year, x='Year', y='Medal', hue='Gender', marker='o')
plt.title("Gender Participation in Medals Over the Years")
plt.ylabel("Number of Medals")
plt.grid(True)
plt.show()
```

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

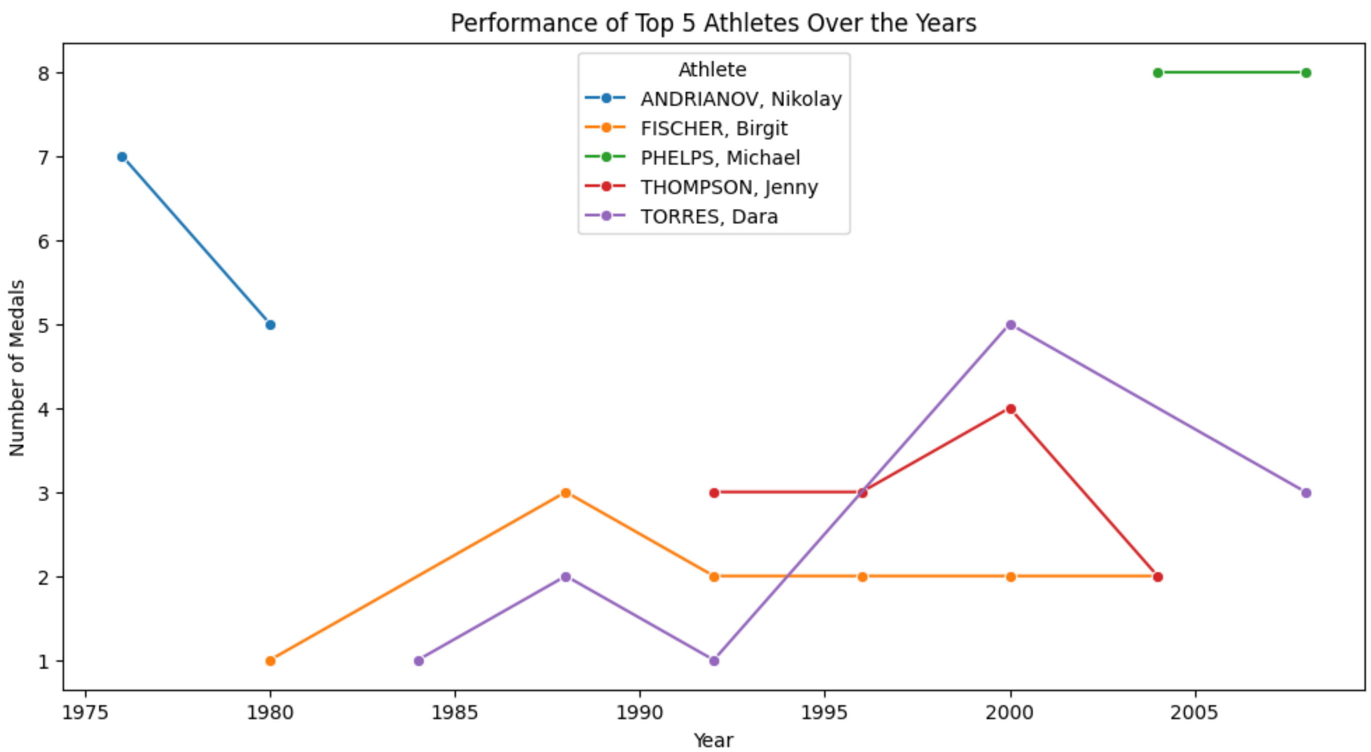


Athlete Performance Over Multiple Olympics

In [ ]:

```
athlete_performance = data.groupby(['Athlete', 'Year'])['Medal'].count().reset_index()
top_athletes = athlete_performance.groupby('Athlete')['Medal'].sum().sort_values(ascendi
athlete_performance = athlete_performance[athlete_performance['Athlete'].isin(top_athlet

