OLYMPIC Q&A ANALYSIS

120 years of Olympic history: atheletes and results

CONTENT

The file athlete_events.csv contains 271116 rows and 15 columns. Each row corresponds to an individual athlete competing in an individual Olympic event (athlete-events). The columns are:

ID - Unique number for each athlete

Name - Athlete's name

Sex - M or F

Age - Integer

Height - In centimeters

Weight - In kilograms

Team - Team name

NOC - National Olympic Committee 3-letter code

Games - Year and season

Year - Integer

Season - Summer or Winter

City - Host city

Sport - Sport

Event - Event

Medal - Gold, Silver, Bronze, or NA

```
In [1]: import pandas as pd
        import numpy as np
```

In [2]:
```python
df = pd.read_csv('athlete_events.csv')
```

In [3]:
```python
df.head()
```

Out[3]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	
0	1	A Dijiang	M	24.0	180.0	80.0	China	CHN	1992 Summer	1992	Summer	Barce
1	2	A Lamusi	M	23.0	170.0	60.0	China	CHN	2012 Summer	2012	Summer	Lor
2	3	Gunnar Nielsen Aaby	M	24.0	NaN	NaN	Denmark	DEN	1920 Summer	1920	Summer	Antwe
3	4	Edgar Lindenau Aabye	M	34.0	NaN	NaN	Denmark/Sweden	DEN	1900 Summer	1900	Summer	F
4	5	Christine Jacoba Aaftink	F	21.0	185.0	82.0	Netherlands	NED	1988 Winter	1988	Winter	Cal

1. How many olympics games have been held?

In [4]:
```python
print('There are {} olympic games held'.format(df['Games'].nunique()))
```
```
There are 51 olympic games held
```

2. List down all Olympics games held so far.

In [5]:
```python
print('The olypic games held are:')
pd.Series(df['Games'].unique().tolist()).to_frame()
```

The olypic games held are:

Out[5]:

	0
0	1992 Summer
1	2012 Summer
2	1920 Summer
3	1900 Summer
4	1988 Winter
5	1992 Winter
6	1994 Winter
7	1932 Summer
8	2002 Winter
9	1952 Summer
10	1980 Winter
11	2000 Summer
12	1996 Summer
13	1912 Summer
14	1924 Summer
15	2014 Winter
16	1948 Summer
17	1998 Winter
18	2006 Winter
19	2008 Summer
20	2016 Summer
21	2004 Summer
22	1960 Winter
23	1964 Winter
24	1984 Winter
25	1984 Summer
26	1968 Summer
27	1972 Summer
28	1988 Summer
29	1936 Summer
30	1952 Winter
31	1956 Winter
32	1956 Summer

	0
33	1960 Summer
34	1928 Summer
35	1976 Summer
36	1980 Summer
37	1964 Summer
38	2010 Winter
39	1968 Winter
40	1906 Summer
41	1972 Winter
42	1976 Winter
43	1924 Winter
44	1904 Summer
45	1928 Winter
46	1908 Summer
47	1948 Winter
48	1932 Winter
49	1936 Winter
50	1896 Summer

3. Mention the total no of nations who participated in each olympics game?

In [6]:
```python
df.groupby('Games').NOC.nunique().to_frame()
```

Out[6]:

Games	NOC
1896 Summer	12
1900 Summer	31
1904 Summer	15
1906 Summer	21
1908 Summer	22
1912 Summer	29
1920 Summer	29
1924 Summer	45
1924 Winter	19
1928 Summer	46
1928 Winter	25
1932 Summer	47
1932 Winter	17
1936 Summer	49
1936 Winter	28
1948 Summer	59
1948 Winter	28
1952 Summer	69
1952 Winter	30
1956 Summer	72
1956 Winter	32
1960 Summer	84
1960 Winter	30
1964 Summer	93
1964 Winter	36
1968 Summer	112
1968 Winter	37
1972 Summer	121
1972 Winter	35
1976 Summer	92
1976 Winter	37
1980 Summer	80
1980 Winter	37
1984 Summer	140

	NOC
Games	
1984 Winter	49
1988 Summer	159
1988 Winter	57
1992 Summer	169
1992 Winter	64
1994 Winter	67
1996 Summer	197
1998 Winter	72
2000 Summer	200
2002 Winter	77
2004 Summer	201
2006 Winter	79
2008 Summer	204
2010 Winter	82
2012 Summer	205
2014 Winter	89
2016 Summer	207

