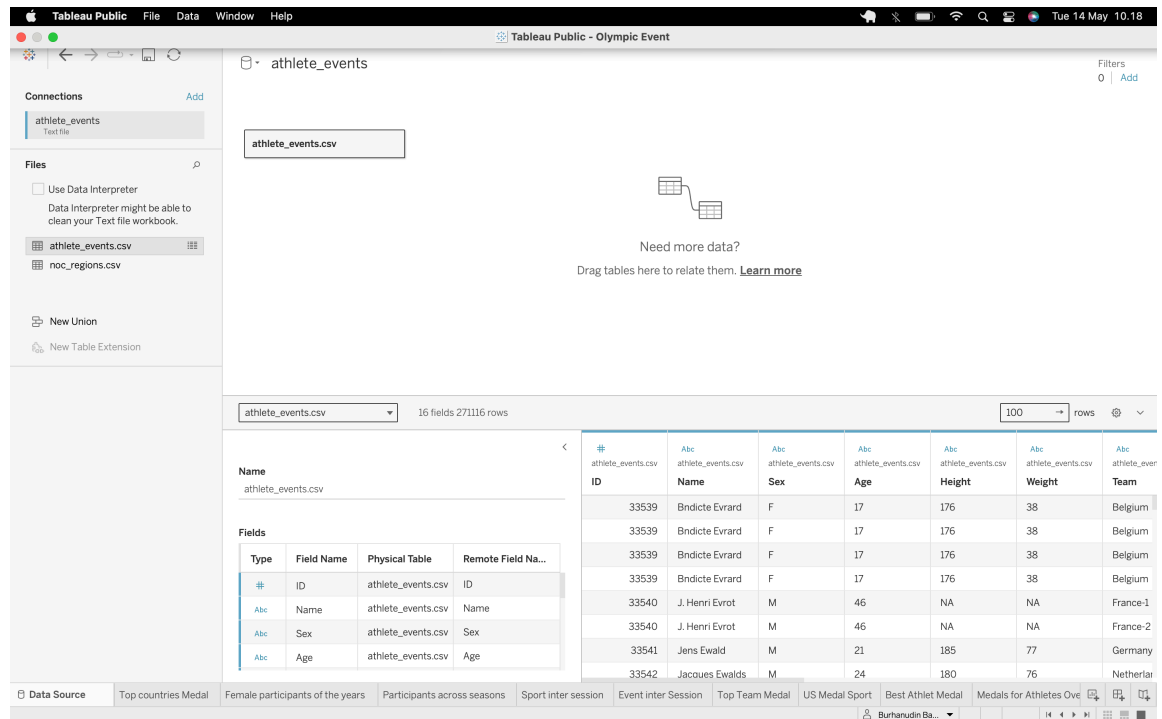


Creating worksheets and dashboards

Early step our study is how to create visualization

- Navigate to [Fundamentals Visualization](#)
- Dashboard will be connected to two [datasource 'Summer-Olympics'](#)
- We create new workbook with the name file of `Olympict Event` and datasource refer to above file *.csv



Combine with python w'll explore about our dataset

```
In [1]: # Import Library
import numpy as np
import pandas as pd
import seaborn as sns
import os
import opendatasets as od
import matplotlib
import matplotlib.pyplot as plt
plt.style.use("ggplot") #using style ggplot

# setting parameter visualization
sns.set_style('darkgrid')
matplotlib.rcParams['font.size'] = 14
matplotlib.rcParams['figure.figsize'] = (9, 5)
matplotlib.rcParams['figure.facecolor'] = '#00000000'
```

```
# disable Anaconda warnings
import warnings
warnings.simplefilter('ignore')

%matplotlib inline
```

```
In [2]: import requests
url = 'https://raw.githubusercontent.com/PacktPublishing/Mastering-Tableau-2
olympics = pd.read_csv(url, encoding="windows_1258", index_col=0) # Add encod
olympics.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 15433 entries, Montreal to Beijing
Data columns (total 10 columns):
#   Column          Non-Null Count  Dtype
---  -
0   Year            15316 non-null  float64
1   Sport           15316 non-null  object
2   Discipline       15316 non-null  object
3   Event           15316 non-null  object
4   Athlete         15316 non-null  object
5   Gender          15316 non-null  object
6   Country_Code    15316 non-null  object
7   Country         15316 non-null  object
8   Event_gender    15316 non-null  object
9   Medal          15316 non-null  object
dtypes: float64(1), object(9)
memory usage: 1.3+ MB
```

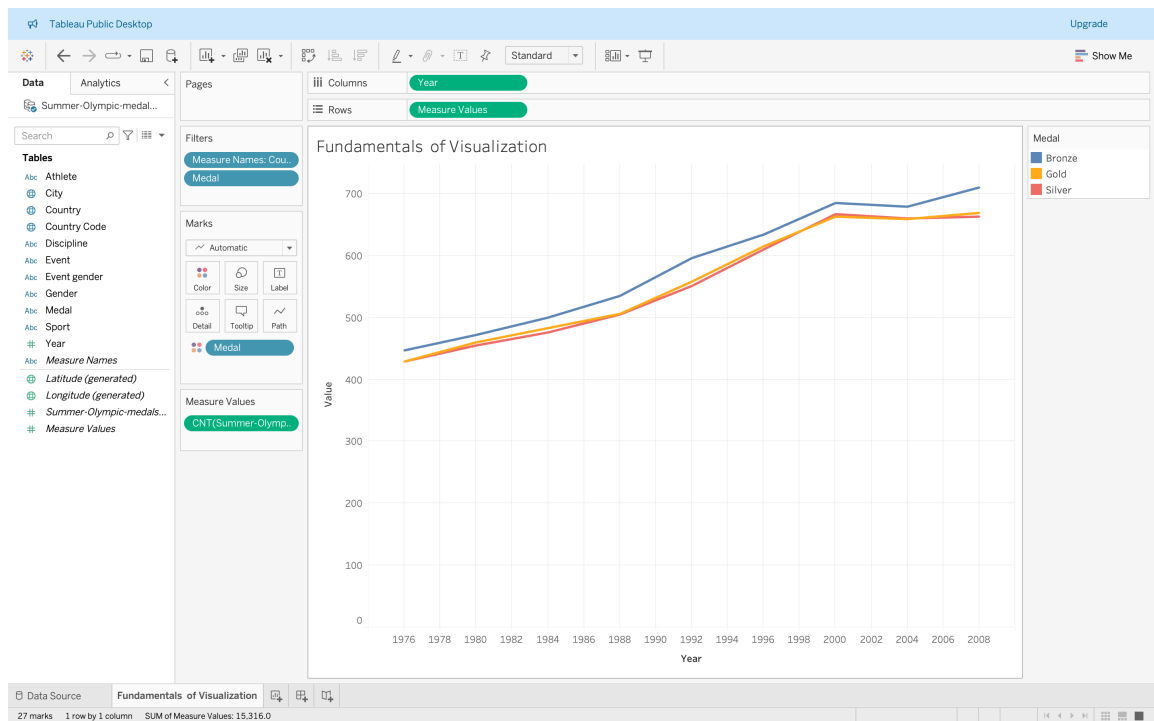
Majority datatype in **Object** format

```
In [3]: # View sampling row of data
olympics.head(3)
```

```
Out[3]:
```

	Year	Sport	Discipline	Event	Athlete	Gender	Country_Code
City							
Montreal	1976.0	Aquatics	Diving	3m springboard	KÖHLER, Christa	Women	GDR
Montreal	1976.0	Aquatics	Diving	3m springboard	KOSENKOV, Aleksandr	Men	URS
Montreal	1976.0	Aquatics	Diving	3m springboard	BOGGS, Philip George	Men	USA

- Display perolehan emas-perak-perunggu di setiap kurun waktu



Sorting and nested sorting

- T to sort the countries by the amount of medals won but separately for gold, silver, and bronze.

