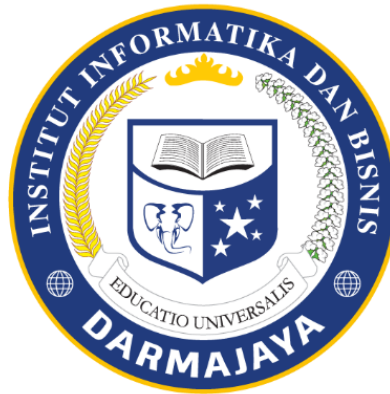


**LAPORAN HASIL ANALISIS**  
***WORKSHOP: PYTHON FOR EXPLORATORY DATA ANALYSIS***



**EKO ZULKARYANTO**  
**2021211013**

**MAGISTER TEKNIK INFORMATIKA**  
**INSTITUT INFORMATIKA DAN BISNIS DARMAJAYA**  
**BANDARLAMPUNG**  
**2021**

## DAFTAR ISI

Mempersiapkan Persyaratan.....	1
Perkenalan Singkat <i>Library</i> yang Di-install.....	2
Pengenalan Data .....	3
Mengunggah <i>Dataset</i> .....	3
Memulai Notebook Baru .....	4
Mengimpor <i>Dataset</i> ke Dalam <i>Dataframe</i> .....	5
Mengumpulkan Informasi Dasar .....	5
Memahami <i>Boolean Indexing</i> .....	6
Eksplorasi Beberapa Fungsi <i>Built-in</i> .....	7
Membersihkan dan Melengkapi Data.....	9
Visualisasi Data.....	11
<i>Barplot</i> .....	15
<i>Scatterplot</i> .....	18
<i>Heatmap</i> .....	19

## Mempersiapkan Persyaratan

Sebelum melakukan *workshop* ini, terlebih dahulu dilakukan persiapan persyaratan *workshop*. Persyaratan *workshop* yang harus dilakukan adalah:

- Mengunduh python di <https://www.python.org/downloads/> dan menginstalasi python.
- Menginstall Jupyter Notebook dengan perintah yang dijalankan di Command Prompt.

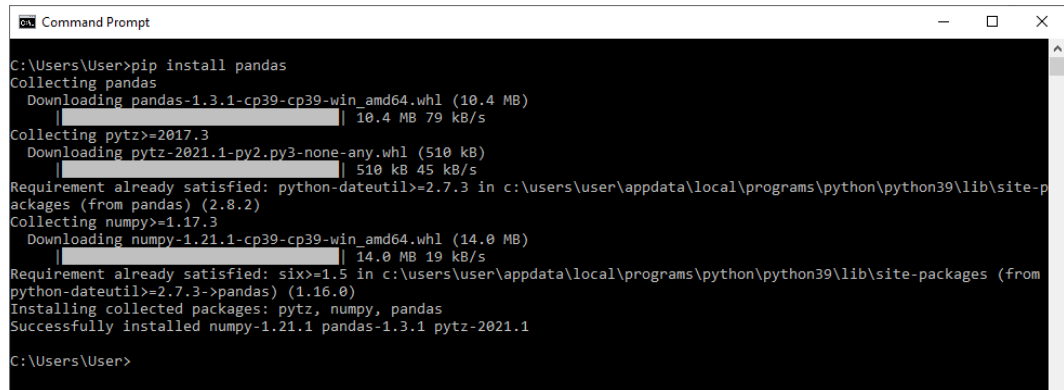
```
python -m pip install jupyter
```

- Menjalankan Jupyter Notebook dengan perintah di Command Prompt.

```
jupyter notebook
```

- Selanjutnya akan terbuka <http://localhost:8888/tree> di *browser*.
- Mengunduh dataset di <https://github.com/noraiz-anwar/exploratory-data-analysis>, yaitu *athlete\_events.csv* dan *noc\_regions.csv*.
- Selanjutnya menginstalasi modul-modul yang dibutuhkan seperti *numpy*, *pandas*, *seaborn*, dan *matplotlib*, menggunakan perintah `pip`.  
Contoh: Menginstalasi modul *pandas* dengan perintah di Command Prompt.

```
pip install pandas
```



```
Command Prompt
C:\Users\User>pip install pandas
Collecting pandas
  Downloading pandas-1.3.1-cp39-cp39-win_amd64.whl (10.4 MB)
    | 10.4 MB 79 kB/s
Collecting pytz>=2017.3
  Downloading pytz-2021.1-py2.py3-none-any.whl (510 kB)
    | 510 kB 45 kB/s
Requirement already satisfied: python-dateutil>=2.7.3 in c:\users\user\appdata\local\programs\python\python39\lib\site-packages (from pandas) (2.8.2)
Collecting numpy>=1.17.3
  Downloading numpy-1.21.1-cp39-cp39-win_amd64.whl (14.0 MB)
    | 14.0 MB 19 kB/s
Requirement already satisfied: six>=1.5 in c:\users\user\appdata\local\programs\python\python39\lib\site-packages (from python-dateutil>=2.7.3->pandas) (1.16.0)
Installing collected packages: pytz, numpy, pandas
Successfully installed numpy-1.21.1 pandas-1.3.1 pytz-2021.1
C:\Users\User>
```

## Perkenalan Singkat *Library* yang Di-install

Beberapa *library* yang di-install pada *workshop* ini adalah *numpy*, *pandas*, *seaborn*, dan *matplotlib*. Berikut daftar *import* yang digunakan:

```
import numpy as np
import pandas as pd
import seaborn as sb
from matplotlib import pyplot as plot
```

*Numpy* (<http://www.numpy.org/>) adalah pustaka untuk bahasa pemrograman Python, menambahkan dukungan untuk *array* dan matriks multidimensi yang besar, bersama dengan koleksi besar fungsi matematika tingkat tinggi untuk beroperasi pada *array* ini.

*Pandas* (<https://pandas.pydata.org/>) adalah paket python yang menyediakan struktur data yang cepat, fleksibel, dan ekspresif yang dirancang untuk membuat bekerja dengan data "relasional" atau "berlabel" menjadi mudah dan intuitif. Ini bertujuan untuk menjadi blok bangunan tingkat tinggi yang mendasar untuk melakukan analisis data dunia nyata yang praktis dengan Python. Selain itu, ia memiliki tujuan yang lebih luas untuk menjadi alat analisis/manipulasi data *opensource* yang paling kuat dan fleksibel yang tersedia dalam bahasa apa pun. Ini sudah berjalan dengan baik menuju tujuan ini.

*Seaborn* (<https://seaborn.pydata.org/>) adalah pustaka visualisasi data Python berdasarkan *matplotlib*. Ini menyediakan antarmuka tingkat tinggi untuk menggambar grafik statistik yang menarik dan informatif.

*Matplotlib* (<https://matplotlib.org/>) adalah pustaka *plot* 2D Python yang menghasilkan angka kualitas publikasi dalam berbagai format *hardcopy* dan lingkungan interaktif di seluruh *platform*.

