Olympic Data Analysis with Pymongo

Why MongoDB?

One of the most popular NoSQL databases is MongoDB. MongoDB uses JSON-like documents that can have varied structures. It uses the MongoDB query language to allow access to the stored data. Since it is schema-free, you can create documents without having to create the structure for the document first.

When a query is made to mongodb,Indexes are preferred for searching the documents within the collection. But if an index is missing, every document within the collection must be searched to select the documents that were requested in the query.

What is Pymongo?

PyMongo is a Python distribution containing tools for working with MongoDB.

Data Collection

The Data files of csv format have been downloaded from kaggle which includes the details of all the olympic games from Athens 1896 to Rio 2016.
There are two csv files namely athlete_events.csv and noc_regions.csv with key column as NOC.
We worked on the combined csv of the above files which are joined based on the NOC column.

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 the following:

1. ID - Unique number for each athlete;
2. Name - Athlete's name;
3. Sex - M or F;
4. Age - Integer;
5. Height - In centimeters;
6. Weight - In kilograms;

7. Team - Team name;
8. NOC - National Olympic Committee 3-letter code;
9. Games - Year and season;
10. Year - Integer;
11. Season - Summer or Winter;
12. City - Host city;
13. Sport - Sport;
14. Event - Event;
15. Medal - Gold, Silver, Bronze, or NA.

Now let's start the process!

Before getting started we need to preprocess the data.

Preprocessing includes
1.Splitting the attributes into independent and dependent attributes.
2.Handling the missing values.
3.Feature Scaling.
4.Splitting the dataset into a training and testing set.

1.Splitting the attributes into independent and dependent attributes.

```
# libraries
import numpy as np # used for handling numbers
import pandas as pd # used for handling the dataset
from sklearn.impute import SimpleImputer # used for handling missing data
from sklearn.preprocessing import LabelEncoder, OneHotEncoder # used for encoding categorical data
from sklearn.model_selection import train_test_split # used for splitting training and testing data
from sklearn.preprocessing import StandardScaler # used for feature scaling
```

```
dataset = pd.read_csv('/Users/apple/output.csv') # to import the dataset into a variable
# Splitting the attributes into independent and dependent attributes
X = dataset.iloc[:, :-1].values # attributes to determine dependent variable / Class
Y = dataset.iloc[:, -1].values # dependent variable / Class
```

```
X
```

```
array([[1, 'A Dijiang', 'M', ..., "Basketball Men's Basketball", nan,
        'China'],
       [2, 'A Lamusi', 'M', ..., "Judo Men's Extra-Lightweight", nan,
        'China'],
       [3, 'Gunnar Nielsen Aaby', 'M', ..., "Football Men's Football",
        nan, 'Denmark'],
       ...,
       [135570, 'Piotr ya', 'M', ...,
        "Ski Jumping Men's Large Hill, Team", nan, 'Poland'],
       [135571, 'Tomasz Ireneusz ya', 'M', ..., "Bobsleigh Men's Four",
        nan, 'Poland'],
       [135571, 'Tomasz Ireneusz ya', 'M', ..., "Bobsleigh Men's Four",
        nan, 'Poland']], dtype=object)
```

```
Y
```

2.Handling the missing data by replacing values with mean of all the other values.

```
# handling the missing data and replace missing values with nan from numpy and replace with mean of all the other value
imputer = SimpleImputer(missing_values=np.nan, strategy='mean')
imputer = imputer.fit(X[:, 3:6])
X[:, 3:6] = imputer.transform(X[:, 3:6])
```

```
X[:, 3:6]
```

```
array([[24.0, 180.0, 80.0],
       [23.0, 170.0, 60.0],
       [24.0, 175.33896987366376, 70.70239290053351],
       ...,
       [27.0, 176.0, 59.0],
       [30.0, 185.0, 96.0],
       [34.0, 185.0, 96.0]], dtype=object)
```

3.Splitting the dataset into training and testing

```
# splitting the dataset into training set and test set
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=0)
```

```
X_train, X_test, Y_train, Y_test
```

```
(array([[107795, 'Uta Schtz', 'F', ...,
         "Swimming Women's 200 metres Freestyle", nan, 'Germany'],
        [2931, 'Rose "Rosie" Allwood (-Morrison)', 'F', ...,
         "Athletics Women's 4 x 100 metres Relay", nan, 'Jamaica'],
        [19866, 'Chaput', 'M', ..., "Cycling Men's Sprint", nan, 'France'],
        ...,
        [61965, 'Leopold Kohl', 'M', ...,
         "Cross Country Skiing Men's 18 kilometres", nan, 'Austria'],
        [76445, 'Mulomowandau Erick Mathoho', 'M', ...,
         "Football Men's Football", nan, 'South Africa'],
        [59683, 'David Taro Kikuchi', 'M', ...,
         "Gymnastics Men's Individual All-Around", nan, 'Canada']],
       dtype=object), array([[109485, 'Shek Wai Hung', 'M', ...,
         "Gymnastics Men's Pommelled Horse", nan, 'China'],
        [116150, 'Mlanie Suchet', 'F', ...,
         "Alpine Skiing Women's Downhill", nan, 'France'],
        [92674, 'Susan Jane "Sue" Pedersen (-Pankey)', 'F', ...,
         "Swimming Women's 100 metres Freestyle", 'Silver', 'USA'],
        ...,
        [129606, 'Michael Vincent "Mike" Wenden', 'M', ...,
         "Swimming Men's 4 x 100 metres Freestyle Relay", 'Bronze',
         'Australia'],
        [39329, 'Wilfried Geeroms', 'M', ...,
         "Athletics Men's 400 metres Hurdles", nan, 'Belgium'],
        [5977, 'Klaus Auhuber', 'M', ..., "Ice Hockey Men's Ice Hockey",
         nan, 'Germany']], dtype=object), array([nan, nan, nan, ..., nan, nan, nan], dtype=object), array(['Hong Kon
g', nan, nan, ..., nan, nan, nan], dtype=object))
```

Analysing the data with Pymongo

1.Importing the required modules-

```
import pandas as pd
import pymongo
import json
import os
from pymongo import MongoClient
import chart_studio.plotly as py
import plotly.graph_objs as go
import plotly
from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot
import cufflinks as cf
```

2.Merging two csv files-

```
#merging two csv files on key column NOC

a = pd.read_csv("/Users/apple/Desktop/120-years-of-olympic-history-athletes-and-results/athlete_events.csv")
b = pd.read_csv("/Users/apple/Desktop/120-years-of-olympic-history-athletes-and-results/noc_regions.csv")
merged = pd.merge(a,b, on='NOC',how='left')
merged.to_csv("output.csv", index=False)
```

Now we will be using the merged csv file for our analysis.

3.A glance into sample data.

```
merged.head()
```

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	Sport	Event	Medal	region	notes
0	1	A Dijiang	M	24.0	180.0	80.0	China	CHN	1992 Summer	1992	Summer	Barcelona	Basketball	Basketball Men's Basketball	NaN	China	NaN
1	2	A Lamusi	M	23.0	170.0	60.0	China	CHN	2012 Summer	2012	Summer	London	Judo	Judo Men's Extra-Lightweight	NaN	China	NaN
2	3	Gunnar Nielsen Aaby	M	24.0	NaN	NaN	Denmark	DEN	1920 Summer	1920	Summer	Antwerpen	Football	Football Men's Football	NaN	Denmark	NaN
3	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	Denmark	NaN
4	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	Netherlands	NaN

4.Connecting to MongoDB locally and uploading csv file into it-

```
import pandas as pd
import pymongo
import json
import os
from pymongo import MongoClient
def import_csvfile(filepath):
 #Connecting to MongoDB using port 27017
 mng_client = pymongo.MongoClient('localhost', 27017)
 #Creating a database
 mng_db = mng_client['Project']
 #Creating a collection to store all documents
 collection_name = 'collection'
 db_cm = mng_db[collection_name]
 cdir = os.path.dirname('__file__')
 file_res = os.path.join(cdir, filepath)
 #Reading csv file
 data = pd.read_csv(file_res)
 #Converting file into json format
 data_json = json.loads(data.to_json(orient='records'))
 #Insering data into MongoDB
 db_cm.insert_many(data_json)
if __name__ == "__main__":
 filepath = '/Users/apple/output.csv'
 import_csvfile(filepath)
```

5.Now let's check the connection and have a look into all the databases managed by the client.

```
#List of names of the databases managed by the client
client = MongoClient()
db_names = client.list_database_names()
print(db_names)
```

```
['Assignment_2', 'BigData_A2', 'Desktop', 'Project', 'admin', 'bushfire', 'business', 'config', 'local', 'mongo', 'my
db']
```

These are the databases existing in the system and our current working database is “Project”.

6.Let’s have a look into the collections in Project Database.

```
# Save a list of names of the collections managed by the "Project" database
Project_coll_names = client.Project.list_collection_names()
print(Project_coll_names)
```

```
['collection', 'python_test', 'things']
```

These are the collections in the Project Database and our present collection is “Collection”.

7. Connecting to Project Database

```
# Connect to the "Project" database
db = client.Project
```

As the connection is made,Now let's check the documents present in the database.

8.Collections in the Database.

```
# Retrieve sample collection documents
collection = db.collection.find_one()
# Print the sample collection documents
print(collection)
print(type(collection))
```

```
{'_id': ObjectId('5e974ac2122d8c72d5e5500e'), 'ID': 1, 'Name': 'A Dijiang', 'Sex': 'M', 'Age': 24.0, 'Height': 180.0,
'Weight': 80.0, 'Team': 'China', 'NOC': 'CHN', 'Games': '1992 Summer', 'Year': 1992, 'Season': 'Summer', 'City': 'Bar
celona', 'Sport': 'Basketball', 'Event': "Basketball Men's Basketball", 'Medal': None, 'region': 'China', 'notes': No
ne}
<class 'dict'>
```

9.Displaying fields present in each document.

```
# Get the fields present in each type of document
collection_fields = list(collection.keys())

print(collection_fields)
```

```
['_id', 'ID', 'Name', 'Sex', 'Age', 'Height', 'Weight', 'Team', 'NOC', 'Games', 'Year', 'Season', 'City', 'Sport', 'E
vent', 'Medal', 'region', 'notes']
```

10.Counting all the documents in the database.

```
#use empty document{} as a filter
filter = {}
#counting documents in a collection
n_project = db.collection.count_documents(filter)
print(n_project)
```

```
271116
```

11.Displaying only Gold Medalists.