4. Which year saw the highest and lowest no of countries participating in olympics

In [7]:
```python
# country with mininum participation
df.groupby('Games').NOC.nunique().reset_index().min().to_frame()
```

Out[7]:

	0
Games	1896 Summer
NOC	12

In [8]:
```python
# country with mininum participation
df.groupby('Games').NOC.nunique().reset_index().max().to_frame()
```

Out[8]:

	0
Games	2016 Summer
NOC	207

5. Which nation has participated in all of the olympic games

In [9]: `df.groupby('NOC').Games.nunique().reset_index()[df.groupby('NOC').Games.nunique()`

Out[9]:

	NOC	Games
69	FRA	51
74	GBR	51
100	ITA	51
190	SUI	51

6. Identify the sport which was played in all summer olympics.

In [10]:
```python
pd.Series(df[df.Season == 'Summer'].Sport.unique()).to_frame()
```

Out[10]:

	0
0	Basketball
1	Judo
2	Football
3	Tug-Of-War
4	Athletics
5	Swimming
6	Badminton
7	Sailing
8	Gymnastics
9	Art Competitions
10	Handball
11	Weightlifting
12	Wrestling
13	Water Polo
14	Hockey
15	Rowing
16	Fencing
17	Equestrianism
18	Shooting
19	Boxing
20	Taekwondo
21	Cycling
22	Diving
23	Canoeing
24	Tennis
25	Modern Pentathlon
26	Golf
27	Softball
28	Archery
29	Volleyball
30	Synchronized Swimming
31	Table Tennis
32	Baseball
33	Rhythmic Gymnastics
34	Rugby Sevens

	0
35	Trampolining
36	Beach Volleyball
37	Triathlon
38	Rugby
39	Lacrosse
40	Polo
41	Cricket
42	Ice Hockey
43	Racquets
44	Motorboating
45	Croquet
46	Figure Skating
47	Jeu De Paume
48	Roque
49	Basque Pelota
50	Alpinism
51	Aeronautics

7. Which Sports were just played only once in the olympics.

```
In [11]: df.groupby('Sport').Games.nunique()[df.groupby('Sport').Games.nunique() == 1].to_
```

Out[11]:

Sport	Games
Aeronautics	1
Basque Pelota	1
Cricket	1
Croquet	1
Jeu De Paume	1
Military Ski Patrol	1
Motorboating	1
Racquets	1
Roque	1
Rugby Sevens	1

8. Fetch the total no of sports played in each olympic games.

In [12]:
```python
df.groupby('Games').Sport.nunique().to_frame()
```

Out[12]:

	Sport
Games	
1896 Summer	9
1900 Summer	20
1904 Summer	18
1906 Summer	13
1908 Summer	24
1912 Summer	17
1920 Summer	25
1924 Summer	20
1924 Winter	10
1928 Summer	17
1928 Winter	8
1932 Summer	18
1932 Winter	7
1936 Summer	24
1936 Winter	8
1948 Summer	20
1948 Winter	9
1952 Summer	19
1952 Winter	8
1956 Summer	19
1956 Winter	8
1960 Summer	19
1960 Winter	8
1964 Summer	21
1964 Winter	10
1968 Summer	20
1968 Winter	10
1972 Summer	23
1972 Winter	10
1976 Summer	23
1976 Winter	10
1980 Summer	23
1980 Winter	10
1984 Summer	25

	Sport
Games	
1984 Winter	10
1988 Summer	27
1988 Winter	10
1992 Summer	29
1992 Winter	12
1994 Winter	12
1996 Summer	31
1998 Winter	14
2000 Summer	34
2002 Winter	15
2004 Summer	34
2006 Winter	15
2008 Summer	34
2010 Winter	15
2012 Summer	32
2014 Winter	15
2016 Summer	34

9. Fetch oldest athletes to win a gold medal

In [13]:
```python
df[(df.Medal == 'Gold') & (df.Age == df[(df.Medal == 'Gold')].Age.max())]
```

Out[13]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	
105199	53238	Charles Jacobus	M	64.0	NaN	NaN	United States	USA	1904 Summer	1904	Summer	St.
233390	117046	Oscar Gomer Swahn	M	64.0	NaN	NaN	Sweden	SWE	1912 Summer	1912	Summer	Stock

10. Find the Ratio of male and female athletes participated in all olympic games.

In [14]:
```python
df.groupby('Sex').Name.nunique().to_frame()
```

Out[14]:

Sex	Name
F	33808
M	100979

In [15]:
```python
ratio = round(df.groupby('Sex').Name.nunique()[1] / df.groupby('Sex').Name.nuniqu
'Ratio of male and female athletes participated in all olympic games is {} : {}'.
```

Out[15]: 'Ratio of male and female athletes participated in all olympic games is 1 : 2.9
9'

11. Fetch the top 5 athletes who have won the most gold medals.

In [16]:
```python
df[df.Medal == 'Gold'].groupby('Name').Medal.count().sort_values(ascending = Fals
```

Out[16]:

Name	Medal
Michael Fred Phelps, II	23
Raymond Clarence "Ray" Ewry	10
Paavo Johannes Nurmi	9
Larysa Semenivna Latynina (Diriy-)	9
Mark Andrew Spitz	9

12. Fetch the top 5 athletes who have won the most medals (gold/silver/bronze).

In [17]:
```python
df[df.Medal.notna()].groupby('Name').Medal.count().sort_values(ascending = False)
```

Out[17]:

Name	Medal
Michael Fred Phelps, II	28
Larysa Semenivna Latynina (Diriy-)	18
Nikolay Yefimovich Andrianov	15
Borys Anfiyanovych Shakhlin	13
Takashi Ono	13

13. Fetch the top 5 most successful countries in olympics. Success is defined by no of medals won.

In [18]:
```python
df[df.Medal.notna()].groupby('NOC').Medal.count().sort_values(ascending = False).
```

Out[18]:

	Medal
NOC	
USA	5637
URS	2503
GER	2165
GBR	2068
FRA	1777

14. List down total gold, silver and bronze medals won by each country.

In [19]:
```python
# country by gold count
gold_count = pd.Series(df[df.Medal == 'Gold'].groupby('NOC').Medal.count(), name
# country by silver count
silver_count = pd.Series(df[df.Medal == 'Silver'].groupby('NOC').Medal.count(), n
# country by bronze count
bronze_count = pd.Series(df[df.Medal == 'Bronze'].groupby('NOC').Medal.count(), n
# country by medal count
medal_count = pd.Series(df[df.Medal.notna()].groupby('NOC').Medal.count(), name =
```

In [20]: `pd.concat([gold_count, silver_count, bronze_count, medal_count], axis=1)`

Out[20]:

NOC	gold_count	silver_count	bronze_count	medal_count
ALG	5.0	4.0	8.0	17
ANZ	20.0	4.0	5.0	29
ARG	91.0	92.0	91.0	274
ARM	2.0	5.0	9.0	16
AUS	348.0	455.0	517.0	1320
...	...	...	...	...
MKD	NaN	NaN	1.0	1
MON	NaN	NaN	1.0	1
MRI	NaN	NaN	1.0	1
TOG	NaN	NaN	1.0	1
WIF	NaN	NaN	5.0	5

149 rows × 4 columns

15. List down total gold, silver and bronze medals won by each country corresponding to each olympic games.

In [21]:
```python
# total gold by country in each olympic game
no_gold = pd.Series(df[df.Medal == 'Gold'].groupby(['Games', 'NOC']).Medal.count(
# total silver by country in each olympic game
no_silver = pd.Series(df[df.Medal == 'Silver'].groupby(['Games', 'NOC']).Medal.co
# total bronze by country in each olympic game
no_bronze = pd.Series(df[df.Medal == 'Bronze'].groupby(['Games', 'NOC']).Medal.co

pd.concat([no_gold, no_silver, no_bronze], axis = 1)
```

Out[21]:

Games	NOC	no_gold	no_silver	no_bronze
	AUS	2.0	NaN	1.0
	AUT	2.0	1.0	2.0
1896 Summer	**DEN**	1.0	2.0	3.0
	FRA	5.0	4.0	2.0
	GBR	3.0	3.0	3.0
...	...	...	...	...
	NOR	NaN	NaN	19.0
	POR	NaN	NaN	1.0
2016 Summer	**TTO**	NaN	NaN	1.0
	TUN	NaN	NaN	3.0
	UAE	NaN	NaN	1.0

