# Olympics & Its Evolution - First Semester 2021

## Prepared by Sujil Kumar K.M (D00242726)

Cross Module Project with Programming, Statistics & Research

In [1]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import os,sys
from scipy import stats
import matplotlib.pyplot as plt
import statsmodels.api as sm
from bokeh.plotting import figure, output_file, show
import bokeh.io
from bokeh.models import ColumnDataSource
from bokeh.plotting import *
from statsmodels.formula.api import ols
from scipy.integrate import quad
%matplotlib inline

olympics = pd.read_csv('E:/Research in Data/Assignments/Proposal/Olympics/data_2/athlete_events.csv')

olympics.head()
```

Out[1]:

| | ID. | Na.me | Sex.rio | A.ge | Heigh.t | Weig.ht | Te am | N OC | Ga.mes | Ye.ar | Season | Ci ty | Spor t | Ev ent | M edal |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | A Dijiang | M | 24.0 | 180.0 | 80.0 | China | CHN | 1992 Summer | 1992 | Summer | Barcelona | Basketball | Basketball Men's Basketball | 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 |
| **2** | 3 | Gunnar Nielsen Aaby | M | 24.0 | NaN | NaN | Denmark | DEN | 1920 Summer | 1920 | Summer | Antwerpen | Football | Football Men's Football | 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 |
| **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 |

In [2]:

```
olympics.describe()
```

Out[2]:

| | ID. | A.ge | Heigh.t | Weig.ht | Ye.ar |
|---|---|---|---|---|---|
| **count** | 271116.000000 | 261642.000000 | 210945.000000 | 208241.000000 | 271116.000000 |
| **mean** | 68248.954396 | 25.556898 | 175.338970 | 70.702393 | 1978.378480 |
| **std** | 39022.286345 | 6.393561 | 10.518462 | 14.348020 | 29.877632 |
| **min** | 1.000000 | 10.000000 | 127.000000 | 25.000000 | 1896.000000 |
| **25%** | 34643.000000 | 21.000000 | 168.000000 | 60.000000 | 1960.000000 |
| **50%** | 68205.000000 | 24.000000 | 175.000000 | 70.000000 | 1988.000000 |
| **75%** | 102097.250000 | 28.000000 | 183.000000 | 79.000000 | 2002.000000 |
| **max** | 135571.000000 | 97.000000 | 226.000000 | 214.000000 | 2016.000000 |

In [3]:

```
olympics.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   Na.me     271116 non-null  object 
 2   Sex.rio   271116 non-null  object 
 3   A.ge      261642 non-null  float64
 4   Heigh.t  210945 non-null  float64
 5   Weig.ht  208241 non-null  float64
 6   Te am    271116 non-null  object
 7   N OC     271116 non-null  object
 8   Ga.mes   271116 non-null  object
 9   Ye.ar    271116 non-null  int64
 10  Season   271116 non-null  object
 11  Ci ty    271116 non-null  object
 12  Spor t   271116 non-null  object
 13  Ev ent   271116 non-null  object
 14  M edal   39783 non-null   object
dtypes: float64(3), int64(2), object(10)
memory usage: 31.0+ MB
```

In [4]:

```
index = olympics.index
number_of_rows = len(index)
print(number_of_rows)
```

```
271116
```

In [5]:

```
duplicate = olympics[olympics.duplicated()]
```

In [6]:

```
duplicate.count()
```

Out[6]:

```
ID.        1385
Na.me      1385
Sex.rio    1385
A.ge       1226
Heigh.t      28
Weig.ht      37
Te am      1385
N OC       1385
Ga.mes     1385
Ye.ar      1385
Season     1385
Ci ty      1385
Spor t     1385
Ev ent     1385
M edal       11
dtype: int64
```

In [7]:

```
duplicate
```

Out[7]:

| | ID. | Na.me | Sex.rio | A.ge | Heigh.t | Weig.ht | Te am | N OC | Ga.mes | Ye.ar | Season | Ci ty | Spor t | Ev ent | M eda |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **1252** | 704 | Dsir Antoine Acket | M | 27.0 | NaN | NaN | Belgium | BEL | 1932 Summer | 1932 | Summer | Los Angeles | Art Competitions | Art Competitions Mixed Painting, Unknown Event | Na |
| **4282** | 2449 | William Truman Aldrich | M | 48.0 | NaN | NaN | United States | USA | 1928 Summer | 1928 | Summer | Amsterdam | Art Competitions | Art Competitions Mixed Painting, Drawings And ... | Na |
| **4283** | 2449 | William Truman Aldrich | M | 48.0 | NaN | NaN | United States | USA | 1928 Summer | 1928 | Summer | Amsterdam | Art Competitions | Art Competitions Mixed Painting, Drawings And ... | Na |
| **4862** | 2777 | Hermann Reinhard Alker | M | 43.0 | NaN | NaN | Germany | GER | 1928 Summer | 1928 | Summer | Amsterdam | Art Competitions | Art Competitions Mixed Architecture, Designs F... | Na |
| **4864** | 2777 | Hermann Reinhard Alker | M | 43.0 | NaN | NaN | Germany | GER | 1928 Summer | 1928 | Summer | Amsterdam | Art Competitions | Art Competitions Mixed Architecture, Architect... | Na |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | .. |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **269994** | 135072 | Anna Katrina Zinkeisen (-Heseltine) | F | 46.0 | NaN | NaN | Great Britain | GBR | 1948 Summer | 1948 | Summer | London | Art Competitions | Art Competitions Mixed Painting, Paintings | NaN |
| **269995** | 135072 | Anna Katrina Zinkeisen (-Heseltine) | F | 46.0 | NaN | NaN | Great Britain | GBR | 1948 Summer | 1948 | Summer | London | Art Competitions | Art Competitions Mixed Painting, Paintings | NaN |
| **269997** | 135072 | Anna Katrina Zinkeisen (-Heseltine) | F | 46.0 | NaN | NaN | Great Britain | GBR | 1948 Summer | 1948 | Summer | London | Art Competitions | Art Competitions Mixed Painting, Unknown Event | NaN |
| **269999** | 135073 | Doris Clare Zinkeisen (-Johnstone) | F | 49.0 | NaN | NaN | Great Britain | GBR | 1948 Summer | 1948 | Summer | London | Art Competitions | Art Competitions Mixed Painting, Unknown Event | NaN |
| **270200** | 135173 | Henri Achille Zo | M | 58.0 | NaN | NaN | France | FRA | 1932 Summer | 1932 | Summer | Los Angeles | Art Competitions | Art Competitions Mixed Painting, Unknown Event | NaN |

1385 rows × 15 columns

Above are the duplicated columns which we need to clean

In [8]:
```
olympics.drop_duplicates(subset = None ,keep = False, inplace = True)
```

In [9]:
```
olympics.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 269119 entries, 0 to 271115
Data columns (total 15 columns):
 #   Column    Non-Null Count   Dtype  
---  ------    --------------   -----  
 0   ID.       269119 non-null  int64  
 1   Na.me     269119 non-null  object 
 2   Sex.rio   269119 non-null  object 
 3   A.ge      259896 non-null  float64
 4   Heigh.t   210904 non-null  float64
 5   Weig.ht   208196 non-null  float64
 6   Te am     269119 non-null  object 
 7   N OC      269119 non-null  object 
 8   Ga.mes    269119 non-null  object 
 9   Ye.ar     269119 non-null  int64  
 10  Season    269119 non-null  object 
 11  Ci ty     269119 non-null  object 
 12  Spor t    269119 non-null  object 
 13  Ev ent    269119 non-null  object 
 14  M edal    39761 non-null   object 
dtypes: float64(3), int64(2), object(10)
memory usage: 32.9+ MB
```

In [10]:
```
olympics
```

Out[10]:

|  | ID. | Na.me | Sex.rio | A.ge | Heigh.t | Weig.ht | Te am | N OC | Ga.mes | Ye.ar | Season | Ci ty | Spor t | Ev ent | ed |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | A Dijiang | M | 24.0 | 180.0 | 80.0 | China | CHN | 1992 Summer | 1992 | Summer | Barcelona | Basketball | Basketball Men's Basketball | Na |
| **1** | 2 | A Lamusi | M | 23.0 | 170.0 | 60.0 | China | CHN | 2012 Summer | 2012 | Summer | London | Judo | Judo Men's Extra-Lightweight | Na |
| **2** | 3 | Gunnar Nielsen Aaby | M | 24.0 | NaN | NaN | Denmark | DEN | 1920 Summer | 1920 | Summer | Antwerpen | Football | Football Men's Football | Na |
| **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- | G |

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | War | |
| **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 | Na |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| **271111** | 135569 | Andrzej ya | M | 29.0 | 179.0 | 89.0 | Poland-1 | POL | 1976 Winter | 1976 | Winter | Innsbruck | Luge | Luge Mixed (Men)'s Doubles | Na |
| **271112** | 135570 | Piotr ya | M | 27.0 | 176.0 | 59.0 | Poland | POL | 2014 Winter | 2014 | Winter | Sochi | Ski Jumping | Ski Jumping Men's Large Hill, Individual | Na |
| **271113** | 135570 | Piotr ya | M | 27.0 | 176.0 | 59.0 | Poland | POL | 2014 Winter | 2014 | Winter | Sochi | Ski Jumping | Ski Jumping Men's Large Hill, Team | Na |
| **271114** | 135571 | Tomasz Ireneusz ya | M | 30.0 | 185.0 | 96.0 | Poland | POL | 1998 Winter | 1998 | Winter | Nagano | Bobsleigh | Bobsleigh Men's Four | Na |
| **271115** | 135571 | Tomasz Ireneusz ya | M | 34.0 | 185.0 | 96.0 | Poland | POL | 2002 Winter | 2002 | Winter | Salt Lake City | Bobsleigh | Bobsleigh Men's Four | Na |

269119 rows × 15 columns

In [11]:

```
olympics.duplicated().sum()
```

Out[11]:

```
0
```

In [12]:

```
index = olympics.index
number_of_rows = len(index)
print(number_of_rows)
```

```
269119
```

Now we have cleaned all the duplicates from the dataset

Renaming columns into meaningful ones

In [13]:

```
olympics.columns = olympics.columns.str.replace(' ','')
olympics.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 269119 entries, 0 to 271115
Data columns (total 15 columns):
 #   Column   Non-Null Count   Dtype  
---  ------   --------------   -----  
 0   ID.      269119 non-null  int64  
 1   Na.me    269119 non-null  object 
 2   Sex.rio  269119 non-null  object 
 3   A.ge     259896 non-null  float64
 4   Heigh.t  210904 non-null  float64
 5   Weig.ht  208196 non-null  float64
 6   Team     269119 non-null  object 
 7   NOC      269119 non-null  object 
 8   Ga.mes   269119 non-null  object 
 9   Ye.ar    269119 non-null  int64  
 10  Season   269119 non-null  object 
 11  City     269119 non-null  object 
 12  Sport    269119 non-null  object 
 13  Event    269119 non-null  object 
 14  Medal    39761 non-null   object 
dtypes: float64(3), int64(2), object(10)
memory usage: 32.9+ MB
```

In above code removed spaces from columns

In [14]:

```
olympics.columns = olympics.columns.str.replace('.','')
olympics.info()
```

```
<ipython-input-14-7f8ee8154c45>:1: FutureWarning: The default value of regex will change from True to False in a
future version. In addition, single character regular expressions will*not* be treated as literal strings when re
gex=True.
  olympics.columns = olympics.columns.str.replace('.','')
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 269119 entries, 0 to 271115
Data columns (total 15 columns):
 #   Column  Non-Null Count   Dtype  
---  ------  --------------   -----  
 0   ID      269119 non-null  int64  
 1   Name    269119 non-null  object 
 2   Sexrio  269119 non-null  object 
 3   Age     259896 non-null  float64
 4   Height  210904 non-null  float64
 5   Weight  208196 non-null  float64
 6   Team    269119 non-null  object 
 7   NOC     269119 non-null  object 
 8   Games   269119 non-null  object 
 9   Year    269119 non-null  int64  
 10  Season  269119 non-null  object 
 11  City    269119 non-null  object 
 12  Sport   269119 non-null  object 
 13  Event   269119 non-null  object 
 14  Medal   39761 non-null   object 
dtypes: float64(3), int64(2), object(10)
memory usage: 32.9+ MB
```

In [15]:

```
olympics.columns = map(str.lower, olympics.columns)
olympics.head()
```

Out[15]:

| | id | name | sexrio | age | height | weight | team | noc | games | year | season | city | sport | event | medal |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | A Dijiang | M | 24.0 | 180.0 | 80.0 | China | CHN | 1992 Summer | 1992 | Summer | Barcelona | Basketball | Basketball Men's Basketball | 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 |
| **2** | 3 | Gunnar Nielsen Aaby | M | 24.0 | NaN | NaN | Denmark | DEN | 1920 Summer | 1920 | Summer | Antwerpen | Football | Football Men's Football | 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 |
| **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 |

In [16]:

```
olympics.rename( {'sexrio':'sex' } , axis=1 , inplace = True)
olympics.describe()
```

Out[16]:

| | id | age | height | weight | year |
|---|---|---|---|---|---|
| **count** | 269119.000000 | 259896.000000 | 210904.000000 | 208196.000000 | 269119.000000 |
| **mean** | 68269.049279 | 25.414662 | 175.338941 | 70.701618 | 1978.735236 |
| **std** | 39028.599815 | 6.076501 | 10.518581 | 14.349204 | 29.688086 |
| **min** | 1.000000 | 10.000000 | 127.000000 | 25.000000 | 1896.000000 |
| **25%** | 34656.000000 | 21.000000 | 168.000000 | 60.000000 | 1960.000000 |
| **50%** | 68244.000000 | 24.000000 | 175.000000 | 70.000000 | 1988.000000 |
| **75%** | 102117.500000 | 28.000000 | 183.000000 | 79.000000 | 2004.000000 |
| **max** | 135571.000000 | 97.000000 | 226.000000 | 214.000000 | 2016.000000 |

## Loading Second Dataset

In [17]:

```
regions = pd.read_csv('E:/Research in Data/Assignments/Proposal/Olympics/data_2/noc_regions.csv')

regions.head()
```

Out[17]:

| | NOC | region | notes |
|---|---|---|---|
| **0** | AFG | Afghanistan | NaN |
| **1** | AHO | Curacao | Netherlands Antilles |
| **2** | ALB | Albania | NaN |
| **3** | ALG | Algeria | NaN |
| **4** | AND | Andorra | NaN |

In [18]:

```
regions.columns = map(str.lower, regions.columns)
regions.head()
```

Out[18]:

| | noc | region | notes |
|---|---|---|---|
| **0** | AFG | Afghanistan | NaN |
| **1** | AHO | Curacao | Netherlands Antilles |
| **2** | ALB | Albania | NaN |
| **3** | ALG | Algeria | NaN |
| **4** | AND | Andorra | NaN |

Left Join Merge of 2 DataFrames

In [19]:

```
data = pd.merge(olympics,regions, on=['noc'], how='left')

data.head(5)
```

Out[19]:

| | id | name | sex | age | height | weight | team | noc | games | year | season | city | sport | event | medal | region |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **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 |
| **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 |
| **2** | 3 | Gunnar Nielsen Aaby | M | 24.0 | NaN | NaN | Denmark | DEN | 1920 Summer | 1920 | Summer | Antwerpen | Football | Football Men's Football | NaN | Denmark |
| **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 |
| **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 |

In [20]:

```
data.sort_values(by=['year'], inplace=True, ascending=False)
data.head()
```

Out[20]:

| | id | name | sex | age | height | weight | team | noc | games | year | season | city | sport | event | medal | reg |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **194731** | 98525 | Maksim Sergeyevich Rakov | M | 30.0 | 181.0 | 100.0 | Kazakhstan | KAZ | 2016 Summer | 2016 | Summer | Rio de Janeiro | Judo | Judo Men's Half-Heavyweight | NaN | Kazakhs |
| **162035** | 82008 | Alexandra Patricia "Alex" Morgan | F | 27.0 | 173.0 | 62.0 | United States | USA | 2016 Summer | 2016 | Summer | Rio de Janeiro | Football | Football Women's Football | NaN | U |
| **117646** | 59981 | Kim Hyeon-Seop | M | 31.0 | 175.0 | 53.0 | South Korea | KOR | 2016 Summer | 2016 | Summer | Rio de Janeiro | Athletics | Athletics Men's 20 kilometres Walk | NaN | Sc Kc |
| **117647** | 59981 | Kim Hyeon-Seop | M | 31.0 | 175.0 | 53.0 | South Korea | KOR | 2016 Summer | 2016 | Summer | Rio de Janeiro | Athletics | Athletics Men's 50 kilometres Walk | NaN | Sc Kc |

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **38150** | 19756 | Chang Hao | M | 25.0 | 173.0 | 72.0 | Chinese Taipei | TPE | 2016 Summer | 2016 | Summer | Rio de Janeiro | Sailing | Sailing Men's Windsurfer | NaN | Tai |

In [21]:

```
#data = data[data['year'].between(1964, 2015)] - This code is used to only exact 50 Years .... currently I am pla
```

Search for null values

In [22]:

```
data.isnull().sum().sum()
```

Out[22]:

```
622177
```

In [23]:

```
data.isnull().sum()
```

Out[23]:

```
id             0
name           0
sex            0
age         9223
height     58215
weight     60923
team           0
noc            0
games          0
year           0
season         0
city           0
sport          0
event          0
medal     229358
region       370
notes     264088
dtype: int64
```

Correcting Variable's Data Dype

In [25]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 269119 entries, 194731 to 69539
Data columns (total 17 columns):
 #   Column  Non-Null Count   Dtype  
---  ------  --------------   -----  
 0   id      269119 non-null  int64  
 1   name    269119 non-null  object 
 2   sex     269119 non-null  object 
 3   age     259896 non-null  float64
 4   height  210904 non-null  float64
 5   weight  208196 non-null  float64
 6   team    269119 non-null  object 
 7   noc     269119 non-null  object 
 8   games   269119 non-null  object 
 9   year    269119 non-null  int64  
 10  season  269119 non-null  object 
 11  city    269119 non-null  object 
 12  sport   269119 non-null  object 
 13  event   269119 non-null  object 
 14  medal   39761 non-null   object 
 15  region  268749 non-null  object 
 16  notes   5031 non-null    object 
dtypes: float64(3), int64(2), object(12)
memory usage: 37.0+ MB
```

In [26]:

```
#data.weight.duplicated().sum()
data['weight'].isnull().sum()

data['weight'].isnull().sum()
```

Out[26]:

```
60923
```

In [27]:

```
data['weight'].unique()
```

Out[27]:

```
array([100.        ,  62.        ,  53.        ,  72.        ,
        98.        ,  50.        ,  80.        ,  45.        ,
        60.        ,  52.        ,  95.        ,  48.        ,
        61.        ,  92.        ,  63.        ,  59.        ,
        76.        , 125.        ,  77.        ,  78.        ,
        54.        ,  73.        ,  42.        ,  70.        ,
        82.        ,  90.        ,  75.        ,  65.        ,
       105.        ,  56.        ,  58.        ,  86.        ,
        34.        ,  91.        ,  57.        ,  67.        ,
        68.        ,  71.        ,  55.        ,  66.        ,
        64.        ,  69.        ,  87.        , 143.        ,
        74.        ,  49.        , 108.        ,          nan,
        88.        ,  79.        ,  85.        ,  94.        ,
        84.        ,  83.        , 118.        ,  99.        ,
        93.        , 107.        ,  51.        , 104.        ,
        39.        ,  44.        ,  46.        ,  97.        ,
        40.        , 110.        ,  81.        , 130.        ,
       103.        ,  47.        ,  96.        , 120.        ,
       101.        , 155.        , 109.        , 102.        ,
       123.        ,  43.        , 124.        ,  89.        ,
       115.        , 140.        ,  41.        , 135.        ,
       112.        , 136.        , 150.        , 127.        ,
       113.        , 106.        , 117.        , 170.        ,
       111.        , 114.        , 122.        , 152.        ,
       160.        , 116.        ,  38.        ,  35.        ,
       142.        ,  37.        , 119.        , 132.        ,
       134.        ,  30.        ,  36.        , 158.        ,
       126.        , 138.        , 128.        , 148.        ,
       141.        , 139.        , 146.        , 147.        ,
       133.        ,  33.        , 137.        , 144.        ,
       121.        ,  31.        , 145.        , 165.        ,
        69.5       , 118.5       ,  55.5       , 154.        ,
        79.5       ,  56.5       ,  64.5       , 103.5       ,
       129.        , 101.5       , 214.        , 102.5       ,
       156.        ,  97.5       ,  65.5       , 131.        ,
        70.5       ,  88.5       ,  81.5       ,  32.        ,
        52.5       ,  90.5       , 131.5       , 167.        ,
        85.5       , 109.5       , 151.        , 121.5       ,
       117.5       , 163.        , 133.5       ,  28.        ,
        72.5       ,  87.5       , 161.        , 175.        ,
       104.5       , 137.5       ,  54.5       ,  76.5       ,
        67.5       ,  74.5       ,  98.5       , 198.        ,
       178.        ,  77.5       ,  71.5       ,  68.5       ,
       138.5       ,  61.5       ,  75.5       ,  80.5       ,
       100.5       ,  63.5       ,  60.5       , 180.        ,
        91.5       ,  48.5       ,  78.5       , 129.5       ,
        66.5       ,  82.5       ,  77.33333333,  73.5       ,
        86.5       ,  93.5       , 149.        ,  89.5       ,
       108.5       , 107.5       , 127.5       , 176.5       ,
        96.5       ,  62.5       ,  57.5       ,  84.5       ,
        49.5       ,  92.5       , 112.5       ,  83.5       ,
       135.5       ,  74.66666667,  25.        , 190.        ,
       122.5       ,  53.5       ,  59.5       ,  51.5       ,
       146.5       ,  58.5       , 182.        , 105.5       ,
       106.5       , 130.5       , 116.5       , 123.5       ,
        95.5       ])
```

In [28]:

```
data['sex'] = data['sex'].astype('category')
data['year'] = data['year'].astype('int')
data['team'] = data['team'].astype('category')
data['season'] = data['season'].astype('category')
data['weight'] = data['weight'].astype('float')
data['height'] = data['height'].astype('float')
data['name'] = data['name'].astype('str')
data['noc'] = data['noc'].astype('category')
data['city'] = data['city'].astype('str')
data['sport'] = data['sport'].astype('str')
data['event'] = data['event'].astype('category')
data['medal'] = data['medal'].astype('category')
data['games'] = data['games'].astype('category')
```

In [29]:

```
data.dtypes
```

Out[29]:

```
id            int64
name         object
```

```
sex       category
age        float64
height     float64
weight     float64
team      category
noc       category
games     category
year         int32
season    category
city        object
sport       object
event     category
medal     category
region      object
notes       object
dtype: object
```

Check for Category Variable Uniqueness

In [30]:

```
data['sex'].unique()
```

Out[30]:

```
['M', 'F']
Categories (2, object): ['M', 'F']
```

In [31]:

```
data['noc'].unique()
```

Out[31]:

```
['KAZ', 'USA', 'KOR', 'TPE', 'RUS', ..., 'BOH', 'ANZ', 'UNK', 'CRT', 'NFL']
Length: 230
Categories (230, object): ['KAZ', 'USA', 'KOR', 'TPE', ..., 'ANZ', 'UNK', 'CRT', 'NFL']
```

In [32]:

```
data['games'].unique()
```

Out[32]:

```
['2016 Summer', '2014 Winter', '2012 Summer', '2010 Winter', '2008 Summer', ..., '1908 Summer', '1906 Summer', '1
904 Summer', '1900 Summer', '1896 Summer']
Length: 51
Categories (51, object): ['2016 Summer', '2014 Winter', '2012 Summer', '2010 Winter', ..., '1906 Summer', '1904 S
ummer', '1900 Summer', '1896 Summer']
```

In [33]:

```
data['season'].unique()
```

Out[33]:

```
['Summer', 'Winter']
Categories (2, object): ['Summer', 'Winter']
```

In [34]:

```
data['sport'].unique()
```

Out[34]:

```
array(['Judo', 'Football', 'Athletics', 'Sailing', 'Rowing', 'Archery',
       'Wrestling', 'Gymnastics', 'Cycling', 'Fencing', 'Swimming',
       'Badminton', 'Hockey', 'Table Tennis', 'Shooting', 'Boxing',
       'Volleyball', 'Taekwondo', 'Rhythmic Gymnastics', 'Golf',
       'Weightlifting', 'Handball', 'Tennis', 'Water Polo',
       'Modern Pentathlon', 'Rugby Sevens', 'Canoeing', 'Triathlon',
       'Equestrianism', 'Synchronized Swimming', 'Diving', 'Basketball',
       'Beach Volleyball', 'Trampolining', 'Cross Country Skiing',
       'Biathlon', 'Alpine Skiing', 'Figure Skating', 'Freestyle Skiing',
       'Ice Hockey', 'Ski Jumping', 'Speed Skating', 'Bobsleigh',
       'Nordic Combined', 'Snowboarding', 'Short Track Speed Skating',
       'Luge', 'Skeleton', 'Curling', 'Baseball', 'Softball',
       'Art Competitions', 'Polo', 'Aeronautics', 'Alpinism',
       'Military Ski Patrol', 'Rugby', 'Tug-Of-War', 'Lacrosse',
       'Jeu De Paume', 'Racquets', 'Motorboating', 'Roque', 'Cricket',
       'Croquet', 'Basque Pelota'], dtype=object)
```

In [35]:

```
data['event'].unique()
```

Out[35]:

```
['Judo Men's Half-Heavyweight', 'Football Women's Football', 'Athletics Men's 20 kilometres Walk', 'Athletics Men
's 50 kilometres Walk', 'Sailing Men's Windsurfer', ..., 'Gymnastics Men's Horizontal Bar, Teams', 'Shooting Men'
s Free Pistol, 30 metres', 'Swimming Men's 1,200 metres Freestyle', 'Swimming Men's 500 metres Freestyle', 'Wrest
ling Men's Unlimited Class, Greco-Roman']
Length: 765
Categories (765, object): ['Judo Men's Half-Heavyweight', 'Football Women's Football', 'Athletics Men's 20 kilome
tres Walk', 'Athletics Men's 50 kilometres Walk', ..., 'Shooting Men's Free Pistol, 30 metres', 'Swimming Men's 1
,200 metres Freestyle', 'Swimming Men's 500 metres Freestyle', 'Wrestling Men's Unlimited Class, Greco-Roman']
```

In [36]:

```
data['medal'].unique()
```

Out[36]:

```
[NaN, 'Gold', 'Bronze', 'Silver']
Categories (3, object): ['Gold', 'Bronze', 'Silver']
```

In [37]:

```
data['team'].unique()
```

Out[37]:

```
['Kazakhstan', 'United States', 'South Korea', 'Chinese Taipei', 'Russia', ..., 'Crocodile-13', 'Fantlet-2', 'Eth
nikos Gymnastikos Syllogos', 'Great Britain/Germany', 'Australia/Great Britain']
Length: 1164
Categories (1164, object): ['Kazakhstan', 'United States', 'South Korea', 'Chinese Taipei', ..., 'Fantlet-2', 'Et
hnikos Gymnastikos Syllogos', 'Great Britain/Germany', 'Australia/Great Britain']
```

Above values seems to be ok , there is no need to modify or remove any values as part of cleaning

### Dealing with Empty Columns & Outliers in the dataset

In [38]:

```
data['id'].count()
```

Out[38]:

```
269119
```

In [39]:

```
data.isnull().sum(axis = 0)
```

Out[39]:

```
id             0
name           0
sex            0
age         9223
height     58215
weight     60923
team           0
noc            0
games          0
year           0
season         0
city           0
sport          0
event          0
medal     229358
region       370
notes     264088
dtype: int64
```

Below is percentage of NaN values for each columns

In [40]:

```
nans = data.isna().mean().mul(100).round()
nans
```

Out[40]:

```
id         0.0
name       0.0
sex        0.0
age        3.0
height    22.0
weight    23.0
team       0.0
noc        0.0
games      0.0
```

```
year        0.0
season      0.0
city        0.0
sport       0.0
event       0.0
medal      85.0
region      0.0
notes      98.0
dtype: float64
```

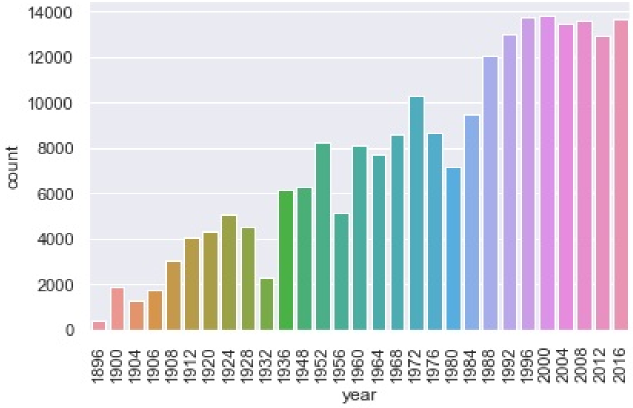
Age : Checking the empty values in the column it's only 3.42% of total rows so , I am not going to replace them with mean or median. Height : 22% of total rows shows empty values. so I neeed to either drop them or replace with mean/median Weight : 23% of total rows shows empty values. Again I neeed to either drop them or replace with mean/median Theoretically, 25% to 30% is the maximum missing values allowed, beyond that we might want to drop the variable from analysis. In practical this may very based on client requirements.In this case I am doing Imputation for missing data. but before deciding on use mean or median I need to check if the data is skewed or not. If it is symmetrical it's appropriate to use mean. In skwed case using median will the best solution. Let me have a look at the boxplot before moving forward.