From this huge data,let’s filter athletes who won Gold medals and begin our analysis.

```
#Filtering only gold medalists
db.collection.count_documents({"Medal" : "Gold"})
```

```
13372
```

```
x=db.collection.find({"Medal" : "Gold"}).limit(5)
for y in x:
 print(y)
```

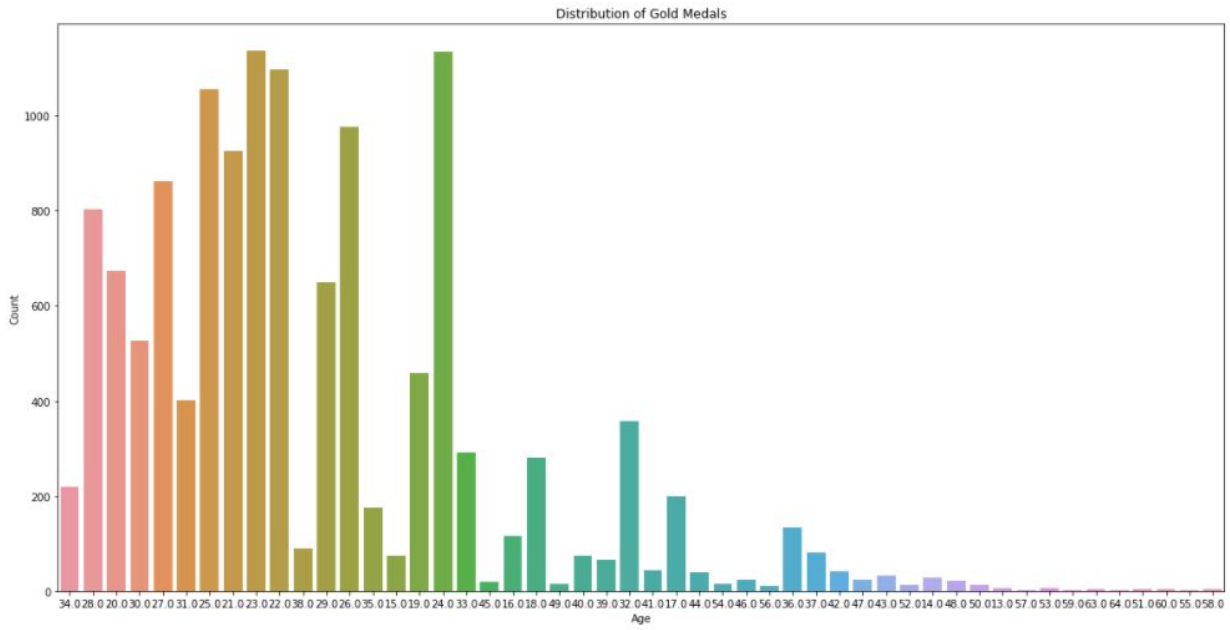
```
{'_id': ObjectId('5e974ac2122d8c72d5e55011'), 'ID': 4, 'Name': 'Edgar Lindenau Aabye', 'Sex': 'M', 'Age': 34.0, 'Heig
ht': None, 'Weight': None, 'Team': 'Denmark/Sweden', 'NOC': 'DEN', 'Games': '1900 Summer', 'Year': 1900, 'Season': 'S
ummer', 'City': 'Paris', 'Sport': 'Tug-Of-War', 'Event': "Tug-Of-War Men's Tug-Of-War", 'Medal': 'Gold', 'region': 'D
enmark', 'notes': None}
{'_id': ObjectId('5e974ac2122d8c72d5e55038'), 'ID': 17, 'Name': 'Paavo Johannes Aaltonen', 'Sex': 'M', 'Age': 28.0,
'Height': 175.0, 'Weight': 64.0, 'Team': 'Finland', 'NOC': 'FIN', 'Games': '1948 Summer', 'Year': 1948, 'Season': 'Su
mmer', 'City': 'London', 'Sport': 'Gymnastics', 'Event': "Gymnastics Men's Team All-Around", 'Medal': 'Gold', 'regio
n': 'Finland', 'notes': None}
{'_id': ObjectId('5e974ac2122d8c72d5e5503a'), 'ID': 17, 'Name': 'Paavo Johannes Aaltonen', 'Sex': 'M', 'Age': 28.0,
'Height': 175.0, 'Weight': 64.0, 'Team': 'Finland', 'NOC': 'FIN', 'Games': '1948 Summer', 'Year': 1948, 'Season': 'Su
mmer', 'City': 'London', 'Sport': 'Gymnastics', 'Event': "Gymnastics Men's Horse Vault", 'Medal': 'Gold', 'region':
'Finland', 'notes': None}
{'_id': ObjectId('5e974ac2122d8c72d5e5503e'), 'ID': 17, 'Name': 'Paavo Johannes Aaltonen', 'Sex': 'M', 'Age': 28.0,
'Height': 175.0, 'Weight': 64.0, 'Team': 'Finland', 'NOC': 'FIN', 'Games': '1948 Summer', 'Year': 1948, 'Season': 'Su
mmer', 'City': 'London', 'Sport': 'Gymnastics', 'Event': "Gymnastics Men's Pommelled Horse", 'Medal': 'Gold', 'regio
n': 'Finland', 'notes': None}
{'_id': ObjectId('5e974ac2122d8c72d5e5504a'), 'ID': 20, 'Name': 'Kjetil Andr Aamodt', 'Sex': 'M', 'Age': 20.0, 'Heigh
t': 176.0, 'Weight': 85.0, 'Team': 'Norway', 'NOC': 'NOR', 'Games': '1992 Winter', 'Year': 1992, 'Season': 'Winter',
'City': 'Albertville', 'Sport': 'Alpine Skiing', 'Event': "Alpine Skiing Men's Super G", 'Medal': 'Gold', 'region':
'Norway', 'notes': None}
```

There are 13372 gold medalists.

12.Age Distribution of Gold Medalists

```
query={
    "Medal":"Gold",
    "Age" : {"$exists" : True}
}
rm = list(db.collection.find(query))
Age=[collection["Age"] for collection in rm]
plt.clf()
plt.figure(figsize=(20, 10))
ax=s.countplot(Age)
plt.title('Distribution of Gold Medals')
plt.xlabel('Age')
plt.ylabel('Count')
plt.show()
```

```
<Figure size 432x288 with 0 Axes>
```



Inference-

1. There are athletes who won gold medals with age greater than 50.
2. Most of the athletes who won gold medals are of age 23 or 24.

Let's continue our analysis with gold medalists with age greater than 50.

Firstly, let's count the number of athletes with above conditions.

```
masterDisciplines = db.collection.count_documents({"Medal":"Gold","Age":{"$gt":50}})
masterDisciplines
```

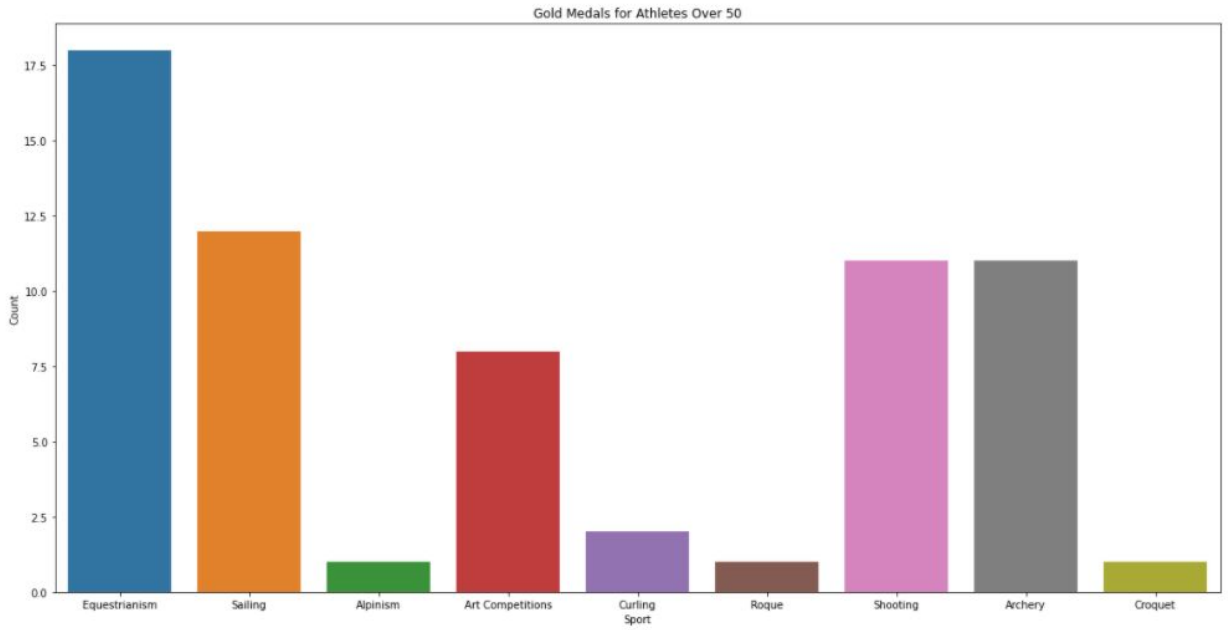
```
65
```

65 athletes,So now let’s learn more about the disciplines which allowed them to win gold medals even after their fifties?

We will now create a new document called masterDisciplines in which we will insert this new set of people and then create a visualization with it.

13.Displaying Gold Medals for Athletes.

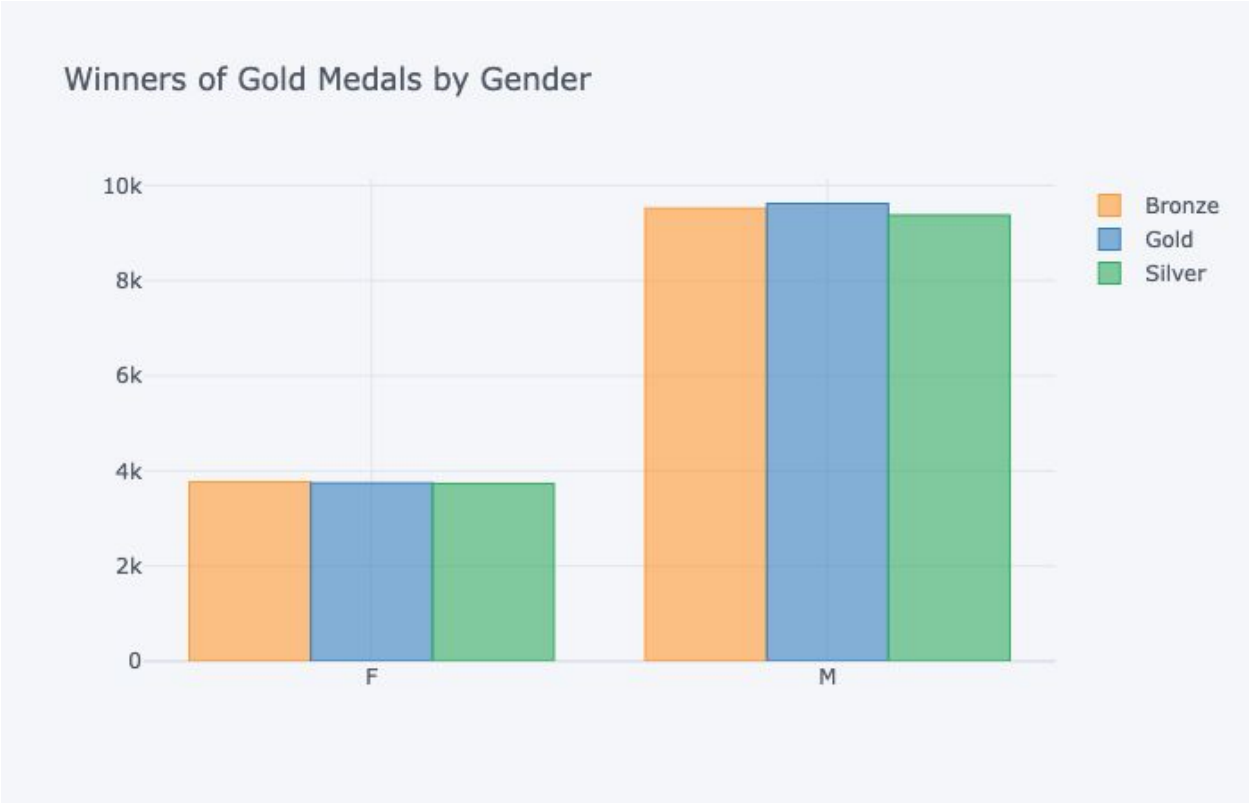
```
MasterDisciplines={
    "Medal":"Gold",
    "Age" : {"$gt":50},
    "Sport" : {"$exists" : True}
}
rm = list(db.collection.find(query))
Sport=[collection["Sport"] for collection in rm]
plt.clf()
plt.figure(figsize=(20, 10))
ax=s.countplot(Sport)
plt.title('Gold Medals for Athletes Over 50')
plt.xlabel('Sport')
plt.ylabel('Count')
plt.show()
```



Inferences-

1.The senior medalists excel in Equestrianism, Shooting, Archery and Sailing.