plt.figure(figsize=(12,6))
sns.lineplot(data=athlete_performance, x='Year', y='Medal', hue='Athlete', marker='o')
plt.title("Performance of Top 5 Athletes Over the Years")
plt.ylabel("Number of Medals")
plt.show()
```



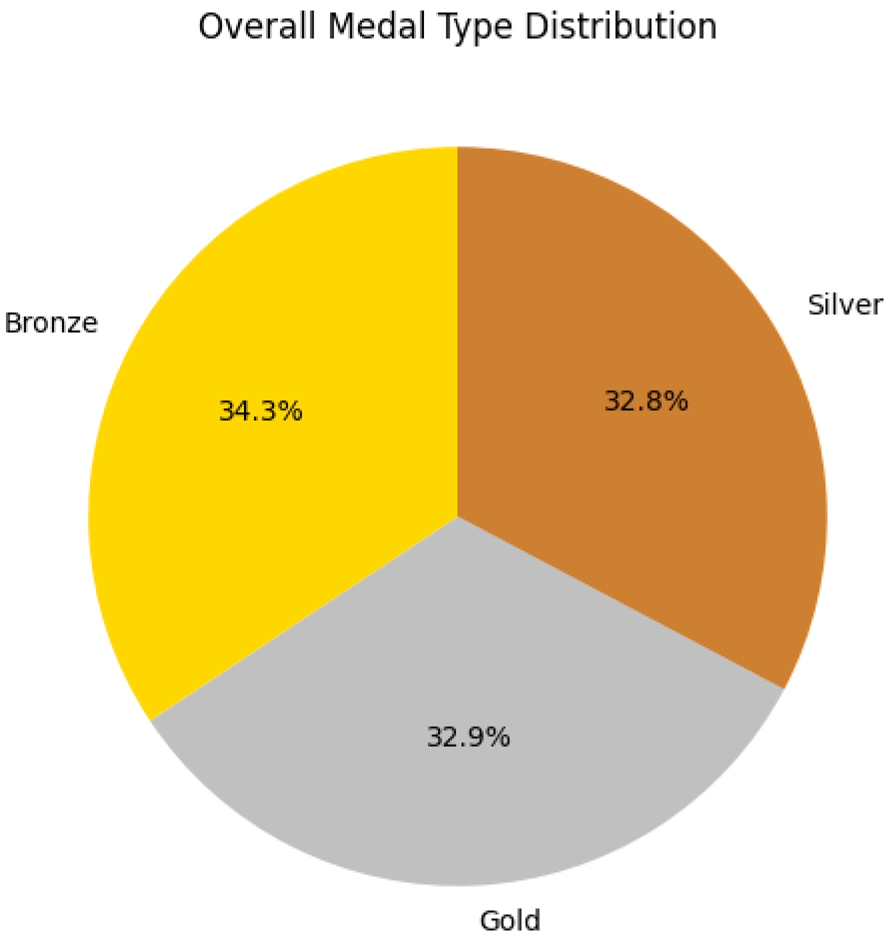
Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

Gold vs Silver vs Bronze Distribution

In [ ]:

```
# Medal type distribution
medal_type = data['Medal'].value_counts()

plt.figure(figsize=(6,6))
medal_type.plot(kind='pie', autopct='%1.1f%%', colors=['gold', 'silver', '#cd7f32'], sta
plt.title("Overall Medal Type Distribution")
plt.ylabel('')
plt.show()
```



Conclusion

The Olympics Data Analysis provided valuable insights into the trends, patterns, and performances of athletes and countries from 1976 to 2008. By examining medal counts, gender participation, sport-wise performances, and country dominance, several key observations emerged:

Top Performing Nations – Countries like the United States, Soviet Union/Russia, China, and Germany consistently dominated the medal tally across multiple Olympic editions, with notable rises and declines influenced by political, social, and economic factors (e.g., boycotts, sports funding, hosting advantages).

Sport Dominance – Certain countries exhibited clear dominance in specific sports, such as Korea in Archery, USA in Athletics & Swimming, and China in Table Tennis & Diving.

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

Gender Participation Gap – While women's participation has grown steadily over the years, the data highlights a historical imbalance, with many male-only events in earlier editions. However, recent trends show progress toward gender parity.

Athlete Excellence – Exceptional athletes like Michael Phelps have redefined performance standards by winning multiple medals across events and years.

Evolving Trends – The medal tallies show that emerging sports powerhouses like China and Australia significantly improved their performance in the 2000s, reshaping the competitive landscape.

Medal Type Distribution – Gold, silver, and bronze medal counts are proportionally balanced across countries, but nations with greater sports diversity tend to have a broader medal spread.

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF