

```
import numpy as np
import pandas as pd
```

```
olympic = pd.read_csv("/content/drive/MyDrive/Unified internship 2025/Project/Summer-Olympic-medals-1976-to-2008.csv")
olympic
```

	City	Year	Sport	Discipline	Event	Athlete	Gender	Country_Code	Country	Event_gender	Medal
0	Montreal	1976.0	Aquatics	Diving	3m springboard	KÖHLER, Christa	Women	GDR	East Germany	W	Silver
1	Montreal	1976.0	Aquatics	Diving	3m springboard	KOSENKOV, Aleksandr	Men	URS	Soviet Union	M	Bronze
2	Montreal	1976.0	Aquatics	Diving	3m springboard	BOGGS, Philip George	Men	USA	United States	M	Gold
3	Montreal	1976.0	Aquatics	Diving	3m springboard	CAGNOTTO, Giorgio Franco	Men	ITA	Italy	M	Silver
4	Montreal	1976.0	Aquatics	Diving	10m platform	WILSON, Deborah Keplar	Women	USA	United States	W	Bronze
...	...	...	...	...	...	...	...	...	...	...	...
15428	Beijing	2008.0	Wrestling	Wrestling Gre-R	66 - 74kg	GUENOT, Christophe	Men	FRA	France	M	Bronze
15429	Beijing	2008.0	Wrestling	Wrestling Gre-R	66 - 74kg	KVIRKELIA, Manuchar	Men	GEO	Georgia	M	Gold

Next steps: [Generate code with olympic](#) [View recommended plots](#) [New interactive sheet](#)

```
olympic.shape
```

(15433, 11)

```
olympic.isnull().sum()
```

	0
City	117
Year	117
Sport	117
Discipline	117
Event	117
Athlete	117
Gender	117
Country_Code	117
Country	117
Event_gender	117
Medal	117

```
olympic[olympic["City"].isnull()]
```

	City	Year	Sport	Discipline	Event	Athlete	Gender	Country_Code	Country	Event_gender	Medal
770	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
771	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
772	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
773	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
774	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...
882	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
883	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
884	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
885	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
886	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

117 rows x 11 columns

```
olympic= olympic.dropna()
olympic.isnull().sum()
```

	0
City	0
Year	0
Sport	0
Discipline	0
Event	0
Athlete	0
Gender	0
Country_Code	0
Country	0
Event_gender	0
Medal	0

olympic

	City	Year	Sport	Discipline	Event	Athlete	Gender	Country_Code	Country	Event_gender	Medal
0	Montreal	1976.0	Aquatics	Diving	3m springboard	KÖHLER, Christa	Women	GDR	East Germany	W	Silver
1	Montreal	1976.0	Aquatics	Diving	3m springboard	KOSENKOV, Aleksandr	Men	URS	Soviet Union	M	Bronze
2	Montreal	1976.0	Aquatics	Diving	3m springboard	BOGGS, Philip George	Men	USA	United States	M	Gold
3	Montreal	1976.0	Aquatics	Diving	3m springboard	CAGNOTTO, Giorgio Franco	Men	ITA	Italy	M	Silver
4	Montreal	1976.0	Aquatics	Diving	10m platform	WILSON, Deborah Keplar	Women	USA	United States	W	Bronze
...	...	...	...	...	...	...	...	...	...	...	...
15428	Beijing	2008.0	Wrestling	Wrestling Gre-R	66 - 74kg	GUENOT, Christophe	Men	FRA	France	M	Bronze
15429	Beijing	2008.0	Wrestling	Wrestling Gre-R	66 - 74kg	KVIRKELIA, Manuchar	Men	GEO	Georgia	M	Gold

```
olympic.duplicated().sum()
```

```
np.int64(1)
```

```
olympic[olympic.duplicated()]
```

	City	Year	Sport	Discipline	Event	Athlete	Gender	Country_Code	Country	Event_gender	Medal
2320	Moscow	1980.0	Hockey	Hockey	hockey	SINGH. Sindh	Men	IND	India	M	Gold

```
olympic = olympic.drop_duplicates()
olympic
```

	City	Year	Sport	Discipline	Event	Athlete	Gender	Country_Code	Country	Event_gender	Medal
0	Montreal	1976.0	Aquatics	Diving	3m springboard	KÖHLER, Christa	Women	GDR	East Germany	W	Silver
1	Montreal	1976.0	Aquatics	Diving	3m springboard	KOSENKOV, Aleksandr	Men	URS	Soviet Union	M	Bronze
2	Montreal	1976.0	Aquatics	Diving	3m springboard	BOGGS, Philip George	Men	USA	United States	M	Gold
3	Montreal	1976.0	Aquatics	Diving	3m springboard	CAGNOTTO, Giorgio Franco	Men	ITA	Italy	M	Silver
4	Montreal	1976.0	Aquatics	Diving	10m platform	WILSON, Deborah Keplar	Women	USA	United States	W	Bronze
...	...	...	...	...	...	...	...	...	...	...	...
15428	Beijing	2008.0	Wrestling	Wrestling Gre-R	66 - 74kg	GUENOT, Christophe	Men	FRA	France	M	Bronze
15429	Beijing	2008.0	Wrestling	Wrestling Gre-R	66 - 74kg	KVIRKELIA, Manuchar	Men	GEO	Georgia	M	Gold

Next steps:

Generate code with olympic

View recommended plots

New interactive sheet

```
olympic.nunique()
```


	0
City	9
Year	9
Sport	28
Discipline	41
Event	293
Athlete	11337
Gender	2
Country_Code	128
Country	127
Event_gender	3
Medal	3

```
olympic.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 15315 entries, 0 to 15432
Data columns (total 11 columns):
#   Column      Non-Null Count  Dtype
---  -
0   City         15315 non-null  object
1   Year         15315 non-null  float64
2   Sport        15315 non-null  object
3   Discipline   15315 non-null  object
4   Event        15315 non-null  object
5   Athlete      15315 non-null  object
```

```
6  Gender      15315 non-null  object
7  Country_Code 15315 non-null  object
8  Country      15315 non-null  object
9  Event_gender 15315 non-null  object
10 Medal       15315 non-null  object
dtypes: float64(1), object(10)
memory usage: 1.4+ MB
```


```
olympic.dtypes
```



	0
City	object
Year	float64
Sport	object
Discipline	object
Event	object
Athlete	object
Gender	object
Country_Code	object
Country	object
Event_gender	object
Medal	object

```
olympic = olympic.copy()
olympic["Year"] = olympic["Year"].astype(int)
```

```
olympic.dtypes
```



	0
City	object
Year	int64
Sport	object
Discipline	object
Event	object
Athlete	object
Gender	object
Country_Code	object
Country	object
Event_gender	object
Medal	object

```
olympic.describe()
```

	Year	
<b>count</b>	15315.000000	
<b>mean</b>	1993.621678	
<b>std</b>	10.159586	
<b>min</b>	1976.000000	
<b>25%</b>	1984.000000	
<b>50%</b>	1996.000000	
<b>75%</b>	2004.000000	
<b>max</b>	2008.000000	

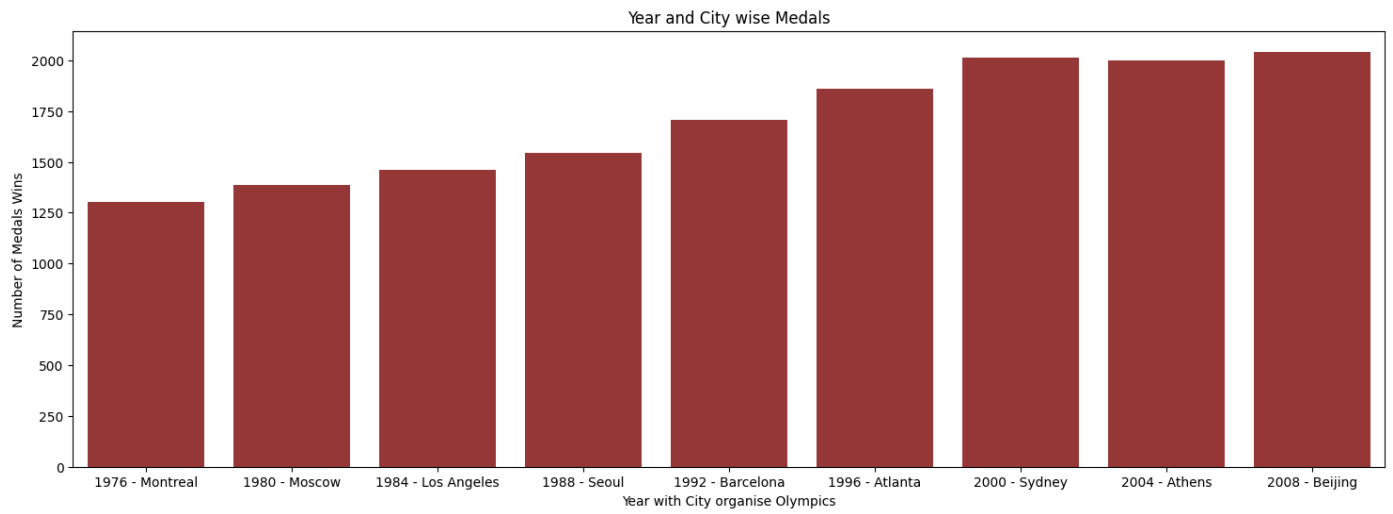
```
Year_City_wise_Medal = olympic.groupby(["Year", "City"]).agg({"Medal" : "count"}).reset_index()
Year_City_wise_Medal
```

	Year	City	Medal	
<b>0</b>	1976	Montreal	1305	
<b>1</b>	1980	Moscow	1386	
<b>2</b>	1984	Los Angeles	1459	
<b>3</b>	1988	Seoul	1546	
<b>4</b>	1992	Barcelona	1705	
<b>5</b>	1996	Atlanta	1859	
<b>6</b>	2000	Sydney	2015	
<b>7</b>	2004	Athens	1998	
<b>8</b>	2008	Beijing	2042	

Next steps: [Generate code with Year\\_City\\_wise\\_Medal](#) [View recommended plots](#) [New interactive sheet](#)

```
import matplotlib.pyplot as plt
import seaborn as sns
```

```
plt.figure(figsize=(18,6))
Year_City_wise_Medal["Year_City"] = Year_City_wise_Medal["Year"].astype(str) + " - " + Year_City_wise_Medal["City"]
sns.barplot(data=Year_City_wise_Medal, x="Year_City", y="Medal", color = "brown")
plt.title("Year and City wise Medals")
plt.xlabel("Year with City organise Olympics")
plt.ylabel("Number of Medals Wins")
plt.show()
```



```
Top_country_year_wise_Medal = olympic.groupby(["Year" , "Country"]).agg({"Medal" : "count"}).reset_index().sort_values(by=["Year", "Medal"], a
Top_5_country_year_wise_Medal = Top_country_year_wise_Medal.groupby("Year").head(5).reset_index(drop=True)
Top_5_country_year_wise_Medal
```

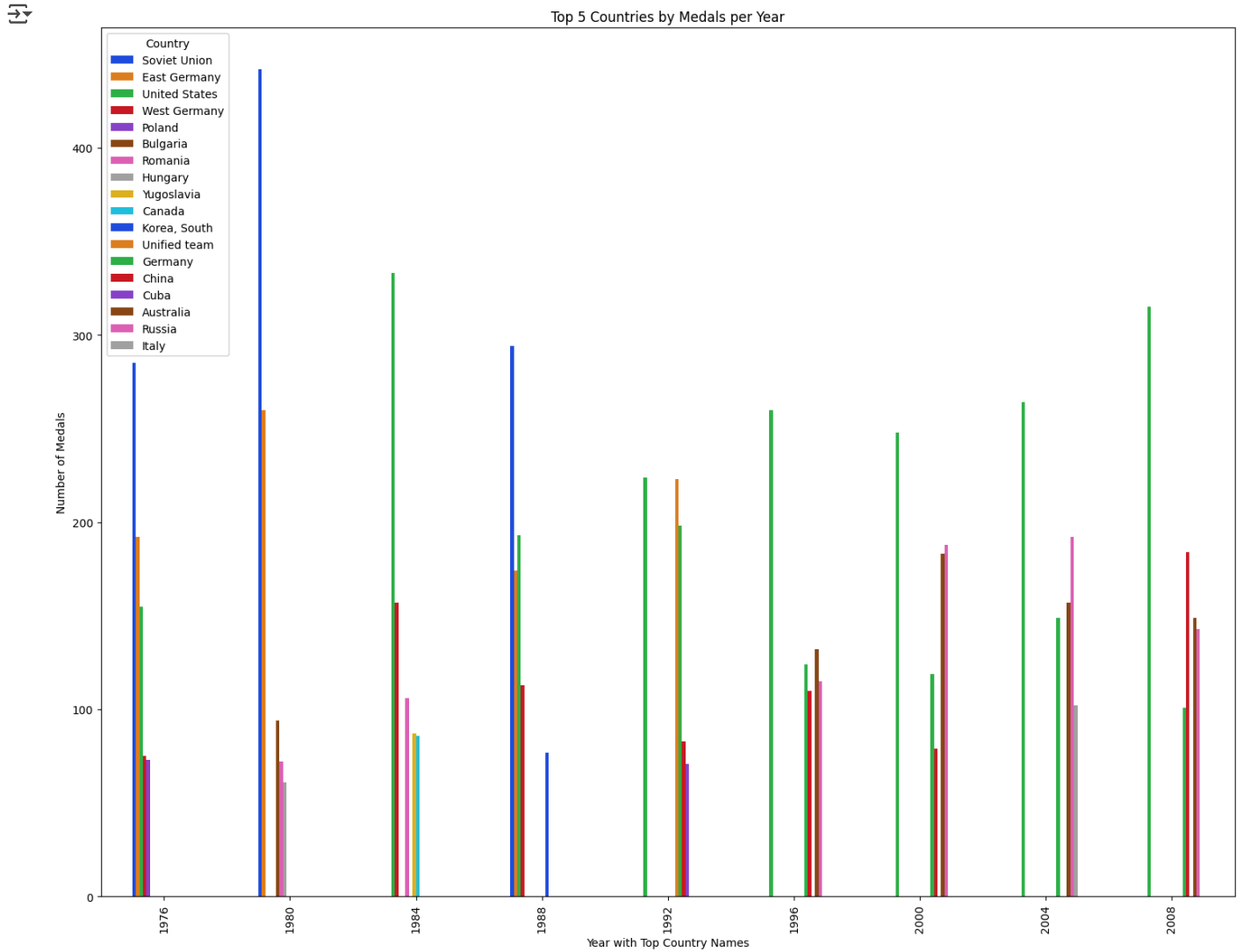


	Year	Country	Medals	
0	1976	Soviet Union	285	
1	1976	East Germany	192	
2	1976	United States	155	
3	1976	West Germany	75	
4	1976	Poland	73	
5	1980	Soviet Union	442	
6	1980	East Germany	260	
7	1980	Bulgaria	94	
8	1980	Romania	72	
9	1980	Hungary	61	
10	1984	United States	333	
11	1984	West Germany	157	
12	1984	Romania	106	
13	1984	Yugoslavia	87	
14	1984	Canada	86	
15	1988	Soviet Union	294	
16	1988	United States	193	
17	1988	East Germany	174	
18	1988	West Germany	113	
19	1988	Korea, South	77	
20	1992	United States	224	
21	1992	Unified team	223	
22	1992	Germany	198	
23	1992	China	83	
24	1992	Cuba	71	
25	1996	United States	260	
26	1996	Australia	132	
27	1996	Germany	124	
28	1996	Russia	115	
29	1996	China	110	
30	2000	United States	248	
31	2000	Russia	188	
32	2000	Australia	183	
33	2000	Germany	119	
34	2000	China	79	
35	2004	United States	264	
36	2004	Russia	192	
37	2004	Australia	157	
38	2004	Germany	149	
39	2004	Italy	102	
40	2008	United States	315	
41	2008	China	184	
42	2008	Australia	149	
43	2008	Russia	143	
44	2008	Germany	101	

Next steps:

[Generate code with Top\\_5\\_country\\_year\\_wise\\_Medal](#)[View recommended plots](#)[New interactive sheet](#)

```
plt.figure(figsize=(15, 12))
sns.barplot(data=Top_5_country_year_wise_Medal, x="Year", y="Medal", hue= "Country", palette="bright", width=0.5)
plt.title("Top 5 Countries by Medals per Year")
plt.xlabel("Year with Top Country Names")
plt.ylabel("Number of Medals")
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```





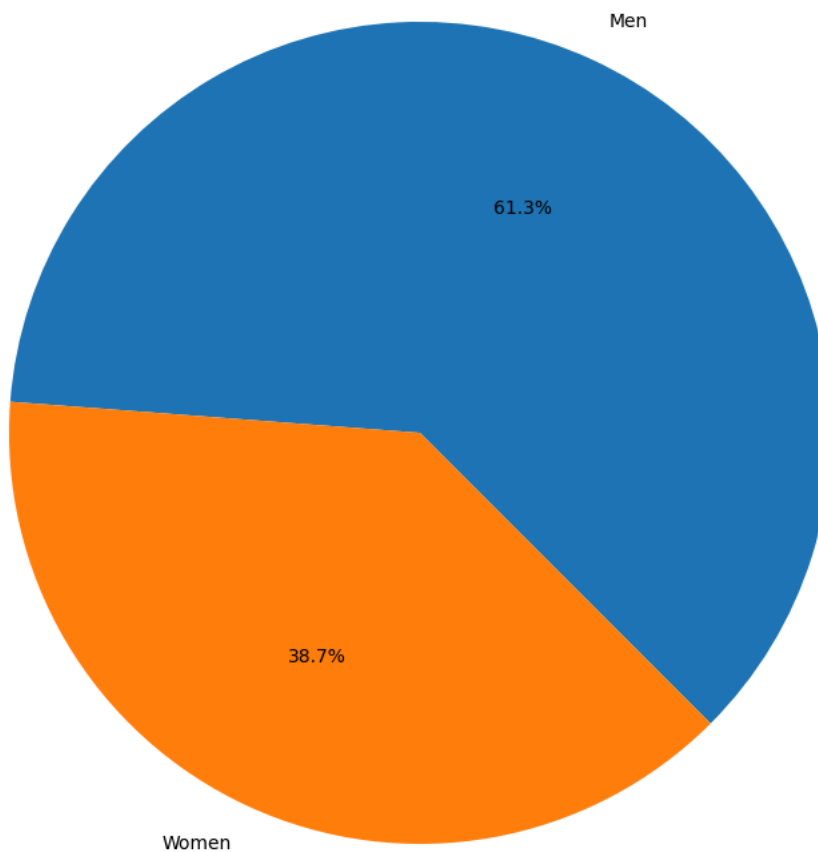
```
Gender_wise_Medal = olympic.groupby(["Gender"]).agg({"Medal" : "count"}).reset_index()
Gender_wise_Medal
```

	Gender	Medal
0	Men	9387
1	Women	5928

Next steps: [Generate code with Gender\\_wise\\_Medal](#) [View recommended plots](#) [New interactive sheet](#)

```
plt.figure(figsize=(12, 10))
plt.pie(Gender_wise_Medal["Medal"], labels= Gender_wise_Medal["Gender"],autopct='%1.1f%%', startangle=-45, labeldistance=1.1)
plt.title("Gender wise Medal Wins", fontsize=14)
plt.show()
```

Gender wise Medal Wins



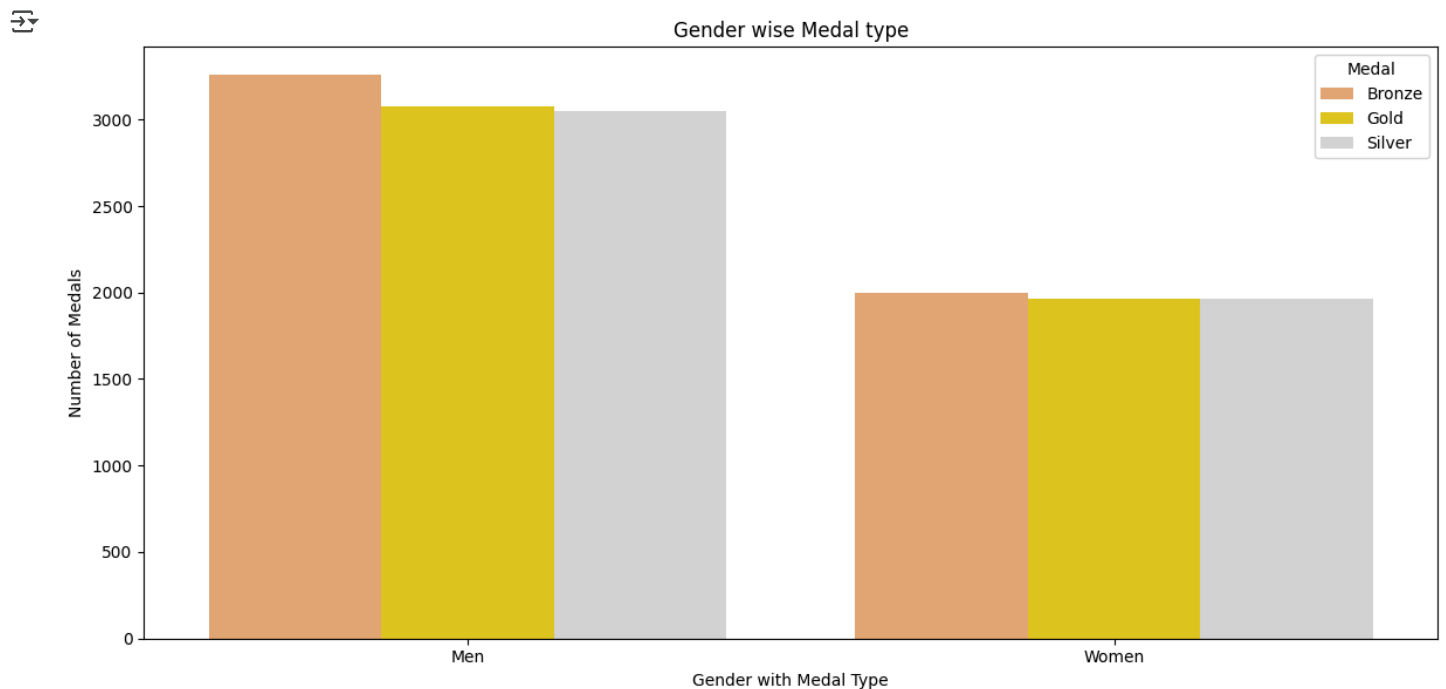
```
Gender_wise_Medal_Type = olympic.groupby(["Gender", "Medal"]).agg({"Medal" : "count"}).rename(columns = {"Medal" : "No of Medal"}).reset_index()
Gender_wise_Medal_Type
```

	Gender	Medal	No of Medal
0	Men	Bronze	3257
1	Men	Gold	3079
2	Men	Silver	3051
3	Women	Bronze	2001
4	Women	Gold	1962
5	Women	Silver	1965


Next steps:




[Generate code with Gender\\_wise\\_Medal\\_Type](#)[View recommended plots](#)[New interactive sheet](#)

```
plt.figure(figsize=(12,6))
medal_color = {"Gold" : "#ffdf00", "Silver" : "#d3d3d3", "Bronze": "#f4a460"}
sns.barplot(data = Gender_wise_Medal_Type, x="Gender", y="No of Medal", hue = "Medal", palette= medal_color)
plt.title("Gender wise Medal type")
plt.xlabel("Gender with Medal Type")
plt.ylabel("Number of Medals")
plt.tight_layout()
plt.show()
```



```
Year_Gender_wise_Medal = olympic.groupby(["Year", "Gender"]).agg({"Medal" : "count"}).reset_index()
Year_Gender_wise_Medal
```

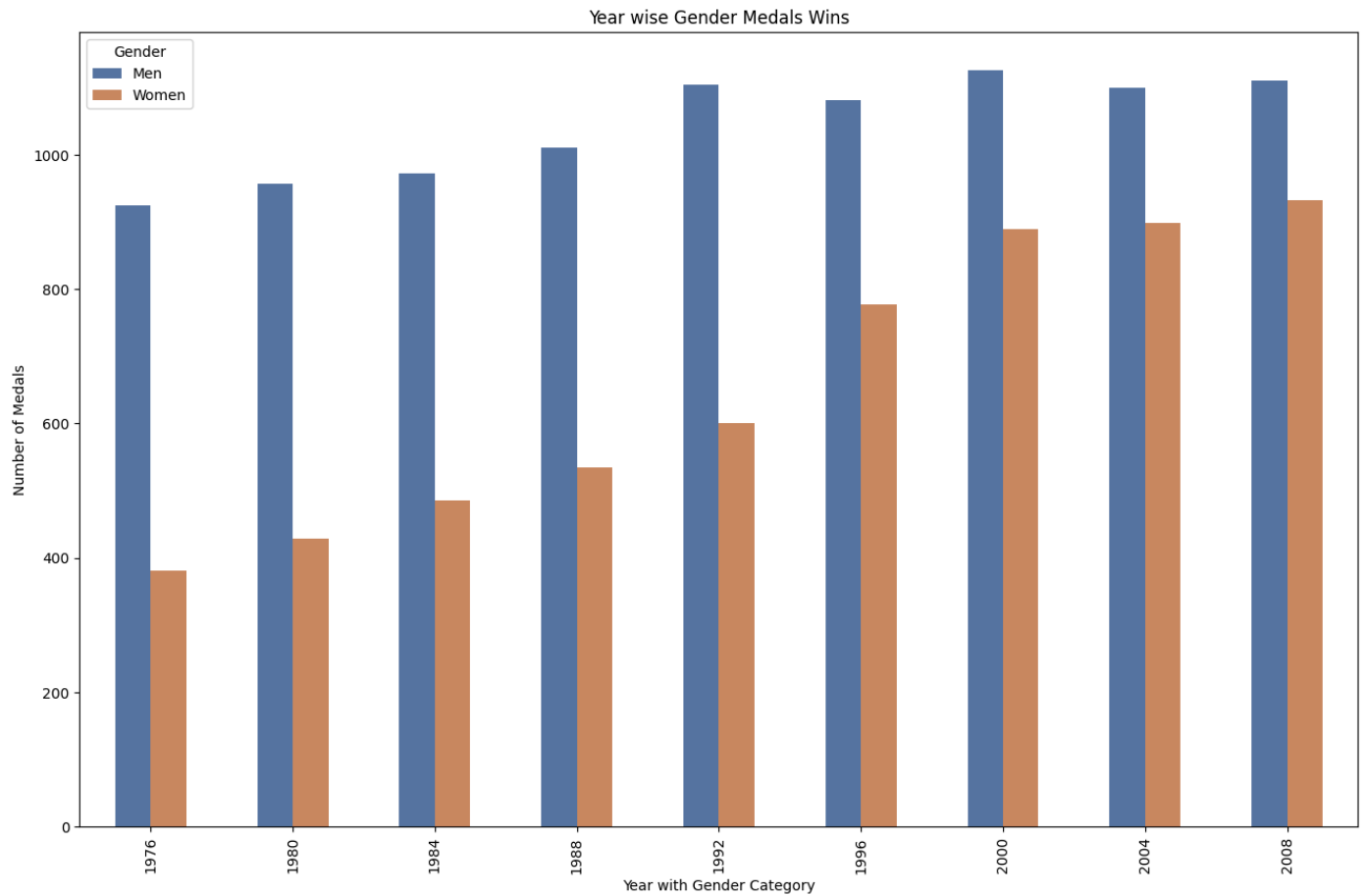


	Year	Gender	Medal	
0	1976	Men	924	
1	1976	Women	381	
2	1980	Men	957	
3	1980	Women	429	
4	1984	Men	973	
5	1984	Women	486	
6	1988	Men	1011	
7	1988	Women	535	
8	1992	Men	1105	
9	1992	Women	600	
10	1996	Men	1082	
11	1996	Women	777	
12	2000	Men	1126	
13	2000	Women	889	
14	2004	Men	1099	
15	2004	Women	899	
16	2008	Men	1110	
17	2008	Women	932	


Next steps:



[Generate code with Year\\_Gender\\_wise\\_Medal](#)[View recommended plots](#)[New interactive sheet](#)

```
plt.figure(figsize=(16, 10))
sns.barplot(data=Year_Gender_wise_Medal, x="Year", y="Medal", hue= "Gender", palette="deep", width=0.5)
plt.title("Year wise Gender Medals Wins")
plt.xlabel("Year with Gender Category")
plt.ylabel("Number of Medals")
plt.xticks(rotation=90)
plt.show()
```



```
Country_wise_Medal = olympic.groupby("Country").agg({"Medal" : "count"}).reset_index().sort_values(by="Medal", ascending = False)
Country_wise_Medals = Country_wise_Medal.head(20)
Country_wise_Medals
```

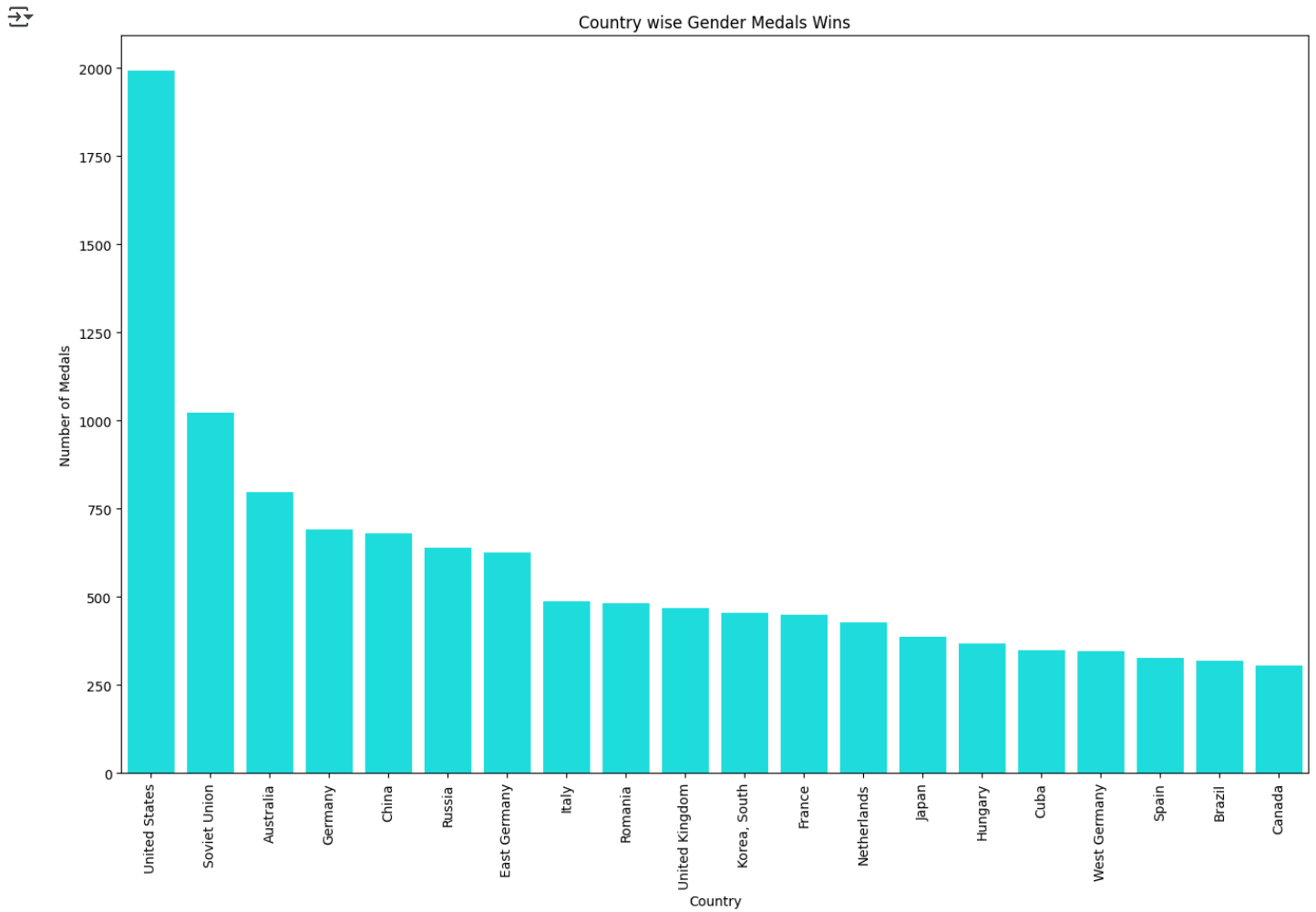


	Country	Medal	
117	United States	1992	
95	Soviet Union	1021	
4	Australia	798	
38	Germany	691	
18	China	679	
87	Russia	638	
29	East Germany	626	
51	Italy	486	
86	Romania	482	
116	United Kingdom	467	
57	Korea, South	454	
36	France	449	
72	Netherlands	428	
53	Japan	388	
43	Hungary	368	
23	Cuba	349	
123	West Germany	345	
96	Spain	328	
12	Brazil	318	
16	Canada	304	


Next steps:




[Generate code with Country\\_wise\\_Medals](#)[View recommended plots](#)[New interactive sheet](#)

```
plt.figure(figsize=(16, 10))
sns.barplot(data=Country_wise_Medals, x="Country", y="Medal", color = "cyan")
plt.title("Country wise Gender Medals Wins")
plt.xlabel("Country")
plt.ylabel("Number of Medals")
plt.xticks(rotation=90)
plt.show()
```



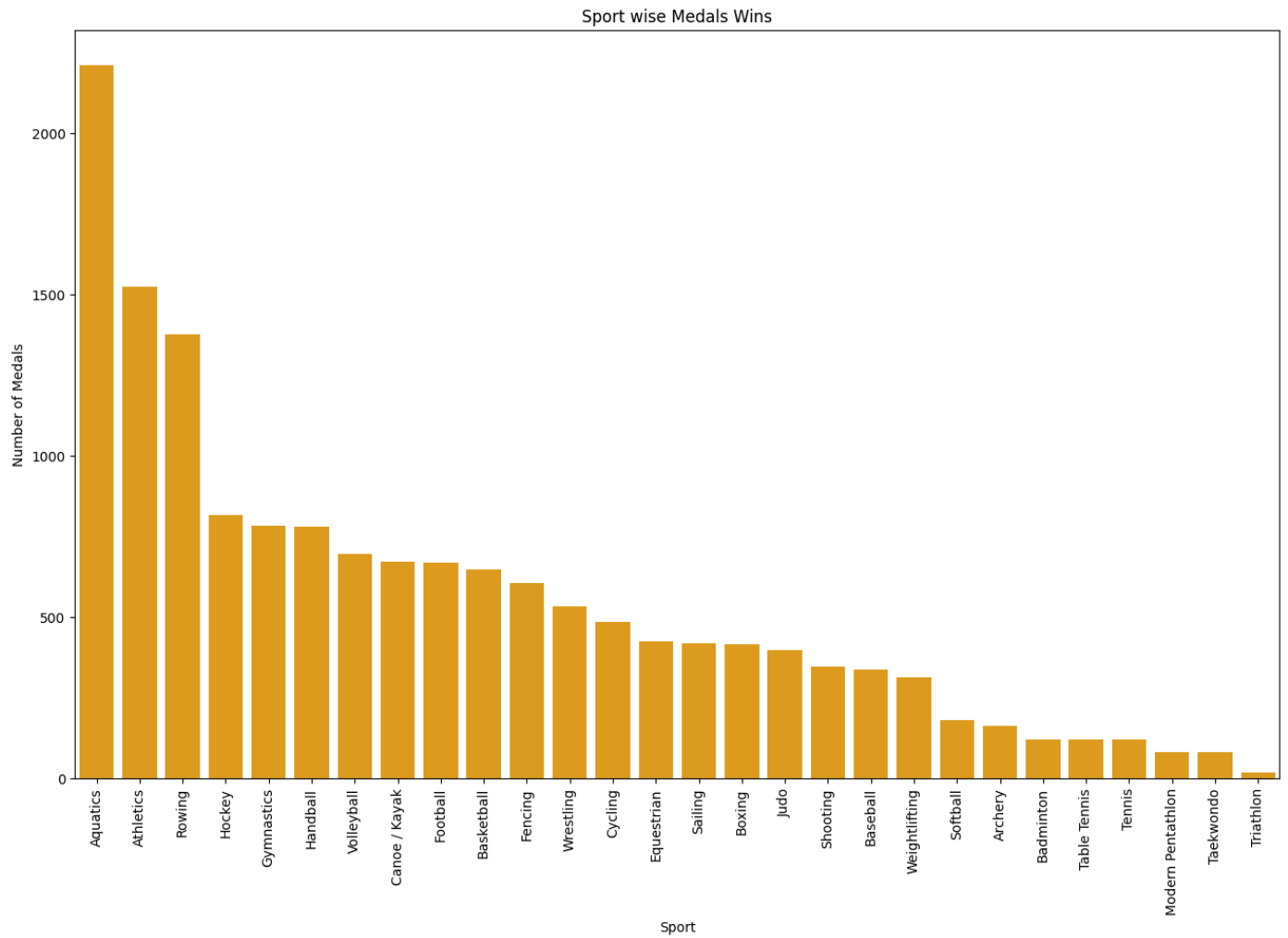
```
Sport_Medal = olympic.groupby(["Sport"]).agg({"Medal" : "count"}).sort_values(by = ["Medal"], ascending = [False]).reset_index()
Sport_Medal
```



	Sport	Medal	
0	Aquatics	2210	
1	Athletics	1523	
2	Rowing	1377	
3	Hockey	816	
4	Gymnastics	783	
5	Handball	780	
6	Volleyball	695	
7	Canoe / Kayak	672	
8	Football	669	
9	Basketball	646	
10	Fencing	606	
11	Wrestling	533	
12	Cycling	483	
13	Equestrian	423	
14	Sailing	417	
15	Boxing	416	
16	Judo	396	
17	Shooting	345	
18	Baseball	335	
19	Weightlifting	311	
20	Softball	180	
21	Archery	162	
22	Badminton	120	
23	Table Tennis	120	
24	Tennis	118	
25	Modern Pentathlon	81	
26	Taekwondo	80	
27	Triathlon	18	

Next steps: [Generate code with Sport\\_Medal](#) [View recommended plots](#) [New interactive sheet](#)