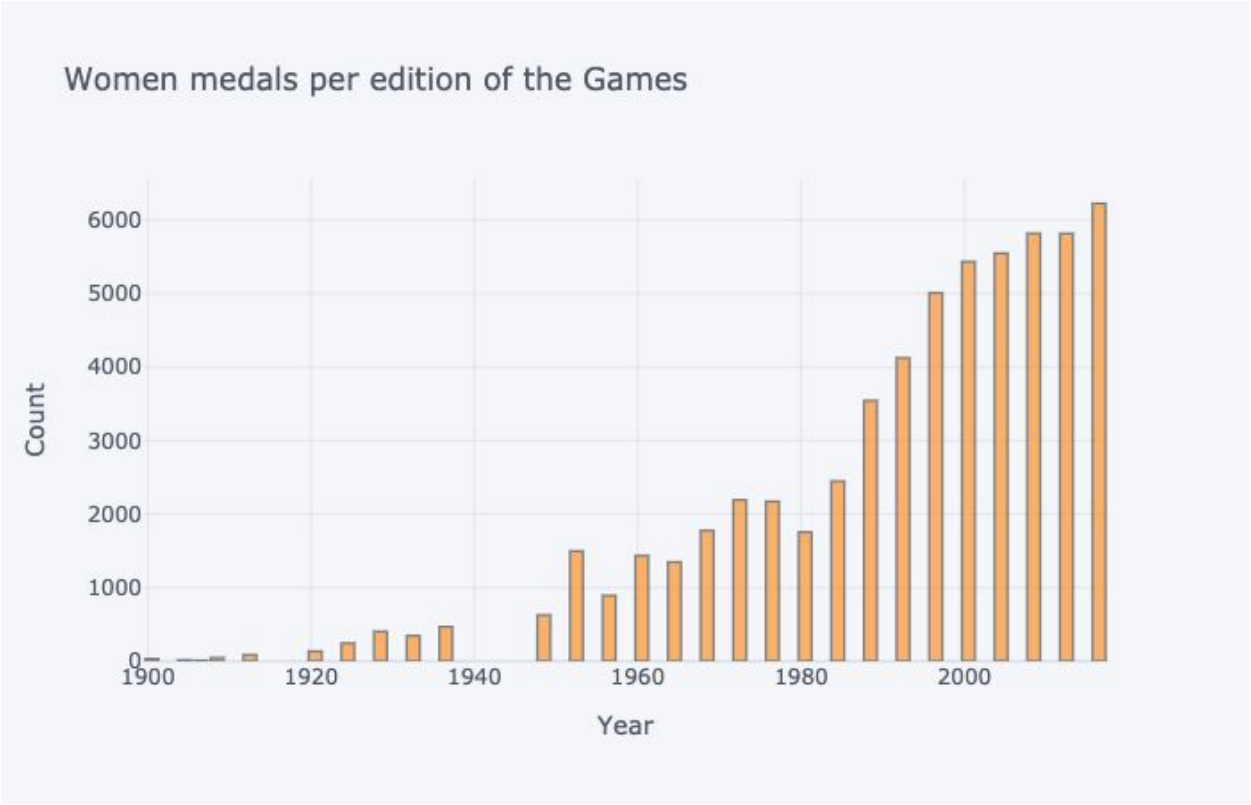
14.Displaying Winners of Gold Medals by Gender



Now let's continue our work focusing on women athletes for a while.

15.Displaying the count of medals women achieved over years.

```
#Women in Athletics
WomenInOlympics = {
    "Sex":"F",
    "Season" : "Summer",
    "Year" : {"$exists" : True}
}
rm = list(db.collection.find(query))
Year=[collection["Year"] for collection in rm]
plt.clf()
plt.figure(figsize=(20, 10))
ax=s.countplot(Year)
plt.title('Women medals per edition of the Games')
plt.xlabel('Year')
plt.ylabel('Count')
plt.show()
```



Inferences-

1. There were very few women athletes in the 1900's.

2. The women population in the Olympics has been rising from 1980.

So let's count the exact number of Women athletes who won medals in the 1900's.

```
db.collection.count_documents({"Sex":"F","Season":"Summer","Year":1900})
```

```
33
```

So there are 33 records of women who won medals in 1900.

16.Displaying Gold Medals per Country

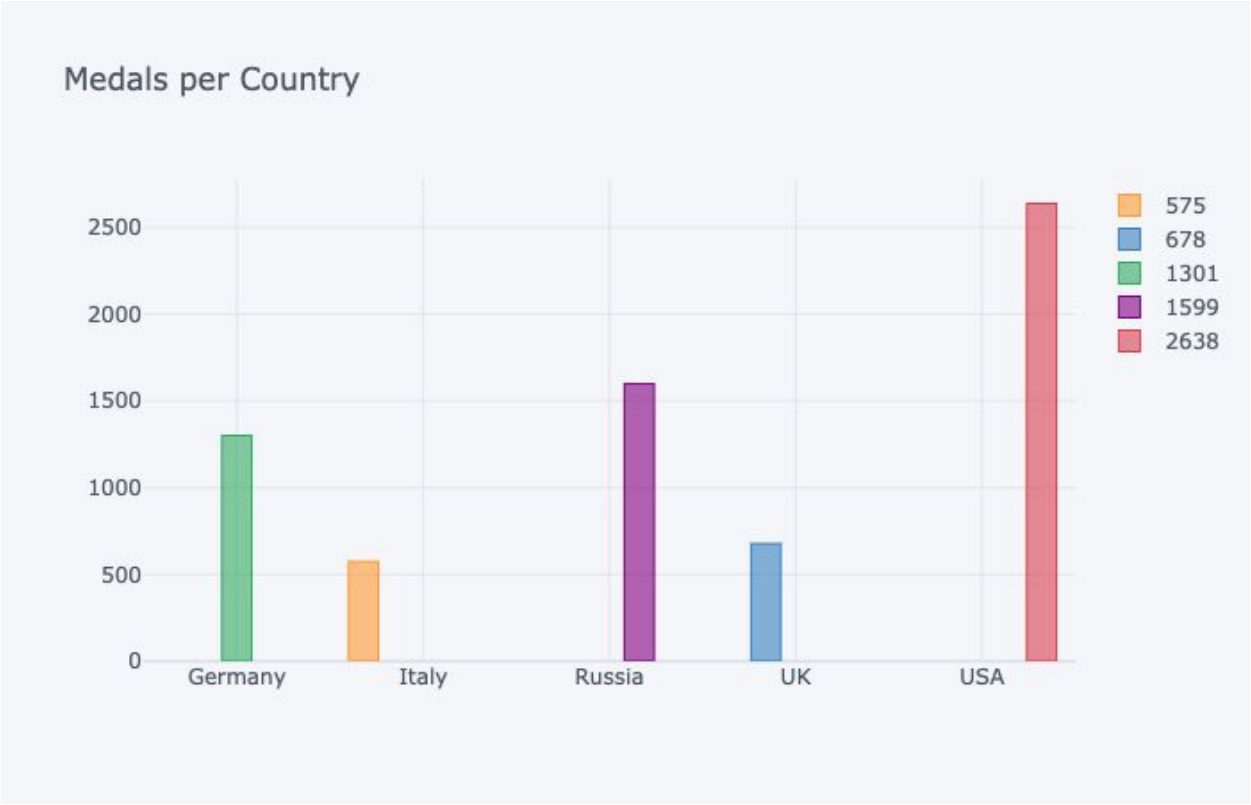
```
from bson.son import SON
import pprint


MedalsperCountry=db.collection.aggregate([
   {
      "$match":{
         "Medal":"Gold"
      }
   },
   {
      "$group":{
         "_id": "$region",

        "Medal":{
            "$sum" : 1}
      }
   },
    { "$sort" : { "Medal" : -1 } }
])
i=0
v=[]
for y in MedalsperCountry:
 if(i==5):
  break
 v.append(y)
 i=i+1
print(v)
```

We are displaying the top 5 countries with a great winning history of gold medals.

	_id	Medal
0	USA	2638
1	Russia	1599
2	Germany	1301
3	UK	678
4	Italy	575



As we can see, the USA performed exceptionally well and stood first.

Lets go deep into their athletics disciplines.

17.Displaying the Medals of the USA per Event.

```
goldMedalsUSA=pd.DataFrame(db.collection.aggregate([
   {
      "$match":{
         "Medal":"Gold",
          "region":"USA"
      }
   },
   {
      "$group":{
         "_id": "$Event",

        "Medal":{
            "$sum" : 1}
      }
   },
    { "$sort" : { "Medal" : -1 } }
]))
goldMedalsUSA
```

	_id	Medal
0	Basketball Men's Basketball	186
1	Swimming Men's 4 x 200 metres Freestyle Relay	111
2	Rowing Men's Coxed Eights	108
3	Swimming Men's 4 x 100 metres Medley Relay	108
4	Basketball Women's Basketball	95
...	...	...
310	Short Track Speed Skating Men's 1,500 metres	1
311	Art Competitions Mixed Painting, Drawings And ...	1
312	Canoeing Men's Canadian Singles, 10,000 metres	1
313	Art Competitions Mixed Sculpturing	1
314	Shooting Men's Small-Bore Rifle, Any Position,...	1

315 rows × 2 columns

And from the results we can see,Basketball is the leading discipline.

Now let’s count the medals of each member of the team instead of counting per team.

18.Displaying Men’s Basketball results.

The medals are not grouped by Team but we were counting the gold medals of each member of the team!

Let's proceed grouping by year the athletes - the idea is to create a new document to make a pre-filter using only the first record for each member of the team.

```
BasketballMen=pd.DataFrame(list(db.collection.aggregate([
   {
      "$match":{
         "Medal":"Gold",
          "region":"USA",
          "Sex":"M",
          "Sport":"Basketball"
      }
   },
    {"$group": {
        "_id":"$Year",
       "ID" : { "$first": '$ID' }, "Name": { "$first": '$Name' }, "Sex" : { "$first": '$Sex' },"Age" : { "$first": '$
    }},
    { "$sort" : { "Year" : 1 } }
])))
BasketballMen
```

	_id	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	Sport	Event	Medal	region	notes
0	1936	7396	Samuel J. "Sam" Balter, Jr.	M	26.0	178.0	68.0	United States	USA	1936 Summer	1936	Summer	Berlin	Basketball	Basketball Men's Basketball	Gold	USA	None
1	1948	7881	Clifford Eugene "Cliff" Barker	M	27.0	188.0	84.0	United States	USA	1948 Summer	1948	Summer	London	Basketball	Basketball Men's Basketball	Gold	USA	None
2	1952	13302	Ronald Yngve "Ron" Bontemps	M	25.0	188.0	79.0	United States	USA	1952 Summer	1952	Summer	Helsinki	Basketball	Basketball Men's Basketball	Gold	USA	None
3	1956	14153	Richard James "Dick" Boushka	M	22.0	195.0	95.0	United States	USA	1956 Summer	1956	Summer	Melbourne	Basketball	Basketball Men's Basketball	Gold	USA	None
4	1960	5212	Jay Joseph Hoyland Arnette	M	21.0	188.0	79.0	United States	USA	1960 Summer	1960	Summer	Roma	Basketball	Basketball Men's Basketball	Gold	USA	None
5	1964	7958	Velvet James "Jim" Barnes	M	23.0	201.0	109.0	United States	USA	1964 Summer	1964	Summer	Tokyo	Basketball	Basketball Men's Basketball	Gold	USA	None
6	1968	8104	Michael Thomas "Mike"	M	25.0	188.0	73.0	United States	USA	1968 Summer	1968	Summer	Mexico City	Basketball	Basketball Men's	Gold	USA	None

So now let's count the number of records.

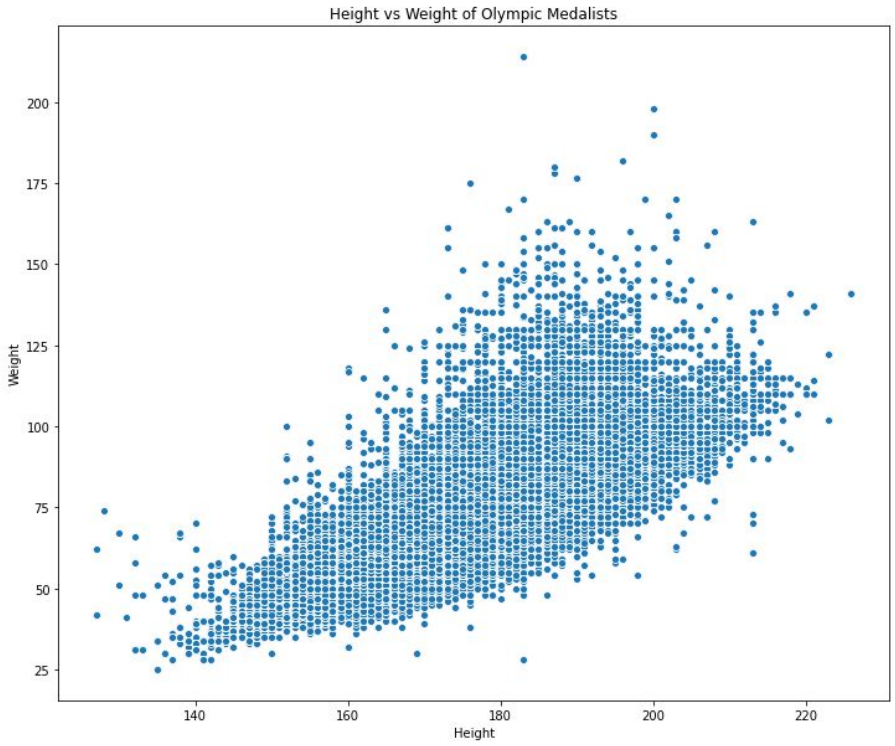
```
BasketballMen['ID'].count()
```

```
15
```

19.Displaying the scatter plot of Height vs Weight of Athletes

```
import matplotlib.pyplot as plt
import seaborn as s
query={
    "Weight" : {"$exists" : True},
    "Height" : {"$exists" : True}
}
rm = list(db.collection.find(query))
Height=[collection["Height"] for collection in rm]
Weight=[collection["Weight"] for collection in rm]

plt.clf()
plt.figure(figsize=(12, 10))
ax=s.scatterplot(Height,Weight,alpha=1)
plt.title('Height vs Weight of Olympic Medalists')
plt.xlabel('Height')
plt.ylabel('Weight')
plt.show()
```



Inference-

1.It seems to be so linear(more the height,more the weight).

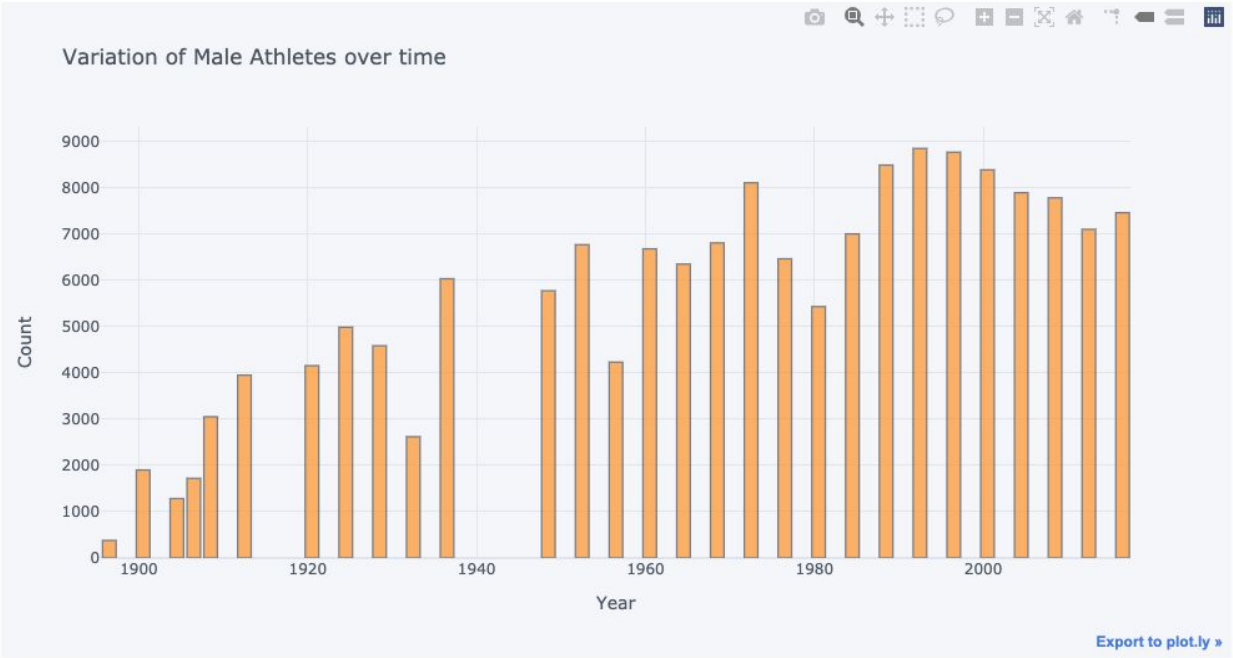
So let us have a check on players who weigh more than 160kg.

```
x=db.collection.find({"Medal" : "Gold","Weight":{"$gt":160}})
for y in x:
 print(y)
```

```
{'_id': ObjectId('5e9727ffa56a9dcc9600ac35'), 'ID': 20144, 'Name': 'Andrey Ivanovich Chemerkin', 'Sex': 'M', 'Age': 2
4.0, 'Height': 183.0, 'Weight': 170.0, 'Team': 'Russia', 'NOC': 'RUS', 'Games': '1996 Summer', 'Year': 1996, 'Seaso
n': 'Summer', 'City': 'Atlanta', 'Sport': 'Weightlifting', 'Event': "Weightlifting Men's Super-Heavyweight", 'Medal':
'Gold', 'region': 'Russia', 'notes': None}
{'_id': ObjectId('5e972801a56a9dcc96042c9b'), 'ID': 134407, 'Name': 'Leonid Ivanovych Zhabotynskiy', 'Sex': 'M', 'Ag
e': 26.0, 'Height': 189.0, 'Weight': 163.0, 'Team': 'Soviet Union', 'NOC': 'URS', 'Games': '1964 Summer', 'Year': 196
4, 'Season': 'Summer', 'City': 'Tokyo', 'Sport': 'Weightlifting', 'Event': "Weightlifting Men's Heavyweight", 'Meda
l': 'Gold', 'region': 'Russia', 'notes': None}
{'_id': ObjectId('5e972801a56a9dcc96042c9c'), 'ID': 134407, 'Name': 'Leonid Ivanovych Zhabotynskiy', 'Sex': 'M', 'Ag
e': 30.0, 'Height': 189.0, 'Weight': 163.0, 'Team': 'Soviet Union', 'NOC': 'URS', 'Games': '1968 Summer', 'Year': 196
8, 'Season': 'Summer', 'City': 'Mexico City', 'Sport': 'Weightlifting', 'Event': "Weightlifting Men's Heavyweight",
'Medal': 'Gold', 'region': 'Russia', 'notes': None}
```

20.Displaying the variation of Men over time.

```
MenOverTime=pd.DataFrame(list(db.collection.find({"Sex" : "M","Season":"Summer"})))
MenOverTime['Year'].iplot(kind='hist', opacity=0.75, bargap = 0.20,title="Variation of Male Athletes over time", xTitle
```

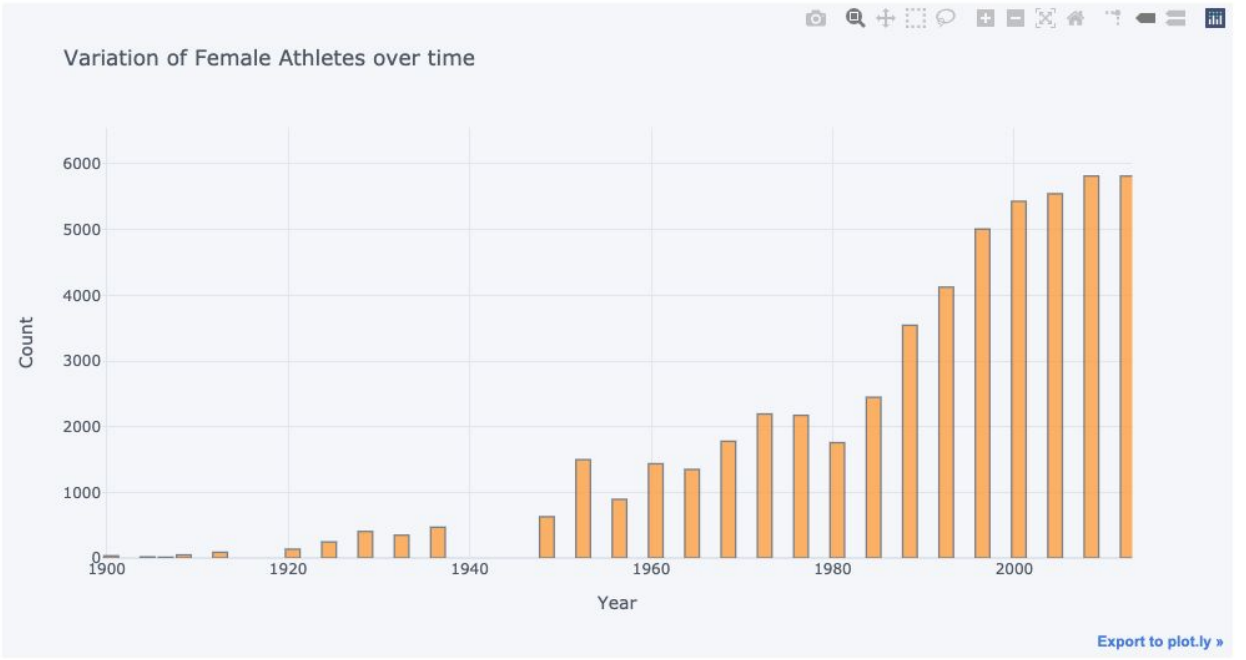


Inference-

1. There is a decrease in the male players after 1900 and then the growth has started in the recent years.

21.Displaying the variation of Women over time.

```
WomenOverTime=pd.DataFrame(list(db.collection.find({"Sex" : "F","Season":"Summer"})))
WomenOverTime['Year'].iplot(kind='hist', opacity=0.75, bargap = 0.20,title="Variation of Female Athletes over time", x
```



Inference-

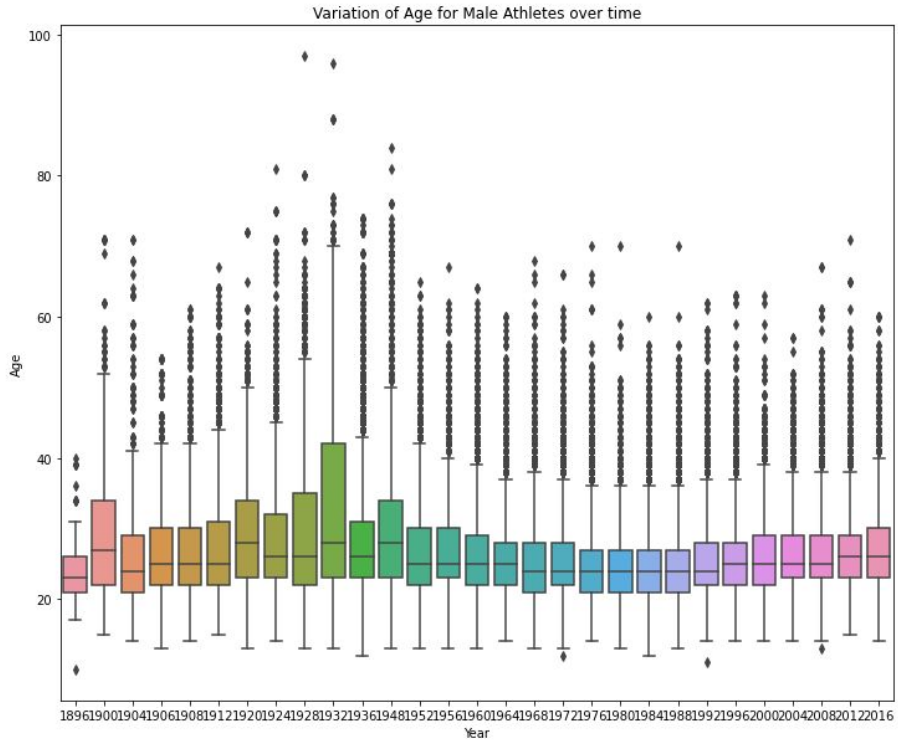
1. The growth of women players is constant.

2.There is an increase in population every year.

<u>22.Displaying the Variation of Age for Male Athletes over time.</u>

```
query={
    "Sex" : "M",
    "Season":"Summer",
    "Year" : {"$exists" : True},
    "Age" : {"$exists" : True}
}
rm = list(db.collection.find(query))
Year=[collection["Year"] for collection in rm]
Age=[collection["Age"] for collection in rm]
plt.clf()
plt.figure(figsize=(12, 10))
ax=s.boxplot(Year,Age)
plt.title('Variation of Age for Male Athletes over time')
plt.xlabel('Year')
plt.ylabel('Age')
plt.show()
```

```
<Figure size 432x288 with 0 Axes>
```



From this graph,we can notice the age of some athletes who won gold medals is greater than 80 between the years 1924 and 1948.

Let’s have more discussion about these senior medalists.

```
x=db.collection.find({"Sex" : "M","Season":"Summer","Age":{"$gt":80}})
for y in x:
 print(y)
```

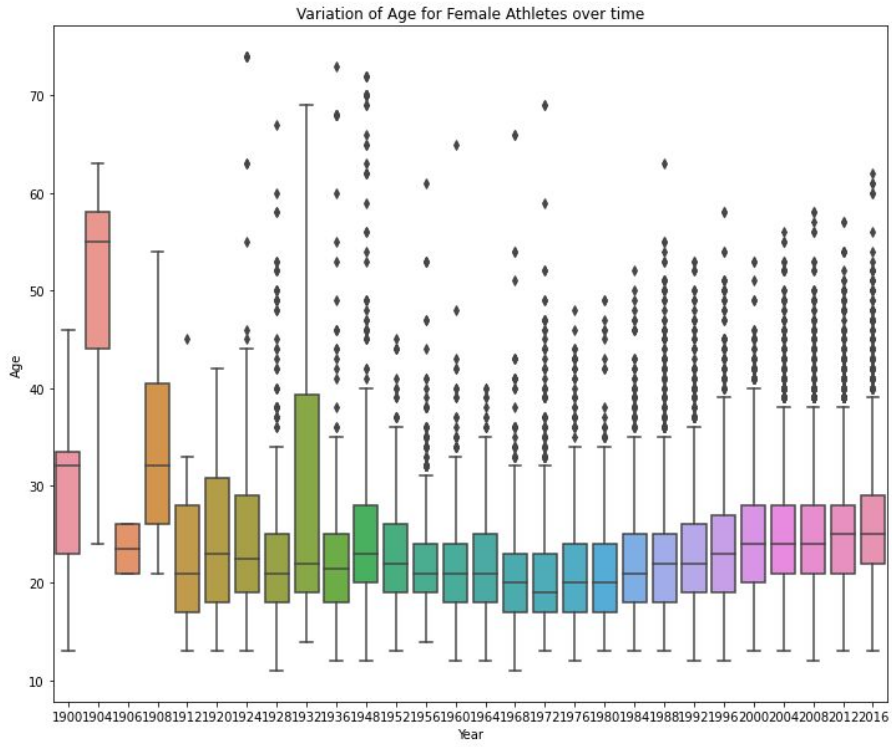
```
{'_id': ObjectId('5e9884a2122d8c72d5e997b7'), 'ID': 5146, 'Name': 'George Denholm Armour', 'Sex': 'M', 'Age': 84.0,
'Height': None, 'Weight': None, 'Team': 'Great Britain', 'NOC': 'GBR', 'Games': '1948 Summer', 'Year': 1948, 'Seaso
n': 'Summer', 'City': 'London', 'Sport': 'Art Competitions', 'Event': 'Art Competitions Mixed Painting, Unknown Even
t', 'Medal': None, 'region': 'UK', 'notes': None}
{'_id': ObjectId('5e9884a7122d8c72d5ea60d9'), 'ID': 31173, 'Name': 'Thomas Cowperthwait Eakins', 'Sex': 'M', 'Age': 8
8.0, 'Height': None, 'Weight': None, 'Team': 'United States', 'NOC': 'USA', 'Games': '1932 Summer', 'Year': 1932, 'Se
ason': 'Summer', 'City': 'Los Angeles', 'Sport': 'Art Competitions', 'Event': 'Art Competitions Mixed Painting, Unkno
wn Event', 'Medal': None, 'region': 'USA', 'notes': None}
{'_id': ObjectId('5e9884a7122d8c72d5ea60da'), 'ID': 31173, 'Name': 'Thomas Cowperthwait Eakins', 'Sex': 'M', 'Age': 8
8.0, 'Height': None, 'Weight': None, 'Team': 'United States', 'NOC': 'USA', 'Games': '1932 Summer', 'Year': 1932, 'Se
ason': 'Summer', 'City': 'Los Angeles', 'Sport': 'Art Competitions', 'Event': 'Art Competitions Mixed Painting, Unkno
wn Event', 'Medal': None, 'region': 'USA', 'notes': None}
{'_id': ObjectId('5e9884a7122d8c72d5ea60db'), 'ID': 31173, 'Name': 'Thomas Cowperthwait Eakins', 'Sex': 'M', 'Age': 8
8.0, 'Height': None, 'Weight': None, 'Team': 'United States', 'NOC': 'USA', 'Games': '1932 Summer', 'Year': 1932, 'Se
ason': 'Summer', 'City': 'Los Angeles', 'Sport': 'Art Competitions', 'Event': 'Art Competitions Mixed Painting, Unkno
wn Event', 'Medal': None, 'region': 'USA', 'notes': None}
{'_id': ObjectId('5e9884a7122d8c72d5eaf262'), 'ID': 49663, 'Name': 'Winslow Homer', 'Sex': 'M', 'Age': 96.0, 'Heigh
t': None, 'Weight': None, 'Team': 'United States', 'NOC': 'USA', 'Games': '1932 Summer', 'Year': 1932, 'Season': 'Sum
mer', 'City': 'Los Angeles', 'Sport': 'Art Competitions', 'Event': 'Art Competitions Mixed Painting, Unknown Event',
'Medal': None, 'region': 'USA', 'notes': None}
{'_id': ObjectId('5e9884a8122d8c72d5ebd003'), 'ID': 77710, 'Name': 'Robert Tait McKenzie', 'Sex': 'M', 'Age': 81.0,
'Height': None, 'Weight': None, 'Team': 'Canada', 'NOC': 'CAN', 'Games': '1948 Summer', 'Year': 1948, 'Season': 'Summ
er', 'City': 'London', 'Sport': 'Art Competitions', 'Event': 'Art Competitions Mixed Sculpturing, Unknown Event', 'Me
dal': None, 'region': 'Canada', 'notes': None}
{'_id': ObjectId('5e9884aa122d8c72d5ed108c'), 'ID': 118789, 'Name': 'Louis Tauzin', 'Sex': 'M', 'Age': 81.0, 'Heigh
t': None, 'Weight': None, 'Team': 'France', 'NOC': 'FRA', 'Games': '1924 Summer', 'Year': 1924, 'Season': 'Summer',
'City': 'Paris', 'Sport': 'Art Competitions', 'Event': 'Art Competitions Mixed Sculpturing', 'Medal': None, 'region':
'France', 'notes': None}
{'_id': ObjectId('5e9884aa122d8c72d5ed5f3a'), 'ID': 128719, 'Name': 'John Quincy Adams Ward', 'Sex': 'M', 'Age': 97.
0, 'Height': None, 'Weight': None, 'Team': 'United States', 'NOC': 'USA', 'Games': '1928 Summer', 'Year': 1928, 'Seas
on': 'Summer', 'City': 'Amsterdam', 'Sport': 'Art Competitions', 'Event': 'Art Competitions Mixed Sculpturing, Statue
s', 'Medal': None, 'region': 'USA', 'notes': None}
```

Art competitions are part of the modern Olympic Games during its early years, from 1912 to 1948. The competitions were part of the original intention of the Olympic Movement's founder, Pierre de Frédy, Baron de Coubertin. Medals were awarded for works of art inspired by sport, divided into five categories: architecture, literature, music, painting, and sculpture.

23.Displaying the Variation of Age for Female Athletes over time.

```
query={
    "Sex" : "F",
    "Season":"Summer",
    "Year" : {"$exists" : True},
    "Age" : {"$exists" : True}
}
rm = list(db.collection.find(query))
Year=[collection["Year"] for collection in rm]
Age=[collection["Age"] for collection in rm]
plt.clf()
plt.figure(figsize=(12, 10))
ax=s.boxplot(Year,Age)
plt.title('Variation of Age for Female Athletes over time')
plt.xlabel('Year')
plt.ylabel('Age')
plt.show()
```

```
<Figure size 432x288 with 0 Axes>
```



Inference-

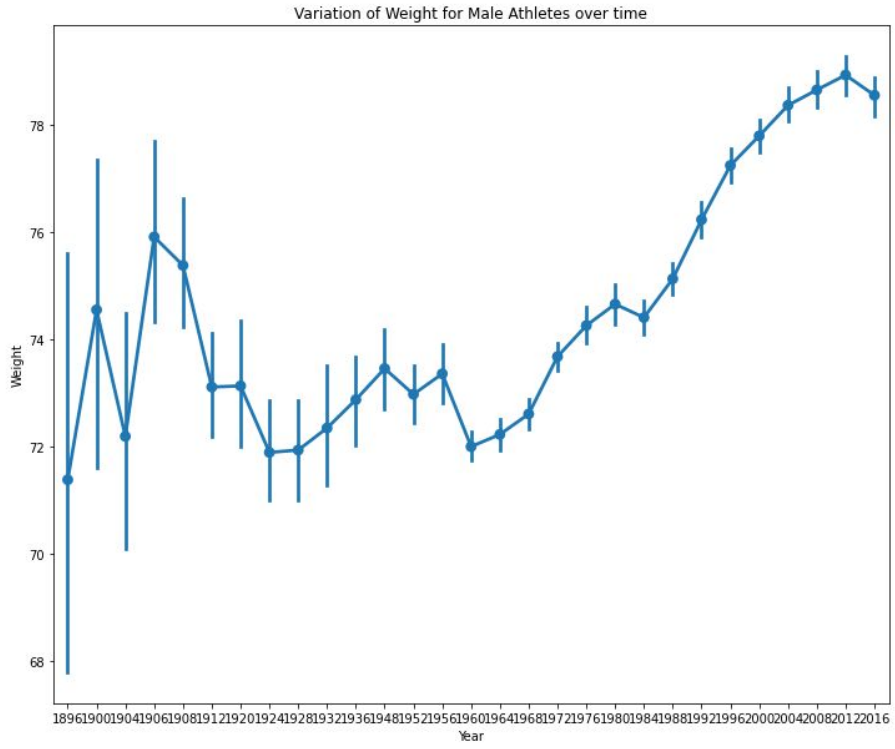
1.Normally, the age distribution starts with a lower minimum and a lower maximum.

2.In 1904 the age distribution is strongly different from the other Olympics.

24.Displaying the Variation of Weight for Male Athletes over time.

```
query={
    "Sex" : "M",
    "Season":"Summer",
    "Year" : {"$exists" : True},
    "Weight" : {"$exists" : True}
}
rm = list(db.collection.find(query))
Year=[collection["Year"] for collection in rm]
Weight=[collection["Weight"] for collection in rm]
plt.clf()
plt.figure(figsize=(12, 10))
ax=s.pointplot(Year,Weight)
plt.title('Variation of Weight for Male Athletes over time')
plt.xlabel('Year')
plt.ylabel('Weight')
plt.show()
```

```
<Figure size 432x288 with 0 Axes>
```



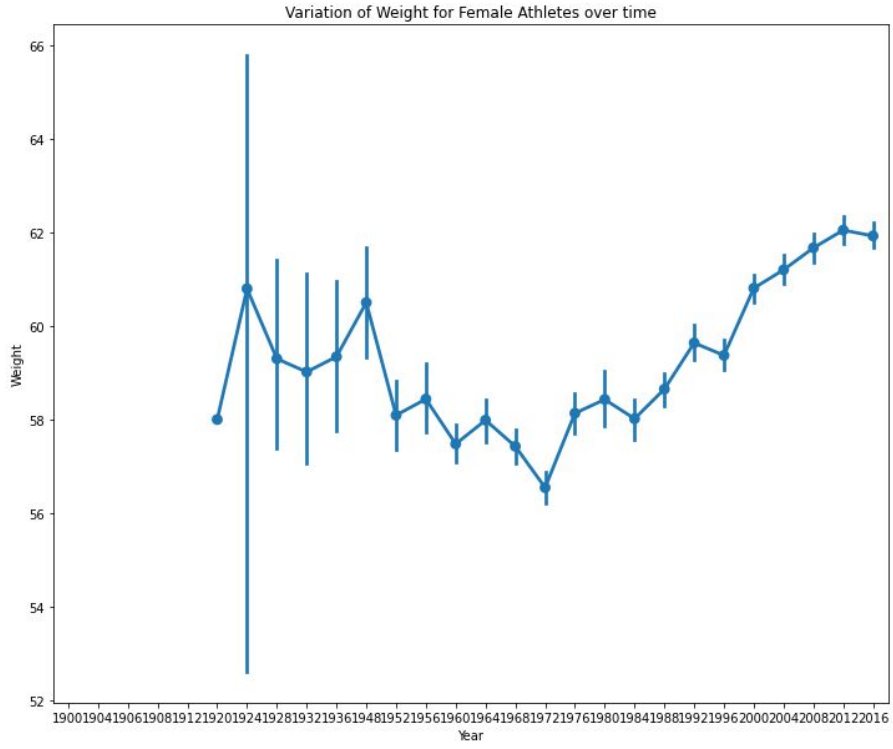
Inference-

1.As we see, In the starting years there are athletes weighing from 60 to 80 and as it goes on most of the athletes weigh above 75 kilograms.

25.Displaying the Variation of Weight for Female Athletes over time.

```
query={
    "Sex" : "F",
    "Season":"Summer",
    "Year" : {"$exists" : True},
    "Weight" : {"$exists" : True}
}
rm = list(db.collection.find(query))
Year=[collection["Year"] for collection in rm]
Weight=[collection["Weight"] for collection in rm]
plt.clf()
plt.figure(figsize=(12, 10))
ax=s.pointplot(Year,Weight)
plt.title('Variation of Weight for Female Athletes over time')
plt.xlabel('Year')
plt.ylabel('Weight')
plt.show()
```

```
<Figure size 432x288 with 0 Axes>
```



Inference-

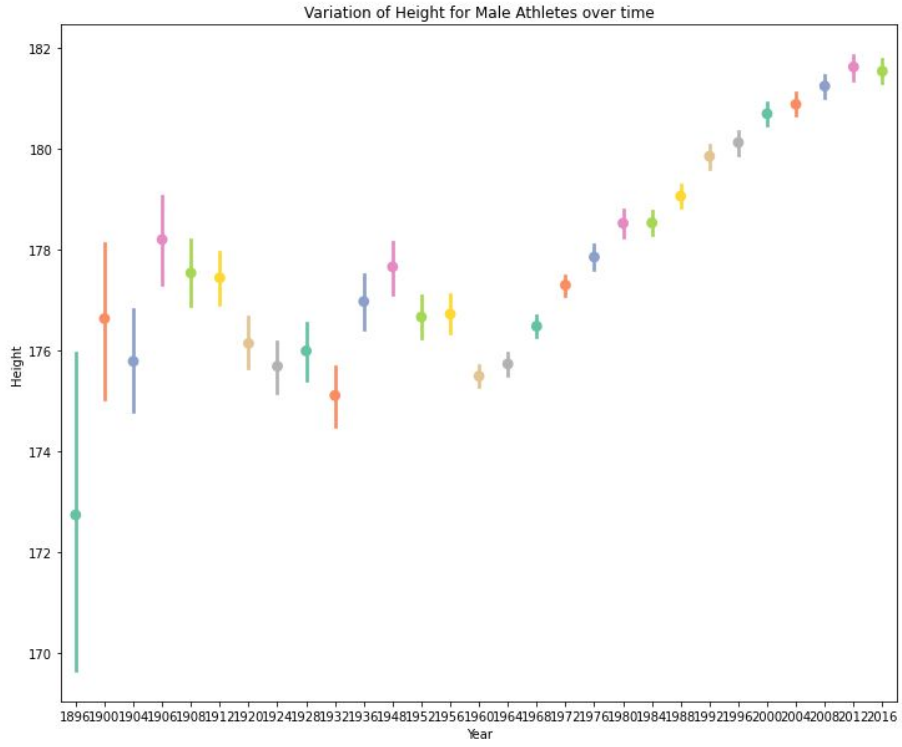
1.The data of women before 1924 is missing.

2.Most of the women athletes weigh between 50 to 60.

26.Displaying the Variation of Height for Male Athletes over time.

```
query={
    "Sex" : "M",
    "Season":"Summer",
    "Year" : {"$exists" : True},
    "Height" : {"$exists" : True}
}
rm = list(db.collection.find(query))
Year=[collection["Year"] for collection in rm]
Height=[collection["Height"] for collection in rm]
plt.clf()
plt.figure(figsize=(12, 10))
ax=s.pointplot(Year,Height,palette='Set2')
plt.title('Variation of Height for Male Athletes over time')
plt.xlabel('Year')
plt.ylabel('Height')
plt.show()
```

```
<Figure size 432x288 with 0 Axes>
```



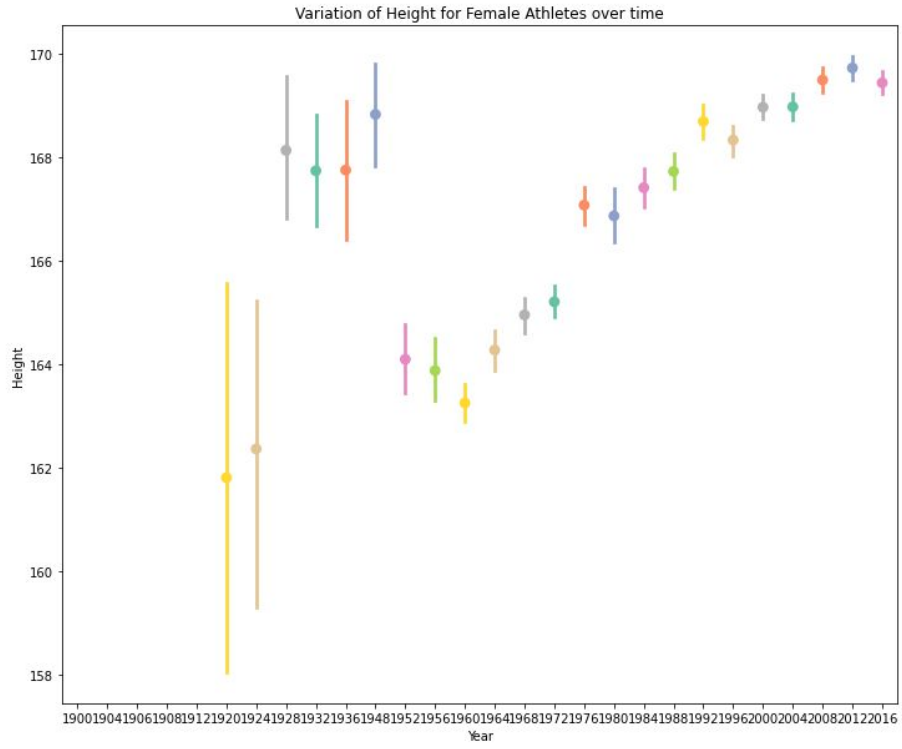
Inference-

1.Height is increasing over time but there is a decline in graph between 2012and 2016.

27.Displaying the Variation of Height for Female Athletes over time.

```
query={
    "Sex" : "F",
    "Season":"Summer",
    "Year" : {"$exists" : True},
    "Height" : {"$exists" : True}
}
rm = list(db.collection.find(query))
Year=[collection["Year"] for collection in rm]
Height=[collection["Height"] for collection in rm]
plt.clf()
plt.figure(figsize=(12, 10))
ax=s.pointplot(Year,Height,palette='Set2')
plt.title('Variation of Height for Female Athletes over time')
plt.xlabel('Year')
plt.ylabel('Height')
plt.show()
```

```
<Figure size 432x288 with 0 Axes>
```



Inference-

1.Height is increasing over time but there is a decline in graph between 2012and 2016.

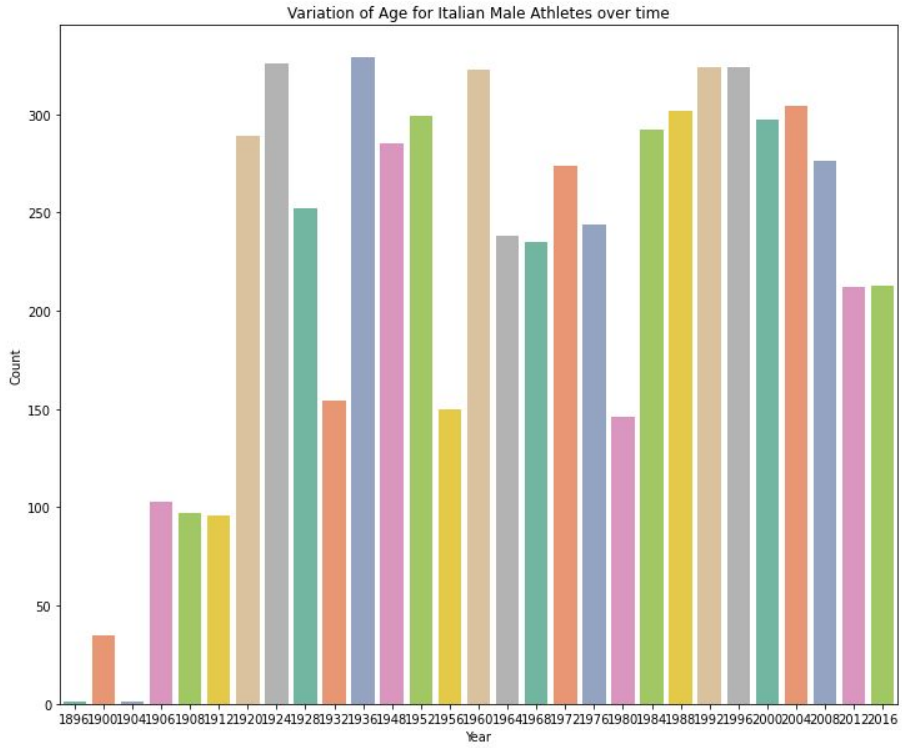
2.From 1928 to 1948,the graph of height was at its peak.

3.Data was missing from 1900 to 1912.

28.Displaying the Variation of Age for Italian Male Athletes over time.

```
query={
    "Sex" : "M",
    "Season":"Summer",
    "region":"Italy",
    "Year" : {"$exists" : True}
}
rm = list(db.collection.find(query))
Year=[collection["Year"] for collection in rm]
plt.clf()
plt.figure(figsize=(12, 10))
ax=s.countplot(Year,palette='Set2')
plt.title('Variation of Age for Italian Male Athletes over time')
plt.xlabel('Year')
plt.ylabel('Count')
plt.show()
```

```
<Figure size 432x288 with 0 Axes>
```



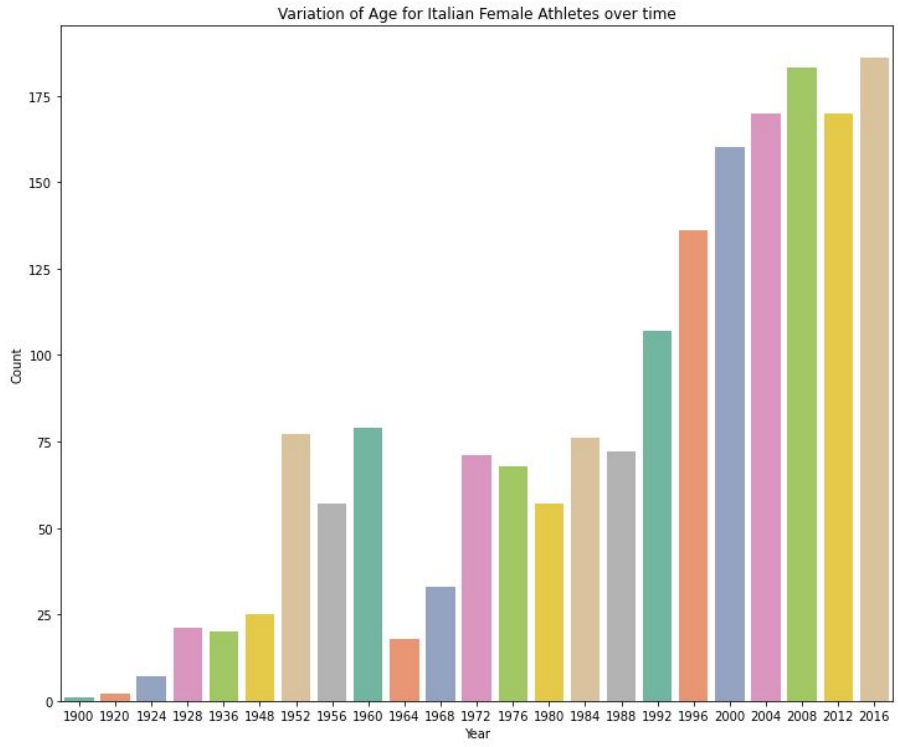
Inference-

1. The men participation is decreasing starting from the 2008 games.

29.Displaying the Variation of Age for Italian Female Athletes over time.

```
query={
    "Sex" : "F",
    "Season":"Summer",
    "region":"Italy",
    "Year" : {"$exists" : True}
}
rm = list(db.collection.find(query))
Year=[collection["Year"] for collection in rm]
plt.clf()
plt.figure(figsize=(12, 10))
ax=s.countplot(Year,palette='Set2')
plt.title('Variation of Age for Italian Female Athletes over time')
plt.xlabel('Year')
plt.ylabel('Count')
plt.show()
```

```
<Figure size 432x288 with 0 Axes>
```



