Chp2_challenge_practice

March 13, 2022

1 Chapter2

```
[1]: import pandas as pd
```

List our Olympic DataFrame

```
[2]: oo = pd.read_csv('../data/olympics.csv', skiprows=4)
     oo.head()
```

```
[2]:      City  Edition     Sport Discipline             Athlete  NOC Gender  \
     0  Athens     1896  Aquatics   Swimming       HAJOS, Alfred  HUN    Men
     1  Athens     1896  Aquatics   Swimming    HERSCHMANN, Otto  AUT    Men
     2  Athens     1896  Aquatics   Swimming   DRIVAS, Dimitrios  GRE    Men
     3  Athens     1896  Aquatics   Swimming  MALOKINIS, Ioannis  GRE    Men
     4  Athens     1896  Aquatics   Swimming  CHASAPIS, Spiridon  GRE    Men

                            Event Event_gender   Medal
     0             100m freestyle            M    Gold
     1             100m freestyle            M  Silver
     2  100m freestyle for sailors            M  Bronze
     3  100m freestyle for sailors            M    Gold
     4  100m freestyle for sailors            M  Silver
```

List Only NOC column using [‘..’] and dot notation. what Type is this object?

```
[4]: oo.NOC
```

```
[4]: 0        HUN
     1        AUT
     2        GRE
     3        GRE
     4        GRE
             …
     29211    GER
     29212    LTU
     29213    ARM
     29214    CUB
     29215    RUS
     Name: NOC, Length: 29216, dtype: object
```

[5]:
```
oo['NOC']
```

[5]:
```
0        HUN
1        AUT
2        GRE
3        GRE
4        GRE
        …
29211    GER
29212    LTU
29213    ARM
29214    CUB
29215    RUS
Name: NOC, Length: 29216, dtype: object
```

[6]:
```
type(oo['NOC'])
```

[6]:
```
pandas.core.series.Series
```

[7]:
```
type(oo.NOC)
```

[7]:
```
pandas.core.series.Series
```

List The Edition, City, Athlete, Medal column. What type is this object

[9]:
```
oo_sub =oo[['Edition','City','Athlete','Medal']]
```

[10]:
```
oo_sub
```

[10]:
```
       Edition     City                Athlete   Medal
0         1896   Athens          HAJOS, Alfred    Gold
1         1896   Athens      HERSCHMANN, Otto  Silver
2         1896   Athens     DRIVAS, Dimitrios  Bronze
3         1896   Athens    MALOKINIS, Ioannis    Gold
4         1896   Athens    CHASAPIS, Spiridon  Silver
…          …      …                     …      …
29211     2008  Beijing        ENGLICH, Mirko  Silver
29212     2008  Beijing  MIZGAITIS, Mindaugas  Bronze
29213     2008  Beijing       PATRIKEEV, Yuri  Bronze
29214     2008  Beijing         LOPEZ, Mijain    Gold
29215     2008  Beijing        BAROEV, Khasan  Silver

[29216 rows x 4 columns]
```

[11]:
```
type(oo_sub)
```

[11]:
```
pandas.core.frame.DataFrame
```

[ ]:

```
```

Ch04_Challenge_practice

March 13, 2022

1 Chapter4

```
[1]: import pandas as pd
```

```
[2]: oo = pd.read_csv('../data/olympics.csv', skiprows=4)
     oo.head()
```

```
[2]:      City  Edition     Sport Discipline             Athlete  NOC Gender  \
     0  Athens     1896  Aquatics   Swimming       HAJOS, Alfred  HUN    Men
     1  Athens     1896  Aquatics   Swimming    HERSCHMANN, Otto  AUT    Men
     2  Athens     1896  Aquatics   Swimming   DRIVAS, Dimitrios  GRE    Men
     3  Athens     1896  Aquatics   Swimming  MALOKINIS, Ioannis  GRE    Men
     4  Athens     1896  Aquatics   Swimming  CHASAPIS, Spiridon  GRE    Men

                            Event Event_gender   Medal
     0             100m freestyle            M    Gold
     1             100m freestyle            M  Silver
     2  100m freestyle for sailors            M  Bronze
     3  100m freestyle for sailors            M    Gold
     4  100m freestyle for sailors            M  Silver
```

1.0.1 In which event did jesse Owens win a medal

```
[14]: jo =oo[(oo.Athlete =='OWENS, Jesse')]
```

```
[15]: jo
```

```
[15]:         City  Edition      Sport Discipline       Athlete  NOC Gender  \
      6427  Berlin     1936  Athletics  Athletics  OWENS, Jesse  USA    Men
      6439  Berlin     1936  Athletics  Athletics  OWENS, Jesse  USA    Men
      6456  Berlin     1936  Athletics  Athletics  OWENS, Jesse  USA    Men
      6523  Berlin     1936  Athletics  Athletics  OWENS, Jesse  USA    Men

                   Event Event_gender Medal
      6427          100m            M  Gold
      6439          200m            M  Gold
      6456  4x100m relay            M  Gold
```

```
6523     long jump            M  Gold
```

[16]:
```
jo.Event.value_counts()
```

[16]:
```
100m            1
200m            1
4x100m relay    1
long jump       1
Name: Event, dtype: int64
```

[5]:
```
oo.Discipline.unique()
```

[5]:
```
array(['Swimming', 'Athletics', 'Cycling Road', 'Cycling Track',
       'Fencing', 'Artistic G.', 'Shooting', 'Tennis', 'Weightlifting',
       'Wrestling Gre-R', 'Water polo', 'Archery', 'Basque Pelota',
       'Cricket', 'Croquet', 'Jumping', 'Football', 'Golf', 'Polo',
       'Rowing', 'Rugby', 'Sailing', 'Tug of War', 'Diving', 'Boxing',
       'Lacrosse', 'Roque', 'Wrestling Free.', 'Hockey', 'Jeu de Paume',
       'Rackets', 'Figure skating', 'Water Motorspor', 'Dressage',
       'Eventing', 'Modern Pentath.', 'Vaulting', 'Ice Hockey',
       'Basketball', 'Canoe / Kayak F', 'Handball', 'Judo', 'Volleyball',
       'Canoe / Kayak S', 'Synchronized S.', 'Rhythmic G.',
       'Table Tennis', 'Badminton', 'Baseball', 'Mountain Bike',
       'Softball', 'Beach volley.', 'Trampoline', 'Taekwondo',
       'Triathlon', 'BMX'], dtype=object)
```

1.0.2 which country has won the most men's gold medals in single badminton over the years

[6]:
```
cont_male_sin_badmnt = oo[(oo.Gender =='Men')& (oo.Medal =='Gold') &(oo.
 ↪Discipline =='Badminton') &(oo.Event =='singles')]
# cont_male_sin_badmnt.Athlete.sort_values()
cont_male_sin_badmnt.Athlete.sort_values()
```

[6]:
```
20045          BUDI KUSUMA, Alan
25734            HIDAYAT, Taufik
21787    HOYER-LARSEN, Poul Erik
23717                JI, Xinpeng
27741                   LIN, Dan
Name: Athlete, dtype: object
```

1.0.3 Which three countries have won the most medals in recent years(from 1984 to 2008)

[23]:
```
rec =oo[(oo.Edition >1984)]
rec.NOC.value_counts().head(3)
```

```
[23]: USA    1504
      AUS     712
      GER     691
      Name: NOC, dtype: int64
```

```
[8]: oo.Sport.unique()
```

```
[8]: array(['Aquatics', 'Athletics', 'Cycling', 'Fencing', 'Gymnastics',
            'Shooting', 'Tennis', 'Weightlifting', 'Wrestling', 'Archery',
            'Basque Pelota', 'Cricket', 'Croquet', 'Equestrian', 'Football',
            'Golf', 'Polo', 'Rowing', 'Rugby', 'Sailing', 'Tug of War',
            'Boxing', 'Lacrosse', 'Roque', 'Hockey', 'Jeu de paume', 'Rackets',
            'Skating', 'Water Motorsports', 'Modern Pentathlon', 'Ice Hockey',
            'Basketball', 'Canoe / Kayak', 'Handball', 'Judo', 'Volleyball',
            'Table Tennis', 'Badminton', 'Baseball', 'Softball', 'Taekwondo',
            'Triathlon'], dtype=object)
```