Let's try analysis this datasheet mor detile in case Historical Olympics dataset

```
In [4]: # Download the data set files

# Assign the Kaggle data set URL into variable
dataset = 'https://www.kaggle.com/datasets/heesoo37/120-years-of-olympic-his
# Using opendatasets let's download the data sets
od.download(dataset)
```

Skipping, found downloaded files in "./120-years-of-olympic-history-athletes-and-results" (use force=True to force download)

```
In [5]: data_dir = './120-years-of-olympic-history-athletes-and-results'
```

```
In [6]: os.listdir(data_dir)
```

```
Out[6]: ['noc_regions.csv', 'athlete_events.csv']
```

```
In [7]: #Importing the dataset and view column
```

```
olympic_event = pd.read_csv('./120-years-of-olympic-history-athletes-and-res
event.columns
```

```
Out[7]: Index(['ID', 'Name', 'Sex', 'Age', 'Height', 'Weight', 'Team', 'NOC', 'Games',  
            'Year', 'Season', 'City', 'Sport', 'Event', 'Medal'],  
            dtype='object')
```

```
In [8]: olympic_event.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 271116 entries, 0 to 271115  
Data columns (total 15 columns):  
#   Column      Non-Null Count  Dtype  
---  ---  
0   ID          271116 non-null  int64  
1   Name        271116 non-null  object  
2   Sex         271116 non-null  object  
3   Age         261642 non-null  float64  
4   Height      210945 non-null  float64  
5   Weight      208241 non-null  float64  
6   Team        271116 non-null  object  
7   NOC         271116 non-null  object  
8   Games       271116 non-null  object  
9   Year        271116 non-null  int64  
10  Season      271116 non-null  object  
11  City        271116 non-null  object  
12  Sport       271116 non-null  object  
13  Event       271116 non-null  object  
14  Medal       39783 non-null   object  
dtypes: float64(3), int64(2), object(10)  
memory usage: 31.0+ MB
```

```
In [9]: olympic_event.shape
```

```
Out[9]: (271116, 15)
```

```
In [10]: olympic_event.head()
```

Out [10]:

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

```
In [11]: # identify the columns containing null values

nan_values = olympic_event.isna()           # define variable contain NaN values
nan_columns = nan_values.any()              # define column contain NaN

columns_with_nan = olympic_event.columns[nan_columns].tolist()    # list of columns with NaN
print(columns_with_nan)

['Age', 'Height', 'Weight', 'Medal']
```

As `Age`, `height`, `weight` are numerical columns. Replacing those values by **zero**.

```
In [12]: olympic_event[['Age', 'Height', 'Weight']] = olympic_event[['Age', 'Height', 'Weight']].fillna(0)
```

For `Medal` I will replace the NaN values by None

```
In [13]: olympic_event.Medal = olympic_event.Medal.fillna('None')
```

Also converting the `Age` fielding to integer

```
In [14]: olympic_event.Age = olympic_event.Age.astype(int)
```

```
In [15]: # View again dataset
olympic_event.head()
```

Out[15]:	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Se
0	1	A Dijiang	M	24	180.0	80.0	China	CHN	1992 Summer	1992	Su
1	2	A Lamusi	M	23	170.0	60.0	China	CHN	2012 Summer	2012	Su
2	3	Gunnar Nielsen Aaby	M	24	0.0	0.0	Denmark	DEN	1920 Summer	1920	Su
3	4	Edgar Lindenau Aabye	M	34	0.0	0.0	Denmark/Sweden	DEN	1900 Summer	1900	Su
4	5	Christine Jacoba Aaftink	F	21	185.0	82.0	Netherlands	NED	1988 Winter	1988	V

Exploratory Analysis and Visualization Olympics

Participants ' demographics , i.e., country, age, gender etc. It's essential to explore these variables to understand how representative the participants is of the worldwide sports community.

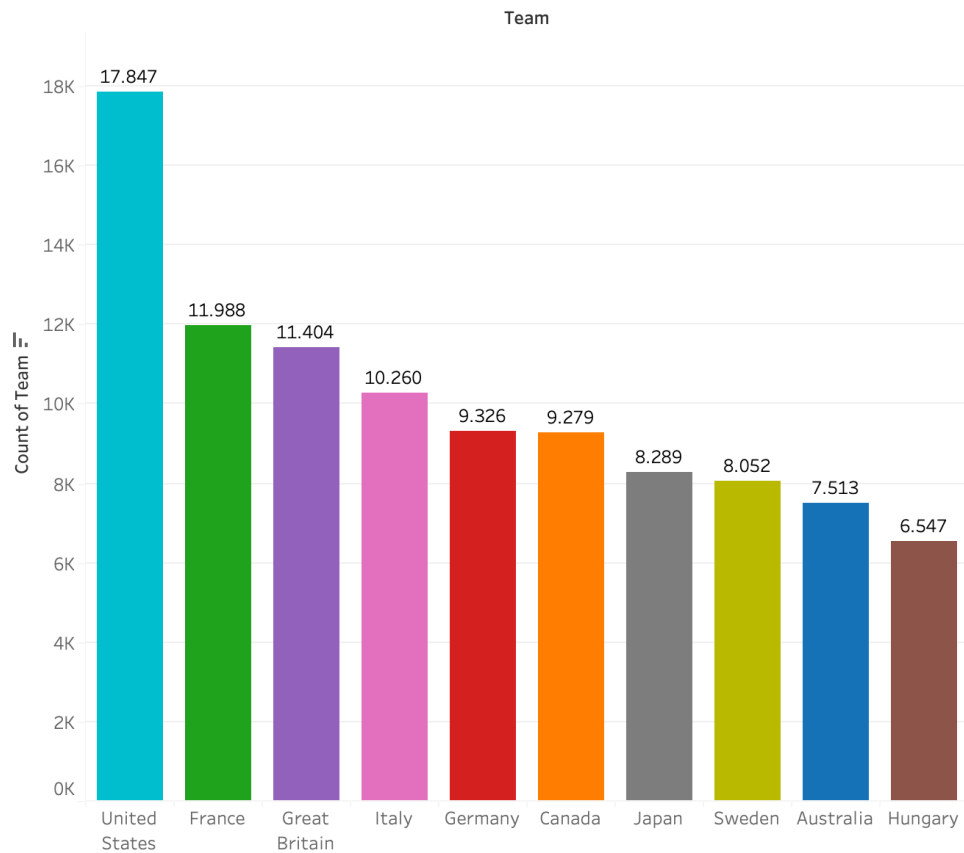
1. Top countries participating in Olympics

```
In [16]: top_countries = olympic_event.Team.value_counts().sort_values(ascending=False)
top_countries
```

```
Out[16]: Team
United States    17847
France           11988
Great Britain    11404
Italy            10260
Germany          9326
Canada           9279
Japan            8289
Sweden           8052
Australia        7513
Hungary          6547
Name: count, dtype: int64
```

Visualization with tableau

Top countries participating



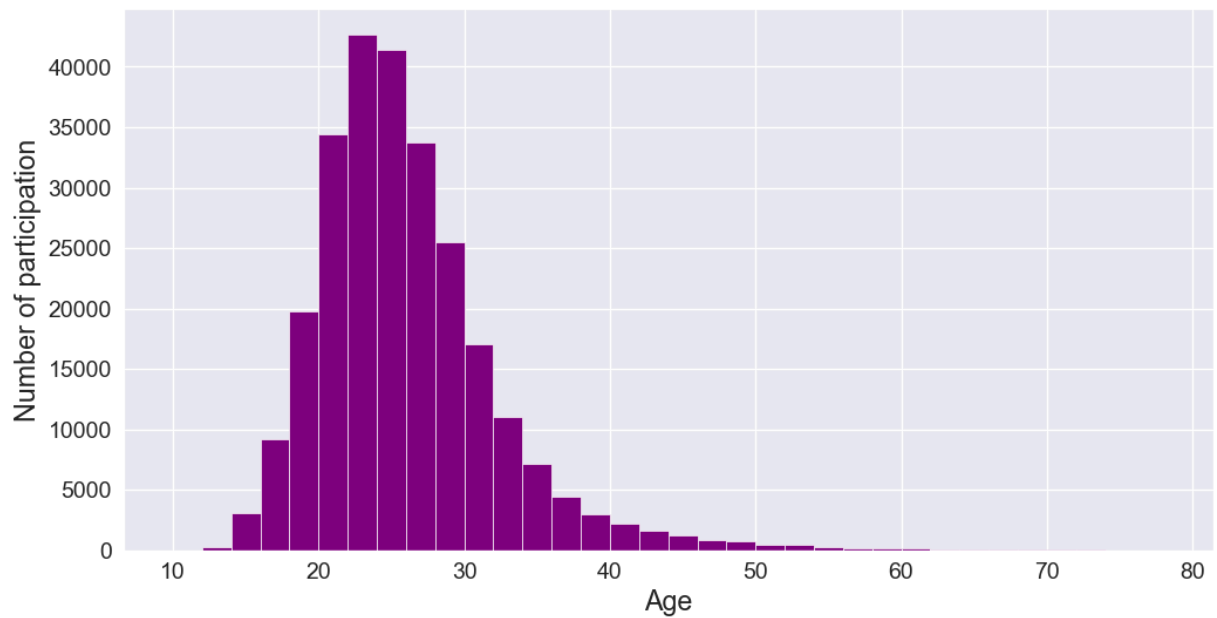
As USA has historically won maximum no of medals it would make sense the participation is highest from US. Surprisingly Soviet Union is not present in the list of top 10 countries.

2. Age Distribution

```
In [17]: import numpy as np
plt.figure(figsize=(12,6))

# plt.title(olympics_event.Age
plt.xlabel('Age')
plt.ylabel('Number of participation')

plt.hist(olympic_event.Age, bins=np.arange(10,80,2), color='purple');
```



From the above distribution we observe maximum participants are of age between 22–26 years, Which would make sense as it is likely for people with less age would perform better in active sport.