In [174…

```
part = data[(data.season == 'Summer')]
```

In [177…

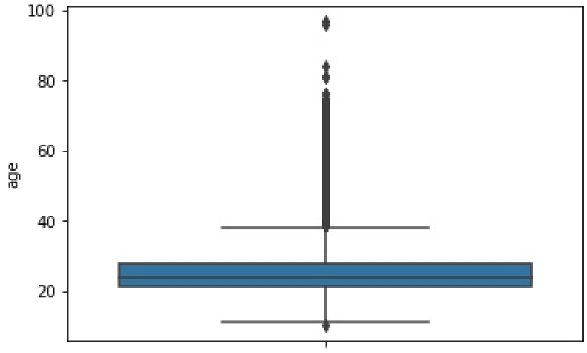
```
sns.countplot(x="year", data=part)
plt.xticks(rotation = 90)
plt.tight_layout()
```



Atheletes Particiation over the years(Including male and female).

In [41]:

```
sns.boxplot(y='age', data=data)
plt.show()
```

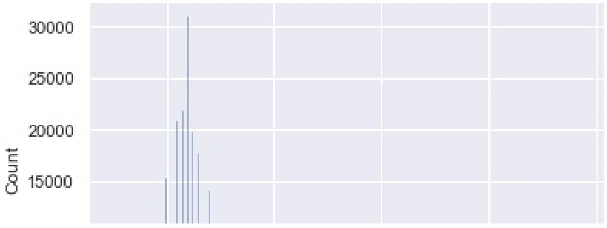


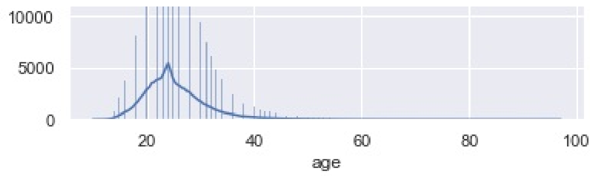
In [146…

```
sns.histplot(data['age'], kde=True)
```

Out[146…

```
<AxesSubplot:xlabel='age', ylabel='Count'>
```

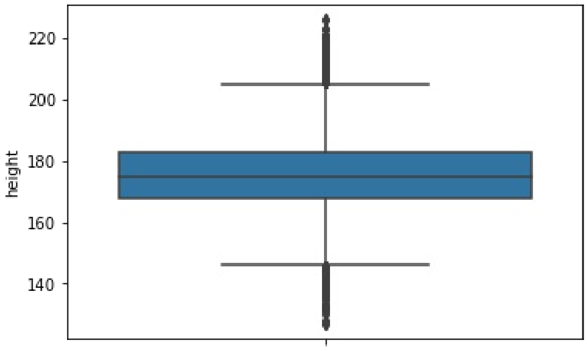




From the above graph I can see the median age is lying between 20 to 30. even though few points near hundred seems like outliers, still we can not remove them as these are possible values for a human age

In [42]:

```
sns.boxplot(y='height', data=data)
plt.show()
```

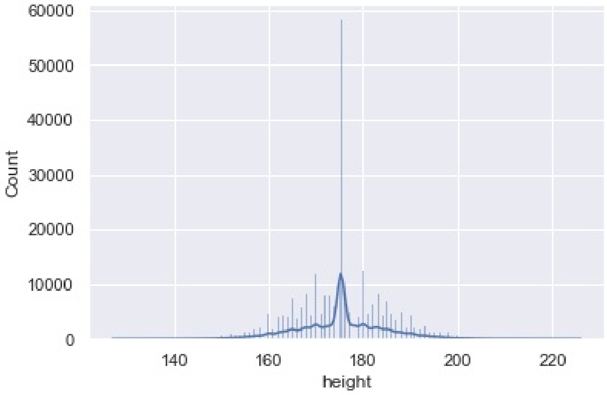


In [147…

```
sns.histplot(data['height'], kde=True)
```

Out[147…

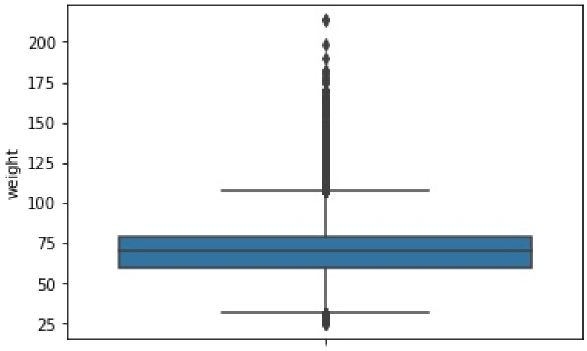
```
<AxesSubplot:xlabel='height', ylabel='Count'>
```



Above graph points doesn't looks like outliers, the points showing are only possible values for height.

In [43]:

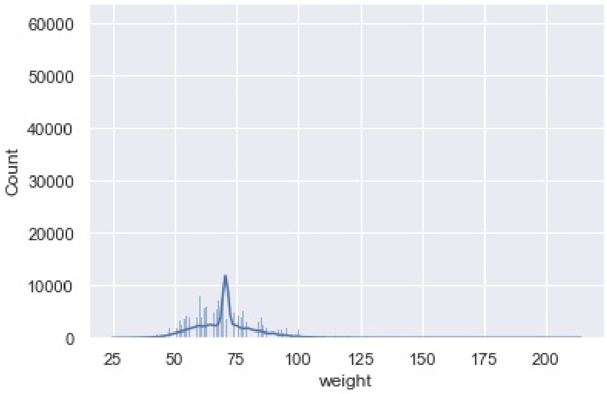
```
sns.boxplot(y='weight', data=data)
plt.show()
```



In [148…

```
sns.histplot(data['weight'], kde=True)
```

Out[148… <AxesSubplot:xlabel='weight', ylabel='Count'>



Above graph points doesn't looks like outliers, the points showing are still possible outcomes, further study is required to make more clarity

## Overview

In my view I looked at 3 Graphs for replacing empty columns with mean/median. and here are my findings.

- Age : Data looks skewed so I have replace it with median
- Height : Looks symmetrical, it is ok to replace them with the mean
- Weight : Data looks skewed so I have replace it with median

Replacing NaN values with mean/median

In [44]:

```
data.update(data['age'].fillna(value=data['age'].median(), inplace=True))
```

In [45]:

```
data.update(data['height'].fillna(value=data['height'].mean(), inplace=True))
```

In [46]:

```
data.update(data['weight'].fillna(value=data['weight'].mean(), inplace=True))
```

In [47]:

```
data['age'].isnull().sum()
```

Out[47]:

```
0
```

In [48]:

```
data['height'].isnull().sum()
```

Out[48]:

```
0
```

In [49]:

```
data['weight'].isnull().sum()
```

Out[49]:

```
0
```

In [50]:

```
data['medal'].value_counts().sum()
```

Out[50]:

```
39761
```

No need to worry about the medals for now, because it will show values for only those who won the competetion.

In [51]:

```
data['region'].value_counts().sum()
```

Out[51]:

```
268749
```

Region has only 0.13% of empty data which doesn't really affect the study compared to the total data we have.

Data Cleaning has been completed

## Distribution of the age of winning gold medals

In [52]:

```
gold_medalists = data[(data.medal == 'Gold')]
gold_medalists.head()
```

Out[52]:

| | id | name | sex | age | height | weight | team | noc | games | year | season | city | sport | event | medal | region | note |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **38164** | 19760 | Chang Hye-Jin | F | 29.0 | 158.0 | 50.0 | South Korea | KOR | 2016 Summer | 2016 | Summer | Rio de Janeiro | Archery | Archery Women's Individual | Gold | South Korea | Na |
| **38165** | 19760 | Chang Hye-Jin | F | 29.0 | 158.0 | 50.0 | South Korea | KOR | 2016 Summer | 2016 | Summer | Rio de Janeiro | Archery | Archery Women's Team | Gold | South Korea | Na |
| **10150** | 5561 | Nickel Ashmeade | M | 26.0 | 183.0 | 77.0 | Jamaica | JAM | 2016 Summer | 2016 | Summer | Rio de Janeiro | Athletics | Athletics Men's 4 x 100 metres Relay | Gold | Jamaica | Na |
| **242984** | 122548 | Josua Tuisova Ratulevu | M | 22.0 | 180.0 | 108.0 | Fiji | FIJ | 2016 Summer | 2016 | Summer | Rio de Janeiro | Rugby Sevens | Rugby Sevens Men's Rugby Sevens | Gold | Fiji | Na |
| **243000** | 122556 | Blair Tuke | M | 27.0 | 181.0 | 78.0 | New Zealand | NZL | 2016 Summer | 2016 | Summer | Rio de Janeiro | Sailing | Sailing Men's Skiff | Gold | New Zealand | Na |

### Exporting data for API Creation

In [53]:

```
data.to_csv('E:/Research in Data/Assignments/Proposal/Olympics/data_2/api_dataset.csv')
```

# Describe Variables

id = continueous Variables - Stands for each id of athlete

name = categorical - nominal - name of athletes

age = continueous numeric - age of athlete

height = continueous numeric - height of athlete in cm

weight = continueous numeric - weight of athlete in kg

sex = categorical , norminal - male/female info about athlete

team = categorical , norminal - name of the team athlete represent

noc = categorical , norminal - short name of the team athlete represent

games = categorical , norminal - Year and season of the game

year = continueous numeric - Year of the event

season = categorical , norminal - Summer/winter seaons of olympic games

city = categorical , norminal - name of the city olympics conducted

sport = categorical , norminal - which sports item the athlete participated

event = categorical , norminal - name of the sports in detail format

medal = categorical , ordinal - gold/silver/bronze medals for the athelete, empty if no medal

region = categorical , norminal - regios is same as the team

Above code creates a dataframe only for Gold Medalists

In [54]:

```
gold_medalists['age'].isnull().any()
```

```
gold_medalists['age'].isnull().any()
```

Out[54]:

```
False
```

In [55]:

```
gold_medalists = gold_medalists[np.isfinite(gold_medalists['age'])]
```

In [56]:

```
gold_medalists['age'].isnull().any()
```

Out[56]:

```
False
```

I am avoiding NaN values from the gold winners dataset as they will be affecting the mean and median of the varibales.(Another chance was to fill them with zero'z or replace with mean or median both of them seems to be unjustifiable)

In [159…

```
gold_medalists['age'].median()
```

Out[159…

```
25.0
```

In [160…

```
gold_medalists['age'].mean()
```

Out[160…

```
25.878273230585066
```

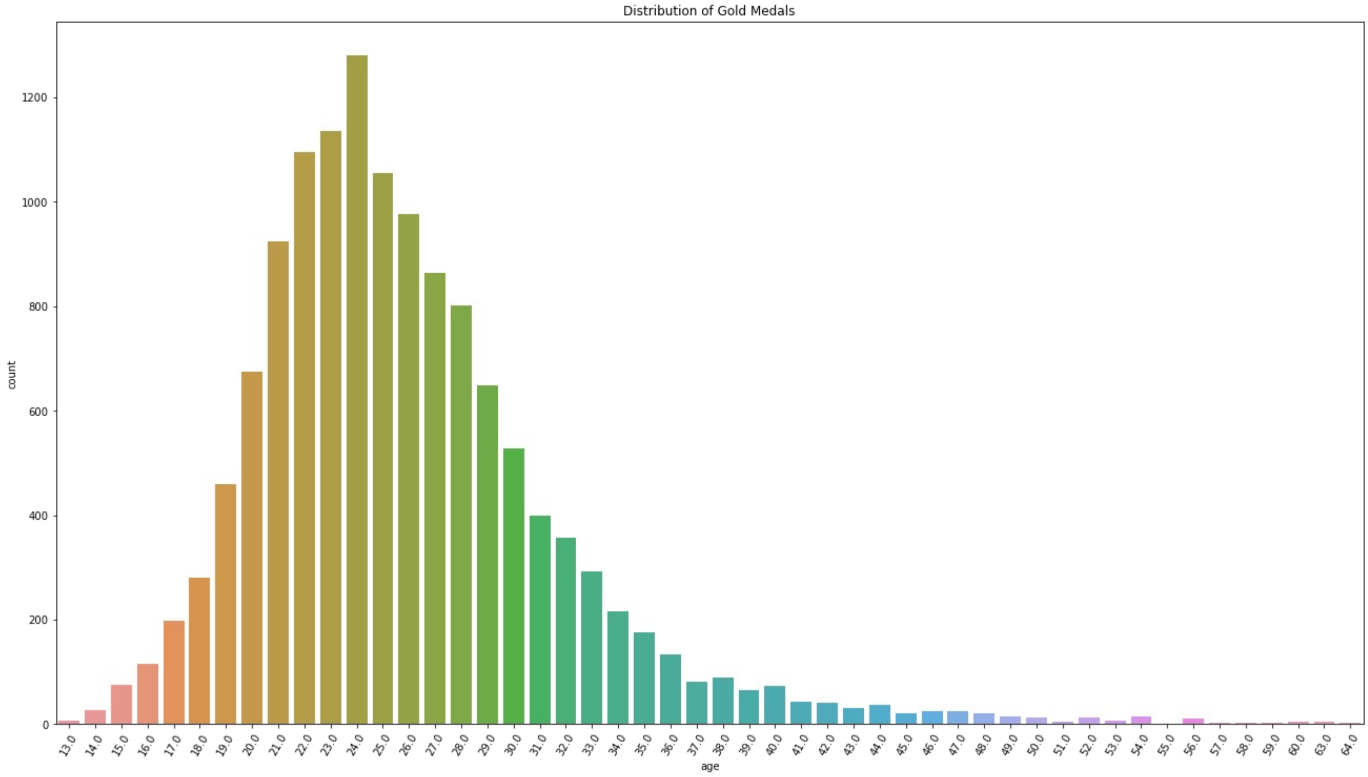
In [57]:

```
plt.figure(figsize=(18, 10))
plt.xticks(rotation = 60)
plt.tight_layout()
sns.countplot(gold_medalists['age'])
plt.title('Distribution of Gold Medals')
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable
as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other argu
ments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[57]:

```
Text(0.5, 1.0, 'Distribution of Gold Medals')
```



Above graph is slightly skewed to the right. Median might be lying between 23 to 24. But since this is skewed mean might very. So we will be taking median for the further study.

So let's try to check now medal winning of athletes whose age are more than 50

That's seems intresting !, there are 201 athletes who won medan aged more than 50

I am gonna create a new Dataframe called master_medalists with these 201 people and try to visualize it

In [58]:

```
master_medalists = gold_medalists['sport'][gold_medalists['age'] > 50]
```

In [59]:

```
master_medalists.count()
```

Out[59]:

```
65
```

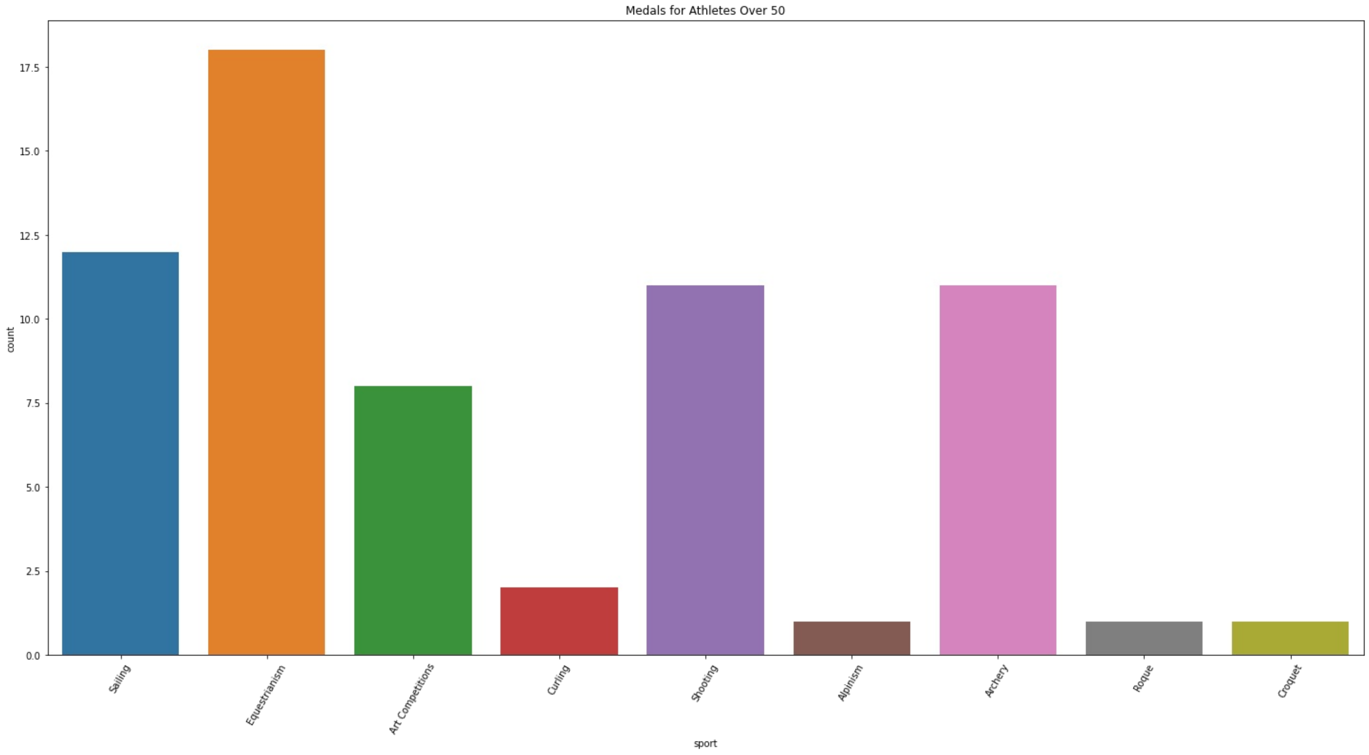
In [60]:

```
plt.figure(figsize=(20, 10))
plt.xticks(rotation = 60)
plt.tight_layout()
sns.countplot(master_medalists)
plt.title('Medals for Athletes Over 50')
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable
as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other argu
ments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[60]:

```
Text(0.5, 1.0, 'Medals for Athletes Over 50')
```



It Seems our senior gold medalists are from shooters, archers, sailors and, above all, horse riders(Equestrianism) !

## Overview into Women Athletes Contributions

In [61]:

```
women_summer = data[(data.sex == 'F') & (data.season == 'Summer')]
women_summer.head()
```

Out[61]:

| | id | name | sex | age | height | weight | team | noc | games | year | season | city | sport | event | medal | region | not |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **162035** | 82008 | Alexandra Patricia "Alex" Morgan | F | 27.0 | 173.0 | 62.0 | United States | USA | 2016 Summer | 2016 | Summer | Rio de Janeiro | Football | Football Women's Football | NaN | USA | Na |

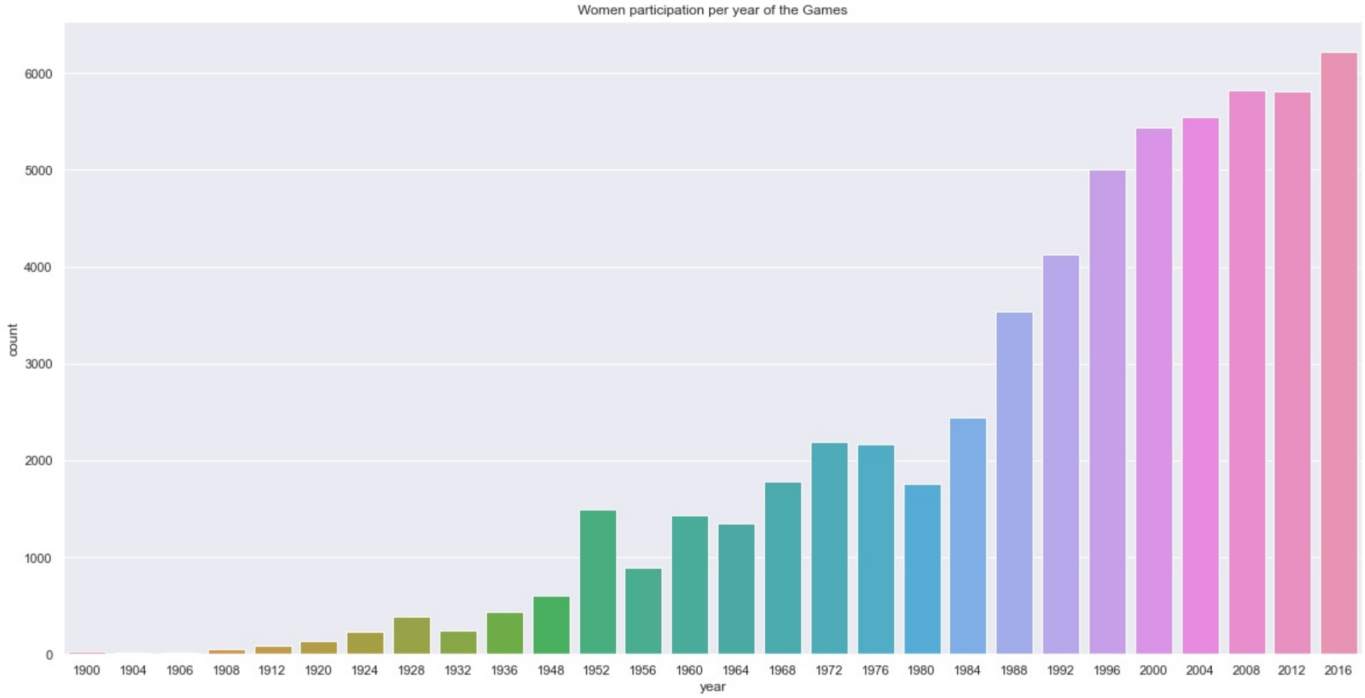
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **38164** | 19760 | Chang Hye-Jin | F | 29.0 | 158.0 | 50.0 | South Korea | KOR | 2016 Summer | 2016 | Summer | Rio de Janeiro | Archery | Archery Women's Individual | Gold | South Korea | Na |
| **38165** | 19760 | Chang Hye-Jin | F | 29.0 | 158.0 | 50.0 | South Korea | KOR | 2016 Summer | 2016 | Summer | Rio de Janeiro | Archery | Archery Women's Team | Gold | South Korea | Na |
| **161905** | 81946 | Alexa Citiali Moreo Medina | F | 21.0 | 147.0 | 45.0 | Mexico | MEX | 2016 Summer | 2016 | Summer | Rio de Janeiro | Gymnastics | Gymnastics Women's Balance Beam | NaN | Mexico | Na |
| **161904** | 81946 | Alexa Citiali Moreo Medina | F | 21.0 | 147.0 | 45.0 | Mexico | MEX | 2016 Summer | 2016 | Summer | Rio de Janeiro | Gymnastics | Gymnastics Women's Uneven Bars | NaN | Mexico | Na |

In [62]:

```
sns.set(style="darkgrid")
plt.figure(figsize=(20, 10))
sns.countplot(x='year', data=women_summer)
plt.title('Women participation per year of the Games')
```

Out[62]:

```
Text(0.5, 1.0, 'Women participation per year of the Games')
```



Above diagram shows a women participation has increased over time in summer olympics

In [63]:

```
women_winter = data[(data.sex == 'F') & (data.season == 'Winter')]
women_winter.head()
```

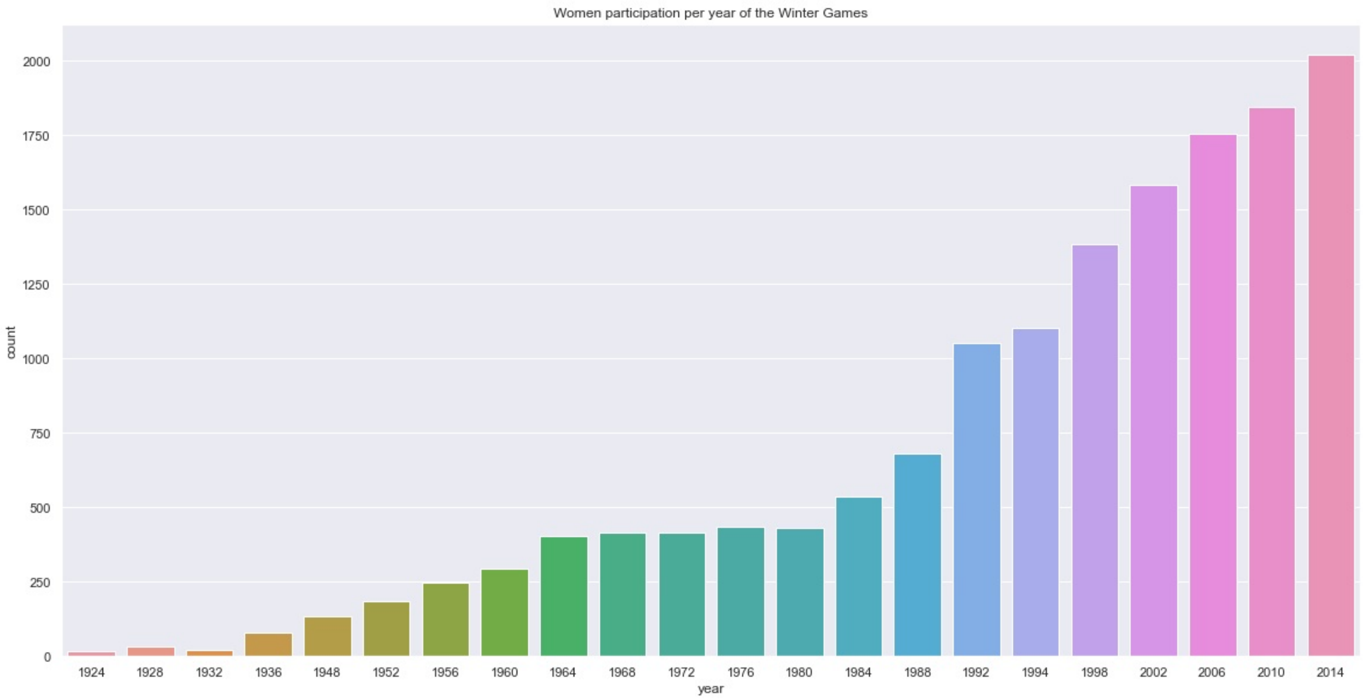
Out[63]:

|  | id | name | sex | age | height | weight | team | noc | games | year | season | city | sport | event | medal | region | notes |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **224004** | 113417 | Gabriela Soukalov (-Koukalov) | F | 24.0 | 170.0 | 62.0 | Czech Republic | CZE | 2014 Winter | 2014 | Winter | Sochi | Biathlon | Biathlon Women's 12.5 kilometres Mass Start | Silver | Czech Republic | NaN |
| **43100** | 22368 | Alexandra Coletti | F | 30.0 | 164.0 | 60.0 | Monaco | MON | 2014 Winter | 2014 | Winter | Sochi | Alpine Skiing | Alpine Skiing Women's Combined | NaN | Monaco | NaN |
| **44002** | 22848 | Emily Cook | F | 34.0 | 160.0 | 52.0 | United States | USA | 2014 Winter | 2014 | Winter | Sochi | Freestyle Skiing | Freestyle Skiing Women's Aerials | NaN | USA | NaN |
| **44126** | 22911 | Penny Jane Coomes | F | 24.0 | 152.0 | 43.0 | Great Britain | GBR | 2014 Winter | 2014 | Winter | Sochi | Figure Skating | Figure Skating Mixed Ice Dancing | NaN | UK | NaN |
|  |  | Alena |  |  |  |  |  |  | 2014 |  |  |  | Cross | Cross Country Skiing |  |  |  |

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **191963** | 97106 | Prochzkov | F | 29.0 | 171.0 | 55.0 | Slovakia | SVK | Winter | 2014 | Winter | Sochi | Country Skiing | Women's 10 kilometres | NaN | Slovakia | NaN |

In [64]:
```
sns.set(style="darkgrid")
plt.figure(figsize=(20, 10))
sns.countplot(x='year', data=women_winter)
plt.title('Women participation per year of the Winter Games')
```

Out[64]:
```
Text(0.5, 1.0, 'Women participation per year of the Winter Games')
```



Above diagram shows a women participation has increased over time in winter olympics as well

Inroder to cross check the above diagram I am just displaying the dataframe again down for year 1956 Year

In [65]:
```
women_winter = data[(data.year == 1956) & (data.medal == 'Gold') & (data.sex == 'F') & (data.id == 2386)]
women_winter
```

Out[65]:

| | id | name | sex | age | height | weight | team | noc | games | year | season | city | sport | event | medal | region | notes |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **4155** | 2386 | Tenley Emma Albright (-Gardiner, -Blakely) | F | 20.0 | 175.338941 | 70.701618 | United States | USA | 1956 Winter | 1956 | Winter | Cortina d'Ampezzo | Figure Skating | Figure Skating Women's Singles | Gold | USA | NaN |

Again to match with the diagram and cross check just taking the count

In [66]:
```
women_winter['id'].loc[women_winter['year'] == 1956].count()
```

Out[66]:
```
1
```

Now it's clearly visible that 246 is matching the count plot year of 1956 , Even I have verified the data in Wikipedia to make sure my research is going in the right direction

## Medals Achieved by each country

Checking which are the top countries leading with gold medals

In [67]:

```
gold_medalists.region.value_counts().reset_index(name='Medals').head(5)
```

Out[67]:

|   | index | Medals |
|---|---|---|
| 0 | USA | 2638 |
| 1 | Russia | 1599 |
| 2 | Germany | 1301 |
| 3 | UK | 676 |
| 4 | Italy | 575 |

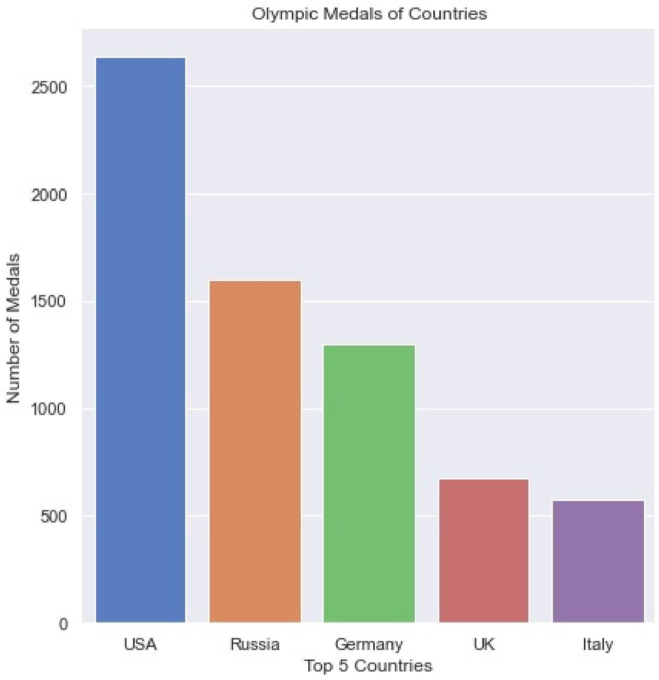
Now I am going to visualize the results in graph

In [68]:

```
totalGoldMedals = gold_medalists.region.value_counts().reset_index(name='medal').head(5)
g = sns.catplot(x="index", y="medal", data=totalGoldMedals,
                height=6, kind="bar", palette="muted")
g.despine(left=True)
g.set_xlabels("Top 5 Countries")
g.set_ylabels("Number of Medals")
plt.title('Olympic Medals of Countries')
```

Out[68]:

```
Text(0.5, 1.0, 'Olympic Medals of Countries')
```



USA Seems to have most number of gold medals compared to other countries

## Disciplines with greatest number of Gold Medals

Creating a dataframe for showing USA Gold medal disciplines

In [69]:

```
us_gld_mdls = gold_medalists.loc[gold_medalists['noc'] == 'USA']
```

In [70]:

```
us_gld_mdls.event.value_counts().reset_index(name='medal').head(20)
```

Out[70]:

|   | index | 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 |
| 5 | Athletics Men's 4 x 400 metres Relay | 81 |
| 6 | Swimming Women's 4 x 100 metres Medley Relay | 79 |

|  |  |  |
|---|---|---|
| **7** | Swimming Women's 4 x 100 metres Freestyle Relay | 78 |
| **8** | Football Women's Football | 66 |
| **9** | Athletics Men's 4 x 100 metres Relay | 63 |
| **10** | Swimming Men's 4 x 100 metres Freestyle Relay | 58 |
| **11** | Athletics Women's 4 x 100 metres Relay | 50 |
| **12** | Softball Women's Softball | 45 |
| **13** | Athletics Women's 4 x 400 metres Relay | 38 |
| **14** | Rowing Women's Coxed Eights | 36 |
| **15** | Volleyball Men's Volleyball | 36 |
| **16** | Ice Hockey Men's Ice Hockey | 36 |
| **17** | Rugby Men's Rugby | 36 |
| **18** | Swimming Women's 4 x 200 metres Freestyle Relay | 33 |
| **19** | Water Polo Women's Water Polo | 25 |

I can see Basketball is the US's most won discipline, In this study we need to more focus on the individual persons charecterstics rather than the team, so we will check success of male athletes to better review it.

I am creating a new dataframe to have more deeper look into the gold winning og USA Male Athletes Basketball Gold Medal Winning

In [71]:

```
basket_us_gld_mdls = us_gld_mdls.loc[(us_gld_mdls['sport'] == 'Basketball') & (us_gld_mdls['sex'] == 'M')].sort_v
```

In [72]:

```
basket_us_gld_mdls.head(15)
```

Out[72]:

|  | id | name | sex | age | height | weight | team | noc | games | year | season | city | sport | event | medal | region | note |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **194273** | 98309 | Jack Williamson Ragland | M | 22.0 | 183.0 | 79.0 | United States | USA | 1936 Summer | 1936 | Summer | Berlin | Basketball | Basketball Men's Basketball | Gold | USA | Na |
| **22247** | 11790 | Ralph English Bishop | M | 20.0 | 193.0 | 86.0 | United States | USA | 1936 Summer | 1936 | Summer | Berlin | Basketball | Basketball Men's Basketball | Gold | USA | Na |
| **211772** | 107150 | Willard Theodore Schmidt | M | 26.0 | 205.0 | 86.0 | United States | USA | 1936 Summer | 1936 | Summer | Berlin | Basketball | Basketball Men's Basketball | Gold | USA | Na |
| **160458** | 81220 | Arthur Owen "Art" Mollner | M | 23.0 | 183.0 | 73.0 | United States | USA | 1936 Summer | 1936 | Summer | Berlin | Basketball | Basketball Men's Basketball | Gold | USA | Na |
| **257597** | 129925 | William John "Bill" Wheatley | M | 27.0 | 188.0 | 79.0 | United States | USA | 1936 Summer | 1936 | Summer | Berlin | Basketball | Basketball Men's Basketball | Gold | USA | Na |
| **108633** | 55375 | Francis Lee Johnson | M | 25.0 | 180.0 | 79.0 | United States | USA | 1936 Summer | 1936 | Summer | Berlin | Basketball | Basketball Men's Basketball | Gold | USA | Na |
| **142260** | 71965 | Frank John Lubin | M | 26.0 | 200.0 | 113.0 | United States | USA | 1936 Summer | 1936 | Summer | Berlin | Basketball | Basketball Men's Basketball | Gold | USA | Na |
| **187885** | 95095 | Donald Arthur "Don" Piper | M | 25.0 | 180.0 | 73.0 | United States | USA | 1936 Summer | 1936 | Summer | Berlin | Basketball | Basketball Men's Basketball | Gold | USA | Na |
| **70856** | 36368 | Joseph Cephis "Joe" Fortenberry | M | 25.0 | 203.0 | 84.0 | United States | USA | 1936 Summer | 1936 | Summer | Berlin | Basketball | Basketball Men's Basketball | Gold | USA | Na |
| **78419** | 40143 | John Haskell "Tex" Gibbons | M | 28.0 | 185.0 | 79.0 | United States | USA | 1936 Summer | 1936 | Summer | Berlin | Basketball | Basketball Men's Basketball | Gold | USA | Na |
| **13614** | 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 | Na |
| **231688** | 117072 | Duane Alexander Swanson | M | 22.0 | 188.0 | 79.0 | United States | USA | 1936 Summer | 1936 | Summer | Berlin | Basketball | Basketball Men's Basketball | Gold | USA | Na |
| **217568** | 110112 | Carl Leslie Shy | M | 27.0 | 183.0 | 77.0 | United States | USA | 1936 Summer | 1936 | Summer | Berlin | Basketball | Basketball Men's Basketball | Gold | USA | Na |
| **120814** | 61570 | Carl Stanley Knowles | M | 26.0 | 188.0 | 75.0 | United States | USA | 1936 Summer | 1936 | Summer | Berlin | Basketball | Basketball Men's Basketball | Gold | USA | Na |

| | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **200428** | 101443 | Robert Lloyd Jackson "Jack" Robinson | M | 21.0 | 183.0 | 82.0 | United States | USA | 1948 Summer | 1948 | Summer | London | Basketball | Basketball Men's Basketball | Gold | USA | Na |

What I suspected was true, I was actually getting the count of team, as we know there are multiple players involved in a basketball team. Now I need to group the data based on year again to check the first record for each member of team

In [73]:

```
bskt_fnl_us = basket_us_gld_mdls.groupby(['year']).first()
bskt_fnl_us.head(5)
```

Out[73]:

| year | id | name | sex | age | height | weight | team | noc | games | season | city | sport | event | medal | region | notes |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **1936** | 98309 | Jack Williamson Ragland | M | 22.0 | 183.0 | 79.0 | United States | USA | 1936 Summer | Summer | Berlin | Basketball | Basketball Men's Basketball | Gold | USA | None |
| **1948** | 101443 | Robert Lloyd Jackson "Jack" Robinson | M | 21.0 | 183.0 | 82.0 | United States | USA | 1948 Summer | Summer | London | Basketball | Basketball Men's Basketball | Gold | USA | None |
| **1952** | 130602 | Howard Earl "Howie" Williams | M | 24.0 | 183.0 | 76.0 | United States | USA | 1952 Summer | Summer | Helsinki | Basketball | Basketball Men's Basketball | Gold | USA | None |
| **1956** | 17312 | Carl Cecil Cain | M | 22.0 | 190.0 | 86.0 | United States | USA | 1956 Summer | Summer | Melbourne | Basketball | Basketball Men's Basketball | Gold | USA | None |
| **1960** | 9670 | Walter Jones "Walt" Bellamy, Jr. | M | 23.0 | 211.0 | 98.0 | United States | USA | 1960 Summer | Summer | Roma | Basketball | Basketball Men's Basketball | Gold | USA | None |

Now the data looks more accurate, Let me count the number of record now

In [74]:

```
bskt_fnl_us['id'].count()
```

Out[74]:

```
15
```

So we have recieved required results from the above, also confirmed with Wikipedia

## Median Height & Weight of Olympics Medalists

In [75]:

```
gold_medalists.head()
```

Out[75]:

| | id | name | sex | age | height | weight | team | noc | games | year | season | city | sport | event | medal | region | note |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **38164** | 19760 | Chang Hye-Jin | F | 29.0 | 158.0 | 50.0 | South Korea | KOR | 2016 Summer | 2016 | Summer | Rio de Janeiro | Archery | Archery Women's Individual | Gold | South Korea | Na |
| **38165** | 19760 | Chang Hye-Jin | F | 29.0 | 158.0 | 50.0 | South Korea | KOR | 2016 Summer | 2016 | Summer | Rio de Janeiro | Archery | Archery Women's Team | Gold | South Korea | Na |
| **10150** | 5561 | Nickel Ashmeade | M | 26.0 | 183.0 | 77.0 | Jamaica | JAM | 2016 Summer | 2016 | Summer | Rio de Janeiro | Athletics | Athletics Men's 4 x 100 metres Relay | Gold | Jamaica | Na |
| **242984** | 122548 | Josua Tuisova Ratulevu | M | 22.0 | 180.0 | 108.0 | Fiji | FIJ | 2016 Summer | 2016 | Summer | Rio de Janeiro | Rugby Sevens | Rugby Sevens Men's Rugby Sevens | Gold | Fiji | Na |
| **243000** | 122556 | Blair Tuke | M | 27.0 | 181.0 | 78.0 | New Zealand | NZL | 2016 Summer | 2016 | Summer | Rio de Janeiro | Sailing | Sailing Men's Skiff | Gold | New Zealand | Na |

In [76]:

```
gold_medalists.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 13366 entries, 38164 to 69539
Data columns (total 17 columns):
 #   Column  Non-Null Count  Dtype   
---  ------  --------------  -----   
 0   id      13366 non-null  int64   
 1   name    13366 non-null  object  
 2   sex     13366 non-null  category
 3   age     13366 non-null  float64 
 4   height  13366 non-null  float64 
 5   weight  13366 non-null  float64 
 6   team    13366 non-null  category
 7   noc     13366 non-null  category
 8   games   13366 non-null  category
 9   year    13366 non-null  int32   
 10  season  13366 non-null  category
 11  city    13366 non-null  object  
 12  sport   13366 non-null  object  
 13  event   13366 non-null  category
 14  medal   13366 non-null  category
 15  region  13365 non-null  object  
 16  notes   171 non-null    object  
dtypes: category(7), float64(3), int32(1), int64(1), object(5)
memory usage: 1.8+ MB
```

Here, I have got more than 13k rows. I only wanna get columns containig values.

In [77]:

```
new_gold_mdls = gold_medalists[(gold_medalists['height'].notnull()) & (gold_medalists['weight'].notnull())]
```

In [78]:

```
new_gold_mdls.isnull().any()
```

Out[78]:

```
id        False
name      False
sex       False
age       False
height    False
weight    False
team      False
noc       False
games     False
year      False
season    False
city      False
sport     False
event     False
medal     False
region     True
notes      True
dtype: bool
```

In [79]:

```
new_gold_mdls.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 13366 entries, 38164 to 69539
Data columns (total 17 columns):
 #   Column  Non-Null Count  Dtype   
---  ------  --------------  -----   
 0   id      13366 non-null  int64   
 1   name    13366 non-null  object  
 2   sex     13366 non-null  category
 3   age     13366 non-null  float64 
 4   height  13366 non-null  float64 
 5   weight  13366 non-null  float64 
 6   team    13366 non-null  category
 7   noc     13366 non-null  category
 8   games   13366 non-null  category
 9   year    13366 non-null  int32   
 10  season  13366 non-null  category
 11  city    13366 non-null  object  
 12  sport   13366 non-null  object  
 13  event   13366 non-null  category
 14  medal   13366 non-null  category
 15  region  13365 non-null  object  
 16  notes   171 non-null    object
dtypes: category(7), float64(3), int32(1), int64(1), object(5)
memory usage: 1.3+ MB
```

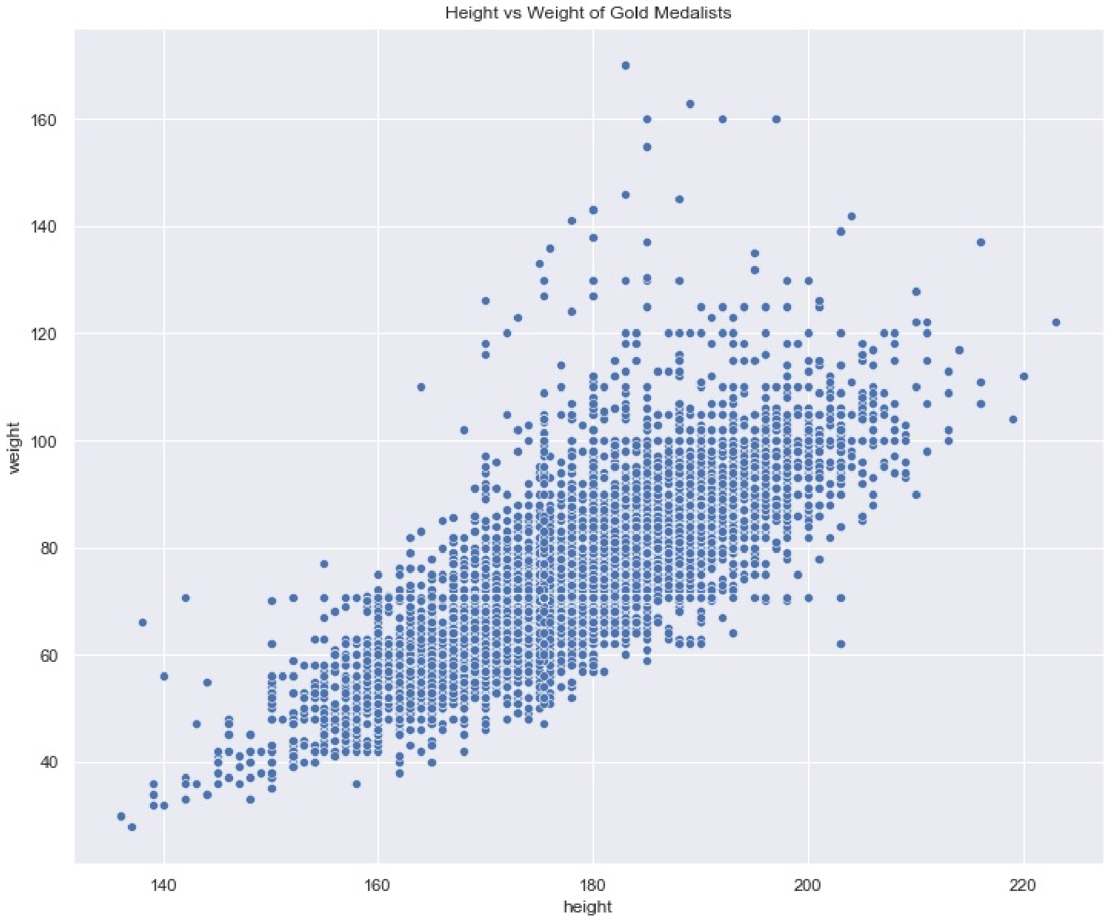
Now I have just above 10k columns, I need to create a scatterplot next to visualize relationships

In [80]:

```
plt.figure(figsize=(12, 10))
ax = sns.scatterplot(x="height", y="weight", data=new_gold_mdls)
plt.title('Height vs Weight of Gold Medalists')
```

Out[80]:

```
Text(0.5, 1.0, 'Height vs Weight of Gold Medalists')
```



The majority of dots represents a linear relatioship between height and weight. As the hieght increases weight also increases.

I just wanna see the number of athletes with more than 150kg weight

In [81]:

```
over_weight = new_gold_mdls.loc[new_gold_mdls['weight'] > 150]
over_weight
```

Out[81]:

| | id | name | sex | age | height | weight | team | noc | games | year | season | city | sport | event | medal | reg |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **233745** | 118074 | Lasha Talakhadze | M | 22.0 | 197.0 | 160.0 | Georgia | GEO | 2016 Summer | 2016 | Summer | Rio de Janeiro | Weightlifting | Weightlifting Men's Super-Heavyweight | Gold | Geo |
| **207047** | 104740 | Behdad Salimi Kordasiabi | M | 22.0 | 192.0 | 160.0 | Iran | IRI | 2012 Summer | 2012 | Summer | London | Weightlifting | Weightlifting Men's Super-Heavyweight | Gold | |
| **198117** | 100282 | Hossein Reza Zadeh | M | 26.0 | 185.0 | 155.0 | Iran | IRI | 2004 Summer | 2004 | Summer | Athina | Weightlifting | Weightlifting Men's Super-Heavyweight | Gold | |
| **198116** | 100282 | Hossein Reza Zadeh | M | 22.0 | 185.0 | 155.0 | Iran | IRI | 2000 Summer | 2000 | Summer | Sydney | Weightlifting | Weightlifting Men's Super-Heavyweight | Gold | |
| **38909** | 20144 | Andrey Ivanovich | M | 24.0 | 183.0 | 170.0 | Russia | RUS | 1996 Summer | 1996 | Summer | Atlanta | Weightlifting | Weightlifting Men's Super- | Gold | Rus |

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Chemerkin | | | | | | | | | | | | Heavyweight | | |
| **4400** | 2511 | Vasily Ivanovich Alekseyev | M | 34.0 | 185.0 | 160.0 | Soviet Union | URS | 1976 Summer | 1976 | Summer | Montreal | Weightlifting | Weightlifting Men's Super-Heavyweight | Gold | Rus |
| **4399** | 2511 | Vasily Ivanovich Alekseyev | M | 30.0 | 185.0 | 160.0 | Soviet Union | URS | 1972 Summer | 1972 | Summer | Munich | Weightlifting | Weightlifting Men's Super-Heavyweight | Gold | Rus |
| **266673** | 134407 | Leonid Ivanovych Zhabotynskiy | M | 30.0 | 189.0 | 163.0 | Soviet Union | URS | 1968 Summer | 1968 | Summer | Mexico City | Weightlifting | Weightlifting Men's Heavyweight | Gold | Rus |
| **266672** | 134407 | Leonid Ivanovych Zhabotynskiy | M | 26.0 | 189.0 | 163.0 | Soviet Union | URS | 1964 Summer | 1964 | Summer | Tokyo | Weightlifting | Weightlifting Men's Heavyweight | Gold | Rus |

In [82]:

```
over_weight['id'].count()
```

Out[82]:

```
9
```

Only 9 Athletes are above 150KG, That's quite interesting as well. I am moving forward with further analysis to know more

## Let's study the Age, Height & Weight relatioship with Medal Winning's

I am only taking medal winners of United States as they got the highest number of medals in the history of olympics. to get more accurate outcome, also It is the ultimate goal of this research to get the best possible outcomes.

Moreover I am not comparing sex and win in this part as Olympics has seperate competitions for Male and Female. So no point in studying winning chance based on sex.

In [83]:

```
data.head()
```

Out[83]:

| | id | name | sex | age | height | weight | team | noc | games | year | season | city | sport | event | medal | reg |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **194731** | 98525 | Maksim Sergeyevich Rakov | M | 30.0 | 181.0 | 100.0 | Kazakhstan | KAZ | 2016 Summer | 2016 | Summer | Rio de Janeiro | Judo | Judo Men's Half-Heavyweight | NaN | Kazakhs |
| **162035** | 82008 | Alexandra Patricia "Alex" Morgan | F | 27.0 | 173.0 | 62.0 | United States | USA | 2016 Summer | 2016 | Summer | Rio de Janeiro | Football | Football Women's Football | NaN | U |
| **117646** | 59981 | Kim Hyeon-Seop | M | 31.0 | 175.0 | 53.0 | South Korea | KOR | 2016 Summer | 2016 | Summer | Rio de Janeiro | Athletics | Athletics Men's 20 kilometres Walk | NaN | Sc Kc |
| **117647** | 59981 | Kim Hyeon-Seop | M | 31.0 | 175.0 | 53.0 | South Korea | KOR | 2016 Summer | 2016 | Summer | Rio de Janeiro | Athletics | Athletics Men's 50 kilometres Walk | NaN | Sc Kc |
| **38150** | 19756 | Chang Hao | M | 25.0 | 173.0 | 72.0 | Chinese Taipei | TPE | 2016 Summer | 2016 | Summer | Rio de Janeiro | Sailing | Sailing Men's Windsurfer | NaN | Tai |

In the below code I have used us_data as daataframe name because I initially decided to study US data then later changed it I see there is no much relationship going on, So later I did not get a chance to change everywhere. Please ignore the name for this reason

In [84]:

```
#us_data = data[(data.team == 'United States')]
us_data = data
us_data.tail()
```

Out[84]:

| | id | name | sex | age | height | weight | team | noc | games | year | season | city | sport | event | medal | regi |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **96389** | 49185 | Fritz Hofmann | M | 24.0 | 167.000000 | 56.000000 | Germany | GER | 1896 Summer | 1896 | Summer | Athina | Athletics | Athletics Men's Triple Jump | NaN | Germa |
| **96390** | 49185 | Fritz Hofmann | M | 24.0 | 167.000000 | 56.000000 | Germany | GER | 1896 Summer | 1896 | Summer | Athina | Athletics | Athletics Men's Shot Put | NaN | Germa |
| | | | | | | | | | | | | | | Shooting Men's | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **54562** | 28169 | Georgios Diamantis | M | 24.0 | 175.338941 | 70.701618 | Greece | GRE | 1896 Summer | 1896 | Summer | Athina | Shooting | Military Rifle, 200 metres | NaN | Gree |
| **135986** | 68911 | Albin Georges Lermusiaux | M | 21.0 | 175.338941 | 70.701618 | France | FRA | 1896 Summer | 1896 | Summer | Athina | Athletics | Athletics Men's Marathon | NaN | Fran |
| **69539** | 35698 | Edwin Harold "Teddy" Flack | M | 22.0 | 175.338941 | 70.701618 | Australia | AUS | 1896 Summer | 1896 | Summer | Athina | Athletics | Athletics Men's 1,500 metres | Gold | Austra |

In [85]:

```
us_data = us_data.sort_values('year', ascending=False).drop_duplicates(subset=['year', 'sport'])
```

In [86]:

```
#us_data = us_data[us_data.medal.notnull()]
```

In [87]:

```
us_data.head()
```

Out[87]:

|  | id | name | sex | age | height | weight | team | noc | games | year | season | city | sport | event | medal |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **194731** | 98525 | Maksim Sergeyevich Rakov | M | 30.0 | 181.0 | 100.0 | Kazakhstan | KAZ | 2016 Summer | 2016 | Summer | Rio de Janeiro | Judo | Judo Men's Half-Heavyweight | NaN |
| **67487** | 34748 | David Rubn Sousa Fernandes | M | 32.0 | 181.0 | 82.0 | Portugal | POR | 2016 Summer | 2016 | Summer | Rio de Janeiro | Canoeing | Canoeing Men's Kayak Fours, 1,000 metres | NaN |
| **198600** | 100515 | Jonelle Richards-Price | F | 35.0 | 162.0 | 57.0 | New Zealand | NZL | 2016 Summer | 2016 | Summer | Rio de Janeiro | Equestrianism | Equestrianism Mixed Three-Day Event, Team | NaN |
| **67483** | 34745 | Bruno Miguel Borges Fernandes | M | 21.0 | 182.0 | 80.0 | Portugal | POR | 2016 Summer | 2016 | Summer | Rio de Janeiro | Football | Football Men's Football | NaN |
| **79381** | 40611 | Niccol Gitto | M | 29.0 | 190.0 | 90.0 | Italy | ITA | 2016 Summer | 2016 | Summer | Rio de Janeiro | Water Polo | Water Polo Men's Water Polo | Bronze |

I need to filter only few variables which are required for my core study, like Height, Weight, Age, Sex. In this study I wanna consider all the three medals as a win for more efficient analysis.

In [88]:

```
#us_data.loc["sex", "height", "weight", "medal"]
columns = ['year','season','sex', 'age', 'height', 'weight','team','event','medal','games','sport']
us_data = pd.DataFrame(us_data, columns=columns)
us_data.tail()
```

Out[88]:

|  | year | season | sex | age | height | weight | team | event | medal | games | sport |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **183735** | 1896 | Summer | M | 24.0 | 175.338941 | 70.701618 | Greece | Swimming Men's 500 metres Freestyle | Silver | 1896 Summer | Swimming |
| **111156** | 1896 | Summer | M | 24.0 | 175.338941 | 70.701618 | Hungary | Gymnastics Men's Horizontal Bar | NaN | 1896 Summer | Gymnastics |
| **36266** | 1896 | Summer | M | 23.0 | 175.338941 | 70.701618 | Greece | Tennis Men's Singles | Silver | 1896 Summer | Tennis |
| **212743** | 1896 | Summer | M | 26.0 | 159.000000 | 70.000000 | Germany | Wrestling Men's Unlimited Class, Greco-Roman | Gold | 1896 Summer | Wrestling |
| **250060** | 1896 | Summer | M | 24.0 | 175.338941 | 70.701618 | Greece | Weightlifting Men's Unlimited, One Hand | NaN | 1896 Summer | Weightlifting |

In [89]:

```
us_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 906 entries, 194731 to 250060
Data columns (total 11 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   year    906 non-null    int32
 1   season  906 non-null    category
 2   sex     906 non-null    category
 3   age     906 non-null    float64
 4   height  906 non-null    float64
 5   weight  906 non-null    float64
 6   team    906 non-null    category
 7   event   906 non-null    category
 8   medal   180 non-null    category
 9   games   906 non-null    category
 10  sport   906 non-null    object
dtypes: category(6), float64(3), int32(1), object(1)
memory usage: 112.4+ KB
```

In [153…

```
data.describe(include='all')
```

Out[153…

| | id | name | sex | age | height | weight | team | noc | games | year | season | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **count** | 269119.000000 | 269119 | 269119 | 269119.000000 | 269119.000000 | 269119.000000 | 269119 | 269119 | 269119 | 269119.000000 | 269119 | 26 |
| **unique** | NaN | 134415 | 2 | NaN | NaN | NaN | 1164 | 230 | 51 | NaN | 2 | |
| **top** | NaN | Heikki Ilmari Savolainen | M | NaN | NaN | NaN | United States | USA | 2000 Summer | NaN | Summer | Lo |
| **freq** | NaN | 39 | 194796 | NaN | NaN | NaN | 17508 | 18514 | 13821 | NaN | 220555 | 2 |
| **mean** | 68269.049279 | NaN | NaN | 25.366180 | 175.338941 | 70.701618 | NaN | NaN | NaN | 1978.735236 | NaN | |
| **std** | 39028.599815 | NaN | NaN | 5.977013 | 9.311661 | 12.620936 | NaN | NaN | NaN | 29.688086 | NaN | |
| **min** | 1.000000 | NaN | NaN | 10.000000 | 127.000000 | 25.000000 | NaN | NaN | NaN | 1896.000000 | NaN | |
| **25%** | 34656.000000 | NaN | NaN | 22.000000 | 170.000000 | 63.000000 | NaN | NaN | NaN | 1960.000000 | NaN | |
| **50%** | 68244.000000 | NaN | NaN | 24.000000 | 175.338941 | 70.701618 | NaN | NaN | NaN | 1988.000000 | NaN | |
| **75%** | 102117.500000 | NaN | NaN | 28.000000 | 180.000000 | 76.000000 | NaN | NaN | NaN | 2004.000000 | NaN | |
| **max** | 135571.000000 | NaN | NaN | 97.000000 | 226.000000 | 214.000000 | NaN | NaN | NaN | 2016.000000 | NaN | |

In [154…

```
data['sex'].describe()
```

Out[154…

```
count     269119
unique         2
top            M
freq      194796
Name: sex, dtype: object
```

In [155…

```
data['age'].describe()
```

Out[155…

```
count    269119.000000
mean         25.366180
std           5.977013
min          10.000000
25%          22.000000
50%          24.000000
75%          28.000000
max          97.000000
Name: age, dtype: float64
```

In [156…

```
data['height'].describe()
```

Out[156…

```
count    269119.000000
mean        175.338941
std           9.311661
min         127.000000
25%         170.000000
50%         175.338941
75%         180.000000
max         226.000000
Name: height, dtype: float64
```

In [157…

```
data['weight'].describe()
```

Out[157…

```
count    269119.000000
```

```
mean         70.701618
std          12.620936
min          25.000000
25%          63.000000
50%          70.701618
75%          76.000000
max         214.000000
Name: weight, dtype: float64
```

In [90]:

```
us_data['win'] = us_data['medal'].notnull()*1
us_data['win']
us_data.tail(10)
```

Out[90]:

| | year | season | sex | age | height | weight | team | event | medal | games | sport | win |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **49932** | 1900 | Summer | M | 26.0 | 175.338941 | 70.701618 | Spain | Basque Pelota Men's Two-Man Teams With Cesta | Gold | 1900 Summer | Basque Pelota | 1 |
| **182196** | 1896 | Summer | M | 24.0 | 175.338941 | 70.701618 | Greece | Shooting Men's Military Rifle, 200 metres | Silver | 1896 Summer | Shooting | 1 |
| **211506** | 1896 | Summer | M | 23.0 | 175.338941 | 70.701618 | Austria | Fencing Men's Sabre, Individual | NaN | 1896 Summer | Fencing | 0 |
| **80415** | 1896 | Summer | M | 21.0 | 175.338941 | 70.701618 | Greece | Athletics Men's 1,500 metres | NaN | 1896 Summer | Athletics | 0 |
| **211505** | 1896 | Summer | M | 23.0 | 175.338941 | 70.701618 | Austria | Cycling Men's 333 metres Time Trial | Bronze | 1896 Summer | Cycling | 1 |
| **183735** | 1896 | Summer | M | 24.0 | 175.338941 | 70.701618 | Greece | Swimming Men's 500 metres Freestyle | Silver | 1896 Summer | Swimming | 1 |
| **111156** | 1896 | Summer | M | 24.0 | 175.338941 | 70.701618 | Hungary | Gymnastics Men's Horizontal Bar | NaN | 1896 Summer | Gymnastics | 0 |
| **36266** | 1896 | Summer | M | 23.0 | 175.338941 | 70.701618 | Greece | Tennis Men's Singles | Silver | 1896 Summer | Tennis | 1 |
| **212743** | 1896 | Summer | M | 26.0 | 159.000000 | 70.000000 | Germany | Wrestling Men's Unlimited Class, Greco-Roman | Gold | 1896 Summer | Wrestling | 1 |
| **250060** | 1896 | Summer | M | 24.0 | 175.338941 | 70.701618 | Greece | Weightlifting Men's Unlimited, One Hand | NaN | 1896 Summer | Weightlifting | 0 |

In [91]:

```
#us_data['medal'] = us_data['medal'].astype('str')
#us_data['medal'] = us_data['medal'].replace(np.nan, 0)
#us_data['medal'] = us_data['medal'].astype('category')
#us_data.tail()
```

In [92]:

```
us_data['win'] = us_data['win'].astype('int')
us_data['medal'] = us_data['medal'].astype('category')
#us_data['medal'] = us_data['medal'].astype('str')
us_data.dtypes
```

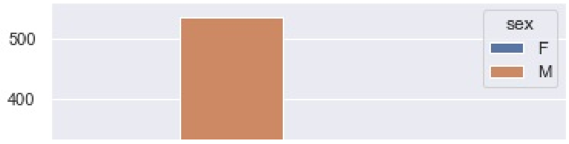
Out[92]:

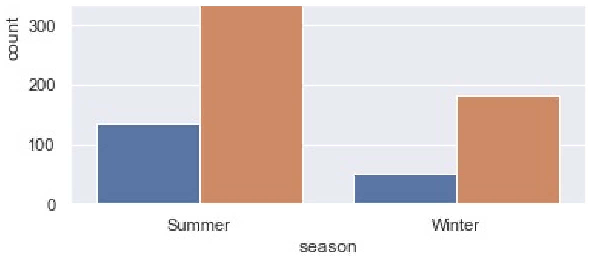
```
year          int32
season     category
sex        category
age         float64
height      float64
weight      float64
team       category
event      category
medal      category
games      category
sport        object
win           int32
dtype: object
```

Comparing between season and sex

In [93]:

```
sns.countplot(x='season',hue='sex',data=us_data)
plt.show()
```





From the above graph It's clear to see more men are participating in both seasons than women and also the ration is almost same in both seasons.

Best way to check 2 categorical varible is to make cross tab. seems to be there is no much relationship going on between season and sex.

In [94]:

```
sex_season = pd.crosstab(index=us_data['sex'], columns=us_data["season"],
margins=True)
sex_season
```

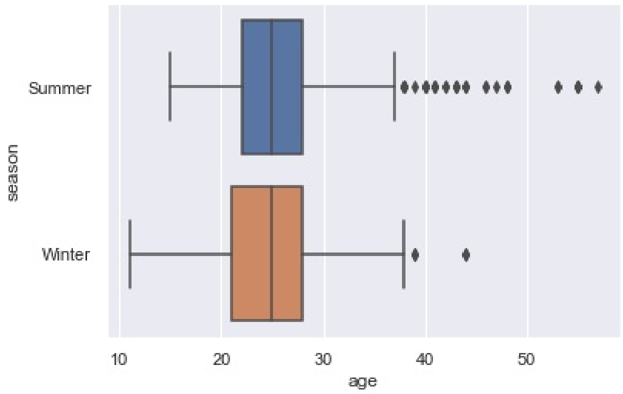
Out[94]:

| season | Summer | Winter | All |
|---|---|---|---|
| sex | | | |
| F | 136 | 51 | 187 |
| M | 536 | 183 | 719 |
| All | 672 | 234 | 906 |

Comparing between season and age

In [95]:

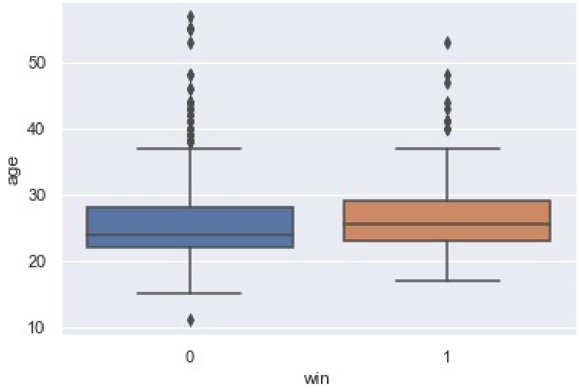
```
sns.boxplot(x='age',y='season',data=us_data)
plt.show()
```



Medians are almost same and IQR's are completely overlapping. no much relation going on

In [96]:

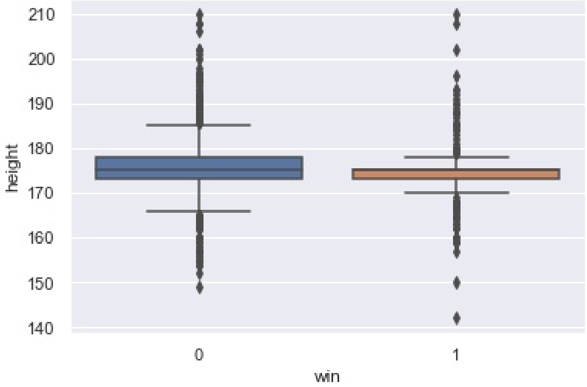
```
sns.boxplot(x="win",y="age", data=us_data)
plt.show()
```



The median age for win is more that median age of no-win. Also IQR's are not completely overlapping. I think there must be a relation between age and win

In [97]:

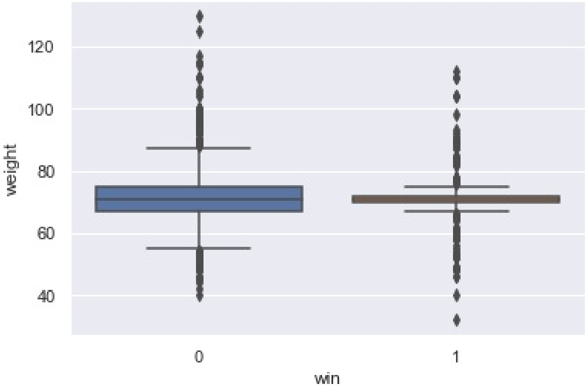
```
sns.boxplot(x="win",y="height", data=us_data)
plt.show()
```



Again the median for win and no win seems to be almost equal. And the IQR's are completely overlapping. Could not find any relation going on here, also it seems to have most winnings lies between 168 to 184 Height.

In [98]:

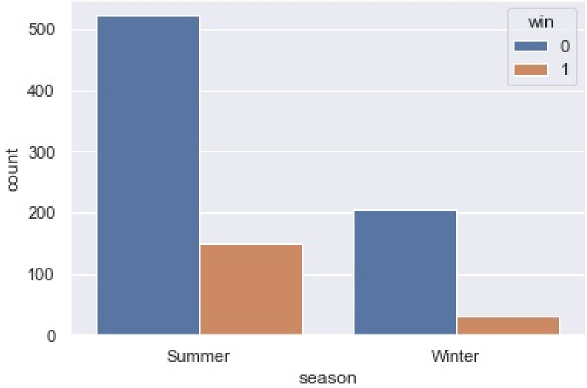
```
sns.boxplot(x="win",y="weight", data=us_data)
plt.show()
```



Between Sex and Win , There is no much relatioship going on here. As both IQR's are completely overlapping and also medians are same for both.