1.0.4 Display the male gold medal winners for the 100m track and field sprint over the years

List the result starting the most recent years show the olympic city edition, athlete, and the country they represent.

```
[32]: oo_ath = oo[(oo.Gender =='Men')& (oo.Medal =='Gold') &(oo.Event =='100m') ]
      oo_ath.
       ↪sort_values('Edition',ascending=False)[['City','Edition','Athlete','NOC']]
```

```
[32]:                       City  Edition                 Athlete  NOC
      27552               Beijing     2008            BOLT, Usain  JAM
      25539                Athens     2004         GATLIN, Justin  USA
      23521                Sydney     2000        GREENE, Maurice  USA
      21598               Atlanta     1996        BAILEY, Donovan  CAN
      19859             Barcelona     1992       CHRISTIE, Linford  GBR
      18284                 Seoul     1988             LEWIS, Carl  USA
      16794           Los Angeles     1984             LEWIS, Carl  USA
      15374                Moscow     1980           WELLS, Allan  GBR
      14069              Montreal     1976        CRAWFORD, Hasely  TRI
      12902                Munich     1972          BORZOV, Valery  URS
      11865                Mexico     1968        HINES, James Ray  USA
      10823                 Tokyo     1964          HAYES, Robert  USA
      9924                   Rome     1960             HARY, Armin  EUA
      9009   Melbourne / Stockholm     1956  MORROW, Robert Joseph  USA
      8121               Helsinki     1952        REMIGINO, Lindy  USA
      7302                 London     1948      DILLARD, Harrison  USA
      6427                 Berlin     1936           OWENS, Jesse  USA
      5806            Los Angeles     1932            TOLAN, Eddie  USA
      5095              Amsterdam     1928         WILLIAMS, Percy  CAN
      4236                  Paris     1924        ABRAHAMS, Harold  GBR
```

```
2996                 Antwerp    1920        PADDOCK, Charles  USA
2022               Stockholm    1912            CRAIG, Ralph  USA
1191                  London    1908        WALKER, Reginald  RSA
737                 St Louis    1904            HAHN, Archie  USA
231                    Paris    1900         JARVIS, Francis  USA
13                    Athens    1896           BURKE, Thomas  USA
```

```
[ ]:
```

Chapter5_Challenge_practise

March 13, 2022

1 Chapter 5

[1]:
```
import pandas as pd
```

[2]:
```
oo = pd.read_csv('../data/olympics.csv', skiprows=4)
oo.head()
```

[2]:
```
     City  Edition     Sport Discipline             Athlete  NOC Gender  \
0  Athens     1896  Aquatics   Swimming       HAJOS, Alfred  HUN    Men
1  Athens     1896  Aquatics   Swimming    HERSCHMANN, Otto  AUT    Men
2  Athens     1896  Aquatics   Swimming   DRIVAS, Dimitrios  GRE    Men
3  Athens     1896  Aquatics   Swimming  MALOKINIS, Ioannis  GRE    Men
4  Athens     1896  Aquatics   Swimming  CHASAPIS, Spiridon  GRE    Men

                       Event Event_gender   Medal
0             100m freestyle            M    Gold
1             100m freestyle            M  Silver
2  100m freestyle for sailors           M  Bronze
3  100m freestyle for sailors           M    Gold
4  100m freestyle for sailors           M  Silver
```

[3]:
```
import matplotlib.pyplot as plt
```

[4]:
```
%matplotlib inline
```

[5]:
```
import seaborn as sns
```

2 Plot the number of medals acheived by Chinese team(men and women) in Beijing 2008 using Matplotlib and Seaborn

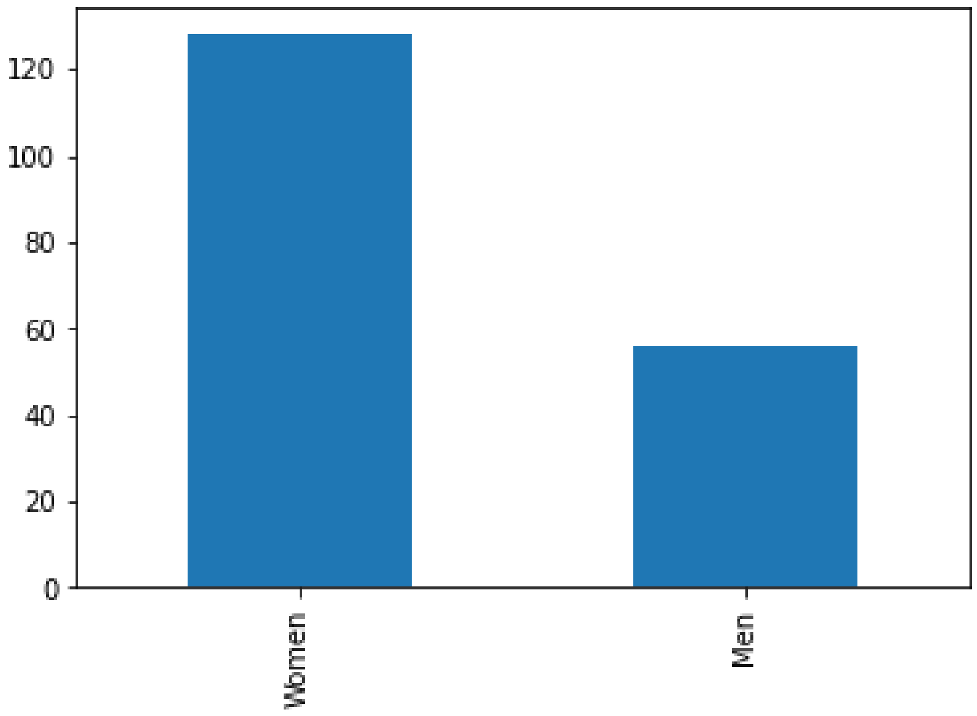
[7]:
```
chn_team = oo[(oo.Edition==2008)&(oo.NOC =='CHN')& (oo.City =='Beijing')]
chn_team
```

[7]:
```
          City  Edition         Sport       Discipline          Athlete  \
27176  Beijing     2008      Aquatics           Diving      ZHOU, Luxin
27177  Beijing     2008      Aquatics           Diving        WANG, Xin
```

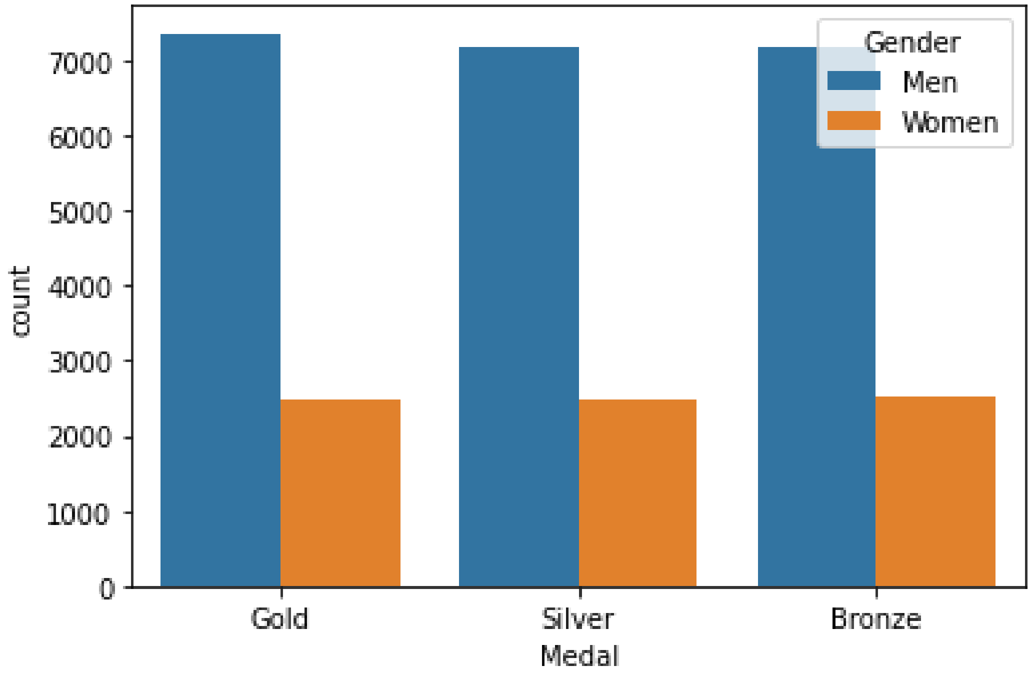
```
27178  Beijing     2008       Aquatics           Diving      CHEN, Ruolin
27180  Beijing     2008       Aquatics           Diving          QIN, Kai
27181  Beijing     2008       Aquatics           Diving         HE, Chong
…        …      …            …              …               …
29138  Beijing     2008  Weightlifting    Weightlifting        LI, Hongli
29140  Beijing     2008  Weightlifting    Weightlifting          LU, Yong
29156  Beijing     2008      Wrestling  Wrestling Free.            XU, Li
29171  Beijing     2008      Wrestling  Wrestling Free.        WANG, Jiao
29204  Beijing     2008      Wrestling  Wrestling Gre-R  CHANG, Yongxiang

       NOC Gender          Event Event_gender   Medal
27176  CHN    Men    10m platform            M  Silver
27177  CHN  Women    10m platform            W  Bronze
27178  CHN  Women    10m platform            W    Gold
27180  CHN    Men  3m springboard            M  Bronze
27181  CHN    Men  3m springboard            M    Gold
…    …    …             …          …     …
29138  CHN    Men            77kg            M  Silver
29140  CHN    Men            85kg            M    Gold
29156  CHN  Women       48 - 55kg            W  Silver
29171  CHN  Women       63 - 72kg            W    Gold
29204  CHN    Men       66 - 74kg            M  Silver

[184 rows x 10 columns]
```

```
[8]: chn_team.Gender.value_counts().plot(kind='bar')
```

```
[8]: <AxesSubplot:>
```