```
plt.figure(figsize=(16, 10))
sns.barplot(data = Sport_Medal, x="Sport", y="Medal", color = "orange")
plt.title("Sport wise Medals Wins")
plt.xlabel("Sport")
plt.ylabel("Number of Medals")
plt.xticks(rotation=90)
plt.show()
```



```
Event_Medal = olympic.groupby(["Event"]).agg({"Medal" : "count"}).sort_values(by = ["Medal"], ascending = [False]).reset_index()
Event_Medals = Event_Medal.head(50)
Event_Medals
```





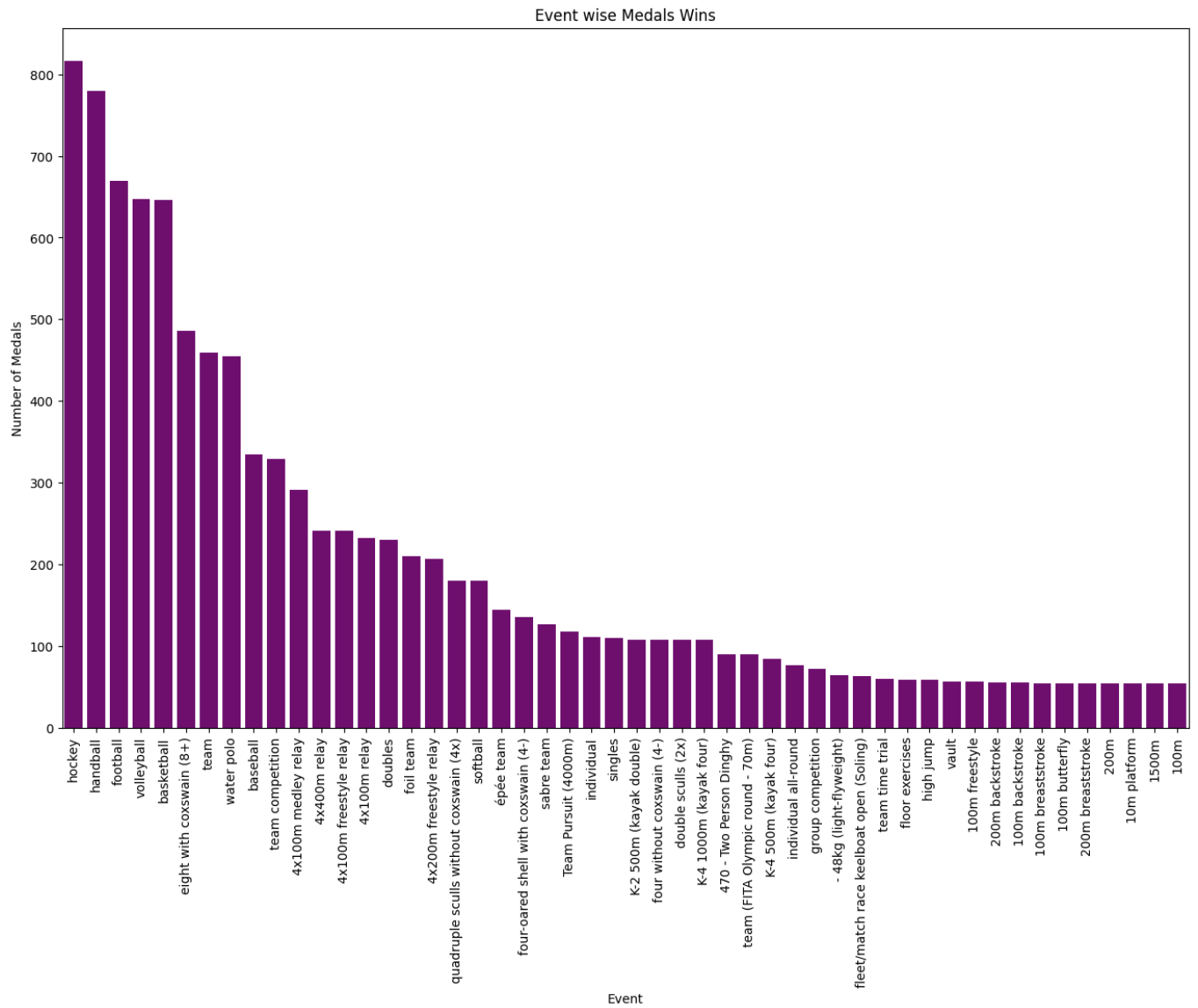
	Event	Medal	
0	hockey	816	
1	handball	780	
2	football	669	
3	volleyball	647	
4	basketball	646	
5	eight with coxswain (8+)	486	
6	team	459	
7	water polo	455	
8	baseball	335	
9	team competition	329	
10	4x100m medley relay	291	
11	4x400m relay	241	
12	4x100m freestyle relay	241	
13	4x100m relay	232	
14	doubles	230	
15	foil team	210	
16	4x200m freestyle relay	207	
17	quadruple sculls without coxswain (4x)	180	
18	softball	180	
19	épée team	144	
20	four-oared shell with coxswain (4-)	135	
21	sabre team	126	
22	Team Pursuit (4000m)	118	
23	individual	111	
24	singles	110	
25	K-2 500m (kayak double)	108	
26	four without coxswain (4-)	108	
27	double sculls (2x)	108	
28	K-4 1000m (kayak four)	108	
29	470 - Two Person Dinghy	90	
30	team (FITA Olympic round - 70m)	90	
31	K-4 500m (kayak four)	84	
32	individual all-round	76	
33	group competition	72	
34	- 48kg (light-flyweight)	64	
35	fleet/match race keelboat open (Soling)	63	
36	team time trial	60	
37	floor exercises	59	
38	high jump	58	
39	vault	56	
40	100m freestyle	56	
41	200m backstroke	55	
42	100m backstroke	55	
43	100m breaststroke	54	
44	100m butterfly	54	
45	200m breaststroke	54	

46	200m	54
47	10m platform	54
48	1500m	54
49	100m	54

Next steps:

[Generate code with Event\\_Medals](#)[View recommended plots](#)[New interactive sheet](#)

```
plt.figure(figsize=(16, 10))
sns.barplot(data = Event_Medals, x="Event", y="Medal", color = "purple")
plt.title("Event wise Medals Wins")
plt.xlabel("Event")
plt.ylabel("Number of Medals")
plt.xticks(rotation=90)
plt.show()
```



```
City_Event = olympic.groupby(["City","Event"]).agg(No_of_Events = ("Event" , "count")).rename(columns={"Event" : "No of Events"}).sort_value
City_Event = City_Event.groupby(["City"]).head(10).reset_index()
City_Event
```

	City	Event	No_of_Events
0	Athens	football	98
1	Athens	hockey	95
2	Athens	handball	88
3	Athens	water polo	78
4	Athens	baseball	71
...	...	...	...
85	Sydney	basketball	72
86	Sydney	volleyball	72
87	Sydney	team	60
88	Sydney	eight with coxswain (8+)	54
89	Sydney	softball	45

90 rows x 3 columns

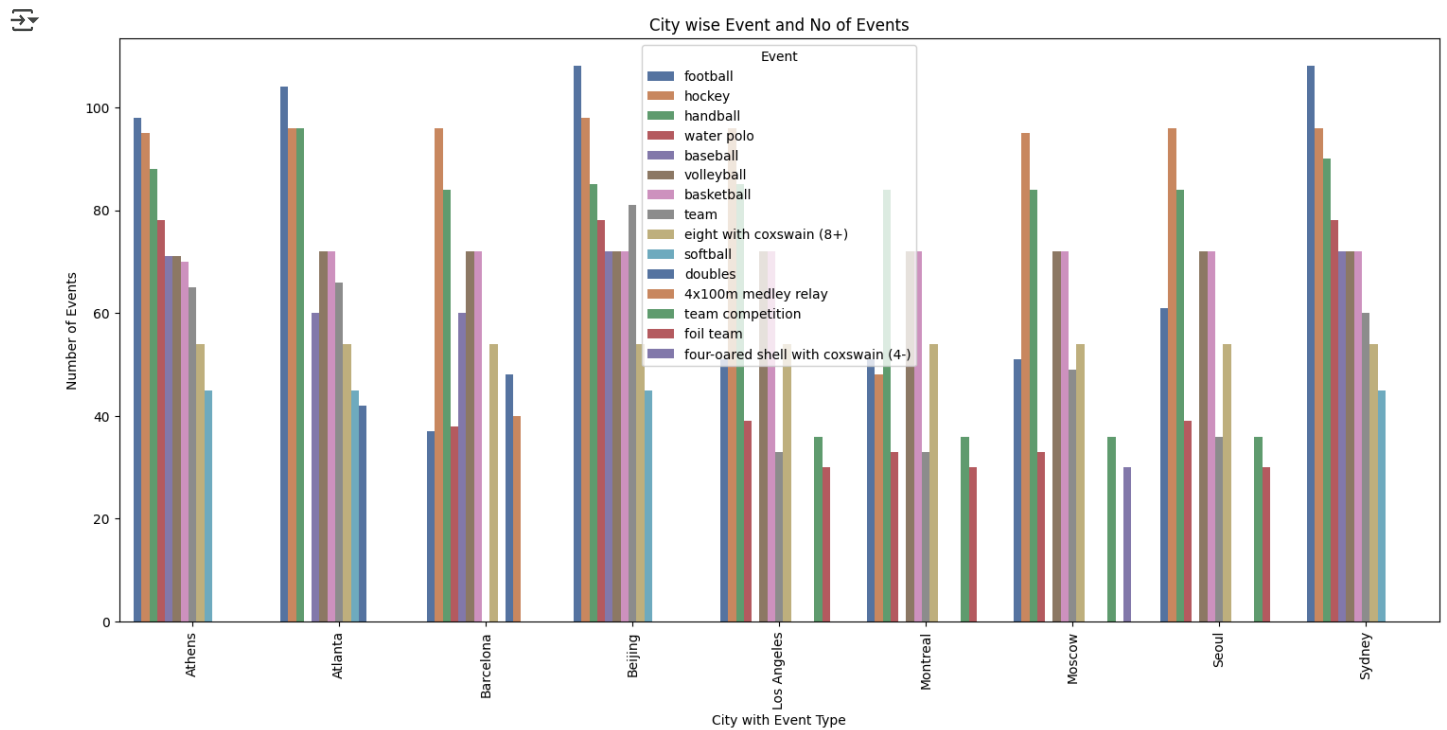
Next steps:

[Generate code with City\\_Event](#)

[View recommended plots](#)

[New interactive sheet](#)