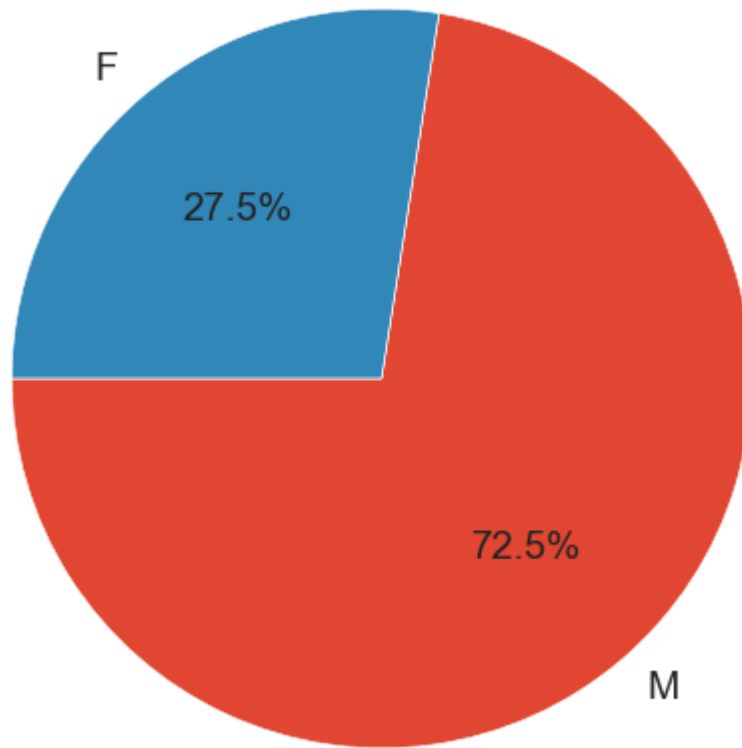
3. Gender Distribution

```
In [18]: gender_account = olympic_event.Sex.value_counts()
gender_account
```

```
Out[18]: Sex
M      196594
F       74522
Name: count, dtype: int64
```

```
In [19]: plt.figure(figsize=(12,6))
plt.title('Gender Distribution')
plt.pie(gender_account, labels=gender_account.index, autopct='%1.1f%%', star
```


Gender Distribution

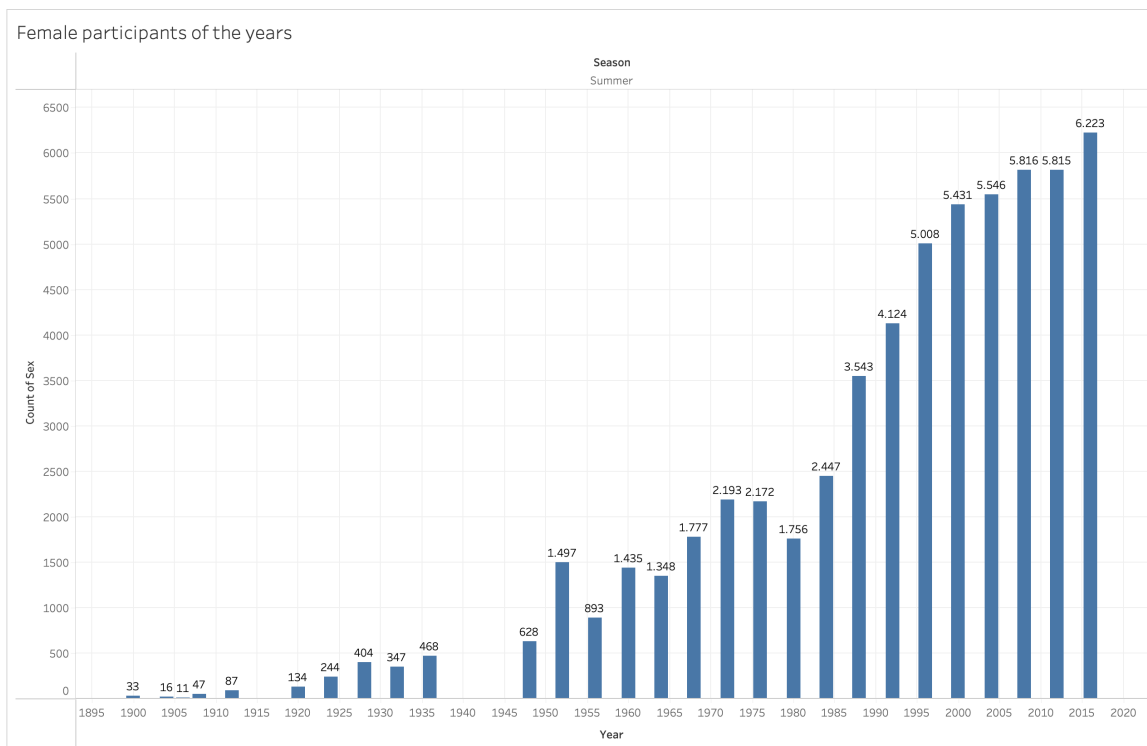


Male seems to be dominating in terms of participation with 72.5% dominated. Let us check the female participants of the years 1900 to 2016

```
In [21]: female_participants = olympic_event[(olympic_event.Sex=='F') & (olympic_event.Year>1900)]
female_participants = female_participants.groupby('Year').count().reset_index()
female_participants
```

Out[21]:

	Year	Sex
0	1900	33
1	1904	16
2	1906	11
3	1908	47
4	1912	87
5	1920	134
6	1924	244
7	1928	404
8	1932	347
9	1936	468
10	1948	628
11	1952	1497
12	1956	893
13	1960	1435
14	1964	1348
15	1968	1777
16	1972	2193
17	1976	2172
18	1980	1756
19	1984	2447
20	1988	3543
21	1992	4124
22	1996	5008
23	2000	5431
24	2004	5546
25	2008	5816
26	2012	5815
27	2016	6223



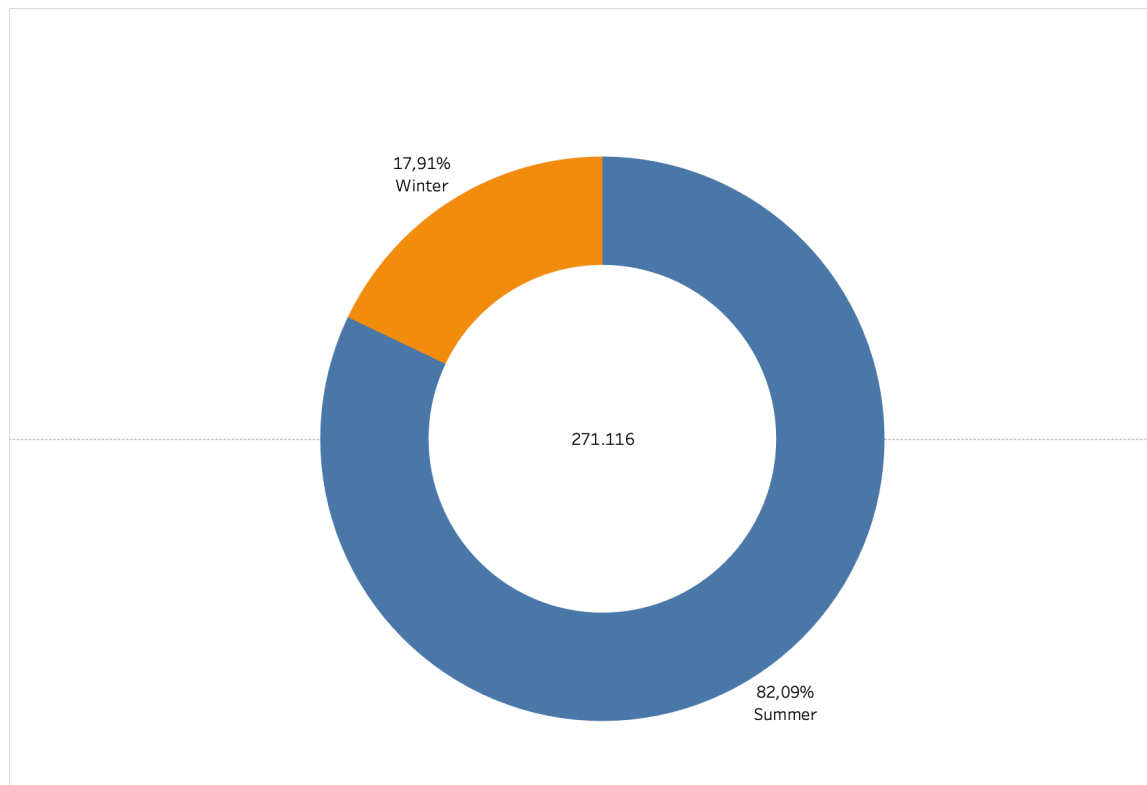
Meskipun partisipasi perempuan sebesar 27,5% selama bertahun-tahun, angka ini telah meningkat secara signifikan seperti yang ditunjukkan di atas

4. Participants across seasons

```
In [22]: Diff_session = olympic_event.Season.value_counts()
Diff_session
```

```
Out[22]: Season
Summer    222552
Winter    48564
Name: count, dtype: int64
```

Participants across seasons



Mengapa Olimpiade Musim Dingin memiliki lebih sedikit peserta, Mari kita coba menjelajahi olahraga dan acara di Olimpiade musim dingin dan musim panas

4.1 Event cabang olah raga musim panas (summer)

```
In [23]: # Menetapkan variable summer_event
summer_olympic = olympic_event[olympic_event.Season=='Summer']

# Menetapkan jumlah secara uniq cabang olahraga
summer_sports = len(summer_olympic[['Sport']].drop_duplicates())

# menetapkan secara uniq jumlah event kompetisi
summer_events = len(summer_olympic[['Event']].drop_duplicates())

#Tampilkan hitungan data kompetisi dengan jumlah cabang olahragayang di adakan
print(f'Sports Played: {summer_sports}, Events held: {summer_events}')
```

Sports Played: 52, Events held: 651

4.2 Event cabang olah raga musim dingin (winter)

```
In [24]: # Menetapkan variable winter_event
summer_olympic = olympic_event[olympic_event.Season=='Winter']

# Menetapkan jumlah secara uniq cabang olahraga
summer_sports = len(summer_olympic[['Sport']].drop_duplicates())
```

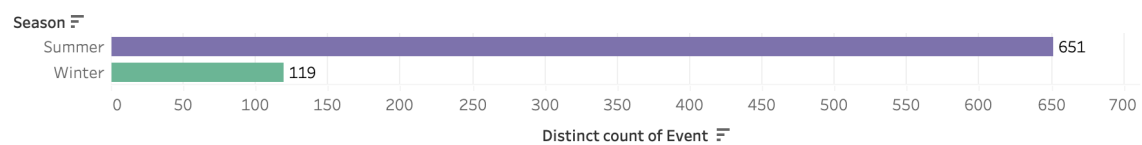
```
# menetapkan secara unik jumlah event kompetisi
summer_events = len(summer_olympic[['Event']].drop_duplicates())

#Tampilkan hitungan data kompetisi dengan jumlah cabang olahragayang di adakan
print(f'Sports Played: {summer_sports}, Events held: {summer_events}')
```

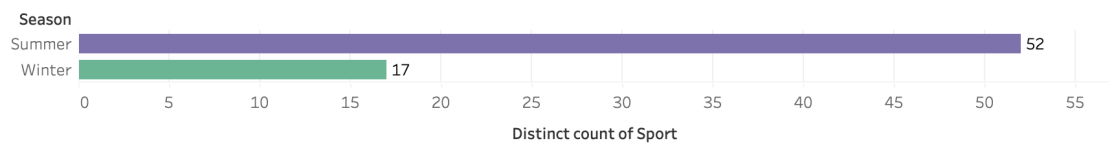
Sports Played: 17, Events held: 119

Berdasarkan data di atas kami memiliki 52 cabang olahraga dan 651 pertandingan di Olimpiade musim panas dan 17 cabang olahraga - 119 pertandingan di Olimpiade musim dingin. Oleh karena itu, kami memiliki jumlah peserta Olimpiade musim panas yang lebih banyak

Event inter Session



Sport inter session



5. Asking and Answering Questions

We've already gained several insights about the participants involved in Olympics. Let's ask some specific questions and try to answer them using data frame operations and visualizations.

5.1 Negara mana yang MEMENANGKAN Medali Emas maksimum di kompetisi Olimpiade yang terakhir diadakan (urutkan 15 teratas)

```
In [83]: year_max = olympic_event.Year.max()
year_max
```

```
Out[83]: 2016
```

```
In [29]: # Tea list
list_Team = olympic_event[(olympic_event.Year==year_max)&(olympic_event.Meda
list_Team
```

```

Out[29]: 1080      Jordan
        1495      Great Britain
        1849      United States
        1850      United States
        2453      Indonesia-1
        ...
        269277     China
        269312     Russia
        269368     China
        269511     China
        270281     Uzbekistan
Name: Team, Length: 665, dtype: object

```

```

In [31]: list_Team.value_counts().head(15)

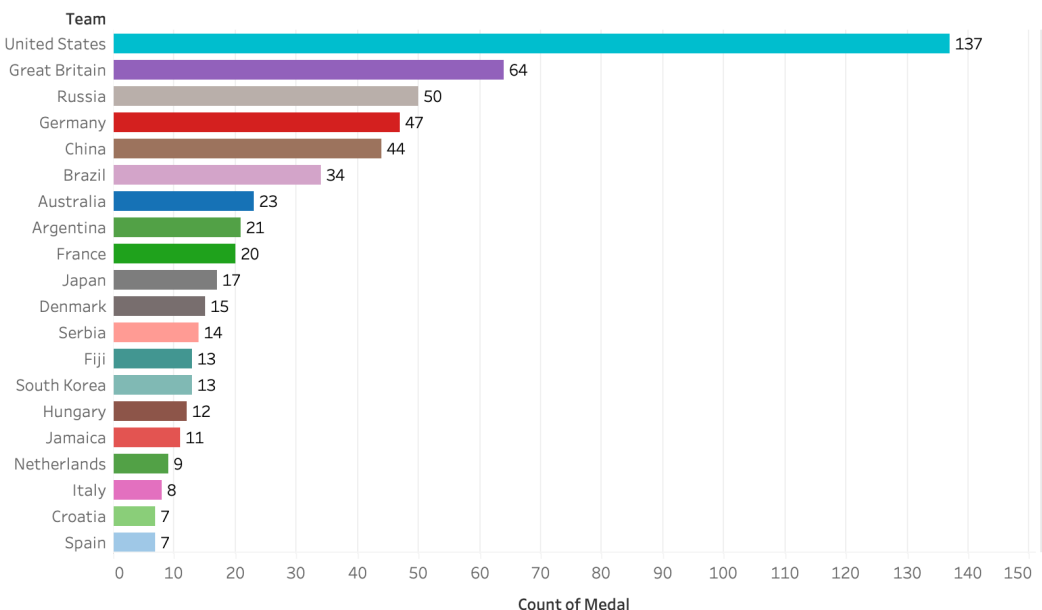
```

```

Out[31]: Team
United States    137
Great Britain    64
Russia           50
Germany          47
China            44
Brazil           34
Australia        23
Argentina        21
France           20
Japan            17
Denmark          15
Serbia           14
Fiji             13
South Korea      13
Hungary          12
Name: count, dtype: int64

```

Top Team Medal



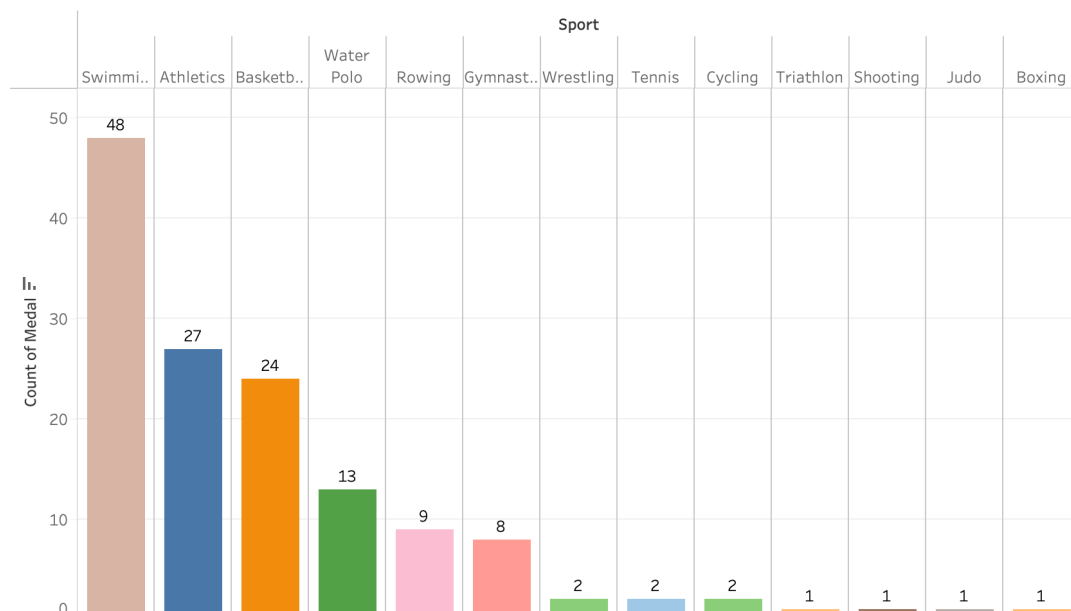
AS tampaknya memimpin perolehan medali emas untuk Olimpiade terakhir yang diadakan pada tahun 2016. Saya ingin tahu olahraga mana yang memperoleh medali emas paling banyak.

```
In [37]: US_Gold = olympic_event[(olympic_event.Year == year_max) & (olympic_event.Medal == 'Gold')]
US_Gold = US_Gold[['Sport', 'Medal']].groupby('Sport').count()
US_Gold.reset_index(inplace=True)
Top_sports = US_Gold.sort_values('Medal', ascending=False)
Top_sports.head(14)
```

```
Out[37]:
```

	Sport	Medal
8	Swimming	48
0	Athletics	27
1	Basketball	24
10	Water Polo	13
6	Rowing	9
4	Gymnastics	8
3	Cycling	2
11	Wrestling	2
2	Boxing	1
5	Judo	1
7	Shooting	1
9	Triathlon	1

US Medal Sport



```
In [35]: fig1, ax1 = plt.subplots()

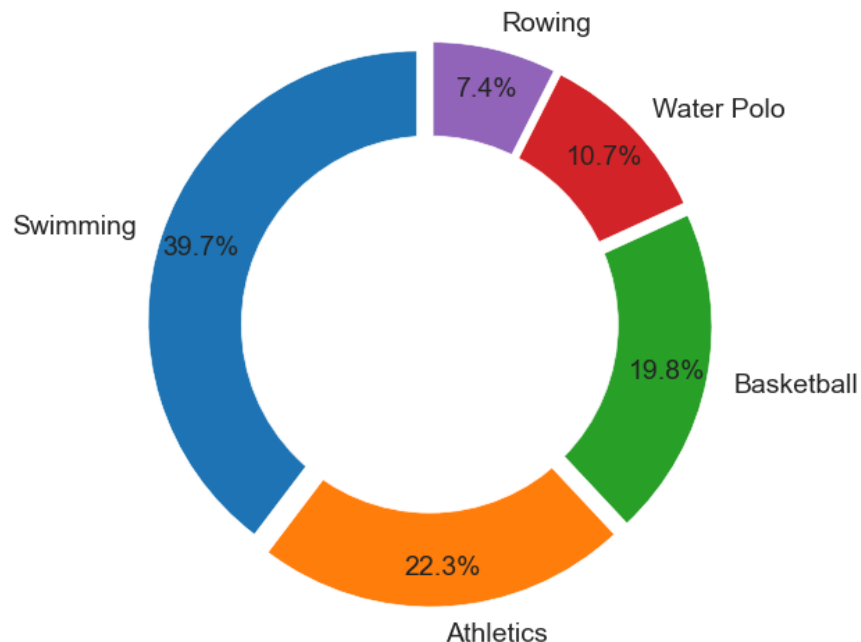
colors = ['#1f77b4', '#ff7f0e', '#2ca02c', '#d62728', '#9467bd']

#explsion
explode = (0.05,0.05,0.05,0.05,0.05)

ax1.pie(Top_sports.head()['Medal'], colors = colors, labels= Top_sports.head
, pctdistance=0.85, explode = explode)

centre_circle = plt.Circle((0,0),0.70,fc='white')
fig = plt.gcf()
fig.gca().add_artist(centre_circle)

ax1.axis('equal')
plt.tight_layout()
plt.show()
```



5.2 Negara-negara mana saja yang memenangkan Medali maksimum per tahun?

```
In [41]: ! pip install pandasql --upgrade --quiet
from pandasql import sqldf
output = sqldf('''
select Team, Year, max(Highest_Number_Of_Medals_per_Year) as 'Highest_Number
select Team, Year, sum(Highest_Number_Of_Medals_per_Year) as 'Highest_Number
select Name, Team, Medal, Year, count(*) as 'Highest_Number_Of_Medals_per_Ye
from olympic_event
where
Medal <> 'None'
and
Season = 'Summer'
group by Year, Medal, Team, Name
```



```
order by Year desc, Highest_Number_Of_Medals_per_Year)
group by Team, Year
order by Year, Highest_Number_Of_Medals_per_Year desc)
group by Year
order by Year desc
'''
```

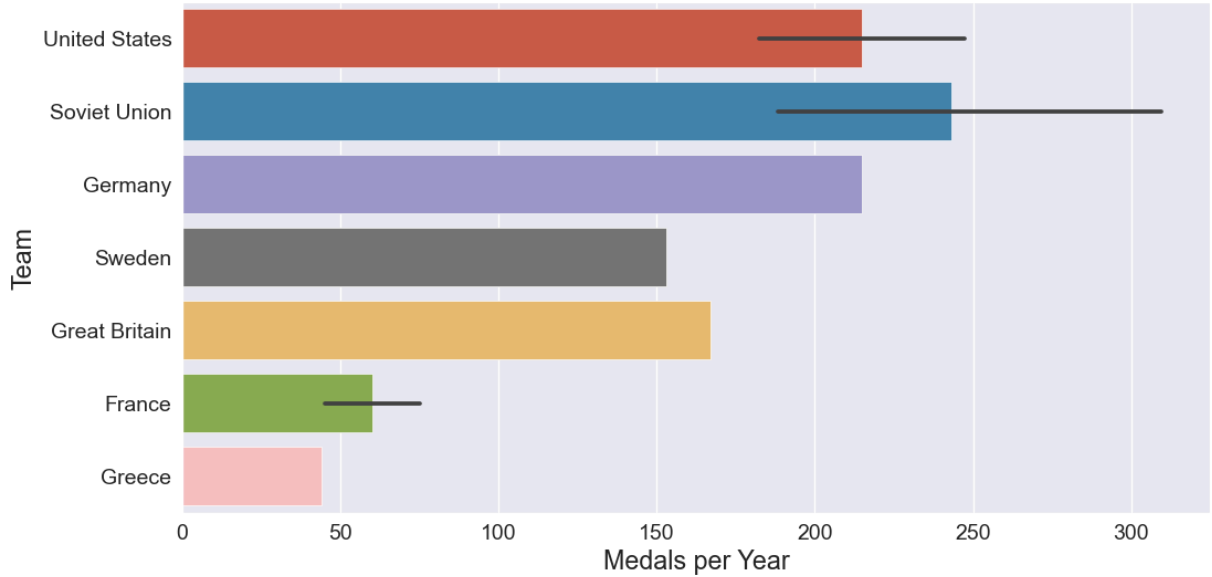
In [42]: output

Out [42]:

	Team	Year	Highest_Number_Of_Medals_per_Year
0	United States	2016	256
1	United States	2012	238
2	United States	2008	309
3	United States	2004	259
4	United States	2000	240
5	United States	1996	255
6	United States	1992	222
7	Soviet Union	1988	300
8	United States	1984	352
9	Soviet Union	1980	442
10	Soviet Union	1976	286
11	Soviet Union	1972	214
12	Soviet Union	1968	192
13	Soviet Union	1964	174
14	Soviet Union	1960	167
15	Soviet Union	1956	169
16	United States	1952	122
17	United States	1948	143
18	Germany	1936	215
19	United States	1932	170
20	United States	1928	88
21	United States	1924	174
22	United States	1920	194
23	Sweden	1912	153
24	Great Britain	1908	167
25	France	1906	45
26	United States	1904	199
27	France	1900	75
28	Greece	1896	44

```
In [43]: plt.figure(figsize=(12, 6))
sns.barplot(x=output.Highest_Number_Of_Medals_per_Year, y=output.Team)
```

```
plt.title('');
plt.xlabel('Medals per Year');
```



Berikut adalah wawasan yang kami peroleh dari visual di atas

- + AS tampaknya telah memenangkan medali maksimum selama beberapa tahun.
- + Uni Soviet tampaknya telah memenangkan jumlah medali terbanyak di ajang Olimpiade.
- + Varians yang menang maksimal dengan Uni Soviet
- + Jerman, Swedia, Inggris Raya dan Yunani pernah meraih medali maksimal satu kali

5.3 Sebutkan 10 atlet teratas yang memenangkan jumlah maksimum Medali Olimpiade untuk negaranya

```
In [50]: df_medal_holders = olympic_event[olympic_event.Medal != 'None']
df_medal_holders['Count_Of_Medals']=1
df_medal_holders
```

Out [50]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Gam
3	4	Edgar Lindenau Aabye	M	34	0.0	0.0	Denmark/Sweden	DEN	19 Sumn
37	15	Arvo Ossian Aaltonen	M	30	0.0	0.0	Finland	FIN	19 Sumn
38	15	Arvo Ossian Aaltonen	M	30	0.0	0.0	Finland	FIN	19 Sumn
40	16	Juhamatti Tapio Aaltonen	M	28	184.0	85.0	Finland	FIN	20 Win
41	17	Paavo Johannes Aaltonen	M	28	175.0	64.0	Finland	FIN	19 Sumn
...	
271078	135553	Galina Ivanovna Zybina (- Fyodorova)	F	25	168.0	80.0	Soviet Union	URS	19 Sumn
271080	135553	Galina Ivanovna Zybina (- Fyodorova)	F	33	168.0	80.0	Soviet Union	URS	19 Sumn
271082	135554	Bogusaw Zych	M	28	182.0	82.0	Poland	POL	19 Sumn
271102	135563	Olesya Nikolayevna Zykina	F	19	171.0	64.0	Russia	RUS	20 Sumn
271103	135563	Olesya Nikolayevna Zykina	F	23	171.0	64.0	Russia	RUS	20 Sumn