1657 rows × 3 columns

16. Identify which country won the most gold, most silver and most bronze medals in each olympic games.

In [22]:
```python
# calculation for gold
gold = df[df.Medal == 'Gold'].groupby(['Games', 'NOC']).Medal.count()
gold_group = gold.reset_index()
#gold_group
max_gold = gold.groupby('Games').max().reset_index()
#max_gold
most_gold = pd.merge(gold_group, max_gold, on=['Games', 'Medal'])
most_gold['most_gold'] = most_gold.NOC + ' - ' + most_gold.Medal.astype(str)
most_gold.drop(['NOC', 'Medal'], axis = 1, inplace = True)
#most_gold
```

In [23]:
```python
# calculation for silver
silver = df[df.Medal == 'Silver'].groupby(['Games', 'NOC']).Medal.count()
silver_group = silver.reset_index()
#silver_group
max_silver = silver.groupby('Games').max().reset_index()
#max_silver
most_silver = pd.merge(silver_group, max_silver, on=['Games', 'Medal'])
most_silver['most_silver'] = most_silver.NOC + ' - ' + most_silver.Medal.astype(s
most_silver.drop(['NOC', 'Medal'], axis = 1, inplace = True)
#most_silver
```

In [24]:
```python
# calculation for bronze
bronze = df[df.Medal == 'Bronze'].groupby(['Games', 'NOC']).Medal.count()
bronze_group = bronze.reset_index()
#bronze_group
max_bronze = bronze.groupby('Games').max().reset_index()
max_bronze
most_bronze = pd.merge(bronze_group, max_bronze, on = ['Games', 'Medal'])
most_bronze['most_bronze'] = most_bronze.NOC + ' - ' + most_bronze.Medal.astype(s
most_bronze.drop(['NOC', 'Medal'], axis = 1, inplace = True)
#most_bronze
```

```
In [25]: most_medal = pd.merge(most_gold, most_silver, on = 'Games')
         most_medal = pd.merge(most_medal, most_bronze, on = 'Games')
         most_medal
```

Out[25]:

	Games	most_gold	most_silver	most_bronze
0	1896 Summer	GER - 25	GRE - 18	GRE - 20
1	1900 Summer	GBR - 59	FRA - 101	FRA - 82
2	1904 Summer	USA - 128	USA - 141	USA - 125
3	1906 Summer	GRE - 24	GRE - 48	GRE - 30
4	1908 Summer	GBR - 147	GBR - 131	GBR - 90
5	1912 Summer	SWE - 103	GBR - 64	GBR - 59
6	1920 Summer	USA - 111	FRA - 71	BEL - 66
7	1924 Summer	USA - 97	FRA - 51	USA - 49
8	1924 Winter	GBR - 16	USA - 10	GBR - 11
9	1928 Summer	USA - 47	NED - 29	GER - 41
10	1928 Winter	CAN - 12	SWE - 13	SUI - 12
11	1932 Summer	USA - 81	USA - 47	USA - 61
12	1932 Winter	CAN - 14	USA - 21	GER - 14
13	1936 Summer	GER - 93	GER - 70	GER - 61
14	1936 Winter	GBR - 12	CAN - 13	USA - 14
15	1948 Summer	USA - 87	GBR - 42	USA - 35
16	1948 Winter	CAN - 13	TCH - 17	SUI - 19
17	1952 Summer	USA - 83	URS - 62	HUN - 32
18	1952 Winter	CAN - 16	USA - 25	SWE - 23
19	1956 Summer	URS - 68	URS - 46	URS - 55
20	1956 Winter	URS - 26	USA - 19	CAN - 18
21	1960 Summer	USA - 81	URS - 63	URS - 45
22	1960 Winter	USA - 19	CAN - 17	URS - 28
23	1964 Summer	USA - 95	URS - 63	URS - 51
24	1964 Winter	URS - 30	SWE - 21	TCH - 17
25	1968 Summer	USA - 99	URS - 63	URS - 64
26	1968 Winter	URS - 26	TCH - 19	CAN - 18
27	1972 Summer	URS - 107	GDR - 64	URS - 60
28	1972 Winter	URS - 36	USA - 18	TCH - 19
29	1976 Summer	URS - 114	URS - 95	URS - 77
30	1976 Winter	URS - 38	TCH - 19	FRG - 23
31	1980 Summer	URS - 187	URS - 129	URS - 126
32	1980 Winter	USA - 24	URS - 29	SWE - 20