```
plt.figure(figsize=(18,8))
sns.barplot(data = City_Event, x="City", y="No_of_Events", hue = "Event", palette= "deep")
plt.title("City wise Event and No of Events")
plt.xlabel("City with Event Type")
plt.ylabel("Number of Events")
plt.xticks(rotation=90)
plt.show()
```



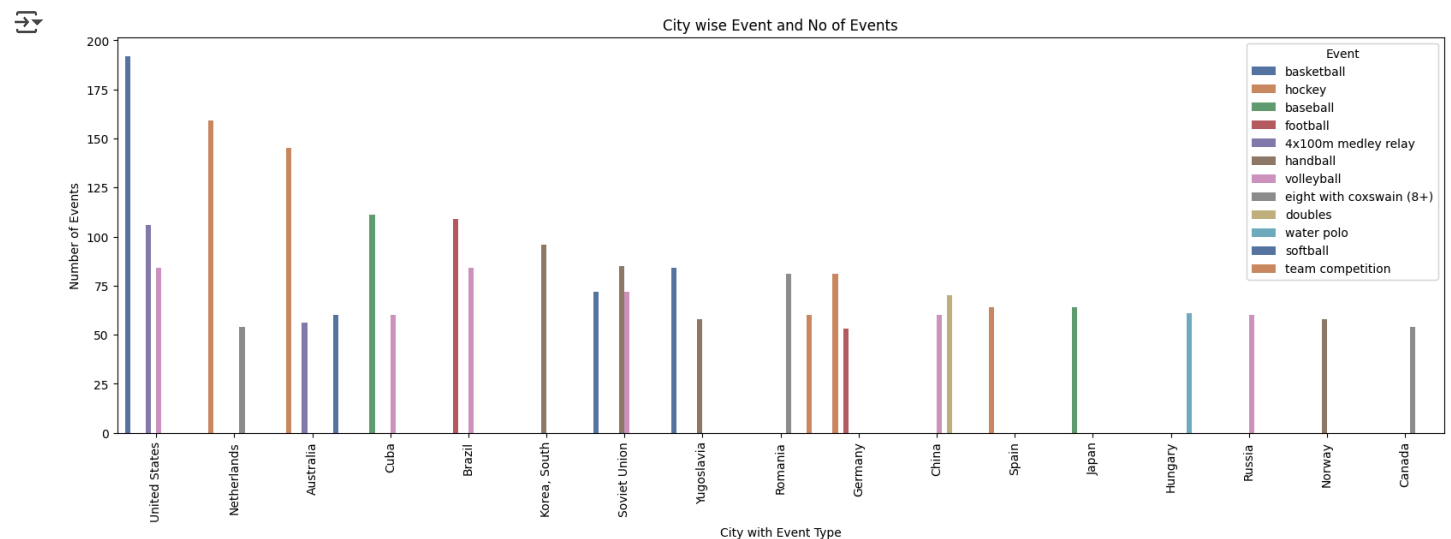
```
Country_Event_wise_Medal = olympic.groupby(["Country", "Event"]).agg({"Medal" : "count"}).sort_values(by = ["Medal"], ascending = (False) )
Country_Event_wise_Medal = Country_Event_wise_Medal.groupby("Country").head(3)
Country_Event_wise_Medals = Country_Event_wise_Medal.reset_index()
Country_Event_wise_Medals
```

	Country	Event	Medal
0	United States	basketball	192
1	Netherlands	hockey	159
2	Australia	hockey	145
3	Cuba	baseball	111
4	Brazil	football	109
...	...	...	...
305	Venezuela	62kg	1
306	Venezuela	54 - 57kg (featherweight)	1
307	Armenia	96 - 120kg	1
308	Armenia	48 - 52kg (flyweight)	1
309	Zambia	- 48kg (light-flyweight)	1

310 rows x 3 columns

Next steps: [Generate code with Country\\_Event\\_wise\\_Medals](#) [View recommended plots](#) [New interactive sheet](#)

```
plt.figure(figsize=(20,6))
sns.barplot(data = Country_Event_wise_Medals.head(30), x="Country", y="Medal", hue = "Event", palette= "deep")
plt.title("City wise Event and No of Events")
plt.xlabel("City with Event Type")
plt.ylabel("Number of Events")
plt.xticks(rotation=90)
plt.show()
```



```
Country_wise_Medal_type = olympic.groupby(["Country", "Medal"]).agg({"Medal" : "count"}).rename(columns={"Medal": "No of Medals"}).sort_value
Country_wise_Medal_type = Country_wise_Medal_type.reset_index()
Country_wise_Medal_type
```

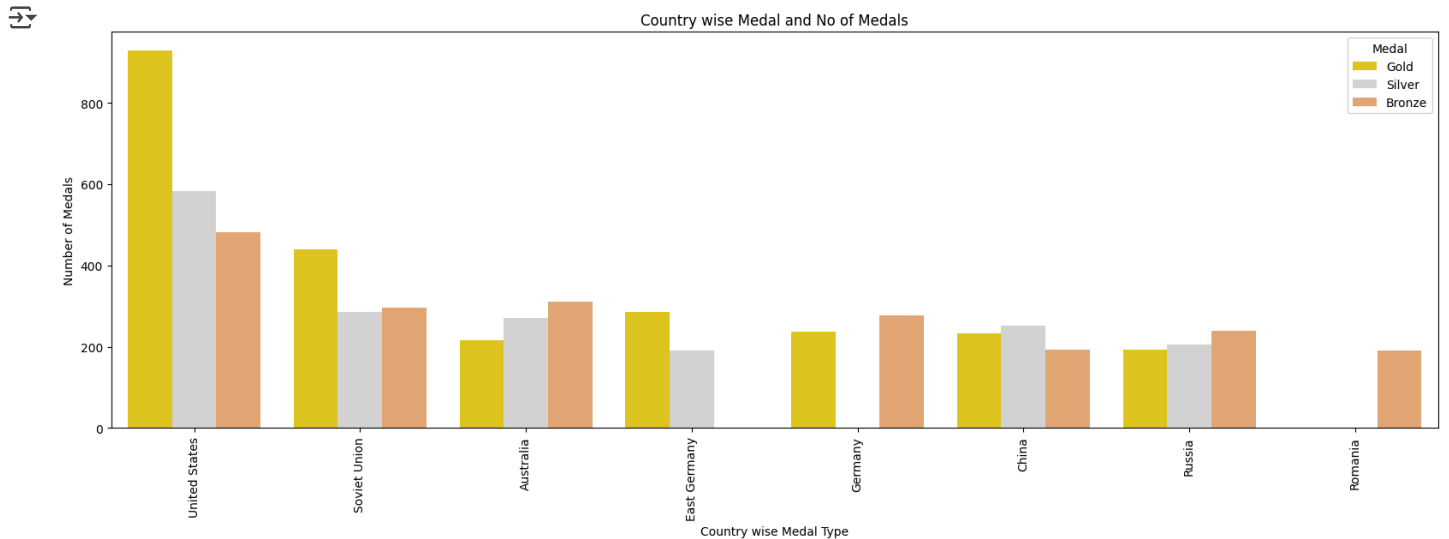
	Country	Medal	No of Medals
0	United States	Gold	928
1	United States	Silver	583
2	United States	Bronze	481
3	Soviet Union	Gold	439
4	Australia	Bronze	312
...	...	...	...
296	Tunisia	Bronze	1
297	Zambia	Bronze	1
298	Zambia	Silver	1
299	Zimbabwe	Bronze	1
300	Afghanistan	Bronze	1

301 rows x 3 columns

Next steps:

[Generate code with Country\\_wise\\_Medal\\_type](#)[View recommended plots](#)[New interactive sheet](#)