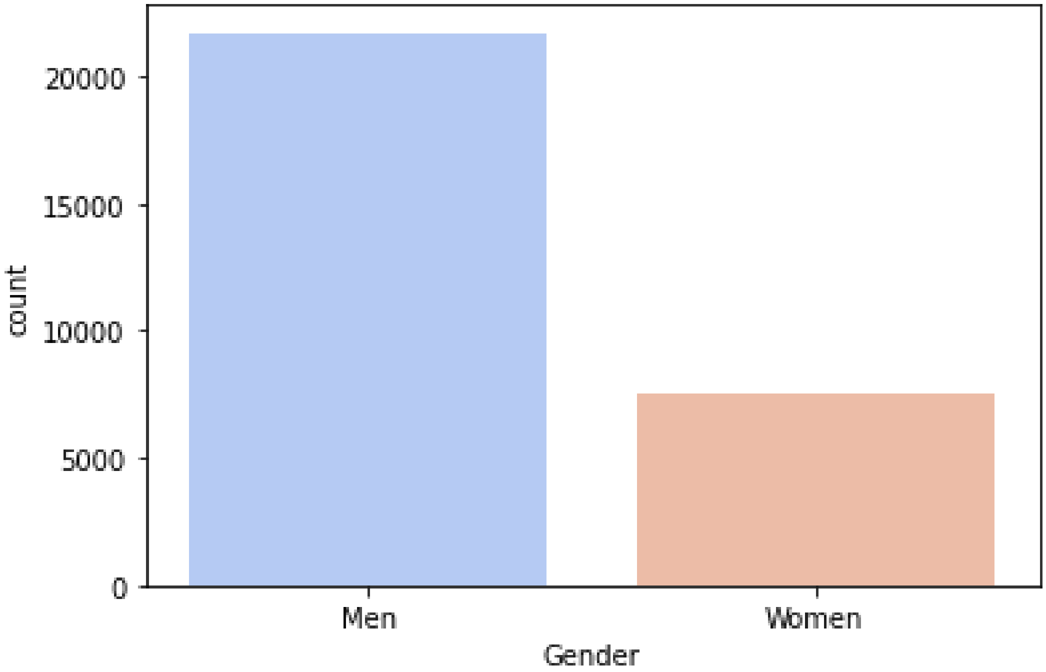
```
[12]: sns.countplot(x='Medal',data=oo, hue='Gender');
```



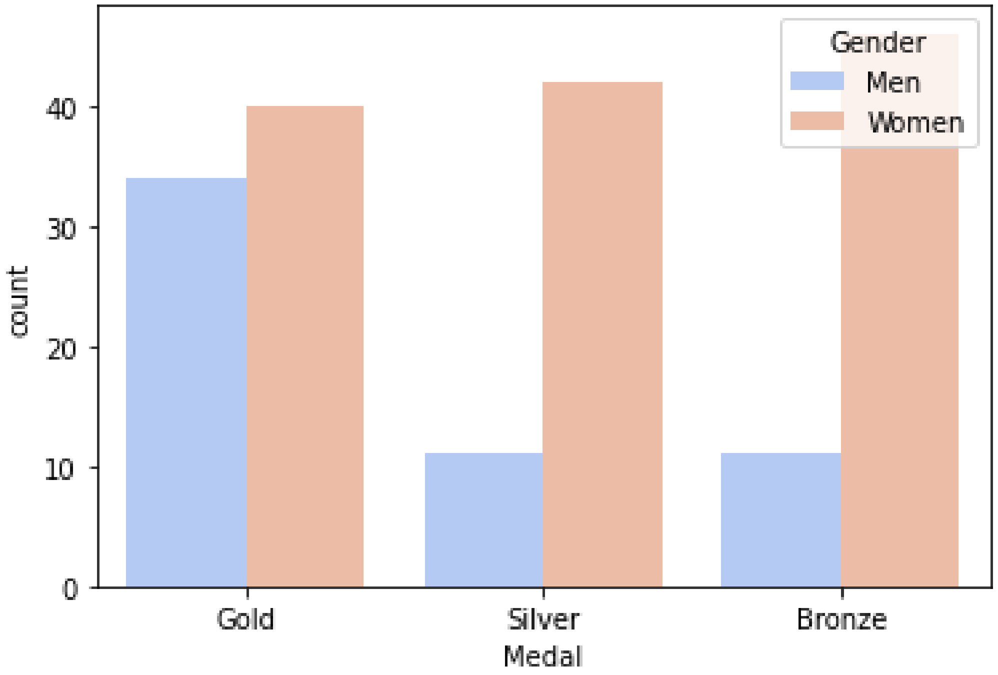
2.0.1 How can you use colormaps to give the data more meaning?

```
[13]: sns.countplot(data=oo, x='Gender',palette='coolwarm')
```

```
[13]: <AxesSubplot:xlabel='Gender', ylabel='count'>
```



```
[14]: sns.countplot(x='Medal',data=chn_team,␣
       ↪hue='Gender',palette='coolwarm',order=['Gold','Silver','Bronze']);
```



```
[10]: oo.City.unique()
```

```
[10]: array(['Athens', 'Paris', 'St Louis', 'London', 'Stockholm', 'Antwerp',
             'Amsterdam', 'Los Angeles', 'Berlin', 'Helsinki',
             'Melbourne / Stockholm', 'Rome', 'Tokyo', 'Mexico', 'Munich',
             'Montreal', 'Moscow', 'Seoul', 'Barcelona', 'Atlanta', 'Sydney',
             'Beijing'], dtype=object)
```

```
[ ]:
```

Chapter_6_challenge

March 13, 2022

1 Chapter 6

[1]:

```
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

[2]:

```
oo = pd.read_csv('../data/olympics.csv',skiprows =4)
oo.head()
```

[2]:

```
     City  Edition     Sport Discipline             Athlete  NOC Gender  \
0  Athens     1896  Aquatics   Swimming       HAJOS, Alfred  HUN    Men
1  Athens     1896  Aquatics   Swimming    HERSCHMANN, Otto  AUT    Men
2  Athens     1896  Aquatics   Swimming   DRIVAS, Dimitrios  GRE    Men
3  Athens     1896  Aquatics   Swimming  MALOKINIS, Ioannis  GRE    Men
4  Athens     1896  Aquatics   Swimming  CHASAPIS, Spiridon  GRE    Men

                       Event Event_gender   Medal
0             100m freestyle            M    Gold
1             100m freestyle            M  Silver
2  100m freestyle for sailors            M  Bronze
3  100m freestyle for sailors            M    Gold
4  100m freestyle for sailors            M  Silver
```

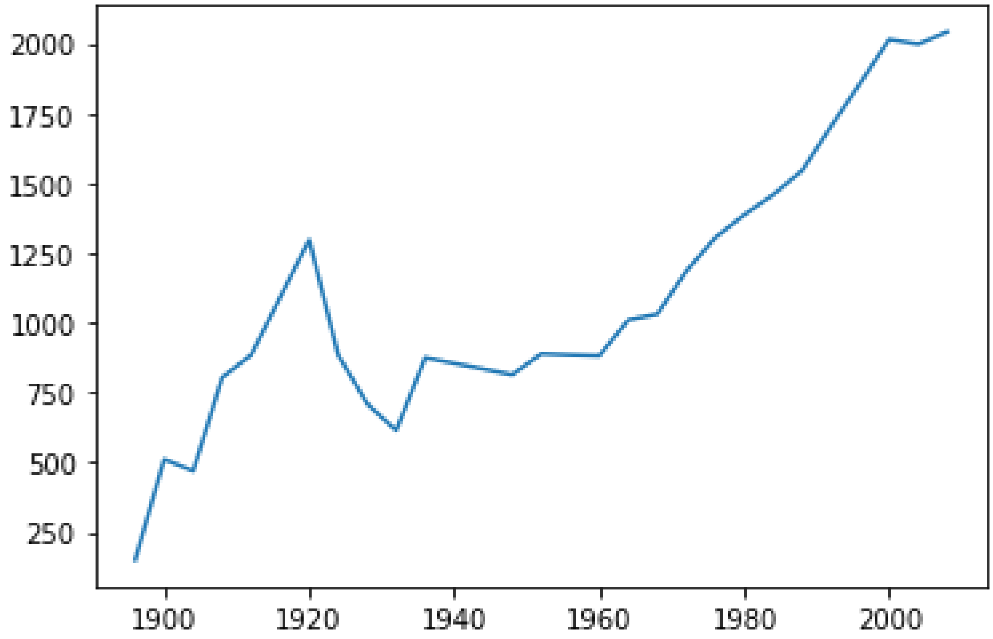
2 Plot the total number of medals awarded at each of the Olympic games throughout history.

[3]:

```
oo.Edition.value_counts().sort_index().plot()
```

[3]:

```
<AxesSubplot:>
```



3 Which countries did not win a medal in the 2008 Olympics? How many countries were there?

[4]:
```
lo =oo[oo.Edition ==2008]
lo
```

[4]:
```
          City  Edition       Sport       Discipline                Athlete  \
27174  Beijing     2008    Aquatics           Diving        GALPERIN, Gleb
27175  Beijing     2008    Aquatics           Diving      MITCHAM, Matthew
27176  Beijing     2008    Aquatics           Diving           ZHOU, Luxin
27177  Beijing     2008    Aquatics           Diving             WANG, Xin
27178  Beijing     2008    Aquatics           Diving          CHEN, Ruolin
…          …        …           …                …                     …
29211  Beijing     2008   Wrestling  Wrestling Gre-R        ENGLICH, Mirko
29212  Beijing     2008   Wrestling  Wrestling Gre-R  MIZGAITIS, Mindaugas
29213  Beijing     2008   Wrestling  Wrestling Gre-R       PATRIKEEV, Yuri
29214  Beijing     2008   Wrestling  Wrestling Gre-R         LOPEZ, Mijain
29215  Beijing     2008   Wrestling  Wrestling Gre-R        BAROEV, Khasan

       NOC Gender          Event Event_gender   Medal
27174  RUS    Men  10m platform             M  Bronze
27175  AUS    Men  10m platform             M    Gold
27176  CHN    Men  10m platform             M  Silver
27177  CHN  Women  10m platform             W  Bronze
```

```
27178  CHN  Women  10m platform            W    Gold
…     …    …           …           …      …
29211  GER    Men      84 - 96kg            M  Silver
29212  LTU    Men    96 - 120kg             M  Bronze
29213  ARM    Men    96 - 120kg             M  Bronze
29214  CUB    Men    96 - 120kg             M    Gold
29215  RUS    Men    96 - 120kg             M  Silver

[2042 rows x 10 columns]
```

[5]:
```
noc =pd.read_csv('../data/Summer Olympic medallists 1896 to 2008 - IOC COUNTRY␣
 ↪CODES.csv')
```

[6]:
```
noc.head()
```

[6]:
```
            Country Int Olympic Committee code ISO code        Country.1
0       Afghanistan                        AFG       AF      Afghanistan
1           Albania                        ALB       AL          Albania
2           Algeria                        ALG       DZ          Algeria
3   American Samoa*                        ASA       AS  American Samoa*
4           Andorra                        AND       AD          Andorra
```

[7]:
```
noc[noc['Country'] != noc['Country.1']]
# it shows that there is no difference between country and country.1 columns
```

[7]:
```
Empty DataFrame
Columns: [Country, Int Olympic Committee code, ISO code, Country.1]
Index: []
```

[8]:
```
noc.set_index('Int Olympic Committee code',inplace =True)
noc.head()
```

[8]:
```
                                    Country ISO code        Country.1
Int Olympic Committee code
AFG                             Afghanistan       AF      Afghanistan
ALB                                 Albania       AL          Albania
ALG                                 Algeria       DZ          Algeria
ASA                         American Samoa*       AS  American Samoa*
AND                                 Andorra       AD          Andorra
```

[9]:
```
medal_2008 =lo.NOC.value_counts()
# medal_2008.index #gives index of the series
```

[10]:
```
medal_2008
```

[10]:
```
USA    315
CHN    184
AUS    149
```

```
RUS    143
GER    101
      …
SUD      1
EGY      1
MDA      1
TOG      1
MRI      1
Name: NOC, Length: 86, dtype: int64
```

[11]:

```
# add new column medal_2008 which is the series object with index of country␣
 ↪code and value with number of medal won by that country
# column is added with reference of index
noc['medal_2008'] =medal_2008
```

[12]:

```
noc.head()
```

[12]:

```
                                    Country ISO code        Country.1  \
Int Olympic Committee code
AFG                             Afghanistan       AF      Afghanistan
ALB                                 Albania       AL          Albania
ALG                                 Algeria       DZ          Algeria
ASA                         American Samoa*       AS  American Samoa*
AND                                 Andorra       AD          Andorra

                            medal_2008
Int Olympic Committee code
AFG                                1.0
ALB                                NaN
ALG                                2.0
ASA                                NaN
AND                                NaN
```

[13]:

```
noc[noc.medal_2008.isnull()]
```

[13]:

```
                                        Country ISO code            Country.1  \
Int Olympic Committee code
ALB                                     Albania       AL              Albania
ASA                             American Samoa*       AS      American Samoa*
AND                                     Andorra       AD              Andorra
ANG                                      Angola       AO               Angola
ANT                         Antigua and Barbuda       AG  Antigua and Barbuda
…                                         …       …                  …
URU                                     Uruguay       UY              Uruguay
VAN                                     Vanuatu       VU              Vanuatu
ISV                             Virgin Islands*       VI      Virgin Islands*
YEM                                       Yemen       YE                Yemen
```

```
ZAM                                      Zambia       ZM                Zambia

                            medal_2008
Int Olympic Committee code
ALB                                NaN
ASA                                NaN
AND                                NaN
ANG                                NaN
ANT                                NaN
…                                 …
URU                                NaN
VAN                                NaN
ISV                                NaN
YEM                                NaN
ZAM                                NaN

[117 rows x 4 columns]
```

[ ]:

Chapter_7_Challenge

March 13, 2022

1 Chapter7

```
[1]: import pandas as pd
```

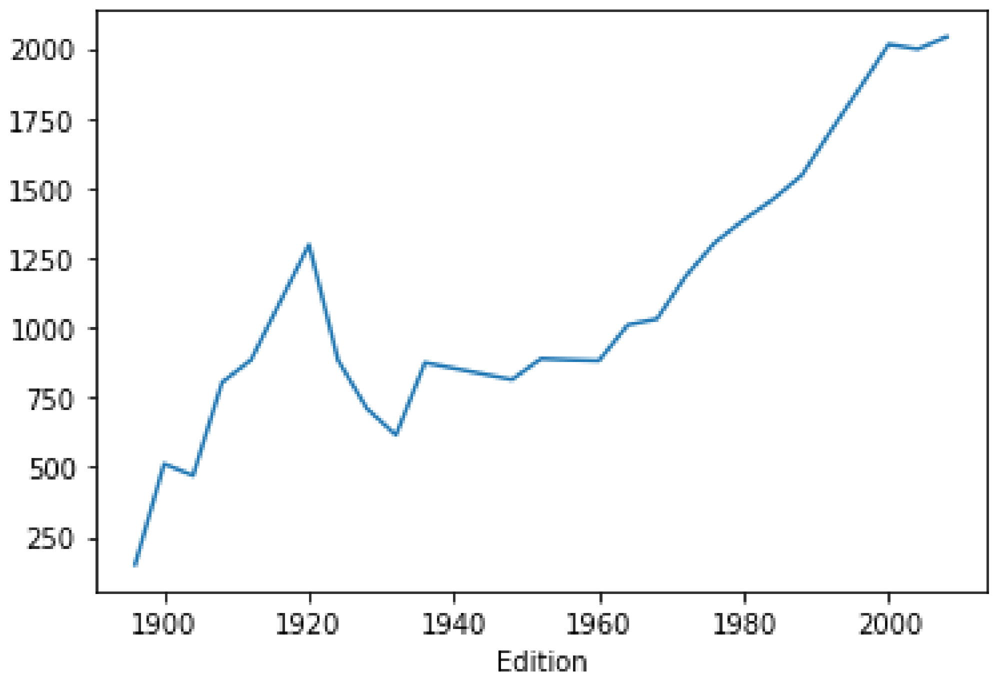
```
[2]: oo = pd.read_csv('../data/olympics.csv',skiprows =4)
     oo.head()
```

```
[2]:      City  Edition     Sport Discipline             Athlete  NOC Gender  \
     0  Athens     1896  Aquatics   Swimming       HAJOS, Alfred  HUN    Men
     1  Athens     1896  Aquatics   Swimming    HERSCHMANN, Otto  AUT    Men
     2  Athens     1896  Aquatics   Swimming   DRIVAS, Dimitrios  GRE    Men
     3  Athens     1896  Aquatics   Swimming  MALOKINIS, Ioannis  GRE    Men
     4  Athens     1896  Aquatics   Swimming  CHASAPIS, Spiridon  GRE    Men

                            Event Event_gender   Medal
     0             100m freestyle            M    Gold
     1             100m freestyle            M  Silver
     2  100m freestyle for sailors            M  Bronze
     3  100m freestyle for sailors            M    Gold
     4  100m freestyle for sailors            M  Silver
```

2 Using groupby(), plot the total number of medals awarded at each of the Olympic games throughout history.

```
[34]: oo.groupby('Edition').size().plot()
```

```
[34]: <AxesSubplot:xlabel='Edition'>
```



2.1 Create a list showing the total number of medals won for each country over the history of the Olympics. For each country, include the year of the first and most recent Olympic medal wins

[38]:
```
oo.groupby(['NOC']).agg({'Edition' :['min','max','count']})
```

[38]:
```
    Edition
        min   max count
NOC
AFG    2008  2008     1
AHO    1988  1988     1
ALG    1984  2008    14
ANZ    1908  1912    29
ARG    1924  2008   239
..      …     …     …
VIE    2000  2008     2
YUG    1924  2000   435
ZAM    1984  1996     2
ZIM    1980  2008    23
ZZX    1896  1904    48

[138 rows x 3 columns]
```

[ ]:

Chapter_8_practise

March 14, 2022

1 Chapter 8

[1]:
```
import pandas as pd
```

[4]:
```
oo = pd.read_csv('../data/olympics.csv',skiprows=4)
```

[5]:
```
oo.head()
```

[5]:
```
     City  Edition     Sport Discipline             Athlete  NOC Gender  \
0  Athens     1896  Aquatics   Swimming       HAJOS, Alfred  HUN    Men
1  Athens     1896  Aquatics   Swimming    HERSCHMANN, Otto  AUT    Men
2  Athens     1896  Aquatics   Swimming   DRIVAS, Dimitrios  GRE    Men
3  Athens     1896  Aquatics   Swimming  MALOKINIS, Ioannis  GRE    Men
4  Athens     1896  Aquatics   Swimming  CHASAPIS, Spiridon  GRE    Men

                       Event Event_gender   Medal
0             100m freestyle            M    Gold
1             100m freestyle            M  Silver
2  100m freestyle for sailors           M  Bronze
3  100m freestyle for sailors           M    Gold
4  100m freestyle for sailors           M  Silver
```

2 Athletes winning medals in Beijing Olympics 100m or 200m track event

[9]:
```
mw =oo[(oo.Edition ==2008)&((oo.Event =='100m')|(oo.Event =='200m'))]
mw
```

[9]:
```
          City  Edition      Sport Discipline                Athlete  NOC  \
27551  Beijing     2008  Athletics  Athletics            DIX, Walter  USA
27552  Beijing     2008  Athletics  Athletics            BOLT, Usain  JAM
27553  Beijing     2008  Athletics  Athletics      THOMPSON, Richard  TRI
27554  Beijing     2008  Athletics  Athletics    FRASER, Shelly-ann  JAM
27555  Beijing     2008  Athletics  Athletics        SIMPSON, Sherone  JAM
27556  Beijing     2008  Athletics  Athletics         STEWART, Kerron  JAM
27569  Beijing     2008  Athletics  Athletics            DIX, Walter  USA
27570  Beijing     2008  Athletics  Athletics            BOLT, Usain  JAM
```

```
27571  Beijing     2008  Athletics  Athletics           CRAWFORD, Shawn  USA
27572  Beijing     2008  Athletics  Athletics           STEWART, Kerron  JAM
27573  Beijing     2008  Athletics  Athletics  CAMPBELL-BROWN, Veronica  JAM
27574  Beijing     2008  Athletics  Athletics            FELIX, Allyson  USA

      Gender Event Event_gender   Medal
27551    Men  100m            M  Bronze
27552    Men  100m            M    Gold
27553    Men  100m            M  Silver
27554  Women  100m            W    Gold
27555  Women  100m            W  Silver
27556  Women  100m            W  Silver
27569    Men  200m            M  Bronze
27570    Men  200m            M    Gold
27571    Men  200m            M  Silver
27572  Women  200m            W  Bronze
27573  Women  200m            W    Gold
27574  Women  200m            W  Silver
```

[11]:

```
g=mw.groupby(['NOC','Gender','Event','Discipline']).size()
g
```

[11]:

```
NOC  Gender  Event  Discipline
JAM  Men     100m   Athletics     1
             200m   Athletics     1
     Women   100m   Athletics     3
             200m   Athletics     2
TRI  Men     100m   Athletics     1
USA  Men     100m   Athletics     1
             200m   Athletics     2
     Women   200m   Athletics     1
dtype: int64
```

[14]:

```
df =g.unstack(['Discipline','Event'])
```

3 Stack()

[15]:

```
df
```

[15]:

```
Discipline Athletics
Event           100m 200m
NOC Gender
JAM Men          1.0  1.0
    Women        3.0  2.0
TRI Men          1.0  NaN
USA Men          1.0  2.0
```

```
    Women         NaN  1.0
```

[16]:

```
df.stack()
```

[16]:

```
Discipline        Athletics
NOC Gender Event
JAM Men    100m         1.0
           200m         1.0
    Women  100m         3.0
           200m         2.0
TRI Men    100m         1.0
USA Men    100m         1.0
           200m         2.0
    Women  200m         1.0
```

[17]:

```
df.stack('Event')
```

[17]:

```
Discipline        Athletics
NOC Gender Event
JAM Men    100m         1.0
           200m         1.0
    Women  100m         3.0
           200m         2.0
TRI Men    100m         1.0
USA Men    100m         1.0
           200m         2.0
    Women  200m         1.0
```

4 Unstack()

[18]:

```
df
```

[18]:

```
Discipline Athletics
Event           100m 200m
NOC Gender
JAM Men          1.0  1.0
    Women        3.0  2.0
TRI Men          1.0  NaN
USA Men          1.0  2.0
    Women        NaN  1.0
```

[19]:

```
df.unstack()
```

[19]:

```
Discipline Athletics
Event           100m       200m
Gender           Men Women  Men Women
NOC
```

```
JAM               1.0   3.0  1.0   2.0
TRI               1.0   NaN  NaN   NaN
USA               1.0   NaN  2.0   1.0
```

[20]:

```
df.unstack('Gender')
```

[20]:

```
Discipline Athletics
Event            100m       200m
Gender            Men Women  Men Women
NOC
JAM               1.0   3.0  1.0   2.0
TRI               1.0   NaN  NaN   NaN
USA               1.0   NaN  2.0   1.0
```

5 Challenge

[21]:

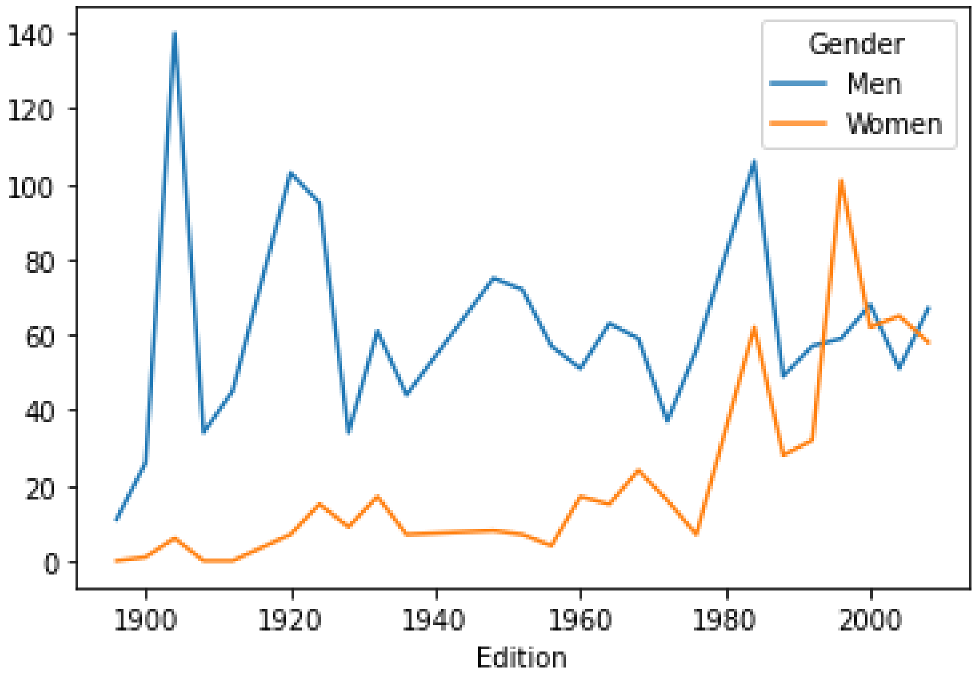
```
### Plot the number of gold medals won by the USA male and female athletes␣
 ↪throughout the history of the Olympics?
```

[27]:

```
gu = oo[(oo.NOC =='USA') &(oo.Medal =='Gold')]
gu.groupby(['Edition','Gender']).size().unstack('Gender',fill_value =0).plot()
```

[27]:

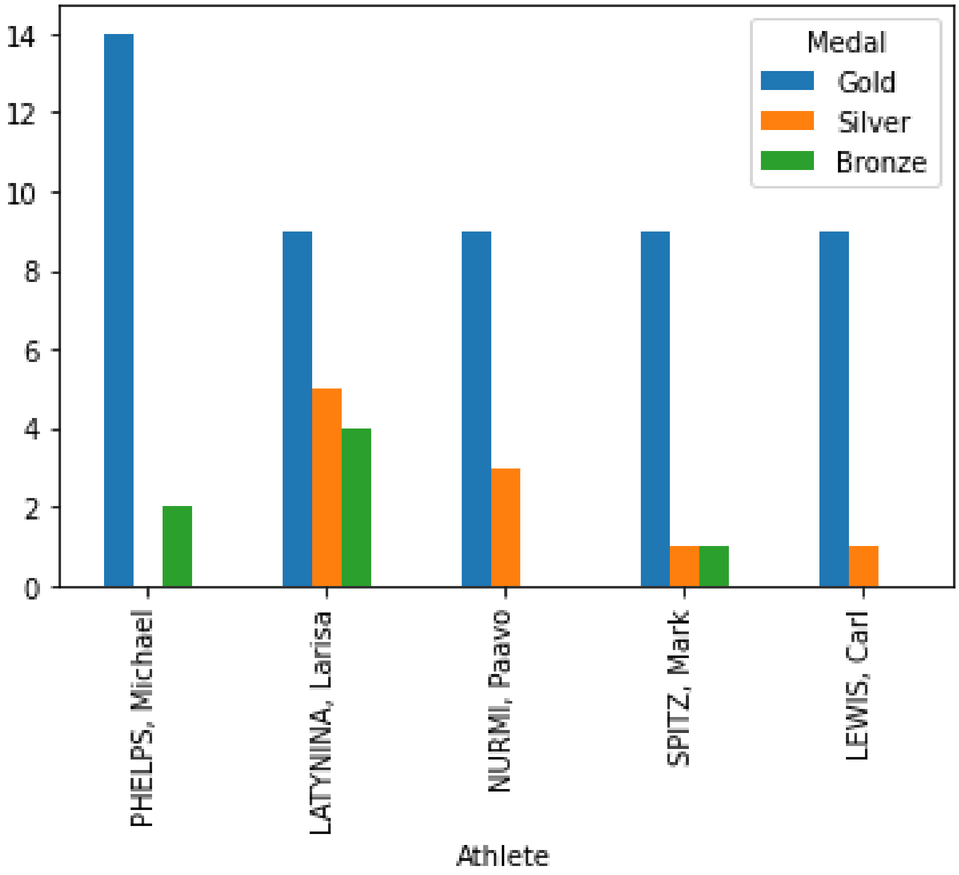
```
<AxesSubplot:xlabel='Edition'>
```



6 Plot the 5 athletes who have won the most gold medals over the history of the Olympics. When there is a tie, consider the number of silver medals, then bronze medals.

```
[43]: g= oo.groupby(['Athlete','Medal']).size().unstack('Medal',fill_value =0)
      g.
       ↪sort_values(['Gold','Silver','Bronze'],ascending=False)[['Gold','Silver','Bronze']].
       ↪head().plot(kind='bar')
```

```
[43]: <AxesSubplot:xlabel='Athlete'>
```



```
[ ]:
```

Chapter9_practise

March 14, 2022

0.1 Chapter 9

[1]:

```
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

[2]:

```
import seaborn as sns
```

[3]:

```
oo = pd.read_csv('../data/olympics.csv',skiprows=4)
oo
```

[3]:

```
          City  Edition     Sport       Discipline                  Athlete  \
0       Athens     1896  Aquatics         Swimming            HAJOS, Alfred
1       Athens     1896  Aquatics         Swimming         HERSCHMANN, Otto
2       Athens     1896  Aquatics         Swimming        DRIVAS, Dimitrios
3       Athens     1896  Aquatics         Swimming       MALOKINIS, Ioannis
4       Athens     1896  Aquatics         Swimming       CHASAPIS, Spiridon
…        …      …        …              …                   …
29211  Beijing     2008 Wrestling  Wrestling Gre-R           ENGLICH, Mirko
29212  Beijing     2008 Wrestling  Wrestling Gre-R     MIZGAITIS, Mindaugas
29213  Beijing     2008 Wrestling  Wrestling Gre-R          PATRIKEEV, Yuri
29214  Beijing     2008 Wrestling  Wrestling Gre-R            LOPEZ, Mijain
29215  Beijing     2008 Wrestling  Wrestling Gre-R           BAROEV, Khasan

       NOC Gender                       Event Event_gender   Medal
0      HUN    Men              100m freestyle            M    Gold
1      AUT    Men              100m freestyle            M  Silver
2      GRE    Men  100m freestyle for sailors            M  Bronze
3      GRE    Men  100m freestyle for sailors            M    Gold
4      GRE    Men  100m freestyle for sailors            M  Silver
…    …    …                          …          …     …
29211  GER    Men                   84 - 96kg            M  Silver
29212  LTU    Men                  96 - 120kg            M  Bronze
29213  ARM    Men                  96 - 120kg            M  Bronze
29214  CUB    Men                  96 - 120kg            M    Gold
29215  RUS    Men                  96 - 120kg            M  Silver

[29216 rows x 10 columns]
```

0.2 Using the Olympic dataset, present a summary of the total medals won by participating countries in the 2008 Olympics.

[4]:
```
lo =oo[oo.Edition==2008]
lo
```

[4]:
```
          City  Edition      Sport       Discipline               Athlete  \
27174  Beijing     2008   Aquatics           Diving        GALPERIN, Gleb
27175  Beijing     2008   Aquatics           Diving      MITCHAM, Matthew
27176  Beijing     2008   Aquatics           Diving           ZHOU, Luxin
27177  Beijing     2008   Aquatics           Diving             WANG, Xin
27178  Beijing     2008   Aquatics           Diving          CHEN, Ruolin
…          …       …          …               …                    …
29211  Beijing     2008  Wrestling  Wrestling Gre-R        ENGLICH, Mirko
29212  Beijing     2008  Wrestling  Wrestling Gre-R  MIZGAITIS, Mindaugas
29213  Beijing     2008  Wrestling  Wrestling Gre-R       PATRIKEEV, Yuri
29214  Beijing     2008  Wrestling  Wrestling Gre-R         LOPEZ, Mijain
29215  Beijing     2008  Wrestling  Wrestling Gre-R        BAROEV, Khasan

       NOC Gender         Event Event_gender   Medal
27174  RUS    Men  10m platform            M  Bronze
27175  AUS    Men  10m platform            M    Gold
27176  CHN    Men  10m platform            M  Silver
27177  CHN  Women  10m platform            W  Bronze
27178  CHN  Women  10m platform            W    Gold
…     …    …          …            …      …
29211  GER    Men     84 - 96kg            M  Silver
29212  LTU    Men    96 - 120kg            M  Bronze
29213  ARM    Men    96 - 120kg            M  Bronze
29214  CUB    Men    96 - 120kg            M    Gold
29215  RUS    Men    96 - 120kg            M  Silver

[2042 rows x 10 columns]
```

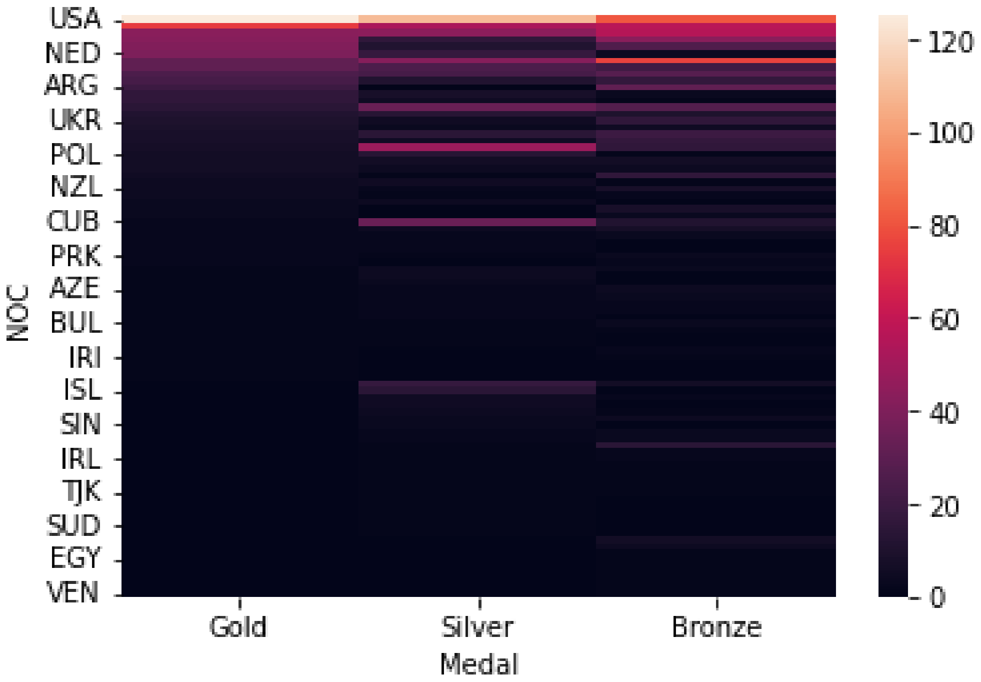
[7]:
```
g = lo.groupby(['NOC','Medal']).size().unstack('Medal',fill_value =0)
g = g.sort_values(['Gold','Silver','Bronze'],ascending␣
 ↪=False)[['Gold','Silver','Bronze']]
g

# g = lo.groupby(['NOC','Medal']).size().unstack('Medal',fill_value=0)
# g = g.
 ↪sort_values(['Gold','Silver','Bronze'],ascending=False)[['Gold','Silver','Bronze']]
# g
```

[7]:
```
Medal  Gold  Silver  Bronze
NOC
USA     125     109      81
```

```
CHN       74       53       57
RUS       43       44       56
GER       42       16       43
KOR       41       11       26
..       …       …       …
ISR        0        0        1
MDA        0        0        1
MRI        0        0        1
TOG        0        0        1
VEN        0        0        1

[86 rows x 3 columns]
```

[8]:
```
sns.heatmap(g)
```

[8]:
```
<AxesSubplot:xlabel='Medal', ylabel='NOC'>
```



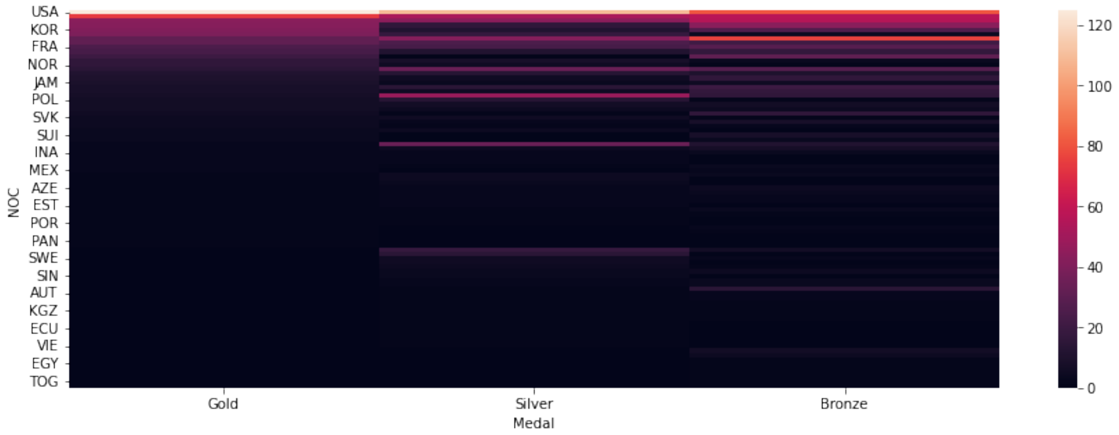
[9]:
```
g.transpose
```

[9]:
```
<bound method DataFrame.transpose of Medal  Gold  Silver  Bronze
NOC
USA      125      109       81
CHN       74       53       57
RUS       43       44       56
```

```
GER      42      16      43
KOR      41      11      26
..       …       …       …
ISR       0       0       1
MDA       0       0       1
MRI       0       0       1
TOG       0       0       1
VEN       0       0       1

[86 rows x 3 columns]>
```

[10]:
```
plt.figure(figsize=(15,5))
sns.heatmap(g)
```

[10]:
```
<AxesSubplot:xlabel='Medal', ylabel='NOC'>
```

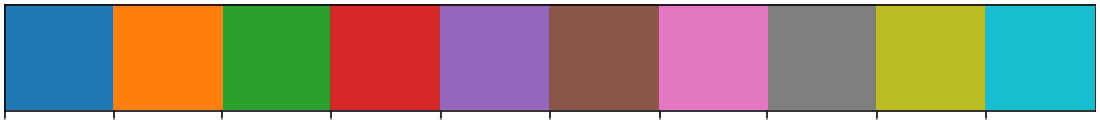


[11]:
```
from matplotlib.colors import ListedColormap
```

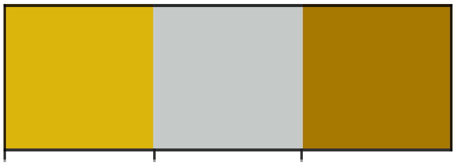
[12]:
```
sns.color_palette()
```

[12]:
```
[(0.12156862745098039, 0.4666666666666667, 0.7058823529411765),
 (1.0, 0.4980392156862745, 0.054901960784313725),
 (0.17254901960784313, 0.6274509803921569, 0.17254901960784313),
 (0.8392156862745098, 0.15294117647058825, 0.1568627450980392),
 (0.5803921568627451, 0.403921568627451, 0.7411764705882353),
 (0.5490196078431373, 0.33725490196078434, 0.29411764705882354),
 (0.8901960784313725, 0.4666666666666667, 0.7607843137254902),
 (0.4980392156862745, 0.4980392156862745, 0.4980392156862745),
 (0.7372549019607844, 0.7411764705882353, 0.13333333333333333),
 (0.09019607843137255, 0.7450980392156863, 0.8117647058823529)]
```