	Games	most_gold	most_silver	most_bronze
33	1984 Summer	USA - 186	USA - 116	FRG - 53
34	1984 Winter	URS - 29	TCH - 24	SWE - 21
35	1988 Summer	URS - 134	URS - 67	URS - 99
36	1988 Winter	URS - 40	FIN - 22	SWE - 23
37	1992 Summer	EUN - 92	EUN - 61	USA - 85
38	1992 Winter	EUN - 35	CAN - 28	TCH - 27
39	1994 Winter	SWE - 23	CAN - 29	FIN - 29
40	1996 Summer	USA - 159	CHN - 70	AUS - 84
41	1998 Winter	USA - 25	RUS - 32	FIN - 49
42	2000 Summer	USA - 130	AUS - 69	GER - 64
43	2002 Winter	CAN - 52	USA - 58	RUS - 27
44	2004 Summer	USA - 117	AUS - 77	RUS - 95
45	2006 Winter	SWE - 35	FIN - 34	USA - 32
46	2008 Summer	USA - 127	USA - 110	USA - 80
47	2010 Winter	CAN - 67	USA - 63	FIN - 46
48	2012 Summer	USA - 145	USA - 57	AUS - 59
49	2014 Winter	CAN - 59	SWE - 32	FIN - 24
50	2014 Winter	CAN - 59	SWE - 32	USA - 24
51	2016 Summer	USA - 139	FRA - 55	USA - 71
52	2016 Summer	USA - 139	GBR - 55	USA - 71

17. Identify which country won the most gold, most silver, most bronze medals and the most medals in each olympic games.

In [26]:
```python
medal = df[df.Medal.notna()].groupby(['Games', 'NOC']).Medal.count()
medal_group = medal.reset_index()
#medal_group
max_medal = medal.groupby('Games').max().reset_index()
#max_medal
most_max_medal = pd.merge(medal_group, max_medal, on =['Games', 'Medal'])
most_max_medal['most_medal'] = most_max_medal.NOC + ' - ' + most_max_medal.Medal.
most_max_medal.drop(['NOC', 'Medal'], axis = 1, inplace = True)
#most_max_medal
```

In [27]:
```python
max_medal_summary = pd.merge(most_medal, most_max_medal, on = 'Games')
max_medal_summary
```

Out[27]:

	Games	most_gold	most_silver	most_bronze	most_medal
0	1896 Summer	GER - 25	GRE - 18	GRE - 20	GRE - 48
1	1900 Summer	GBR - 59	FRA - 101	FRA - 82	FRA - 235
2	1904 Summer	USA - 128	USA - 141	USA - 125	USA - 394
3	1906 Summer	GRE - 24	GRE - 48	GRE - 30	GRE - 102
4	1908 Summer	GBR - 147	GBR - 131	GBR - 90	GBR - 368
5	1912 Summer	SWE - 103	GBR - 64	GBR - 59	SWE - 190
6	1920 Summer	USA - 111	FRA - 71	BEL - 66	USA - 194
7	1924 Summer	USA - 97	FRA - 51	USA - 49	USA - 182
8	1924 Winter	GBR - 16	USA - 10	GBR - 11	GBR - 31
9	1928 Summer	USA - 47	NED - 29	GER - 41	USA - 88
10	1928 Winter	CAN - 12	SWE - 13	SUI - 12	SWE - 16
11	1932 Summer	USA - 81	USA - 47	USA - 61	USA - 189
12	1932 Winter	CAN - 14	USA - 21	GER - 14	USA - 34
13	1936 Summer	GER - 93	GER - 70	GER - 61	GER - 224
14	1936 Winter	GBR - 12	CAN - 13	USA - 14	NOR - 18
15	1948 Summer	USA - 87	GBR - 42	USA - 35	USA - 152
16	1948 Winter	CAN - 13	TCH - 17	SUI - 19	SUI - 28
17	1952 Summer	USA - 83	URS - 62	HUN - 32	USA - 134
18	1952 Winter	CAN - 16	USA - 25	SWE - 23	USA - 30
19	1956 Summer	URS - 68	URS - 46	URS - 55	URS - 169
20	1956 Winter	URS - 26	USA - 19	CAN - 18	URS - 37
21	1960 Summer	USA - 81	URS - 63	URS - 45	URS - 169
22	1960 Winter	USA - 19	CAN - 17	URS - 28	URS - 42
23	1964 Summer	USA - 95	URS - 63	URS - 51	URS - 174
24	1964 Winter	URS - 30	SWE - 21	TCH - 17	URS - 47
25	1968 Summer	USA - 99	URS - 63	URS - 64	URS - 192
26	1968 Winter	URS - 26	TCH - 19	CAN - 18	URS - 37
27	1972 Summer	URS - 107	GDR - 64	URS - 60	URS - 214
28	1972 Winter	URS - 36	USA - 18	TCH - 19	URS - 45
29	1976 Summer	URS - 114	URS - 95	URS - 77	URS - 286
30	1976 Winter	URS - 38	TCH - 19	FRG - 23	URS - 56
31	1980 Summer	URS - 187	URS - 129	URS - 126	URS - 442
32	1980 Winter	USA - 24	URS - 29	SWE - 20	URS - 54
33	1984 Summer	USA - 186	USA - 116	FRG - 53	USA - 352