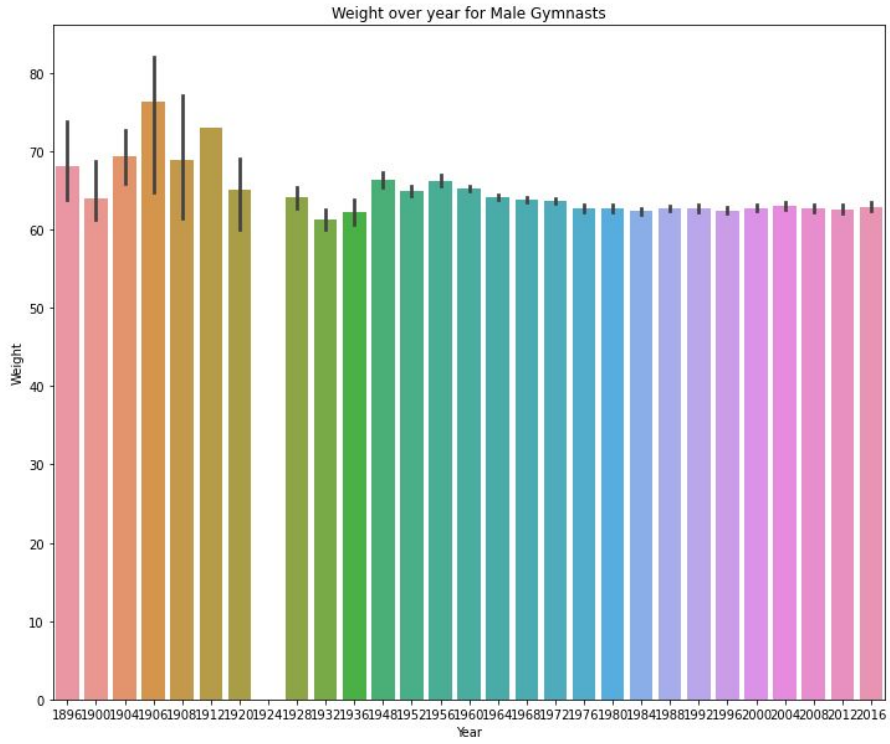
Inference-

1.The participation of Italian women is increasing over the time period.

30.Displaying Weight over year for Male Gymnasts.

```
query={
    "Sex" : "M",
    "Season":"Summer",
    "Sport":"Gymnastics",
    "Year" : {"$exists" : True},
    "Weight" : {"$exists" : True}
}
rm = list(db.collection.find(query))
Year=[collection["Year"] for collection in rm]
Weight=[collection["Weight"] for collection in rm]
plt.clf()
plt.figure(figsize=(12, 10))
s.barplot(Year,Weight)
plt.title('Weight over year for Male Gymnasts')
plt.xlabel('Year')
plt.ylabel('Weight')
plt.show()
```

```
<Figure size 432x288 with 0 Axes>
```



Inference-

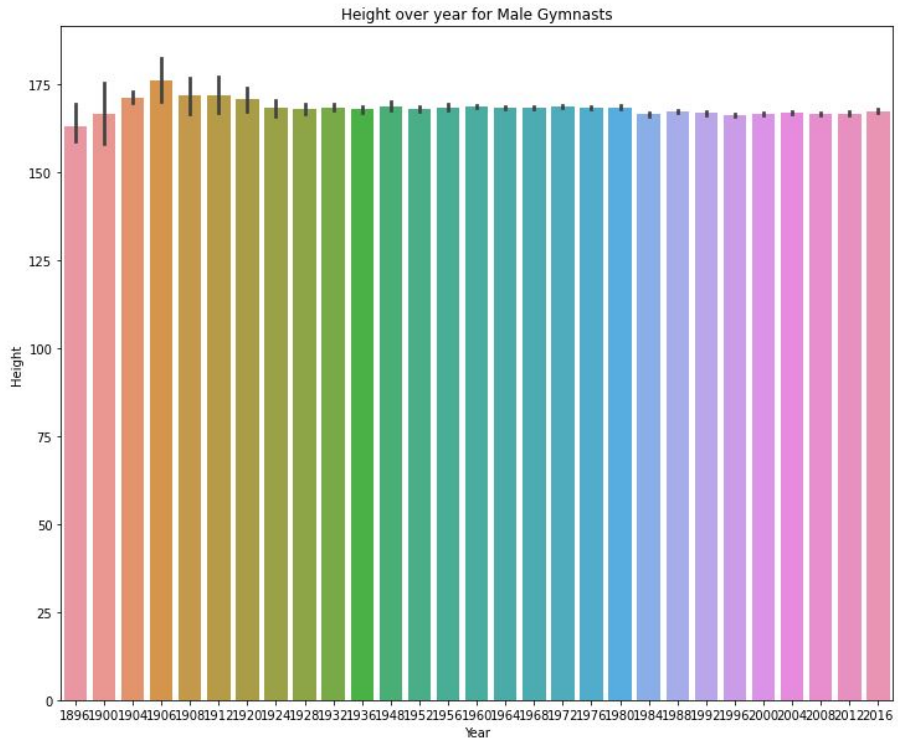
1.The weight for men has been more or less stable since 1964.

2.The data is missing for the year 1924.

31.Displaying Height over year for Male Gymnasts.

```
query={
    "Sex" : "M",
    "Season":"Summer",
    "Sport":"Gymnastics",
    "Year" : {"$exists" : True},
    "Height" : {"$exists" : True}
}
rm = list(db.collection.find(query))
Year=[collection["Year"] for collection in rm]
Height=[collection["Height"] for collection in rm]
plt.clf()
plt.figure(figsize=(12, 10))
s.barplot(Year,Height)
plt.title('Height over year for Male Gymnasts')
plt.xlabel('Year')
plt.ylabel('Height')
plt.show()
```

```
<Figure size 432x288 with 0 Axes>
```



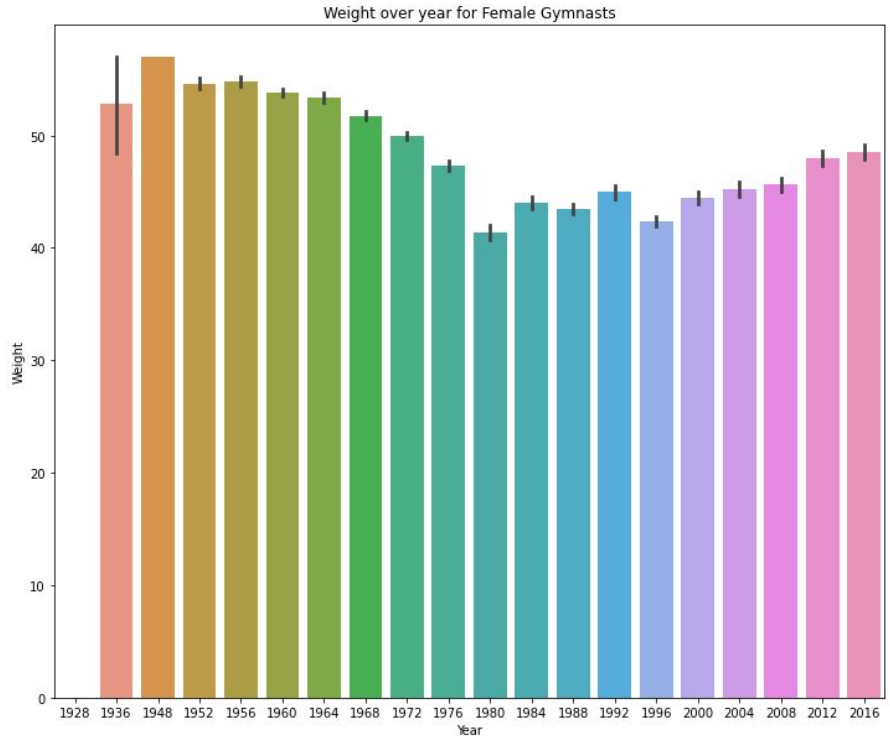
Inference-

1.The graph of height looks more stable over the years.

32.Displaying Weight over year for Female Gymnasts.

```
query={
    "Sex" : "F",
    "Season":"Summer",
    "Sport":"Gymnastics",
    "Year" : {"$exists" : True},
    "Weight" : {"$exists" : True}
}
rm = list(db.collection.find(query))
Year=[collection["Year"] for collection in rm]
Weight=[collection["Weight"] for collection in rm]
plt.clf()
plt.figure(figsize=(12, 10))
s.barplot(Year,Weight)
plt.title('Weight over year for Female Gymnasts')
plt.xlabel('Year')
plt.ylabel('Weight')
plt.show()
```

```
<Figure size 432x288 with 0 Axes>
```



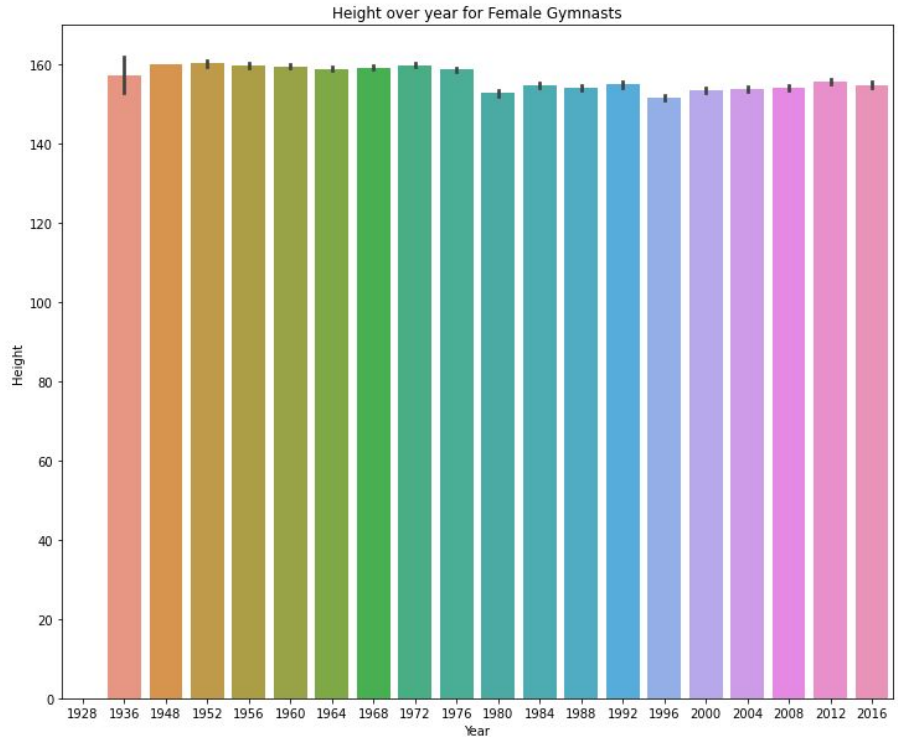
Inference-

1.The weight for female Gymnasts has gone down for 60 to 50 kilograms on average.