In [99]:

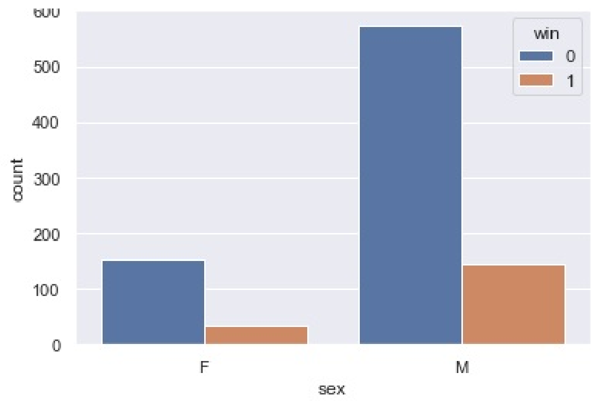
```
sns.countplot(x='season',hue='win',data=us_data)
plt.show()
```



Between Seaon and Win , There is no much relatioship going on here. As both counts are showing equal proportions.

In [100…

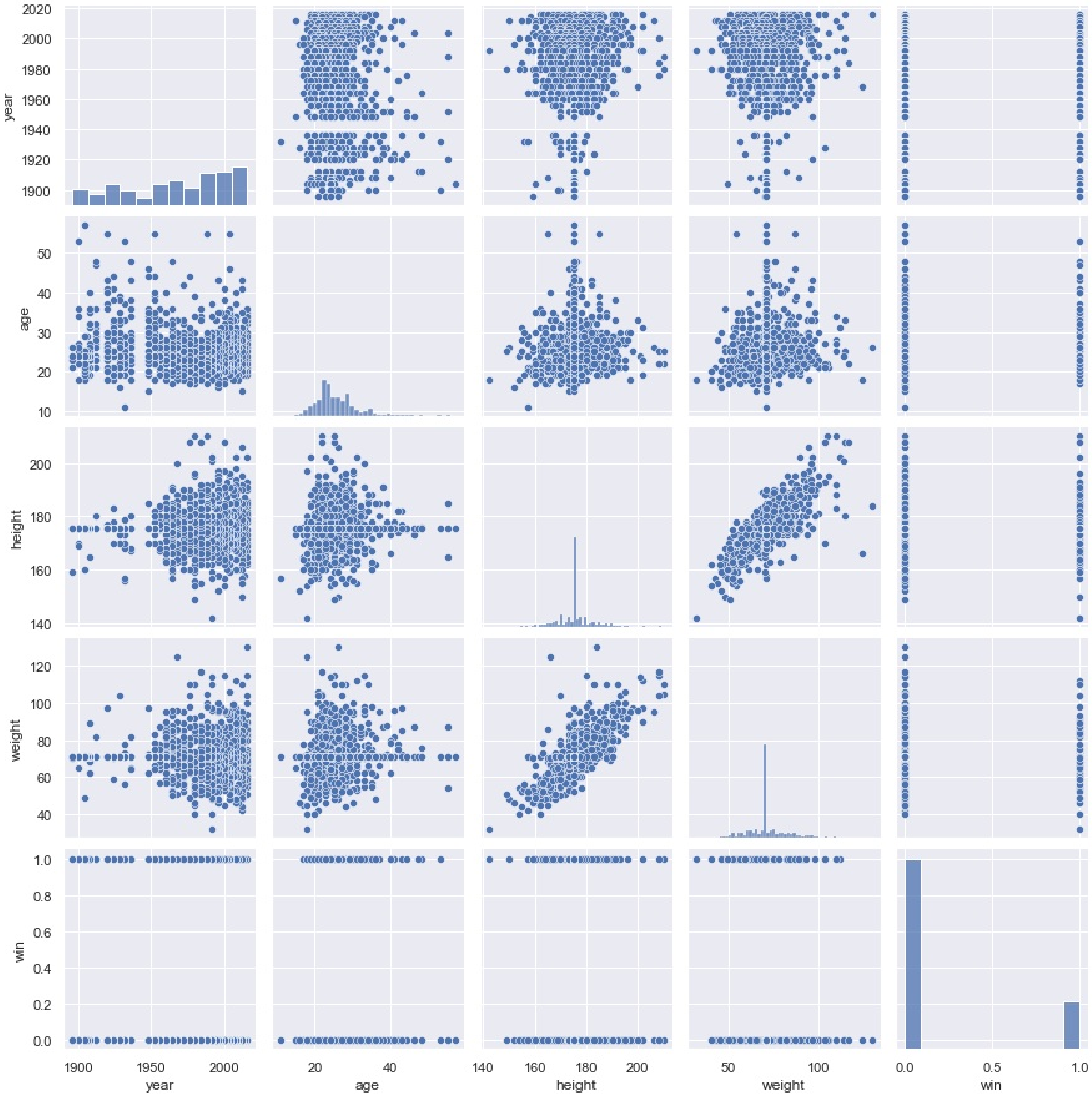
```
sns.countplot(x='sex',hue='win',data=us_data)
plt.show()
```



In this graph we can see for male and female win rates are in equal proportion. no relationship much found between them.

In [101…

```
#multivariate
sns.pairplot(us_data)
plt.show()
```

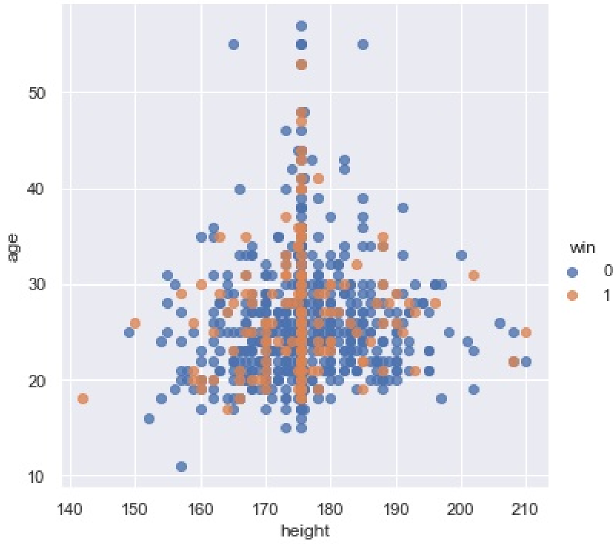


From the above graph I can see there is only one linear relation going on between height and weight

In [102…

```
sns.lmplot(x='height',y='age',hue='win',data=us_data,fit_reg=False)
```

```
plt.show()
```



No much relationship found between age height & win. spead was almost rough. But I can see a clustering happening in 175 Height. I believe most of the US athletes participated in Olympics are of 175cm height.

In [103...

```
us_data = pd.DataFrame(us_data)
rank = us_data['medal']
us_data['rank'] = rank
us_data['rank'] = us_data['rank'].replace(['Gold','Silver','Bronze'], [1, 2, 3])
```

In [104...

```
us_data['rank'] = us_data['rank'].replace('?', np.NaN)

us_data['games'] = us_data['games'].astype('str')
```

In [105...

```
us_data['rank'].head()
```

Out[105...

```
194731    NaN
67487     NaN
198600    NaN
67483     NaN
79381     3.0
Name: rank, dtype: float64
```

In [106...

```
us_data.dtypes
```

Out[106...

```
year         int32
season    category
sex       category
age        float64
height     float64
weight     float64
team      category
event     category
medal     category
games       object
sport       object
win          int32
rank       float64
dtype: object
```

In [107...

```
us_data.head(15)
```

Out[107...

| | year | season | sex | age | height | weight | team | event | medal | games | sport | win | rank |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **194731** | 2016 | Summer | M | 30.0 | 181.0 | 100.0 | Kazakhstan | Judo Men's Half-Heavyweight | NaN | 2016 Summer | Judo | 0 | NaN |
| **67487** | 2016 | Summer | M | 32.0 | 181.0 | 82.0 | Portugal | Canoeing Men's Kayak Fours, 1,000 metres | NaN | 2016 Summer | Canoeing | 0 | NaN |
| | | | | | | | New | Equestrianism Mixed Three-Day | | 2016 | | | |

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **198600** | 2016 | Summer | F | 35.0 | 162.0 | 57.0 | Zealand | Event, Team | NaN | Summer | Equestrianism | 0 | NaN |
| **67483** | 2016 | Summer | M | 21.0 | 182.0 | 80.0 | Portugal | Football Men's Football | NaN | 2016 Summer | Football | 0 | NaN |
| **79381** | 2016 | Summer | M | 29.0 | 190.0 | 90.0 | Italy | Water Polo Men's Water Polo | Bronze | 2016 Summer | Water Polo | 1 | 3.0 |
| **79395** | 2016 | Summer | M | 25.0 | 185.0 | 82.0 | Canada | Badminton Men's Singles | NaN | 2016 Summer | Badminton | 0 | NaN |
| **25775** | 2016 | Summer | M | 28.0 | 180.0 | 67.0 | Belgium | Athletics Men's 4 x 400 metres Relay | NaN | 2016 Summer | Athletics | 0 | NaN |
| **189295** | 2016 | Summer | F | 21.0 | 163.0 | 55.0 | Italy | Swimming Women's 200 metres Butterfly | NaN | 2016 Summer | Swimming | 0 | NaN |
| **189365** | 2016 | Summer | F | 30.0 | 180.0 | 84.0 | United States | Rowing Women's Coxed Eights | Gold | 2016 Summer | Rowing | 1 | 1.0 |
| **28742** | 2016 | Summer | M | 26.0 | 167.0 | 60.0 | Germany | Gymnastics Men's Individual All-Around | NaN | 2016 Summer | Gymnastics | 0 | NaN |
| **198427** | 2016 | Summer | M | 33.0 | 173.0 | 94.0 | Australia | Weightlifting Men's Middle-Heavyweight | NaN | 2016 Summer | Weightlifting | 0 | NaN |
| **67546** | 2016 | Summer | M | 33.0 | 180.0 | 115.0 | Spain | Shooting Men's Trap | NaN | 2016 Summer | Shooting | 0 | NaN |
| **197521** | 2016 | Summer | M | 19.0 | 202.0 | 90.0 | Cuba | Volleyball Men's Volleyball | NaN | 2016 Summer | Volleyball | 0 | NaN |
| **191011** | 2016 | Summer | F | 28.0 | 173.0 | 70.0 | New Zealand | Sailing Women's Two Person Dinghy | Silver | 2016 Summer | Sailing | 1 | 2.0 |
| **265554** | 2016 | Summer | M | 25.0 | 186.0 | 80.0 | Australia | Hockey Men's Hockey | NaN | 2016 Summer | Hockey | 0 | NaN |

In [108…

```
us_data.dtypes
```

Out[108…

```
year          int32
season     category
sex        category
age         float64
height      float64
weight      float64
team       category
event      category
medal      category
games        object
sport        object
win           int32
rank        float64
dtype: object
```

In [109…

```
us_data['rank'].isnull().sum().sum()
```

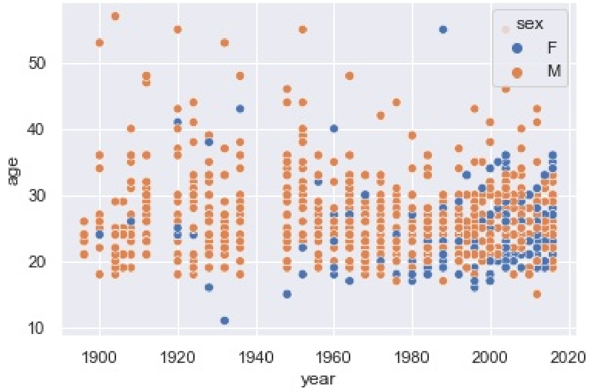
Out[109…

```
726
```

As I already made graphs above between different variables, I can refer them to decide between their replationship

In [110…

```
sns.scatterplot(x='year',y='age',hue='sex',data=us_data)
plt.show()
```

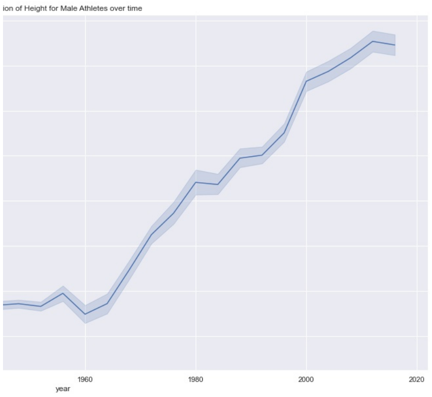


From the above graph random spread has been found, no clustering identified

```
)
nt', data=men_over_time)
ight for Male Athletes over time')
```

```
o\site-packages\seaborn\_decorators.
the following variables as keyword a
2, the only valid positional argumen
ng other arguments without an explic
n error or misinterpretation.
```

```
f Height for Male Athletes over time
```



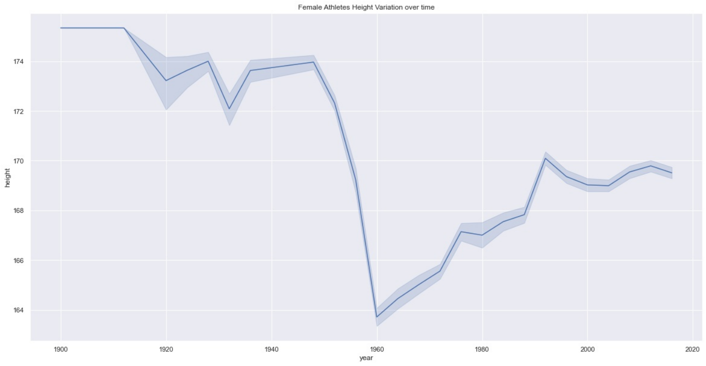
In [142…

```
plt.figure(figsize=(20, 10))
sns.lineplot('year', 'height', data=women_over_time)
plt.title('Female Athletes Height Variation over time')
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decora
tors.py:36: FutureWarning: Pass the following variables as
keyword args: x, y. From version 0.12, the only valid posi
tional argument will be `data`, and passing other argument
s without an explicit keyword will result in an error or m
isinterpretation.
  warnings.warn(
```

Out[142…

```
Text(0.5, 1.0, 'Female Athletes Height Variation over time
')
```



It's
clea
fro
the
dat
tha
me
hei
inc
ove
tim
for
fem
ath
the
was
a
dec
in
hei
fro
190
to
196
the
it
gra
inc