[13]:
```
sns.palplot(sns.color_palette())
```



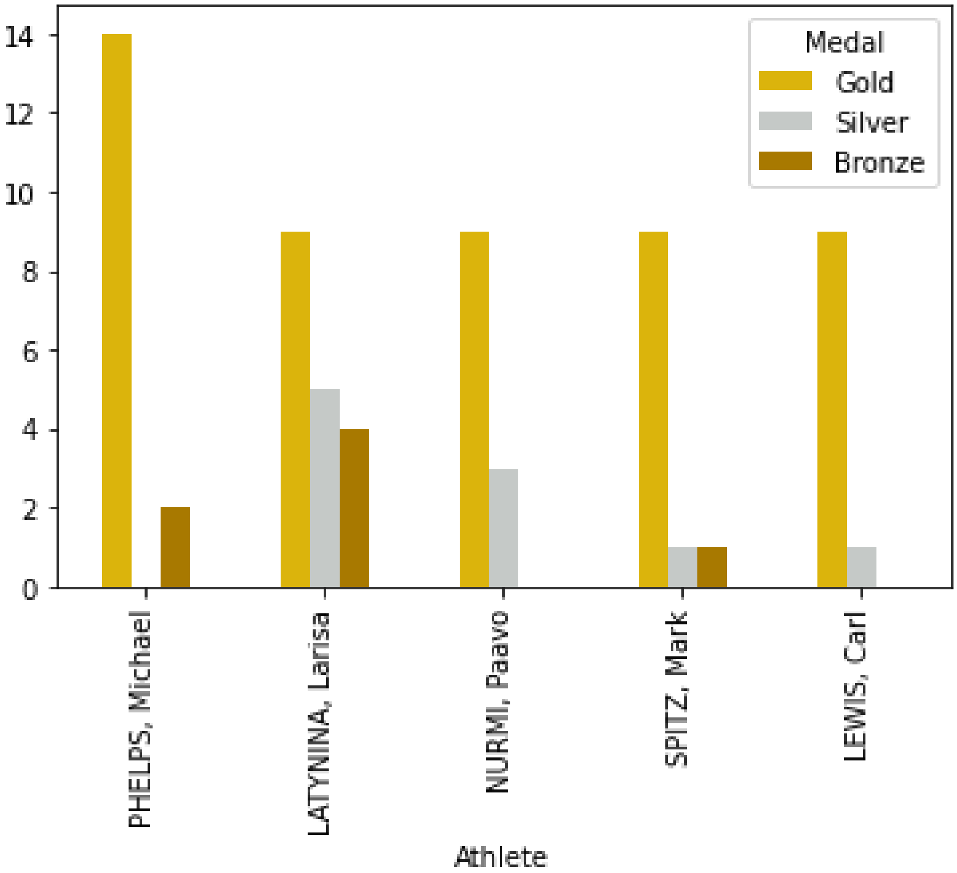
```
[14]: gsb = ['#dbb40c','#c5c9c7','#a87900']
      sns.palplot(sns.color_palette(gsb))
```



```
[15]: my_gsb = ListedColormap(sns.color_palette(gsb))
```

```
[16]: g = oo.groupby(['Athlete','Medal']).size().unstack('Medal',fill_value=0)
      g = g.
       ↪sort_values(['Gold','Silver','Bronze'],ascending=False)[['Gold','Silver','Bronze']].
       ↪head()
      g.plot(kind='bar',colormap=my_gsb)
```

```
[16]: <AxesSubplot:xlabel='Athlete'>
```


Medal
Gold
Silver
Bronze
14
12
10
8
6
4
2
0
PHELPS, Michael
LATYNINA, Larisa
NURMI, Paavo
SPITZ, Mark
LEWIS, Carl
Athlete

```
[ ]:
```

Chapter10_practise

March 14, 2022

[1]:

```
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

[2]:

```
oo = pd.read_csv('../data/olympics.csv',skiprows=4)
oo.head()
```

[2]:

```
     City  Edition     Sport Discipline             Athlete  NOC Gender  \
0  Athens     1896  Aquatics   Swimming       HAJOS, Alfred  HUN    Men
1  Athens     1896  Aquatics   Swimming    HERSCHMANN, Otto  AUT    Men
2  Athens     1896  Aquatics   Swimming   DRIVAS, Dimitrios  GRE    Men
3  Athens     1896  Aquatics   Swimming  MALOKINIS, Ioannis  GRE    Men
4  Athens     1896  Aquatics   Swimming  CHASAPIS, Spiridon  GRE    Men

                       Event Event_gender   Medal
0             100m freestyle            M    Gold
1             100m freestyle            M  Silver
2  100m freestyle for sailors            M  Bronze
3  100m freestyle for sailors            M    Gold
4  100m freestyle for sailors            M  Silver
```

[3]:

```
gy  = oo[oo.NOC == 'USA']
gy = gy.groupby(['Edition','Athlete','Medal']).size().
 ↪unstack('Medal',fill_value=0)
gy['Total'] = gy['Gold'] + gy['Silver'] + gy['Bronze']
gy.reset_index(inplace=True)
tu = [group.sort_values('Total',ascending=False)[:1] for year,group in gy.
 ↪groupby('Edition')]
tu
top = pd.DataFrame()
for i in tu:
    top = top.append(i)
top
```

```
C:\Users\aadar\AppData\Local\Temp\ipykernel_32124\2586981064.py:9:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
  top = top.append(i)
```

```
C:\Users\aadar\AppData\Local\Temp\ipykernel_32124\2586981064.py:9:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
  top = top.append(i)
C:\Users\aadar\AppData\Local\Temp\ipykernel_32124\2586981064.py:9:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
  top = top.append(i)
C:\Users\aadar\AppData\Local\Temp\ipykernel_32124\2586981064.py:9:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
  top = top.append(i)
C:\Users\aadar\AppData\Local\Temp\ipykernel_32124\2586981064.py:9:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
  top = top.append(i)
C:\Users\aadar\AppData\Local\Temp\ipykernel_32124\2586981064.py:9:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
  top = top.append(i)
C:\Users\aadar\AppData\Local\Temp\ipykernel_32124\2586981064.py:9:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
  top = top.append(i)
C:\Users\aadar\AppData\Local\Temp\ipykernel_32124\2586981064.py:9:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
  top = top.append(i)
C:\Users\aadar\AppData\Local\Temp\ipykernel_32124\2586981064.py:9:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
  top = top.append(i)
C:\Users\aadar\AppData\Local\Temp\ipykernel_32124\2586981064.py:9:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
  top = top.append(i)
C:\Users\aadar\AppData\Local\Temp\ipykernel_32124\2586981064.py:9:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
  top = top.append(i)
C:\Users\aadar\AppData\Local\Temp\ipykernel_32124\2586981064.py:9:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
  top = top.append(i)
C:\Users\aadar\AppData\Local\Temp\ipykernel_32124\2586981064.py:9:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
  top = top.append(i)
```

```
C:\Users\aadar\AppData\Local\Temp\ipykernel_32124\2586981064.py:9:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
  top = top.append(i)
C:\Users\aadar\AppData\Local\Temp\ipykernel_32124\2586981064.py:9:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
  top = top.append(i)
C:\Users\aadar\AppData\Local\Temp\ipykernel_32124\2586981064.py:9:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
  top = top.append(i)
C:\Users\aadar\AppData\Local\Temp\ipykernel_32124\2586981064.py:9:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
  top = top.append(i)
C:\Users\aadar\AppData\Local\Temp\ipykernel_32124\2586981064.py:9:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
  top = top.append(i)
C:\Users\aadar\AppData\Local\Temp\ipykernel_32124\2586981064.py:9:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
  top = top.append(i)
C:\Users\aadar\AppData\Local\Temp\ipykernel_32124\2586981064.py:9:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
  top = top.append(i)
C:\Users\aadar\AppData\Local\Temp\ipykernel_32124\2586981064.py:9:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
  top = top.append(i)
C:\Users\aadar\AppData\Local\Temp\ipykernel_32124\2586981064.py:9:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
  top = top.append(i)
C:\Users\aadar\AppData\Local\Temp\ipykernel_32124\2586981064.py:9:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
  top = top.append(i)
C:\Users\aadar\AppData\Local\Temp\ipykernel_32124\2586981064.py:9:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
  top = top.append(i)
C:\Users\aadar\AppData\Local\Temp\ipykernel_32124\2586981064.py:9:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
  top = top.append(i)
```

```
[3]: Medal  Edition                      Athlete  Bronze  Gold  Silver  Total
     5         1896              GARRETT, Robert       0     2       2      4
     14        1900               BAXTER, Irving       0     2       3      5
     153       1904                 HEIDA, Anton       0     5       1      6
     363       1908             SHERIDAN, Martin       1     2       0      3
     421       1912        OSBURN, Carl Townsend       1     1       2      4
     506       1920                  LEE, Willis       1     5       1      7
     733       1924          WEISSMULLER, Johnny       1     3       0      4
     800       1928       OSIPOWICH, Albina Lucy       0     2       0      2
     859       1932           DIDRIKSON, Mildred       0     2       1      3
     1028      1936                 OWENS, Jesse       0     4       0      4
     1128      1948      MCLANE, James Price Jr.       0     2       1      3
     1234      1952          KONNO, Ford Hiroshi       0     2       1      3
     1379      1956        MORROW, Robert Joseph       0     3       0      3
     1500      1960  VON SALTZA, Susan Christina       0     3       1      4
     1619      1964        STOUDER, Sharon Marie       0     3       1      4
     1673      1968    HICKCOX, Charles Buchanan       0     3       1      4
     1844      1972                  SPITZ, Mark       0     7       0      7
     1870      1976    BABASHOFF, Shirley Farber       0     1       4      5
     2207      1984             RETTON, Mary-Lou       2     1       2      5
     2294      1988              BIONDI, Matthew       1     5       1      7
     2567      1992              MILLER, Shannon       3     0       2      5
     2703      1996               HALL, Gary Jr.       0     2       2      4
     3045      2000                 TORRES, Dara       3     2       0      5
     3226      2004              PHELPS, Michael       2     6       0      8
     3463      2008              PHELPS, Michael       0     8       0      8
```

```
[ ]:
```