```
plt.figure(figsize=(20,6))
medal_color = {"Gold" : "#ffdf00", "Silver" : "#d3d3d3", "Bronze": "#f4a460"}
sns.barplot(data =Country_wise_Medal_type.head(20), x="Country", y="No of Medals", hue = "Medal", palette= medal_color)
plt.title("Country wise Medal and No of Medals")
plt.xlabel("Country wise Medal Type")
plt.ylabel("Number of Medals")
plt.xticks(rotation=90)
plt.show()
```

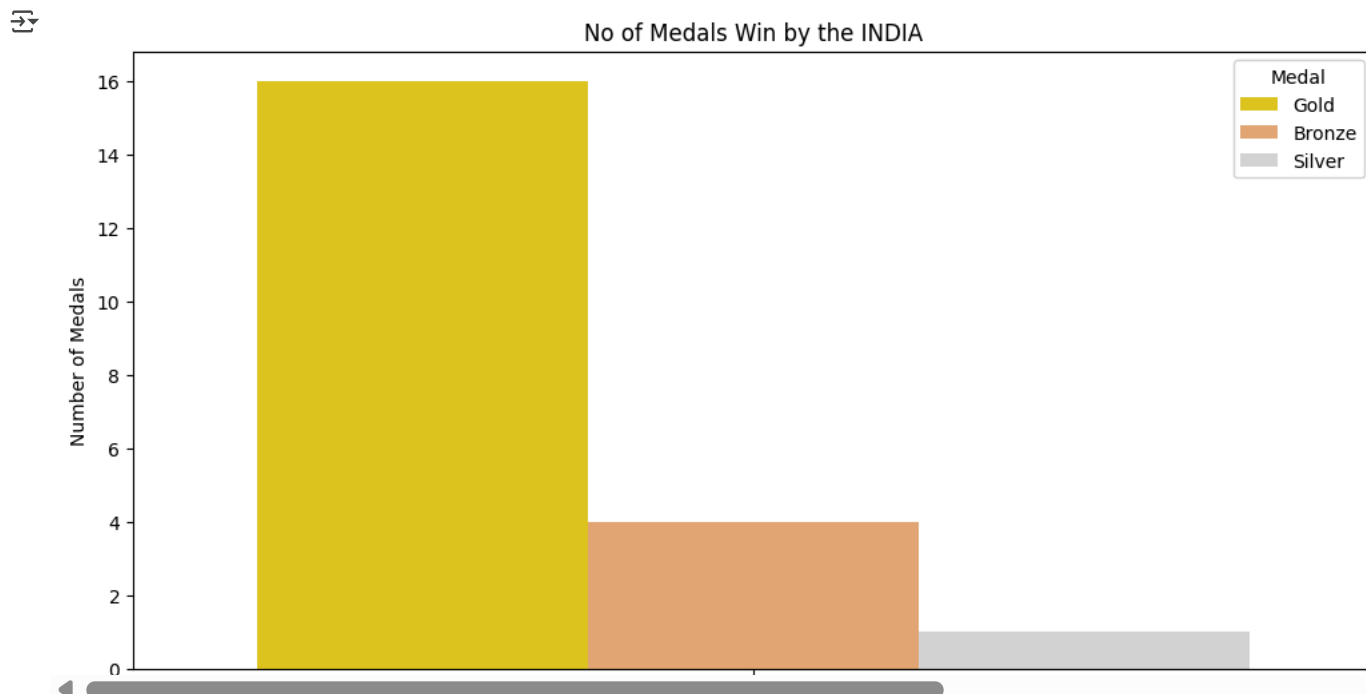


```
India_Medal = Country_wise_Medal_type[Country_wise_Medal_type["Country"]=="India"]
India_Medal
```

	Country	Medal	No of Medals	
127	India	Gold	16	
198	India	Bronze	4	
246	India	Silver	1	

Next steps: [Generate code with India\\_Medal](#) [View recommended plots](#) [New interactive sheet](#)

```
plt.figure(figsize=(12,6))
medal_color = {"Gold" : "#ffdf00", "Silver" : "#d3d3d3", "Bronze": "#f4a460"}
sns.barplot(data =India_Medal, y="No of Medals", hue = "Medal", palette= medal_color)
plt.title("No of Medals Win by the INDIA")
plt.ylabel("Number of Medals")
plt.show()
```



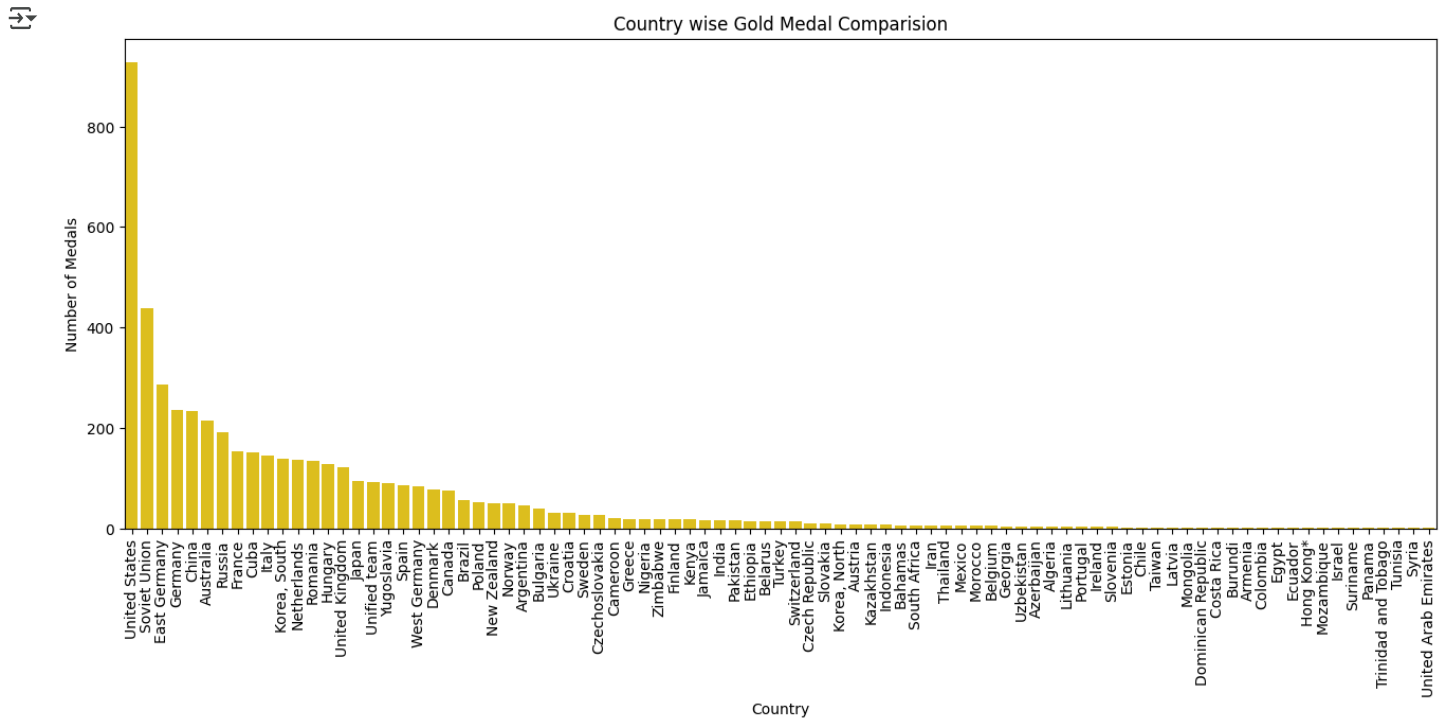
```
Gold_filter= olympic[olympic["Medal"] == "Gold"]
Country_wise_Gold = Gold_filter.groupby(["Country"]).agg({"Medal" : "count"}).reset_index().sort_values(by = "Medal", ascending = False)
Country_wise_Gold
```

	Country	Medal	
82	United States	928	
67	Soviet Union	439	
24	East Germany	286	
32	Germany	237	
15	China	234	
...	...	...	
59	Panama	1	
75	Trinidad and Tobago	1	
76	Tunisia	1	
72	Syria	1	
80	United Arab Emirates	1	

87 rows x 2 columns

Next steps: [Generate code with Country\\_wise\\_Gold](#) [View recommended plots](#) [New interactive sheet](#)

```
plt.figure(figsize=(16,6))
sns.barplot(data = Country_wise_Gold,x = "Country", y="Medal", color = "Gold")
plt.title("Country wise Gold Medal Comparision")
plt.ylabel("Number of Medals")
plt.xticks(rotation=90)
plt.show()
```