39783 rows × 16 columns

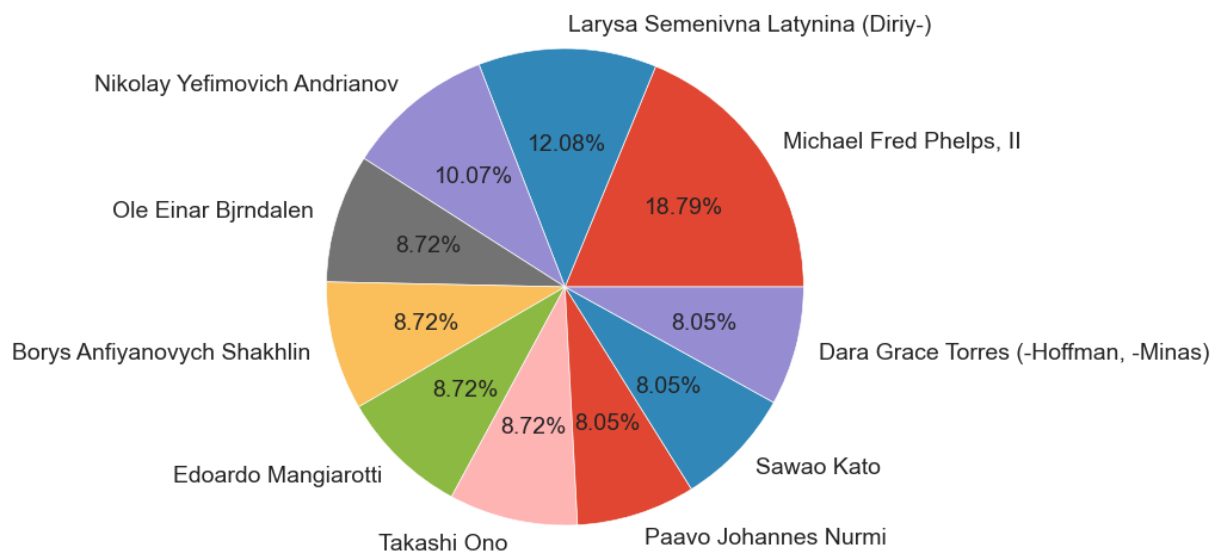
```
In [46]: df_highest_medals = df_medal_holders[['Name', 'Year', 'Team', 'Count_Of_Meda
df_highest_medals = df_highest_medals.groupby(['Name', 'Team']).sum().sort_va
df_highest_medals.reset_index(inplace=True)
```

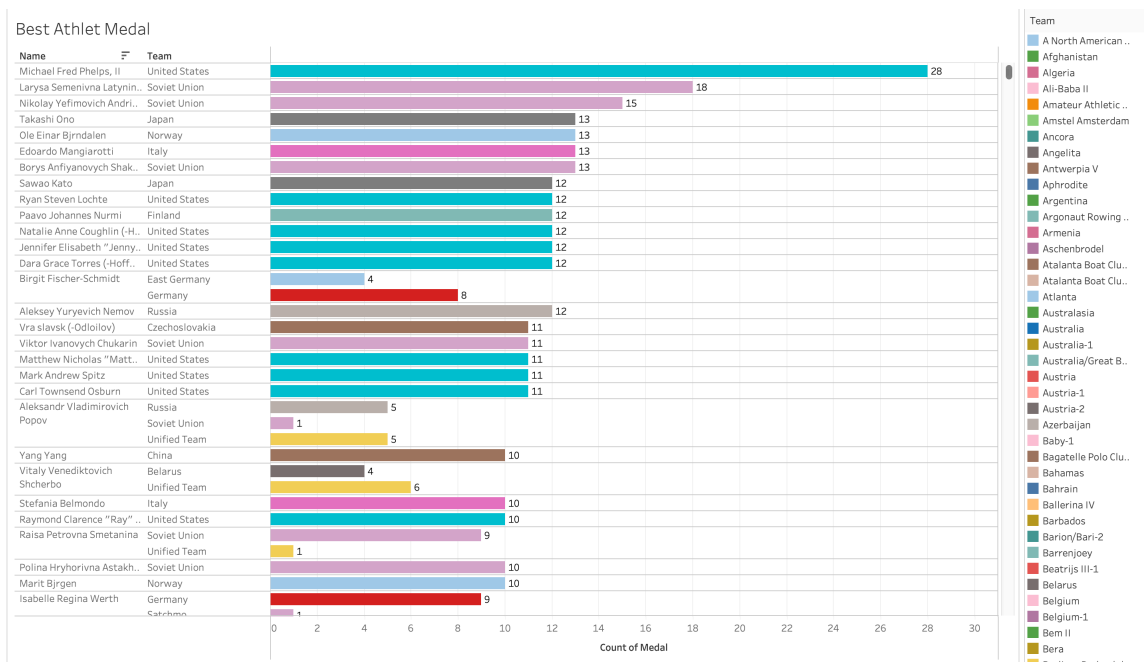
```
In [47]: df_highest_medals
```

```
Out[47]:
```

	Name	Team	Count_Of_Medals
0	Michael Fred Phelps, II	United States	28
1	Larysa Semenivna Latynina (Diriy-)	Soviet Union	18
2	Nikolay Yefimovich Andrianov	Soviet Union	15
3	Ole Einar Bjrndalen	Norway	13
4	Borys Anfiyanovych Shakhlin	Soviet Union	13
5	Edoardo Mangiarotti	Italy	13
6	Takashi Ono	Japan	13
7	Paavo Johannes Nurmi	Finland	12
8	Sawao Kato	Japan	12
9	Dara Grace Torres (-Hoffman, -Minas)	United States	12

```
In [51]: from matplotlib import pyplot as plt
import numpy as np
fig = plt.figure()
ax = fig.add_axes([0,0,1,1])
ax.axis('equal')
Team = df_highest_medals.Name
Count_of_Medal = df_highest_medals.Count_Of_Medals
ax.pie(Count_of_Medal, labels = Team, autopct='%1.2f%%')
plt.show()
```

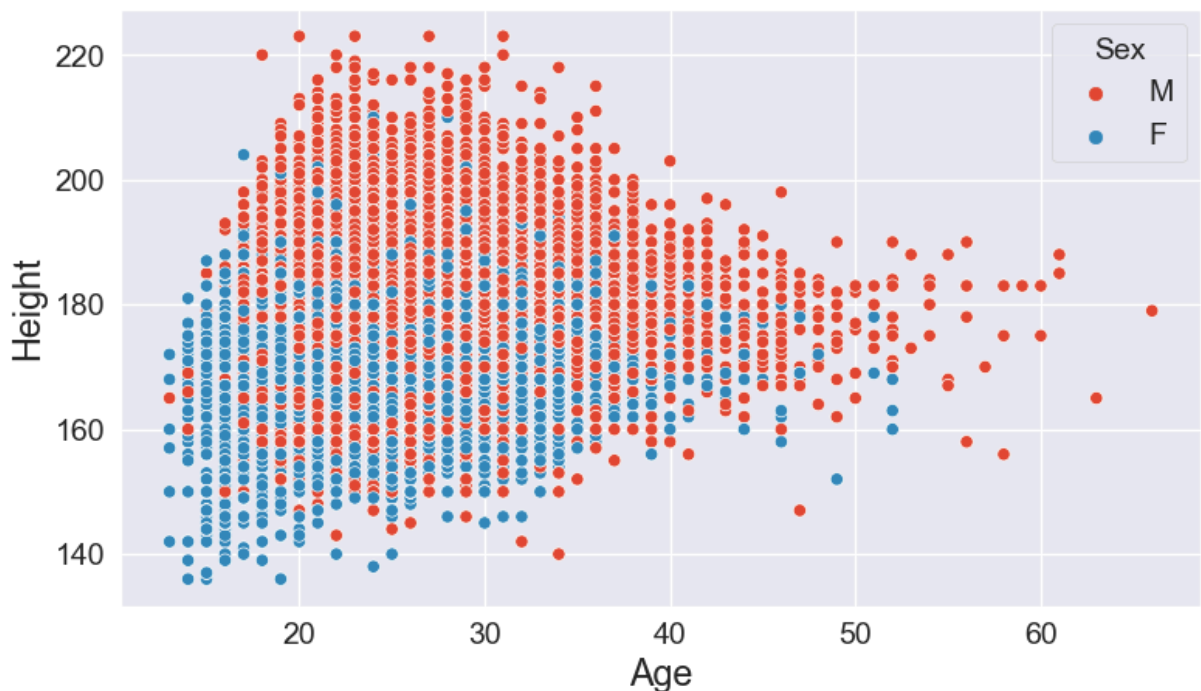




5.4 Pembagian Medali berdasarkan Usia, Tinggi Badan dan Berat Badan

5.4.1 Age and Height

```
In [52]: df1 = olympic_event[(olympic_event.Age != 0) & (olympic_event.Height != 0.0)]
sns.scatterplot(x=df1.Age, y=df1.Height, hue='Sex', data=df1)
plt.xlabel("Age")
plt.ylabel("Height");
```