33.Displaying Height over year for Female Gymnasts.

```
query={
    "Sex" : "F",
    "Season":"Summer",
    "Sport":"Gymnastics",
    "Year" : {"$exists" : True},
    "Height" : {"$exists" : True}
}
rm = list(db.collection.find(query))
Year=[collection["Year"] for collection in rm]
Height=[collection["Height"] for collection in rm]
plt.clf()
plt.figure(figsize=(12, 10))
s.barplot(Year,Height)
plt.title('Height over year for Female Gymnasts')
plt.xlabel('Year')
plt.ylabel('Height')
plt.show()
```

```
<Figure size 432x288 with 0 Axes>
```



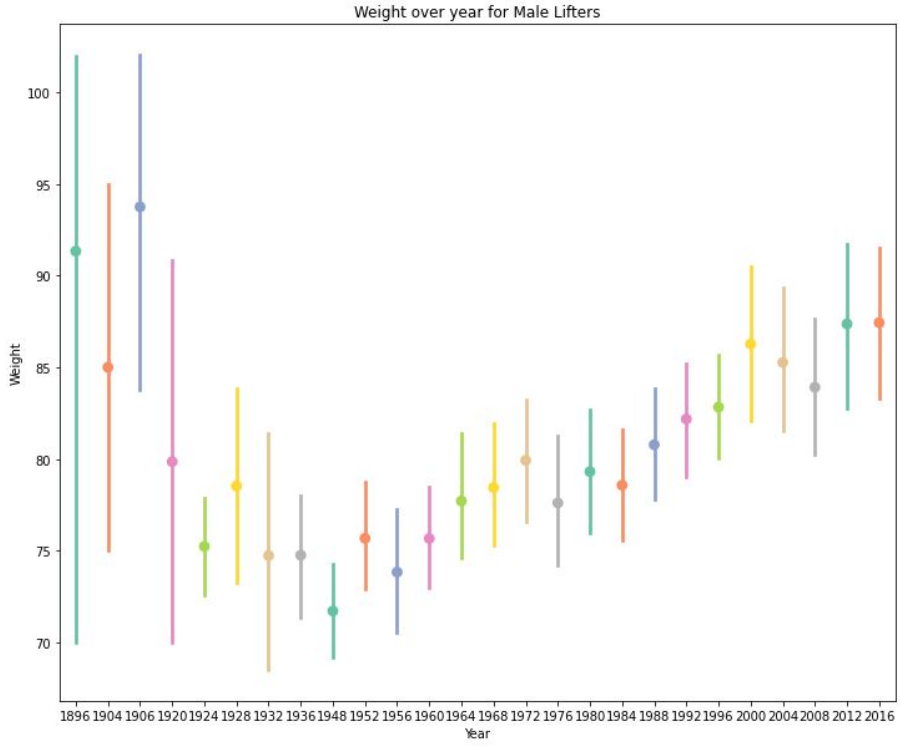
Inference-

1.The graph of height looks more stable over the years.

34.Displaying Weight over year for Male Lifters.

```
query={
    "Sex" : "M",
    "Season":"Summer",
    "Sport":"Weightlifting",
    "Year" : {"$exists" : True},
    "Weight" : {"$exists" : True}
}
rm = list(db.collection.find(query))
Year=[collection["Year"] for collection in rm]
Weight=[collection["Weight"] for collection in rm]
plt.clf()
plt.figure(figsize=(12, 10))
ax=s.pointplot(Year,Weight,palette='Set2')
plt.title('Weight over year for Male Lifters')
plt.xlabel('Year')
plt.ylabel('Weight')
plt.show()
```

```
<Figure size 432x288 with 0 Axes>
```



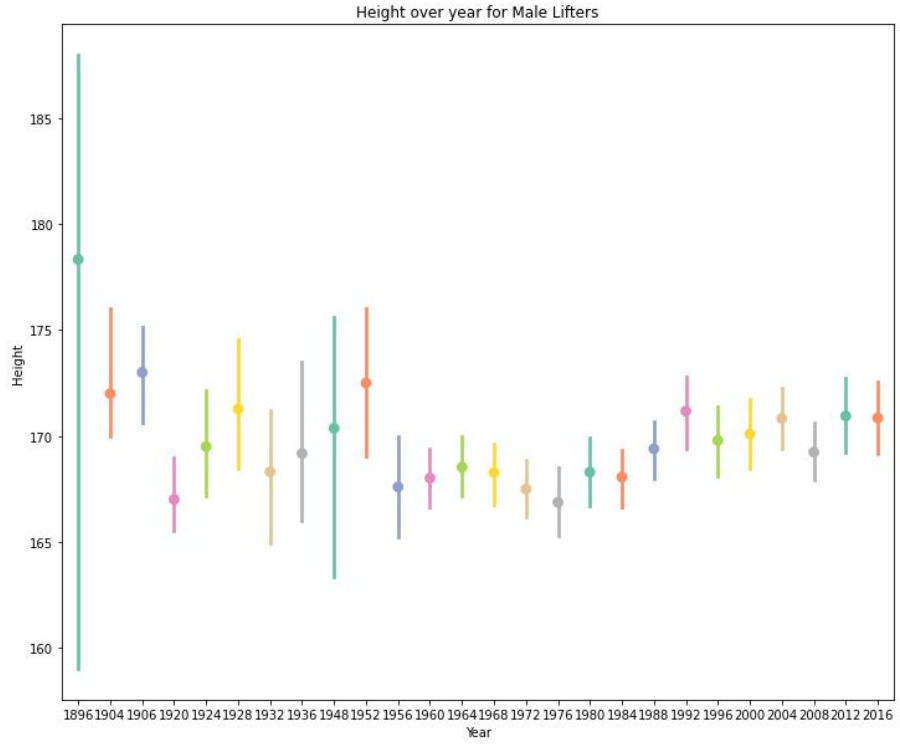
Inference-

1.Weight of Male lifters mostly range between 75 to 90.

35.Displaying Height over year for Male Lifters.

```
query={
    "Sex" : "M",
    "Season":"Summer",
    "Sport":"Weightlifting",
    "Year" : {"$exists" : True},
    "Height" : {"$exists" : True}
}
rm = list(db.collection.find(query))
Year=[collection["Year"] for collection in rm]
Height=[collection["Height"] for collection in rm]
plt.clf()
plt.figure(figsize=(12, 10))
ax=s.pointplot(Year,Height,palette='Set2')
plt.title('Height over year for Male Lifters')
plt.xlabel('Year')
plt.ylabel('Height')
plt.show()
```

```
<Figure size 432x288 with 0 Axes>
```



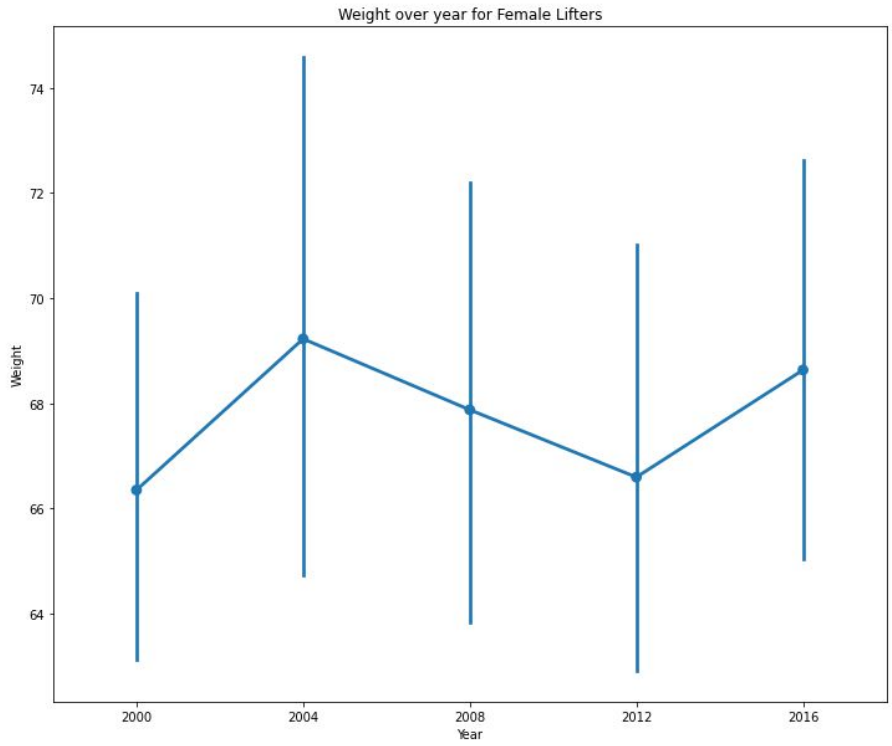
Inference-

1.The height of the lifters mostly lies between 165 to 175.

36.Displaying Weight over year for Female Lifters.

```
query={
    "Sex" : "F",
    "Season":"Summer",
    "Sport":"Weightlifting",
    "Year" : {"$exists" : True},
    "Weight" : {"$exists" : True}
}
rm = list(db.collection.find(query))
Year=[collection["Year"] for collection in rm]
Weight=[collection["Weight"] for collection in rm]
plt.clf()
plt.figure(figsize=(12, 10))
ax=s.pointplot(Year,Weight)
plt.title('Weight over year for Female Lifters')
plt.xlabel('Year')
plt.ylabel('Weight')
plt.show()
```

```
<Figure size 432x288 with 0 Axes>
```



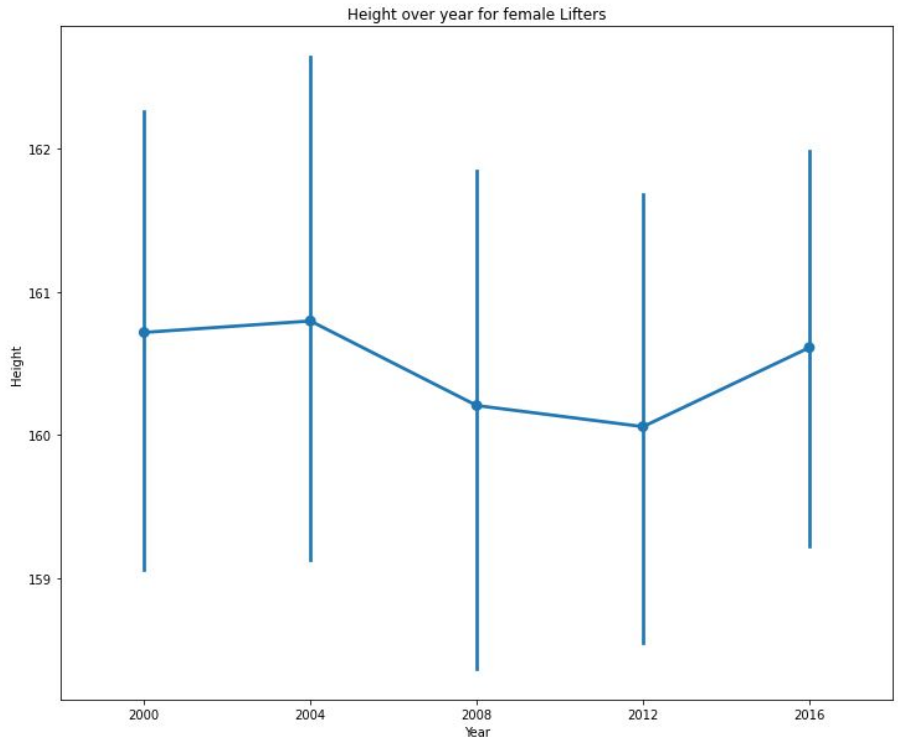
Inference-

1.The data is available only from 2000.

37.Displaying Height over year for Female Lifters.

```
query={
    "Sex" : "F",
    "Season":"Summer",
    "Sport":"Weightlifting",
    "Year" : {"$exists" : True},
    "Height" : {"$exists" : True}
}
rm = list(db.collection.find(query))
Year=[collection["Year"] for collection in rm]
Height=[collection["Height"] for collection in rm]
plt.clf()
plt.figure(figsize=(12, 10))
ax=s.pointplot(Year,Height)
plt.title('Height over year for female Lifters')
plt.xlabel('Year')
plt.ylabel('Height')
plt.show()
```

```
<Figure size 432x288 with 0 Axes>
```



Inference-

1.The data is available only from 2000.