## Pengenalan Data

Data yang digunakan pada *workshop* ini adalah data lengkap atlet dalam mengikuti kegiatan olimpiade yang telah diadakan selama 120 tahun. *File* athlete\_events.csv berisi 271.116 baris dan 15 kolom data. Berikut informasi pada kolom yang tersedia:

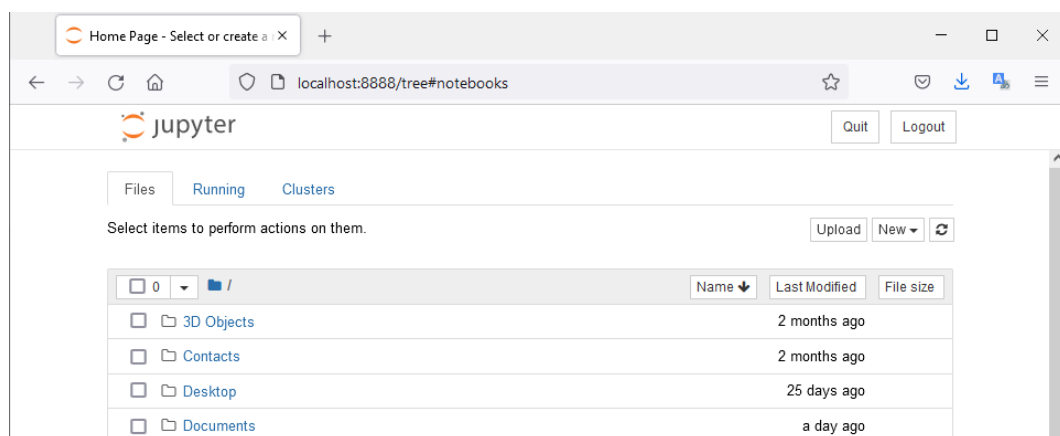
1. ID – nomor unik masing-masing atlet;
2. Name – nama atlet;
3. Sex - M atau F, laki-laki atau perempuan;
4. Age – umur, bilangan bulat;
5. Height – dalam sentimeter;
6. Weight – dalam kilogram;
7. Team – nama tim;
8. NOC - National Olympic Committee 3 huruf kode;
9. Games – tahun dan musim;
10. Year – tahun, bilangan bulat;
11. Season - *Summer* atau *Winter*;
12. City – tuan rumah;
13. Sport – jenis olahraga;
14. Event - kegiatan;
15. Medal - *Gold*, *Silver*, *Bronze*, atau NA.

*File* noc\_regions.csv berisi 230 baris dan 3 kolom data. Berikut informasi kolom yang tersedia:

1. NOC - National Olympic Committee 3 huruf kode;
2. Region – Nama negara
3. Notes – *string* yang berisi informasi tentang negara dan NOC

## Mengunggah Dataset

*Dataset* athlete\_events.csv dan noc\_regions.csv diunggah ke direktori yang terlihat di <http://localhost:8888/tree>.

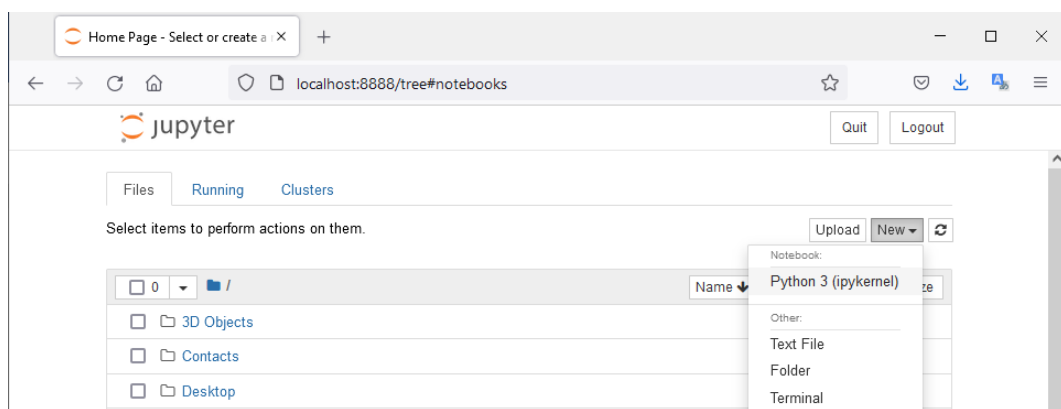


Direktori tersebut terletak di C:/Users/user. Kemudian mengunggah *dataset* yang telah diunduh dengan mengklik tombol *upload*. Setelah pilih *file dataset*-nya, kemudian klik tombol *upload* pada tombol di sebelah kanan *file*.

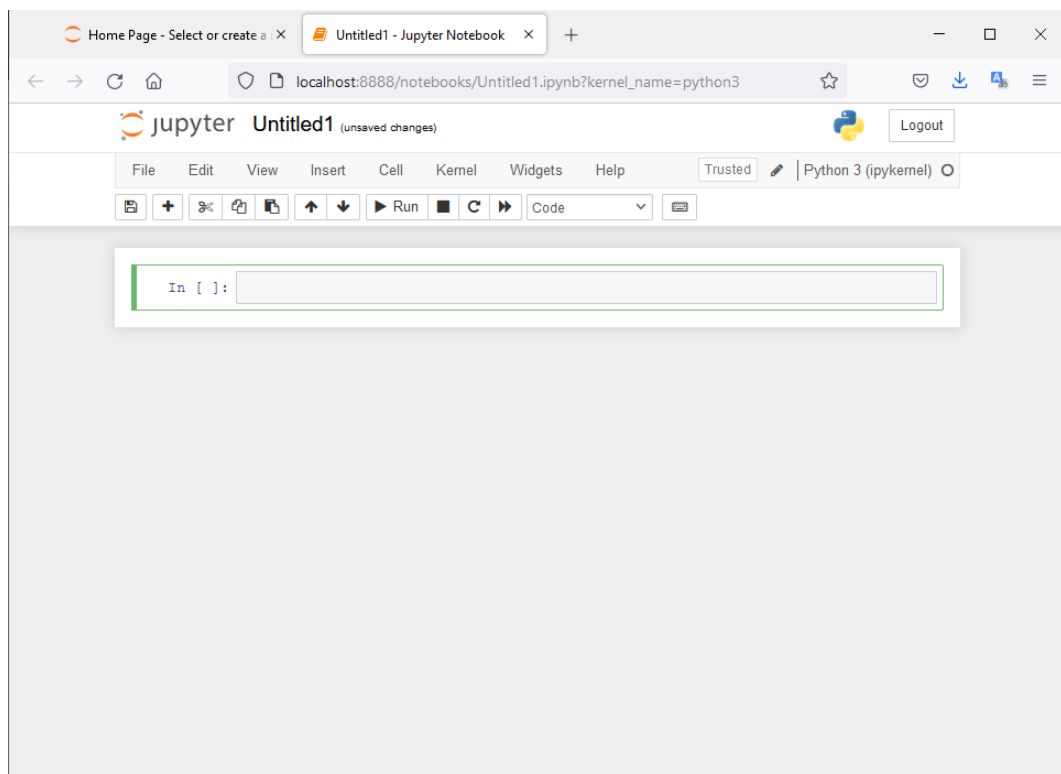


## Memulai Notebook Baru

Selanjutnya memulai membuat *notebook* baru pada **Jupyter Notebook** dengan mengklik New > Python 3 (ipykernel).



Selanjutnya akan terbuka *notebook* baru dan siap untuk memulai program.



## Mengimpor *Dataset* ke Dalam *Dataframe*

Untuk mengimpor *dataset*, menggunakan kode di bawah ini:

```
import pandas as pd
data = pd.read_csv("athlete_events.csv")
negara = pd.read_csv("noc_regions.csv")
```

## Mengumpulkan Informasi Dasar

Untuk memperoleh informasi dasar dari data tersebut, menggunakan fungsi *head*.

```
data.head()
```

ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	Sport	Event	Medal
1	A Dijing	M	24.0	180.0	80.0	China	CHN	1992 Summer	1992	Summer	Barcelona	Basketball	Basketball Men's Basketball	NaN
2	A Lamusi	M	23.0	170.0	60.0	China	CHN	2012 Summer	2012	Summer	London	Judo	Judo Men's Extra-Lightweight	NaN
3	Gunnar Nielsen Aaby	M	24.0	NaN	NaN	Denmark	DEN	1920 Summer	1920	Summer	Antwerpen	Football	Football Men's Football	NaN
4	Edgar Lindenau Aabye	M	34.0	NaN	NaN	Denmark/Sweden	DEN	1900 Summer	1900	Summer	Paris	Tug-Of-War	Tug-Of-War Men's Tug-Of-War	Gold
5	Christine Jacoba Aaftink	F	21.0	185.0	82.0	Netherlands	NED	1988 Winter	1988	Winter	Calgary	Speed Skating	Speed Skating Women's 500 metres	NaN

Untuk memperoleh informasi statistika deskriptif, menggunakan perintah:

```
data.describe()
```

	Age	Height	Weight	Year
count	261642.000000	210945.000000	208241.000000	271116.000000
mean	25.556898	175.338970	70.702393	1978.378480
std	6.393561	10.518462	14.348020	29.877632
min	10.000000	127.000000	25.000000	1896.000000
25%	21.000000	168.000000	60.000000	1960.000000
50%	24.000000	175.000000	70.000000	1988.000000
75%	28.000000	183.000000	79.000000	2002.000000
max	97.000000	226.000000	214.000000	2016.000000

Untuk memperoleh ringkasan informasi dari keseluruhan *dataframe* adalah dengan perintah:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 271116 entries, 1 to 135571
Data columns (total 14 columns):
#   Column      Non-Null Count  Dtype
#   ...
```

```

---
0   Name      271116 non-null object
1   Sex       271116 non-null object
2   Age       261642 non-null float64
3   Height    210945 non-null float64
4   Weight    208241 non-null float64
5   Team      271116 non-null object
6   NOC       271116 non-null object
7   Games     271116 non-null object
8   Year      271116 non-null int64
9   Season    271116 non-null object
10  City      271116 non-null object
11  Sport     271116 non-null object
12  Event     271116 non-null object
13  Medal     39783 non-null object
dtypes: float64(3), int64(1), object(10)
memory usage: 31.0+ MB

```

Untuk mengecek jika ada data yang bernilai *Null* pada kolom tertentu, menggunakan perintah:

```
data.isnull().any()
```

```

Name      False
Sex       False
Age       True
Height    True
Weight    True
Team      False
NOC       False
Games     False
Year      False
Season    False
City      False
Sport     False
Event     False
Medal     True
dtype: bool

```

### Memahami *Boolean Indexing*

Untuk menghitung *record* yang tidak mendapatkan medali yaitu dengan perintah:

```
data['Medal'].isna().sum()
```

```
231333
```

Jadi, ada 231.333 *record* yang tidak mendapatkan medali.

Untuk mendapatkan *record* yang mendapatkan medali *Gold* dengan umur tertua adalah dengan perintah:

```

gold = data.loc[data['Medal']=='Gold']
gold.loc[gold['Age']==gold['Age'].max()]

```

ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	Sport	Event	Medal
53238	Charles Jacobus	M	64.0	NaN	NaN	United States	USA	1904 Summer	1904	Summer	St Louis	Roque	Roque Men's Singles	Gold
117046	Oscar Gomer Swahn	M	64.0	NaN	NaN	Sweden	SWE	1912 Summer	1912	Summer	Stockholm	Shooting	Shooting Men's Running Target, Single Shot, Team	Gold

Untuk menampilkan jumlah peraih medali *Gold* oleh perempuan dari negara dan tahun tertentu adalah dengan perintah berikut:

```
womanGold = gold.loc[gold['Sex']=='F']
womanGold.groupby(['NOC','Year'])['Medal'].count()
NOC  Year
ALG  1992      1
      2000      1
ANZ  1912      1
ARG  2016      2
AUS  1932      1
      ..
YUG  1984     15
      1988      1
ZIM  1980     15
      2004      1
      2008      1
Name: Medal, Length: 507, dtype: int64
```

## Eksplorasi Beberapa Fungsi *Built-in*

Fungsi `notnull` dapat digunakan untuk menampilkan semua yang meraih medali. *Record* yang tidak mendapatkan medali tidak ditampilkan.

```
medal = data.loc[data['Medal'].notnull()]
medal
```

ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	Sport	Event	Medal
4	Edgar Lindenau Aabye	M	34.0	NaN	NaN	Denmark/Sweden	DEN	1900 Summer	1900	Summer	Paris	Tug-Of-War	Tug-Of-War Men's Tug-Of-War	Gold
15	Arvo Ossian Aaltonen	M	30.0	NaN	NaN	Finland	FIN	1920 Summer	1920	Summer	Antwerpen	Swimming	Swimming Men's 200 metres Breaststroke	Bronze
15	Arvo Ossian Aaltonen	M	30.0	NaN	NaN	Finland	FIN	1920 Summer	1920	Summer	Antwerpen	Swimming	Swimming Men's 400 metres Breaststroke	Bronze
16	Juhamatti Tapio Aaltonen	M	28.0	184.0	85.0	Finland	FIN	2014 Winter	2014	Winter	Sochi	Ice Hockey	Ice Hockey Men's Ice Hockey	Bronze
17	Paavo Johannes Aaltonen	M	28.0	175.0	64.0	Finland	FIN	1948 Summer	1948	Summer	London	Gymnastics	Gymnastics Men's Individual All-Around	Bronze
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
135553	Galina Ivanovna Zybina (-Fyodorova)	F	25.0	168.0	80.0	Soviet Union	URS	1956 Summer	1956	Summer	Melbourne	Athletics	Athletics Women's Shot Put	Silver
135553	Galina Ivanovna Zybina (-Fyodorova)	F	33.0	168.0	80.0	Soviet Union	URS	1964 Summer	1964	Summer	Tokyo	Athletics	Athletics Women's Shot Put	Bronze
135554	Bogusaw Zych	M	28.0	182.0	82.0	Poland	POL	1980 Summer	1980	Summer	Moskva	Fencing	Fencing Men's Foil, Team	Bronze
135563	Olesya Nikolayevna Zykina	F	19.0	171.0	64.0	Russia	RUS	2000 Summer	2000	Summer	Sydney	Athletics	Athletics Women's 4 x 400 metres Relay	Bronze
135563	Olesya Nikolayevna Zykina	F	23.0	171.0	64.0	Russia	RUS	2004 Summer	2004	Summer	Athina	Athletics	Athletics Women's 4 x 400 metres Relay	Silver

39783 rows × 14 columns



Fungsi `loc` dapat digunakan untuk memilih kolom tertentu untuk ditampilkan. Contoh menampilkan atlet Indonesia yang mendapat medali.

```
ina = data.loc[data['NOC']=='INA']
ina.loc[ina['Medal'].notnull()]
```

ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	Sport	Event	Medal
1301	Sri Wahyuni Agustiani	F	21.0	147.0	47.0	Indonesia	INA	2016 Summer	2016	Summer	Rio de Janeiro	Weightlifting	Weightlifting Women's Flyweight	Silver
1374	Tontowi Ahmad	M	29.0	179.0	72.0	Indonesia-1	INA	2016 Summer	2016	Summer	Rio de Janeiro	Badminton	Badminton Mixed Doubles	Gold
5040	Antonius Budi Ariantho	M	24.0	170.0	66.0	Indonesia-1	INA	1996 Summer	1996	Summer	Atlanta	Badminton	Badminton Men's Doubles	Bronze
16152	Alexander Alan Budikusuma Wiratama	M	24.0	178.0	71.0	Indonesia	INA	1992 Summer	1992	Summer	Barcelona	Badminton	Badminton Men's Singles	Gold
32609	Hian Eng	M	27.0	175.0	68.0	Indonesia-1	INA	2004 Summer	2004	Summer	Athina	Badminton	Badminton Men's Doubles	Bronze
43863	Rudy Gunawan	M	25.0	185.0	83.0	Indonesia-1	INA	1992 Summer	1992	Summer	Barcelona	Badminton	Badminton Men's Doubles	Silver

Fungsi `groupby` digunakan untuk membuat *group* berdasarkan nilai. Seperti meng-group-kan medali *Gold*, *Silver*, dan *Bronze*. Contoh menampilkan jumlah medali yang diperoleh Indonesia berdasarkan tahun dan jenis medali.

```
ina.groupby(['Year', 'Medal'])['Medal'].count()
```

```
Year  Medal
1988  Silver      3
1992  Bronze      1
      Gold        2
      Silver      3
1996  Bronze      3
      Gold        2
      Silver      1
2000  Bronze      2
      Gold        2
      Silver      4
2004  Bronze      3
      Gold        1
      Silver      1
2008  Bronze      3
      Gold        2
      Silver      2
2012  Bronze      1
      Silver      1
2016  Gold        2
      Silver      2
Name: Medal, dtype: int64
```

Fungsi `value_counts` untuk menjumlahkan unik data pada kolom yang dipilih. Contoh menghitung jumlah medali yang telah diperoleh atlet Indonesia.

```
ina['Medal'].value_counts()
```

```
Silver      17
```

```

Bronze      13
Gold        11
Name: Medal, dtype: int64

```

Fungsi `pivot_table` digunakan untuk membuat *pivot* tabel. Contoh berikut untuk menampilkan jumlah perolehan medali per negara di tahun tertentu.

```

pivot = pd.pivot_table(data[['Year', 'NOC', 'Medal', 'ID']],
index=['Year', 'NOC'], columns='Medal', aggfunc='count')
pivot

```

		ID			
	Medal	Bronze	Gold	No Medal	Silver
Year	NOC				
1896	AUS	1.0	2.0	2.0	NaN
	AUT	2.0	2.0	3.0	1.0
	DEN	3.0	1.0	9.0	2.0
	FRA	2.0	5.0	15.0	4.0
	GBR	3.0	3.0	16.0	3.0
...	...	...	...	...	...
2016	VIE	NaN	1.0	26.0	1.0
	VIN	NaN	NaN	4.0	NaN
	YEM	NaN	NaN	3.0	NaN
	ZAM	NaN	NaN	7.0	NaN
	ZIM	NaN	NaN	31.0	NaN

Fungsi `reindex` digunakan untuk membuat *index* baru. Nilai yang terdapat di *index* yang lama maka akan bernilai *Null/NaN*.

## Membersihkan dan Melengkapi Data

Berikut langkah-langkah yang dilakukan untuk membersihkan dan melengkapi data:

- Mengeluarkan data yang tidak memiliki informasi tentang medali.  
Berikut perintah yang dilakukan:

```

dataMedal = data[data['Medal'].notnull()]
dataMedal

```

ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	Sport	Event	Medal
4	Edgar Lindenau Aabye	M	34.0	NaN	NaN	Denmark/Sweden	DEN	1900 Summer	1900	Summer	Paris	Tug-Of-War	Tug-Of-War Men's Tug-Of-War	Gold
15	Arvo Ossian Aaltonen	M	30.0	NaN	NaN	Finland	FIN	1920 Summer	1920	Summer	Antwerpen	Swimming	Swimming Men's 200 metres Breaststroke	Bronze
15	Arvo Ossian Aaltonen	M	30.0	NaN	NaN	Finland	FIN	1920 Summer	1920	Summer	Antwerpen	Swimming	Swimming Men's 400 metres Breaststroke	Bronze
16	Juhamatti Tapio Aaltonen	M	28.0	184.0	85.0	Finland	FIN	2014 Winter	2014	Winter	Sochi	Ice Hockey	Ice Hockey Men's Ice Hockey	Bronze
17	Paavo Johannes Aaltonen	M	28.0	175.0	64.0	Finland	FIN	1948 Summer	1948	Summer	London	Gymnastics	Gymnastics Men's Individual All-Around	Bronze
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
135553	Galina Ivanovna Zybina (-Fyodorova)	F	25.0	168.0	80.0	Soviet Union	URS	1956 Summer	1956	Summer	Melbourne	Athletics	Athletics Women's Shot Put	Silver
135553	Galina Ivanovna Zybina (-Fyodorova)	F	33.0	168.0	80.0	Soviet Union	URS	1964 Summer	1964	Summer	Tokyo	Athletics	Athletics Women's Shot Put	Bronze
135554	Bogusaw Zych	M	28.0	182.0	82.0	Poland	POL	1980 Summer	1980	Summer	Moskva	Fencing	Fencing Men's Foil, Team	Bronze
135563	Olesya Nikolayevna Zykina	F	19.0	171.0	64.0	Russia	RUS	2000 Summer	2000	Summer	Sydney	Athletics	Athletics Women's 4 x 400 metres Relay	Bronze
135563	Olesya Nikolayevna Zykina	F	23.0	171.0	64.0	Russia	RUS	2004 Summer	2004	Summer	Athina	Athletics	Athletics Women's 4 x 400 metres Relay	Silver

39783 rows × 14 columns

Sehingga dari informasi di atas, diperoleh hanya data yang memiliki informasi medali.

Berikutnya mengisi umur (Age) yang kosong dengan rata-rata umur atlet. Maka kueri yang digunakan adalah:

```
dataMedal['Age'].fillna(value=dataMedal['Age'].mean(), inplace=True)
```

Namun untuk menghasilkan umur yang bulat tidak berupa desimal, perlu dilakukan pembulatan ke atas atau ke bawah. Berikut untuk pembulatan ke bawah:

```
import math
dataMedal['Age'].fillna(value=math.floor(dataMedal['Age'].mean()), inplace=True)
```

Berikutnya mengisi data tinggi (Height) yang kosong dengan ketentuan Tinggi perempuan yang kosong diisi dengan rata-rata tinggi perempuan dan tinggi laki-laki yang kosong diisi dengan rata-rata tinggi laki-laki. Berikut kueri yang digunakan:

```
dataMedal.loc[dataMedal['Sex']=='M']['Height'].fillna(dataMedal.loc[dataMedal['Sex']=='M']['Height'].mean(), inplace=True)
dataMedal.loc[dataMedal['Sex']=='F']['Height'].fillna(dataMedal.loc[dataMedal['Sex']=='F']['Height'].mean(), inplace=True)
```

Dan untuk data berat (*Weight*) yang kosong diisi dengan rata-rata berat pada Sport yang sama. Berikut kueri yang digunakan:

```
dataMedal["Weight"] =
dataMedal.groupby("Sport")["Weight"].transform(lambda x:
x.fillna(x.mean()))
dataMedal
```

Setelah dijalankan perintah-perintah di atas maka *output*-nya adalah sebagai berikut:

ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	Sport	Event	Medal
4	Edgar Lindenau Aabye	M	34.0	181.156113	94.137931	Denmark/Sweden	DEN	1900 Summer	1900	Summer	Paris	Tug-Of-War	Tug-Of-War Men's Tug-Of-War	Gold
15	Arvo Ossian Aaltonen	M	30.0	181.156113	73.251005	Finland	FIN	1920 Summer	1920	Summer	Antwerpen	Swimming	Swimming Men's 200 metres Breaststroke	Bronze
15	Arvo Ossian Aaltonen	M	30.0	181.156113	73.251005	Finland	FIN	1920 Summer	1920	Summer	Antwerpen	Swimming	Swimming Men's 400 metres Breaststroke	Bronze
16	Juhamatti Tapio Aaltonen	M	28.0	181.156113	85.000000	Finland	FIN	2014 Winter	2014	Winter	Sochi	Ice Hockey	Ice Hockey Men's Ice Hockey	Bronze
17	Paavo Johannes Aaltonen	M	28.0	181.156113	64.000000	Finland	FIN	1948 Summer	1948	Summer	London	Gymnastics	Gymnastics Men's Individual All-Around	Bronze
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
135553	Galina Ivanovna Zybina (-Fyodorova)	F	25.0	168.000000	80.000000	Soviet Union	URS	1956 Summer	1956	Summer	Melbourne	Athletics	Athletics Women's Shot Put	Silver
135553	Galina Ivanovna Zybina (-Fyodorova)	F	33.0	168.000000	80.000000	Soviet Union	URS	1964 Summer	1964	Summer	Tokyo	Athletics	Athletics Women's Shot Put	Bronze
135554	Bogusaw Zych	M	28.0	181.156113	82.000000	Poland	POL	1980 Summer	1980	Summer	Moskva	Fencing	Fencing Men's Foil, Team	Bronze

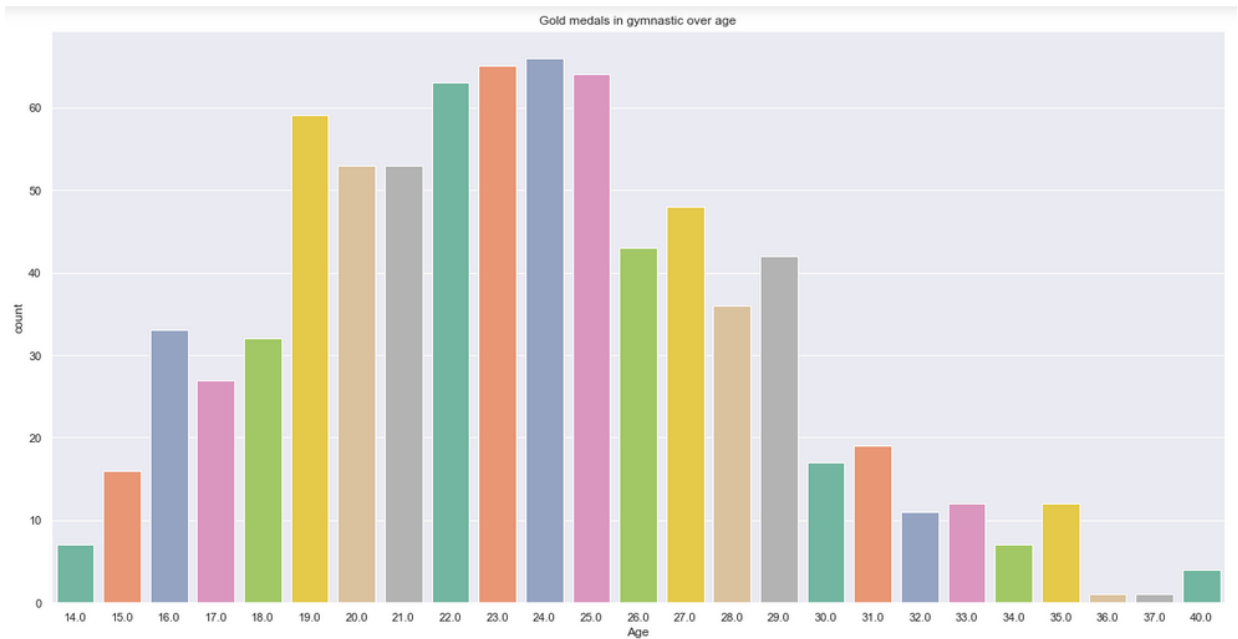
## Visualisasi Data

### 1. Medali emas terhadap usia

Berikut perintah untuk menampilkan diagram batang jumlah medali emas terhadap usia menggunakan *Countplot* pada olahraga Gymnastics.