	Games	most_gold	most_silver	most_bronze	most_medal
34	1984 Winter	URS - 29	TCH - 24	SWE - 21	URS - 56
35	1988 Summer	URS - 134	URS - 67	URS - 99	URS - 300
36	1988 Winter	URS - 40	FIN - 22	SWE - 23	URS - 66
37	1992 Summer	EUN - 92	EUN - 61	USA - 85	USA - 224
38	1992 Winter	EUN - 35	CAN - 28	TCH - 27	EUN - 59
39	1994 Winter	SWE - 23	CAN - 29	FIN - 29	GER - 40
40	1996 Summer	USA - 159	CHN - 70	AUS - 84	USA - 259
41	1998 Winter	USA - 25	RUS - 32	FIN - 49	FIN - 58
42	2000 Summer	USA - 130	AUS - 69	GER - 64	USA - 242
43	2002 Winter	CAN - 52	USA - 58	RUS - 27	USA - 84
44	2004 Summer	USA - 117	AUS - 77	RUS - 95	USA - 263
45	2006 Winter	SWE - 35	FIN - 34	USA - 32	CAN - 69
46	2008 Summer	USA - 127	USA - 110	USA - 80	USA - 317
47	2010 Winter	CAN - 67	USA - 63	FIN - 46	USA - 97
48	2012 Summer	USA - 145	USA - 57	AUS - 59	USA - 248
49	2014 Winter	CAN - 59	SWE - 32	FIN - 24	CAN - 86
50	2014 Winter	CAN - 59	SWE - 32	USA - 24	CAN - 86
51	2016 Summer	USA - 139	FRA - 55	USA - 71	USA - 264
52	2016 Summer	USA - 139	GBR - 55	USA - 71	USA - 264

18. Which countries have never won gold medal but have won silver/bronze medals?

In [28]: `df[df.Medal != 'Gold'].groupby('NOC').Medal.count()[df[df.Medal != 'Gold'].groupt`

Out[28]:

	Medal
NOC	
AFG	2
AHO	1
ALG	12
ANZ	9
ARG	183
...	...
VIE	3
WIF	5
YUG	260
ZAM	2
ZIM	5

145 rows × 1 columns

19. In which Sport/event, India has won highest medals.

In [29]:
```python
event = df[df.Medal.notna()].groupby(['Event', 'NOC']).Medal.count()
event_medal = event.reset_index()
#event_medal
event_max = event.groupby('Event').max().reset_index()
#event_max
event_max_medal = pd.merge(event_medal, event_max, on = ['Event', 'Medal'])
event_max_medal.set_index('NOC').loc['IND'].to_frame()
```

Out[29]:

	IND
Event	Hockey Men's Hockey
Medal	173

20. Break down all olympic games where India won medal for Hockey and how many medals in each olympic games

```
In [30]: pd.Series(df[df.Medal.notna()].groupby(['NOC', 'Sport']).get_group(('IND', 'Hocke
```

Out[30]:

	0
0	1928 Summer
2	1932 Summer
3	1936 Summer
8	1948 Summer
9	1952 Summer
10	1956 Summer
4	1960 Summer
1	1964 Summer
7	1968 Summer
6	1972 Summer
5	1980 Summer