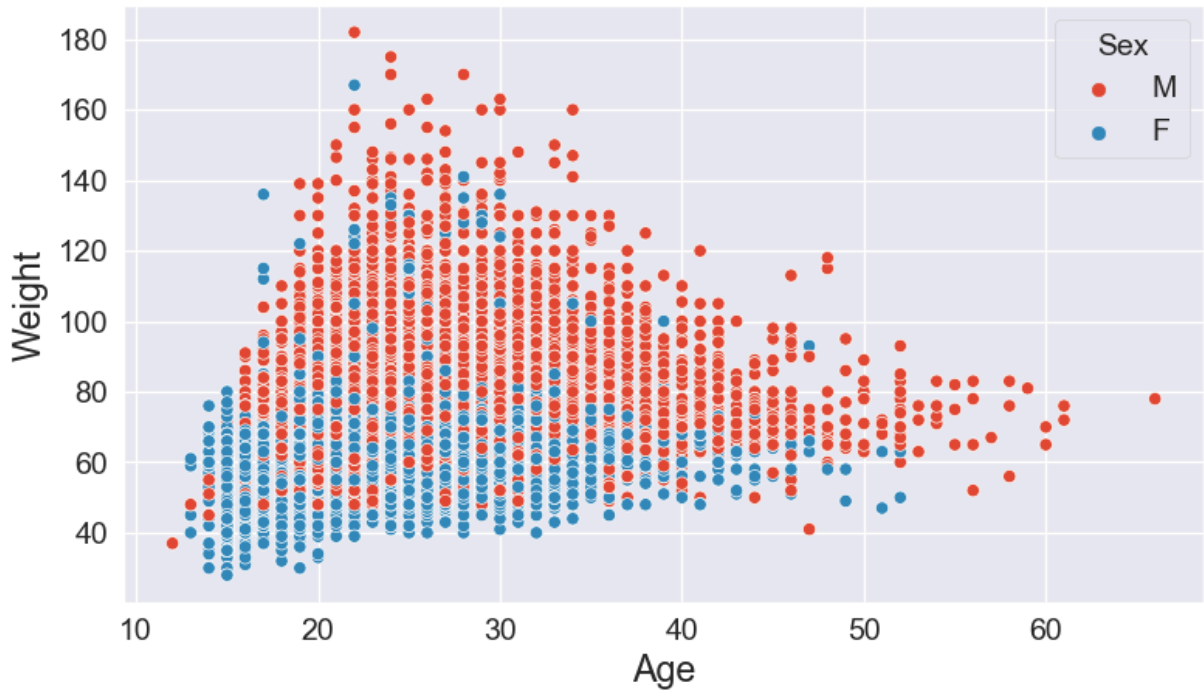
```
In [53]: df1[(df1.Height <= 140.0) & (df1.Age <= 20)][['Name', 'Height', 'Age', 'Sport']].s
```

Out [53]:

	Name	Height	Age	Sport
256836	Wang Xin (Ruoxue-)	137.0	15	Diving
256837	Wang Xin (Ruoxue-)	137.0	15	Diving
13741	Oana Mihaela Ban	139.0	18	Gymnastics
23763	Loredana Boboc	139.0	16	Gymnastics
31837	Diana Laura Bulimar	140.0	16	Gymnastics
69216	Mariya Yevgenyevna Filatova (-Kurbatova)	136.0	14	Gymnastics
69222	Mariya Yevgenyevna Filatova (-Kurbatova)	136.0	19	Gymnastics
69225	Mariya Yevgenyevna Filatova (-Kurbatova)	136.0	19	Gymnastics
108408	Jiang Yuyuan	140.0	16	Gymnastics
143279	Lu Li	136.0	15	Gymnastics
143280	Lu Li	136.0	15	Gymnastics
160840	Mo Huilan	140.0	17	Gymnastics
160920	Dominique Helena Moceanu (-Canales)	139.0	14	Gymnastics
256864	Wang Yan	140.0	16	Gymnastics
270182	Kimberley Lyn "Kim" Zmeskal (-Burdette)	139.0	16	Gymnastics

5.4.2 Age and Weight

```
In [57]: df2 = olympic_event[(olympic_event.Age != 0) & (olympic_event.Weight != 0.0)]
sns.scatterplot(x=df2.Age, y=df2.Weight, hue='Sex', data=df2)
plt.xlabel("Age")
plt.ylabel("Weight");
```

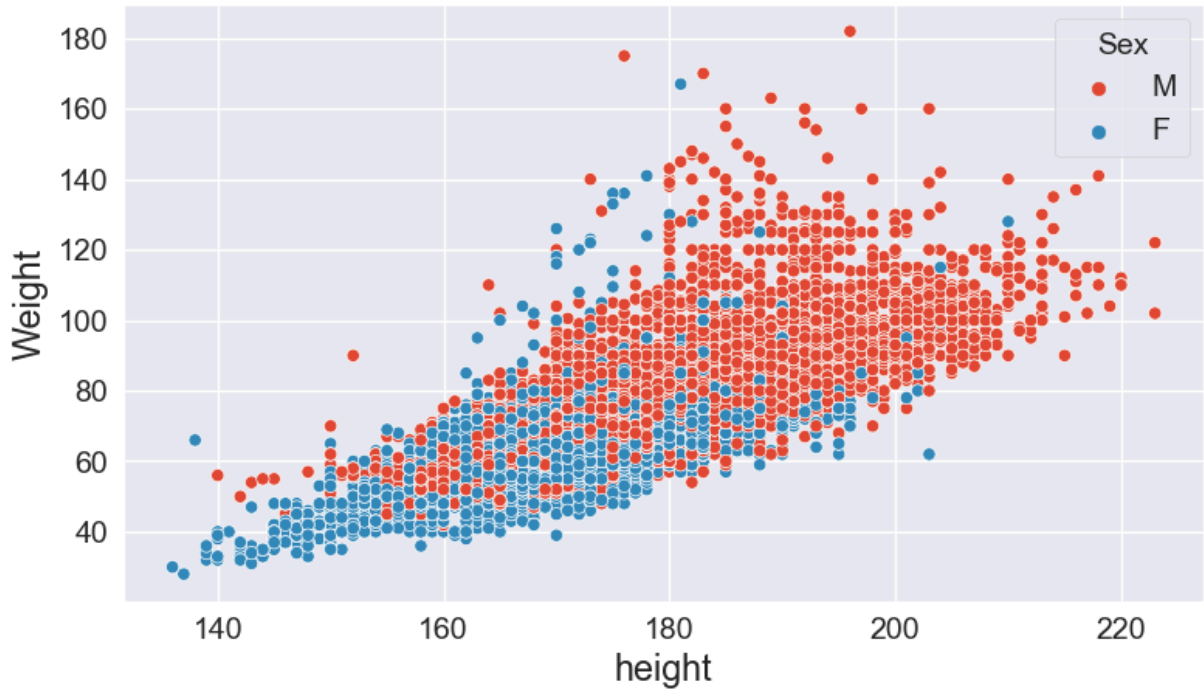


```
In [55]: df3[(olympic_event.Weight>160)]
```

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Ye
39181	20144	Andrey Ivanovich Chemerkin	M	24	183.0	170.0	Russia	RUS	1996 Summer	199
39182	20144	Andrey Ivanovich Chemerkin	M	28	183.0	170.0	Russia	RUS	2000 Summer	200
124420	62843	Olha Vasylivna Korobka	F	22	181.0	167.0	Ukraine	UKR	2008 Summer	200
173166	87041	Dmitry Yuryevich Nosov	M	24	176.0	175.0	Russia	RUS	2004 Summer	200
237040	118869	Christopher J. "Chris" Taylor	M	22	196.0	182.0	United States	USA	1972 Summer	19
268659	134407	Leonid Ivanovich Zhabotynskiy	M	26	189.0	163.0	Soviet Union	URS	1964 Summer	19
268660	134407	Leonid Ivanovich Zhabotynskiy	M	30	189.0	163.0	Soviet Union	URS	1968 Summer	19

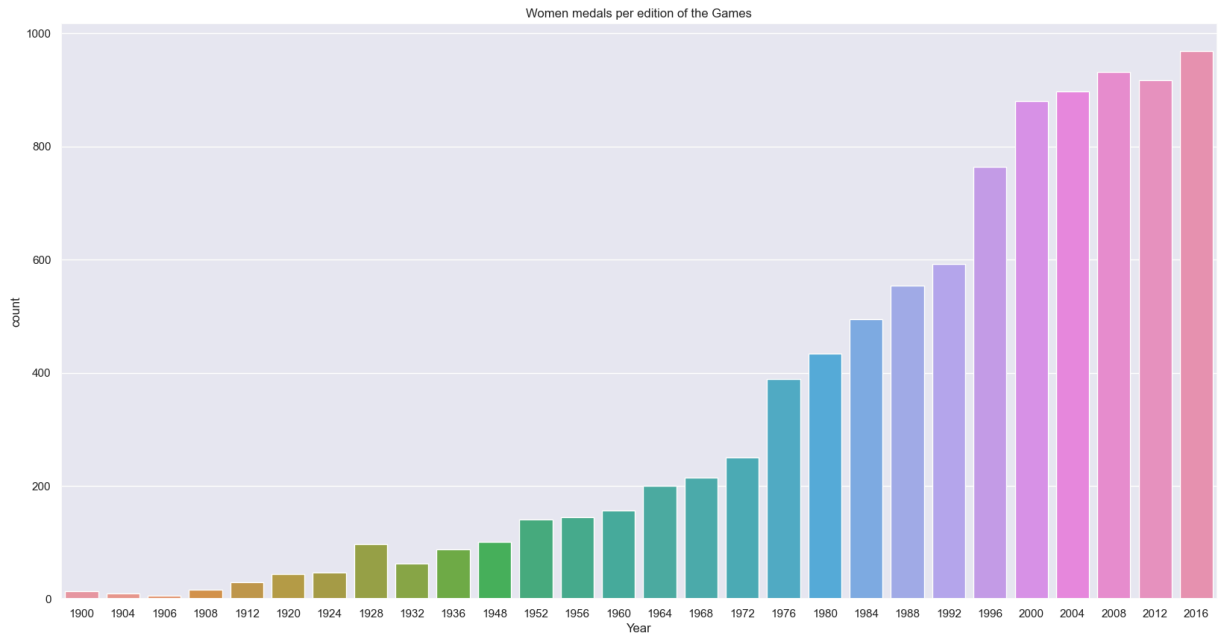
5.4.3 Height and Weight

```
In [56]: df3 = olympic_event[(olympic_event.Age != 0) & (olympic_event.Weight != 0.0)]
sns.scatterplot(x=df3.Height, y=df3.Weight, hue='Sex', data=df3)
plt.xlabel("height")
plt.ylabel("Weight");
```