```
import seaborn as sb
from matplotlib import pyplot as plot

gymData=dataMedal[(dataMedal['Sport']=='Gymnastics') &
(dataMedal['Medal']=='Gold')]
sb.set(style="darkgrid")
plot.figure(figsize=(20, 10))
sb.countplot(x='Age', data=gymData, palette='Set2')
plot.title('Gold medals in gymnastic over age')
```

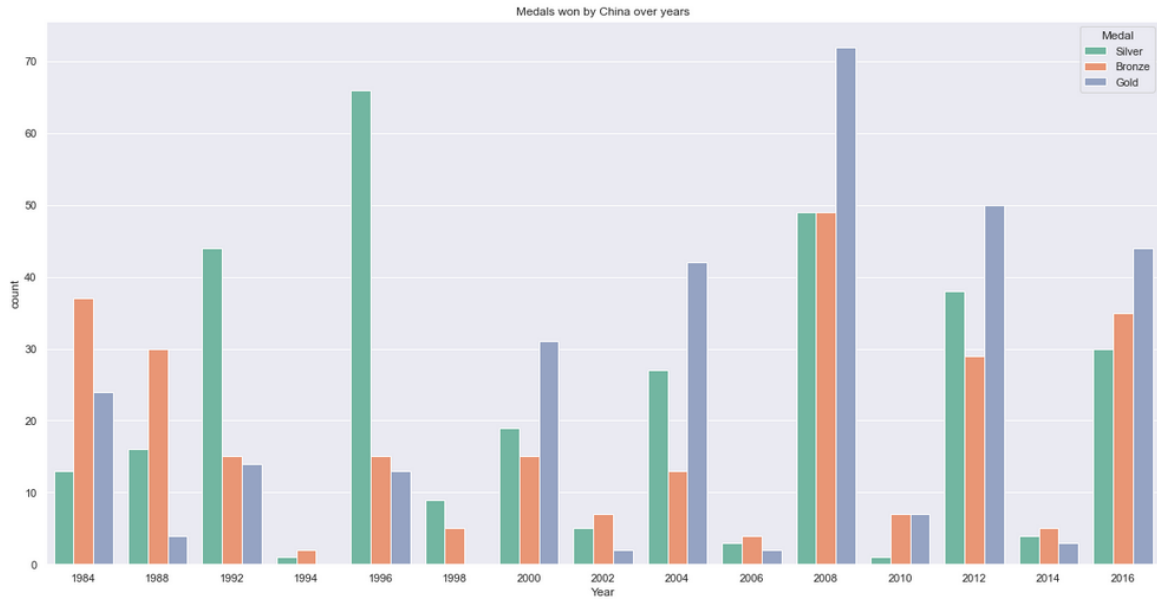


Dari informasi distribusi umur terhadap perolehan medali emas pada permainan *gymnastic*, diketahui bahwa umur 24 yang terbanyak perolehan medali emasnya, diikuti umur 23 dan 25.

## 2. Medali yang telah diperoleh China Terhadap Tahun

Untuk menampilkan data perolehan medali oleh China dari tahun-ketahun dengan menggunakan perintah:

```
chinaMedal=dataMedal[(dataMedal['Team']=='China')]
sb.set(style="darkgrid")
plot.figure(figsize=(20, 10))
sb.countplot(x='Year', hue='Medal', data=chinaMedal,
palette='Set2')
plot.title('Medals won by China over years')
```

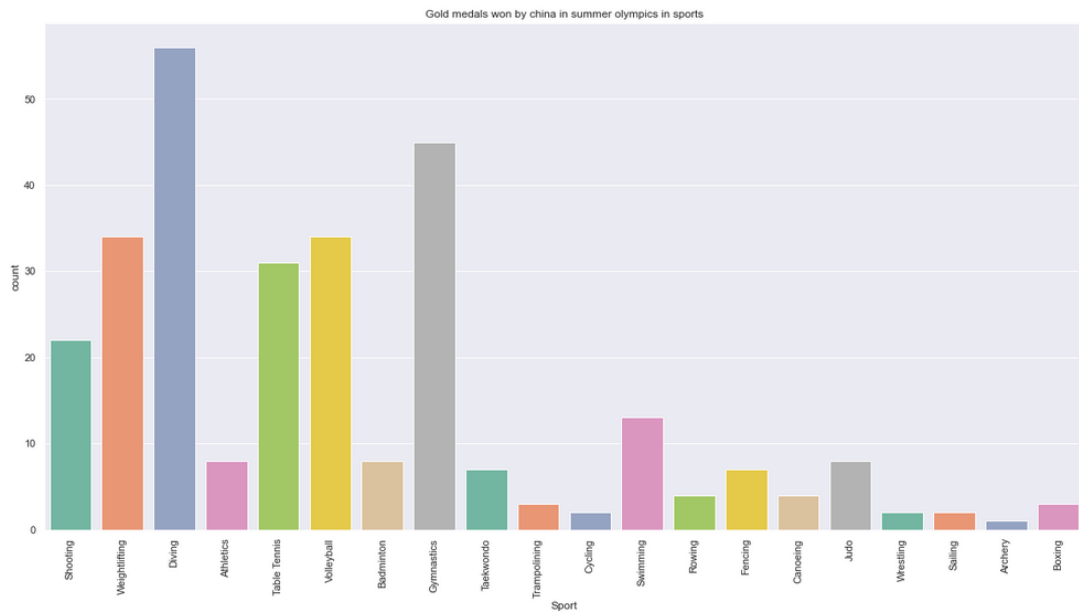


Dari informasi di atas diketahui bahwa perolehan medali terbanyak oleh China adalah pada tahun 2008. Medali emas terbanyak yang diperoleh China juga pada tahun 2008.

### 3. Medali emas yang diperoleh China di Olimpiade Musim Panas terhadap olahraga (Sport)

Untuk menampilkan emas yang diperoleh China terhadap olahraga selama Olimpiade musim panas adalah dengan perintah berikut:

```
chinaSummerGold=dataMedal[(dataMedal['Season']=='Summer') &
(dataMedal['Team']=='China') & (dataMedal['Medal']=='Gold')]
sb.set(style="darkgrid")
plot.figure(figsize=(20, 10))
plot.xticks(rotation=90)
sb.countplot(x='Sport', data=chinaSummerGold,
palette='Set2')
plot.title('Gold medals won by china in summer olympics in
sports')
```

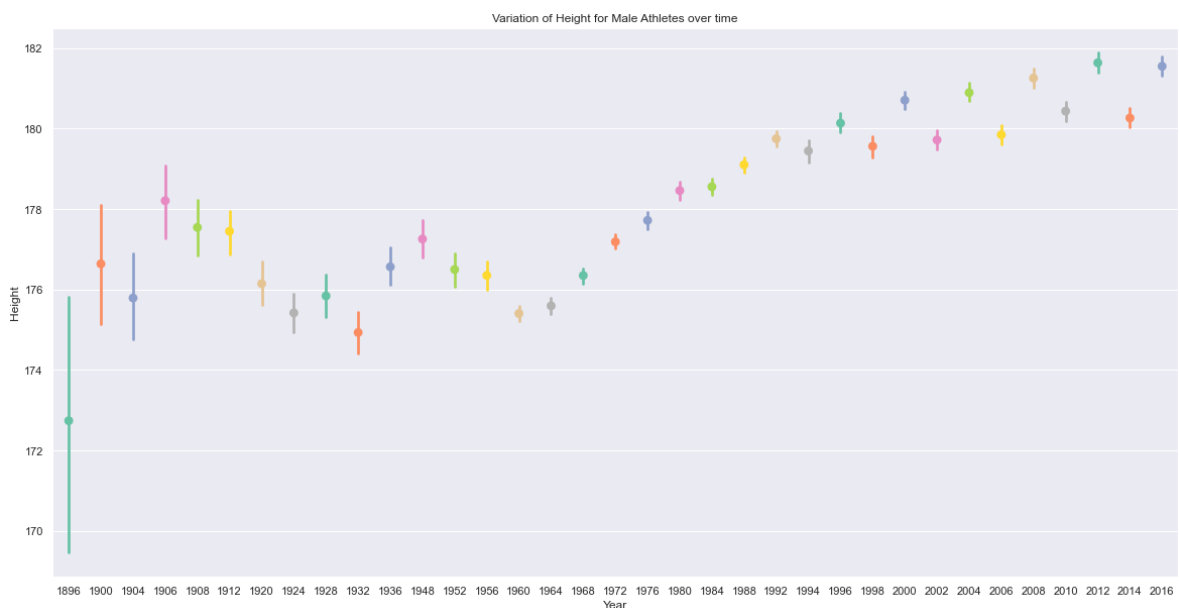


Dari informasi di atas diketahui bahwa perolehan emas oleh China di olimpiade musim panas terbanyak adalah pada olahraga *Diving*, diikuti oleh olahraga *Gymnastic*, *Volleyball*, *Weightlifting*, dan *Table Tennis*.

### 1. Variasi ukuran tinggi atlet laki-laki

Untuk menampilkan variasi tinggi atlet laki-laki dari waktu ke waktu adalah dengan perintah berikut:

```
menData=data[data['Sex']=='M']
plot.figure(figsize=(20, 10))
sb.pointplot('Year', 'Height', data=menData, palette='Set2')
plot.title('Variation of Height for Male Athletes over
time')
```

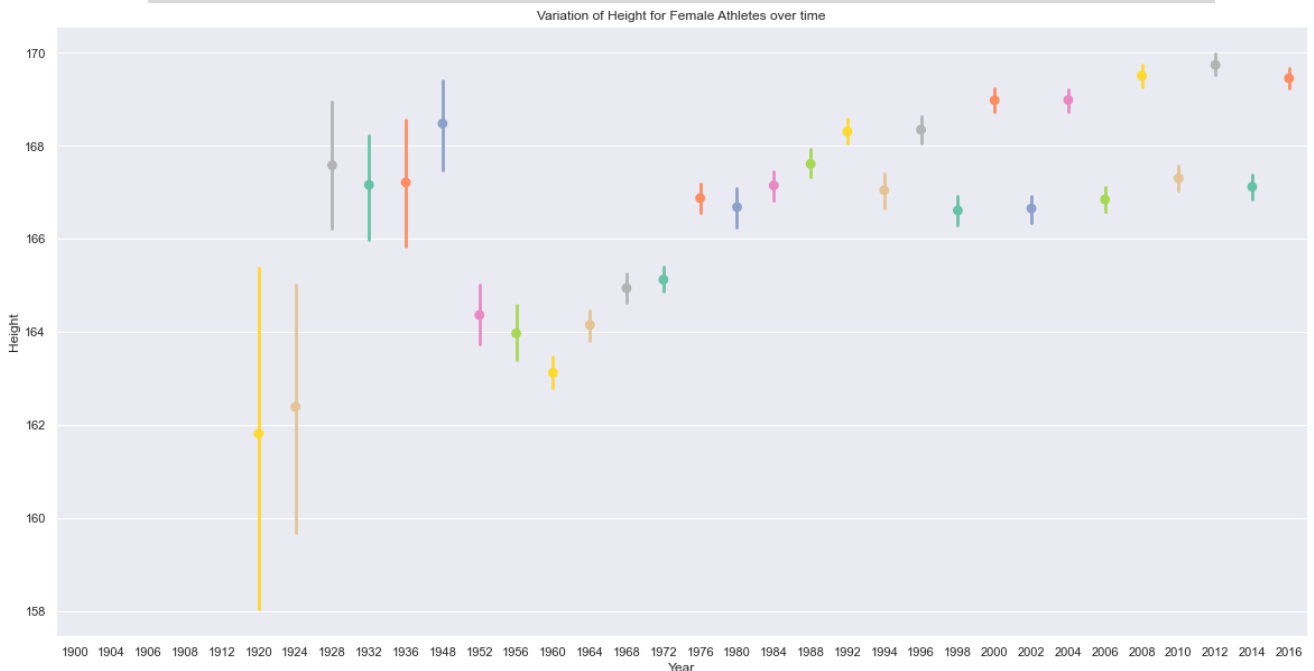


Dari hasil di atas diketahui bahwa terjadi peningkatan rata-rata tinggi badan atlet yang mengikuti olimpiade dari tahun 1964 sampai 2016 di tipe musim yang sama.

## 2. Variasi ukuran tinggi atlet perempuan

Untuk menampilkan variasi tinggi atlet perempuan dari waktu ke waktu adalah dengan perintah berikut:

```
femaleData=data[data['Sex']=='F']
plot.figure(figsize=(20, 10))
sb.pointplot('Year', 'Height', data=femaleData,
palette='Set2')
plot.title('Variation of Height for Female Athletes over
time')
```



Dari hasil di atas diketahui bahwa peningkatan rata-rata tinggi badan atlet perempuan olimpiade mulai tahun 1964 sampai 2016. Namun untuk *Winter*, tidak terjadi peningkatan yang signifikan.

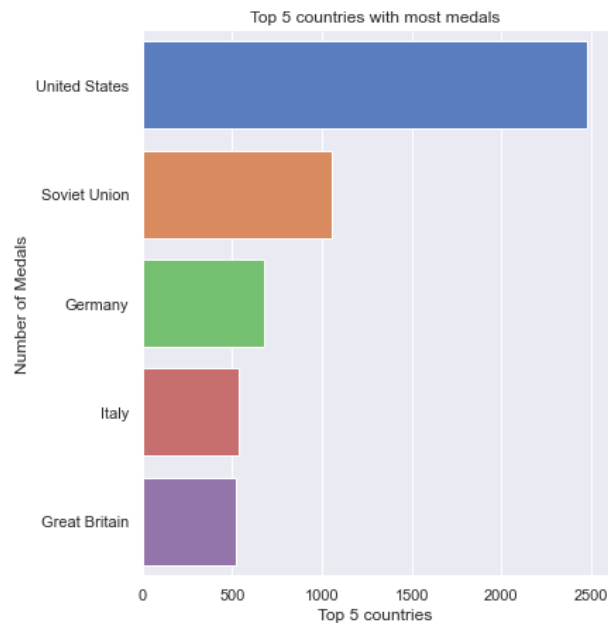
### **Barplot**

Berikut kode *barplot* untuk menampilkan top 5 negara peraih medali emas terbanyak sepanjang masa:



```
goldMedal = dataMedal[dataMedal['Medal']=='Gold']
top5goldMedal =
goldMedal['Team'].value_counts().reset_index(name='Medal').head(5)

g = sb.catplot(x="Medal", y="index", data=top5goldMedal,
               height=6, kind="bar", palette="muted")
g.despine(left=True)
g.set_xlabels("Top 5 countries")
g.set_ylabels("Number of Medals")
plot.title('Top 5 countries with most medals')
```

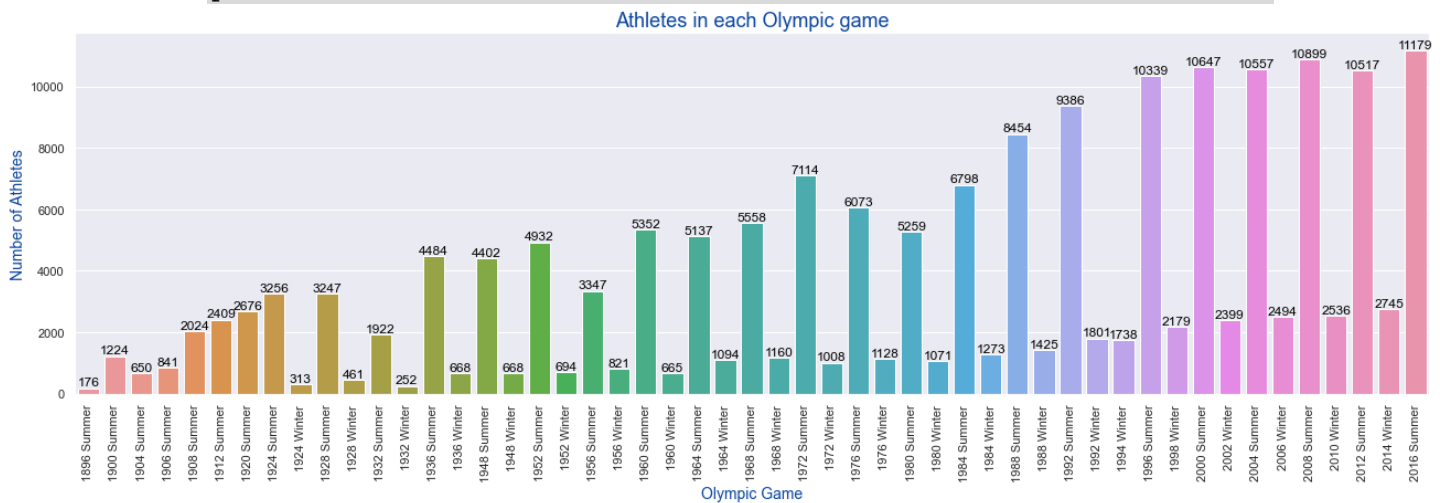


Dari *barplot* di atas diketahui bahwa Amerika Serikat adalah peraih medali emas terbanyak sepanjang masa. Peringkat berikutnya diikuti oleh Uni Soviet, Jerman, Itali dan Inggris.

Berikutnya *barplot* untuk banyaknya atlet di setiap tahunnya adalah dengan perintah berikut:

```
athletes = data.pivot_table(data, index=['Games'],
aggfunc=lambda x:
len(x.unique())).reset_index()[['Games','ID']]
fig, ax = plot.subplots(figsize=(22,6))
a = sb.barplot(x='Games', y='ID', data=athletes, ax=ax)
a.set_xticklabels(labels=athletes['Games'],rotation=90)
for p in ax.patches:
    ax.text(p.get_x() + p.get_width()/2., p.get_height(),
'%d' % int(p.get_height()),
          fontsize=12, color='black', ha='center',
va='bottom')
ax.set_xlabel('Olympic Game', size=14, color="#0D47A1")
ax.set_ylabel('Number of Athletes', size=14,
color="#0D47A1")
```

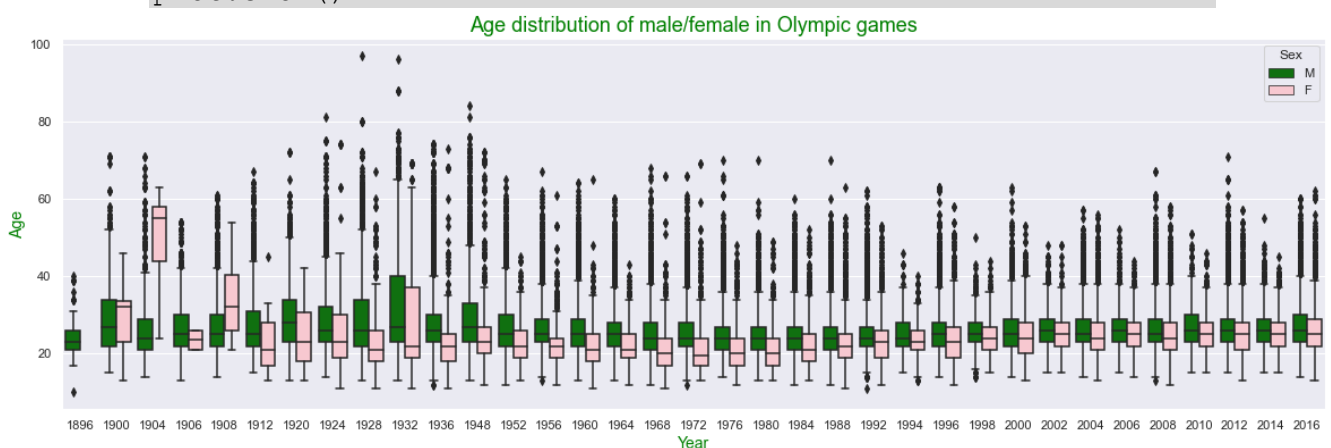
```
ax.set_title('Athletes in each Olympic game', size=18,
color="#0D47A1")
plot.show()
```



Berdasarkan informasi *barplot* di atas diketahui bahwa jumlah atlet terus mengalami peningkatan dari tahun ke tahun. Terlihat bahwa mulai tahun 1996 sudah mencapai di atas 10.000 atlet.

Berikut boxplot untuk untuk menampilkan umur laki-laki dan perempuan.

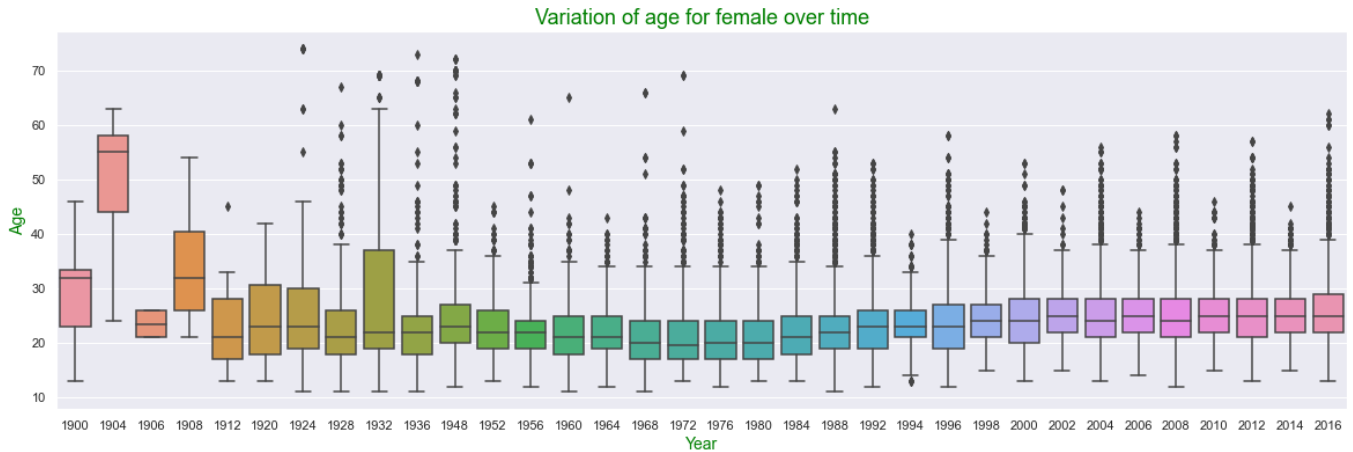
```
fig, ax = plot.subplots(figsize=(20,6))
a = sb.boxplot(x="Year", y="Age", hue="Sex", palette={"M":
"green", "F":"pink"}, data=data, ax=ax)
ax.set_xlabel('Year', size=14, color="green")
ax.set_ylabel('Age', size=14, color="green")
ax.set_title('Age distribution of male/female in Olympic
games', size=18, color="green")
plot.show()
```



Berikut boxplot untuk distribusi umur atlet perempuan tiap tahunnya.