```
Country_Sport_Discipline_Medal = olympic.groupby(["Country_Code", "Sport", "Discipline"]).agg({"Medal" : "count"}).reset_index().sort_values("Medal")
Country_Sport_Discipline_Medals = Country_Sport_Discipline_Medal.reset_index(drop=True)
Country_Sport_Discipline_Medals
```

	Country_Code	Sport	Discipline	Medal
0	USA	Aquatics	Swimming	446
1	USA	Athletics	Athletics	299
2	AUS	Aquatics	Swimming	197
3	USA	Basketball	Basketball	192
4	NED	Hockey	Hockey	159
...	...	...	...	...
968	AUT	Equestrian	Dressage	1
969	AUT	Canoe / Kayak	Canoe / Kayak S	1
970	ARG	Aquatics	Swimming	1
971	AFG	Taekwondo	Taekwondo	1
972	AUS	Judo	Judo	1

973 rows x 4 columns

Next steps:

[Generate code with Country\\_Sport\\_Discipline\\_Medals](#)

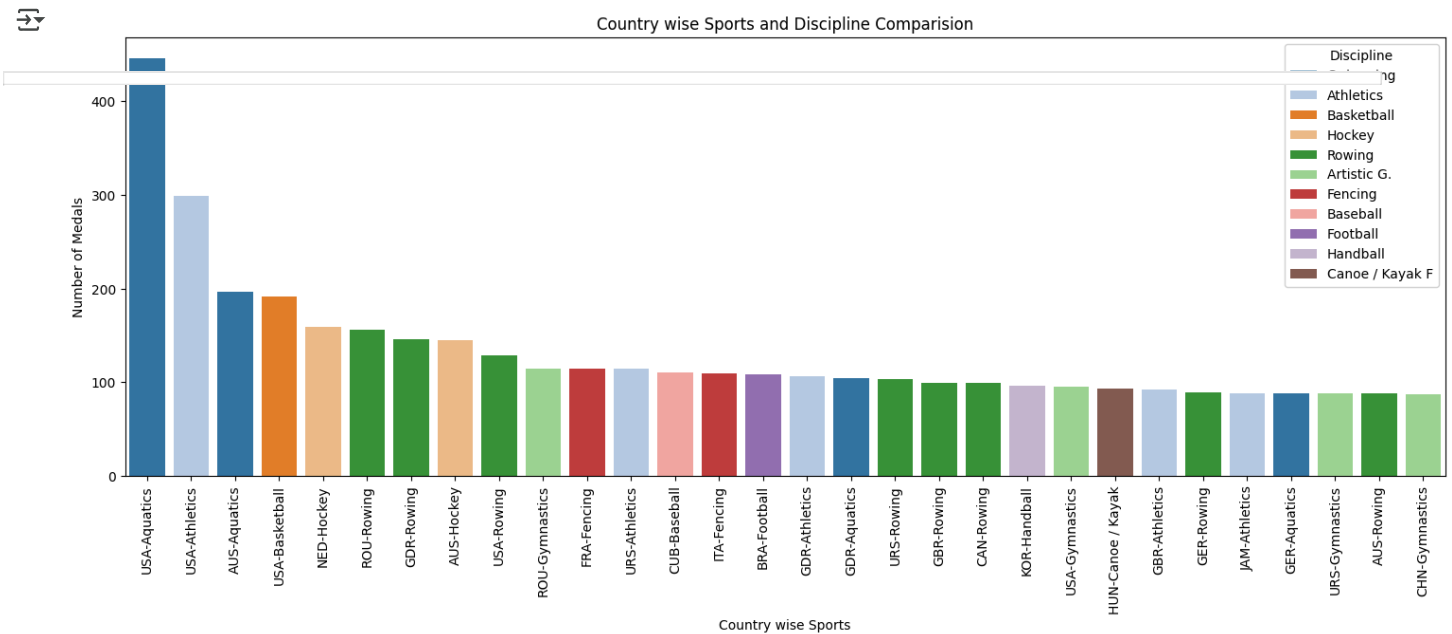
[View recommended plots](#)

[New interactive sheet](#)

```
Country_Sport_Discipline_Medals["Country_Sport"] = Country_Sport_Discipline_Medals["Country_Code"] + "-" +Country_Sport_Discipline_Medals["Sport"]
plt.figure(figsize=(18,6))
sns.barplot(data = Country_Sport_Discipline_Medals.head(30),x = "Country_Sport", y="Medal", hue = "Discipline" , palette = "tab20")
```



```
plt.title("Country wise Sports and Discipline Comparison")
plt.xlabel("Country wise Sports")
plt.ylabel("Number of Medals")
plt.xticks(rotation=90)
plt.show()
```



```
Top_Athlete_Medal = olympic.groupby(["Country", "Athlete"]).agg({"Medal" : "count"}).sort_values(by = "Medal", ascending=False).reset_index()
Top_Athlete_Medal
```

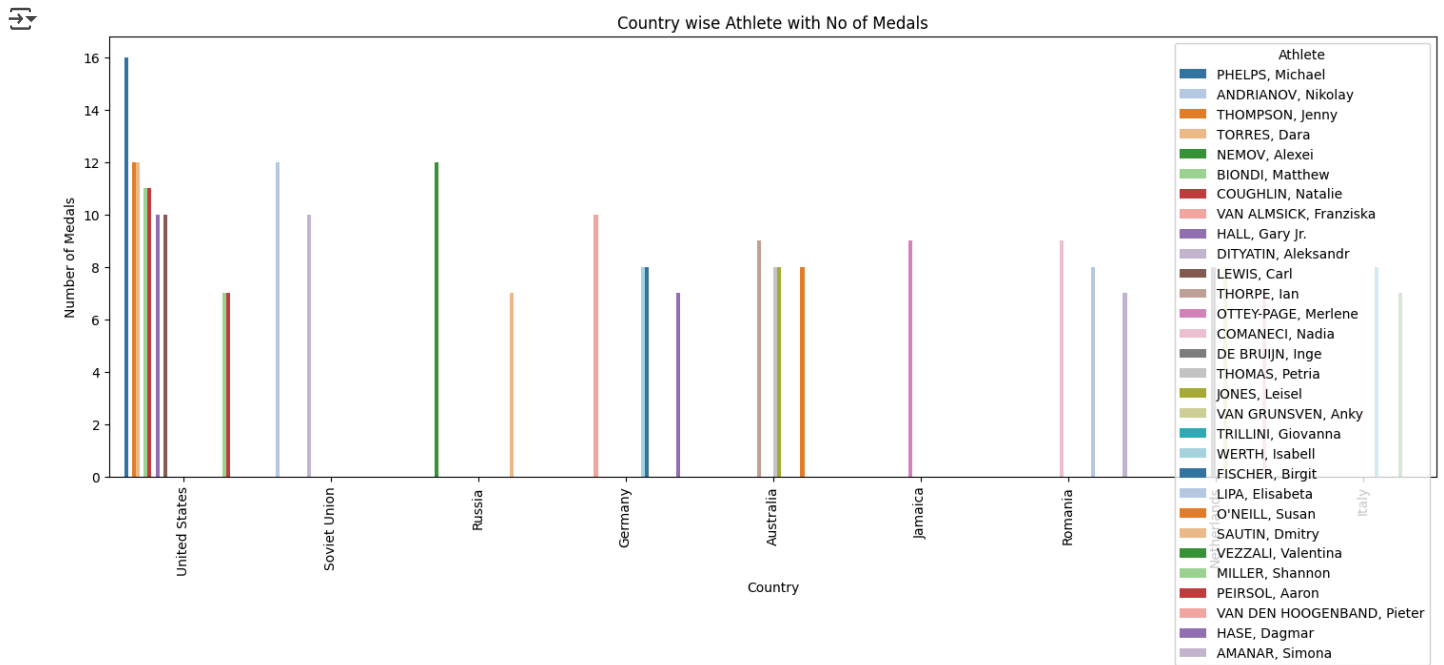
	Country	Athlete	Medal
0	United States	PHELPS, Michael	16
1	Soviet Union	ANDRIANOV, Nikolay	12
2	United States	THOMPSON, Jenny	12
3	United States	TORRES, Dara	12
4	Russia	NEMOV, Alexei	12
...	...	...	...
11535	Hungary	VARGA, Zsolt	1
11536	Hungary	VASKUTI, Istvan	1
11537	Hungary	VEREB, Krisztian	1
11538	Hungary	VERMES, Alban	1
11539	Hungary	VAJDA, Attila Sandor	1

11540 rows x 3 columns

Next steps: [Generate code with Top\\_Athlete\\_Medal](#) [View recommended plots](#) [New interactive sheet](#)

```
plt.figure(figsize=(18,6))
sns.barplot(data = Top_Athlete_Medal.head(30) ,x = "Country", y="Medal", hue = "Athlete" , palette = "tab20")
plt.title("Country wise Athlete with No of Medals")
plt.xlabel("Country")
plt.ylabel("Number of Medals")
```

```
plt.xticks(rotation=90)
plt.show()
```



```
Gold_filter = olympic[olympic["Medal"]=="Gold"]
Gold_Medalist_Athlete_Name = Gold_filter.groupby(["Country", "Athlete"]).agg({"Medal" : "count"}).rename(columns={"Medal": "No of Medals"}).s
Gold_Medalist_Athlete_Name
```

	Country	Athlete	No of Medals
0	United States	PHELPS, Michael	14
1	United States	LEWIS, Carl	9
2	United States	THOMPSON, Jenny	8
3	United States	BIONDI, Matthew	8
4	United States	VAN DYKEN, Amy	6
...	...	...	...
4056	Hungary	VAJDA, Attila Sandor	1
4057	Hungary	VARGA, Daniel Rudolf	1
4058	Hungary	VARGA, Denes Andor	1
4059	Hungary	VARGA, Karoly	1
4060	Hungary	PULAI, Imre	1

4061 rows x 3 columns

Next steps: [Generate code with Gold\\_Medalist\\_Athlete\\_Name](#) [View recommended plots](#) [New interactive sheet](#)

```
plt.figure(figsize=(18,6))
sns.barplot(data = Gold_Medalist_Athlete_Name.head(20) ,x ="Country", y="No of Medals", hue = "Athlete" , palette = "tab20")
plt.title("Country wise Gold Medalist Athlete")
plt.xlabel("Country")
plt.ylabel("Number of Medals")
```