5.5 Partisipasi perempuan di Olimpiade

```
In [58]: Women_In_Olympics = olympic_event[(olympic_event.Sex == 'F') & (olympic_event.Year > 1900)]
sns.set(style="darkgrid")
plt.figure(figsize=(20, 10))
sns.countplot(x='Year', data=Women_In_Olympics)
plt.title('Women medals per edition of the Games');
```



- Berdasarkan tren yang ada, partisipasi perempuan rata-rata meningkat dari tahun ke tahun

5.6 Siapakah atlet peraih Medali secara Perorangan dengan Usia lebih dari 50 tahun

```
In [60]: df_medal_holders = olympic_event[(olympic_event.Medal != 'None') & (olympic_e
df_medal_holders['Count_Of_Medals'] = 1

df_medal_holders_above50 = df_medal_holders[olympic_event.Age >= 50]

df_medal_holders_above50.head(10)
```

Out [60]:

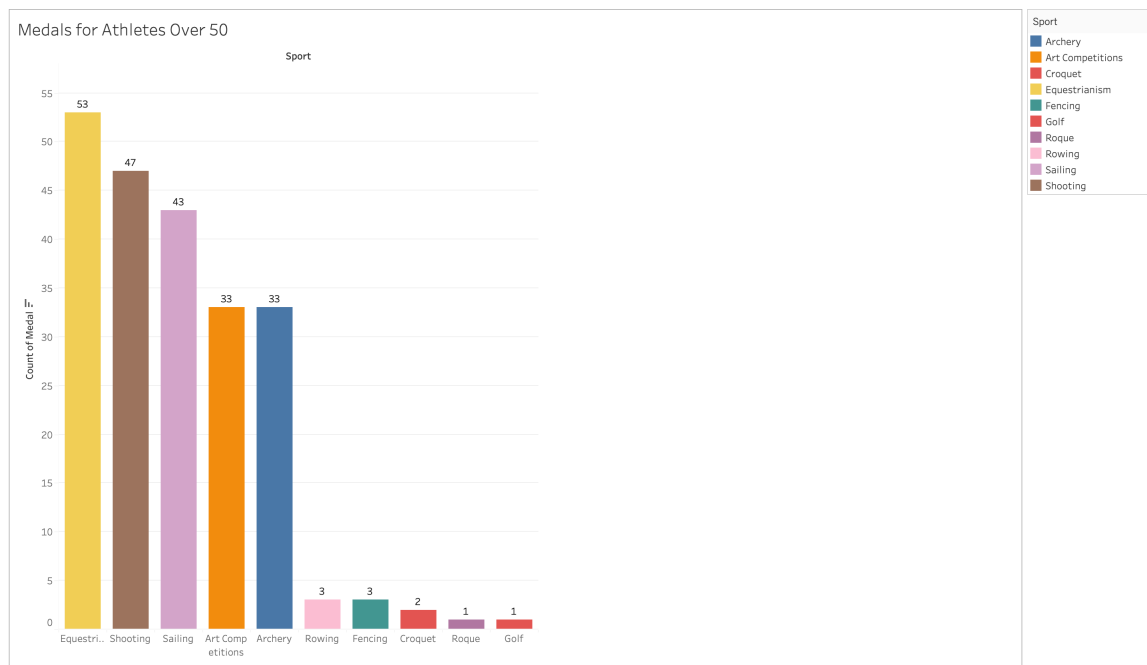
	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	
3680	2112	Abdullah Al-Rashidi	M	52	183.0	83.0	Individual Olympic Athletes	IOA	2016 Summer	2016	5
5077	2894	Derek Swithin Allhusen	M	54	180.0	71.0	Great Britain	GBR	1968 Summer	1968	5
5078	2894	Derek Swithin Allhusen	M	54	180.0	71.0	Great Britain	GBR	1968 Summer	1968	5
7961	4404	Johan August Anker	M	56	0.0	0.0	Norna	NOR	1928 Summer	1928	5
13393	7272	Nikolaus "Klaus" Balkenhol	M	52	178.0	83.0	Germany	GER	1992 Summer	1992	5
13394	7272	Nikolaus "Klaus" Balkenhol	M	52	178.0	83.0	Germany	GER	1992 Summer	1992	5
13396	7272	Nikolaus "Klaus" Balkenhol	M	56	178.0	83.0	Germany	GER	1996 Summer	1996	5
14364	7744	Ernest Barberolle	M	58	156.0	56.0	France	FRA	1920 Summer	1920	5
17552	9349	Ludger Beerbaum	M	52	190.0	85.0	Germany	GER	2016 Summer	2016	5
21999	11599	Rudolf Georg Binding	M	60	0.0	0.0	Germany	GER	1928 Summer	1928	5

```
In [72]: df_medal_holders_above50_list = df_medal_holders_above50.groupby(['Sport']).
df_medal_holders_above50_reset_index = df_medal_holders_above50_list.reset_i
df_medal_holders_above50_reset_index[['Sport', 'Count_Of_Medals']].head()
```

Out [72]:

	Sport	Count_Of_Medals
0	Equestrianism	53
1	Shooting	50
2	Sailing	46
3	Art Competitions	37
4	Archery	34

Medals for Athletes Over 50

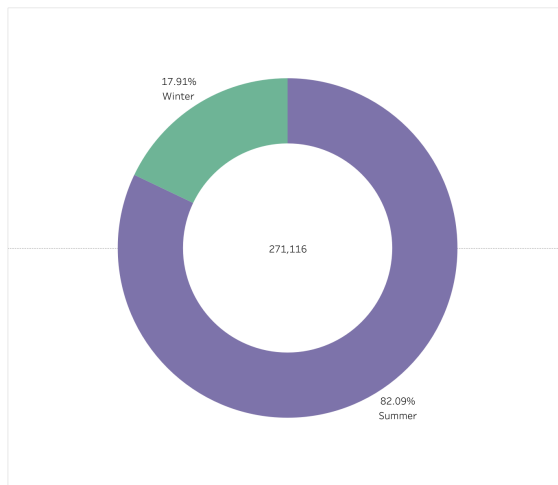


Dashboard Olympic Event

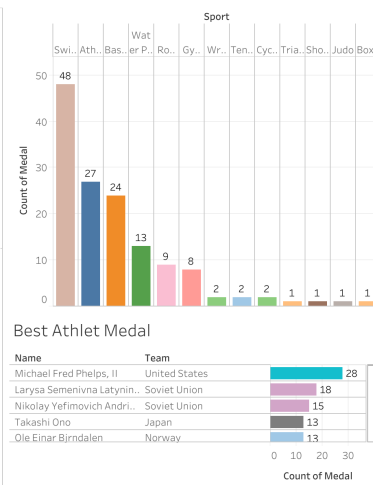
(

<https://public.tableau.com/app/profile/burhanudin.badiuzaman/viz/OlympicEvent/Dashboard?publish=yes>)

Participants across seasons



US Medal Sport



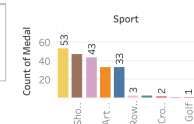
Top Team Medal



Best Athlet Medal



Medals for Athletes Over 50



Kesimpulan dan Kesimpulan

Kami telah menarik banyak kesimpulan dari survei tersebut. Berikut ringkasan beberapa di antaranya:

- AS tampaknya dominan dalam hal partisipasi emas maksimum serta partisipasi keseluruhan dalam permainan.
- Kami mengamati atlet dari usia 12 hingga usia 58 tahun memenangkan medali.
- Olimpiade Musim Panas mempunyai jumlah acara dan olahraga yang lebih banyak dibandingkan dengan Olimpiade musim dingin.
- Dalam sejarah 120 tahun Olimpiade, Michael Fred Phelps, II telah meraih medali maksimal untuk negaranya yaitu 28 Medali
- Kami melihat tren peserta perempuan selama bertahun-tahun mengalami peningkatan.
- Berpartisipasi dengan beban tinggi (seperti > 150) tampaknya berhasil dengan baik dalam Gulat, Angkat Berat, dan Judo.

```
In [86]: !jupyter nbconvert --to webpdf --allow-chromium-download ReviewingBasic.ipynb
```

```
[NbConvertApp] Converting notebook ReviewingBasic.ipynb to webpdf
[NbConvertApp] WARNING | Alternative text is missing on 9 image(s).
[NbConvertApp] Building PDF
[NbConvertApp] PDF successfully created
[NbConvertApp] Writing 2680769 bytes to ReviewingBasic.pdf
```

```
In [ ]:
```