```
fig, ax = plot.subplots(figsize=(20,6))
a = sb.boxplot(x="Year",y="Age",
data=data[data['Sex']=='F'], ax=ax)
```

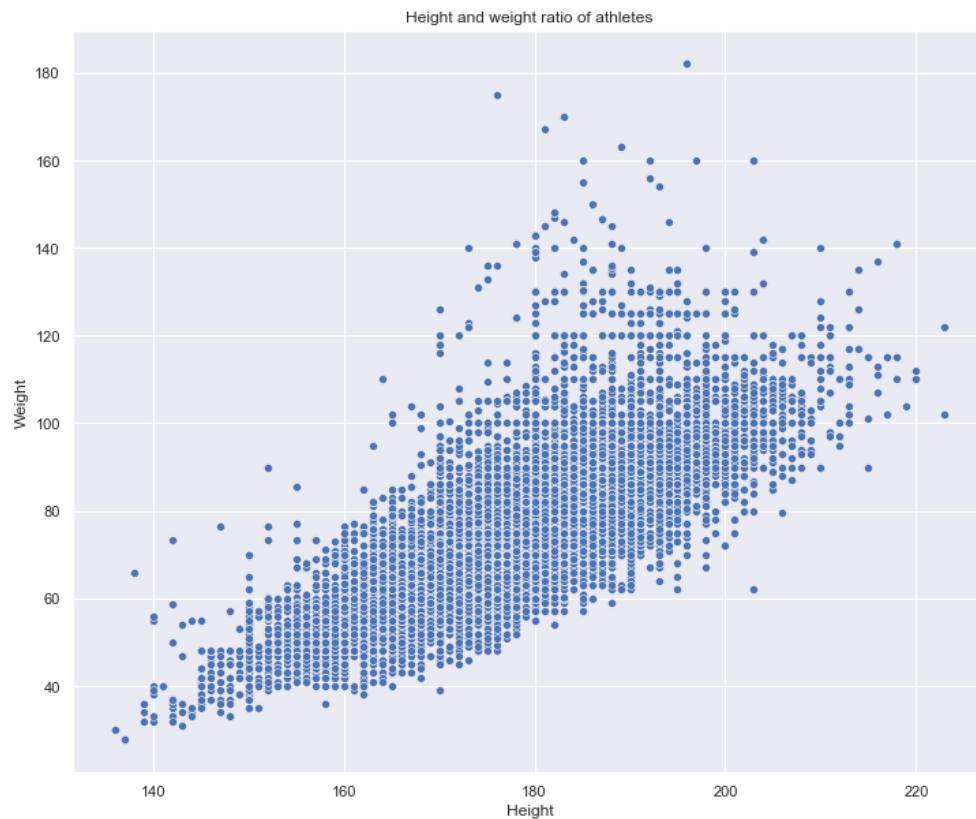
```
ax.set_xlabel('Year', size=14, color="green")
ax.set_ylabel('Age', size=14, color="green")
ax.set_title('Variation of age for female over time',
size=18, color="green")
plot.show()
```



### Scatterplot

Berikut *scatterplot* untuk melihat rasio berat terhadap tinggi atlet.

```
plot.figure(figsize=(12, 10))
ax = sb.scatterplot(x="Height", y="Weight", data=dataMedal)
plot.title('Height and weight ratio of athletes')
```



Dari *scatterplot* di atas dapat diketahui bahwa tinggi badan atlet 160 – 200 cm dan berat badan atlet 60-100 kg mendominasi.

### Heatmap

Berikut heatmap untuk memetakan rata-rata umur terhadap peralihan medali.

```
import numpy as np

data['Medal'].fillna(value="No Medal",inplace=True)

yearMedal = data.pivot_table(data, index=['Year','Medal'],
aggfunc=np.mean).reset_index()[['Year','Medal','Age']]

yearMedal = yearMedal.pivot("Medal", "Year", "Age")

yearMedal = yearMedal.reindex(["Gold","Silver","Bronze","No Medal"])

f, ax = plot.subplots(figsize=(25, 5))

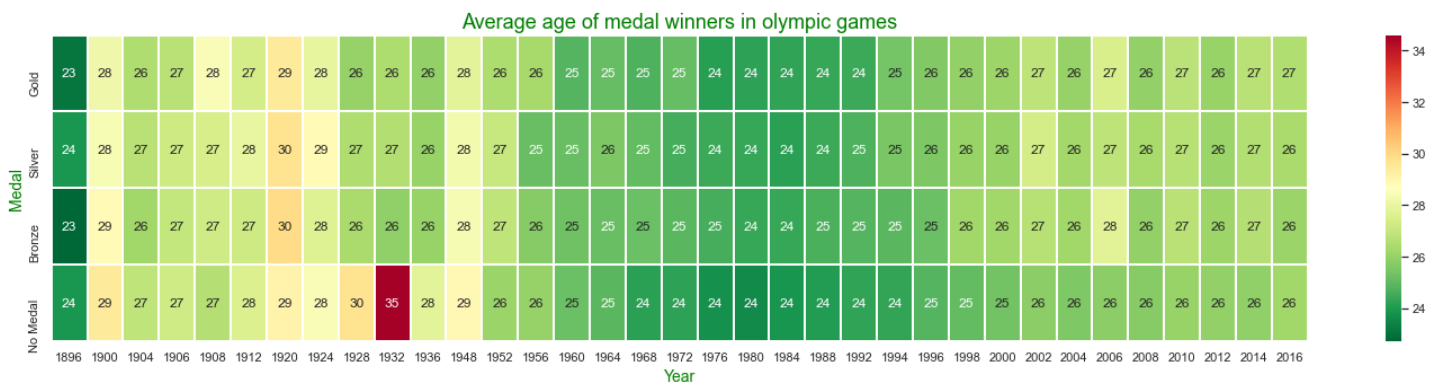
sb.heatmap(yearMedal, annot=True, linewidths=0.05, ax=ax,
cmap="RdYlGn_r")

ax.set_xlabel('Year', size=14, color="green")

ax.set_ylabel('Medal', size=14, color="green")

ax.set_title('Average age of medal winners in olympic games', size=18, color="green")

plot.show()
```



Berdasarkan informasi dari heatmap di atas, rata-rata umur peraih medal paling muda adalah 23 tahun di tahu 1896, kemudian diikuti rata-rata umur 24 tahun mulai dari tahun 1976 sampai dengan 1992. Mulai tahun 1994 sampai dengan 2016 peraih medali rata-rata oleh umur 25-27 tahun.