C

January 24, 2025

1

[107]:

```
#   package
import geopandas as gpd
import pandas as pd
import matplotlib.pyplot as plt
import chardet
```

[108]:

```
#
def read_csv_with_detected_encoding(file_path):
    #
    with open(file_path, 'rb') as f:
        result = chardet.detect(f.read())
        encoding = result['encoding']
        print(f"    {file_path}     {encoding}")
    #
    return pd.read_csv(file_path, encoding=encoding)

#   data_dictionary.csv
csv_content =␣
 ↪read_csv_with_detected_encoding('2025_Problem_C_Data\\data_dictionary.csv')
print("data_dictionary.csv    ")
print(csv_content.head())

#   summerOly_medal_counts.csv
medal_counts =␣
 ↪read_csv_with_detected_encoding('2025_Problem_C_Data\\summerOly_medal_counts.
 ↪csv')
print("\nsummerOly_medal_counts.csv    ")
print(medal_counts.head())

#   summerOly_hosts.csv
olympic_hosts =␣
 ↪read_csv_with_detected_encoding('2025_Problem_C_Data\\summerOly_hosts.csv')
print("\nsummerOly_hosts.csv    ")
print(olympic_hosts.head())
```

```
#   summerOly_programs.csv
olympic_programs =␣
 ↪read_csv_with_detected_encoding('2025_Problem_C_Data\\summerOly_programs.
 ↪csv')
print("\nsummerOly_programs.csv    ")
print(olympic_programs.head())

#   summerOly_athletes.csv
olympic_athletes =␣
 ↪read_csv_with_detected_encoding('2025_Problem_C_Data\\summerOly_athletes.
 ↪csv')
print("\nsummerOly_athletes.csv    ")
print(olympic_athletes.head())
```

```
   2025_Problem_C_Data\data_dictionary.csv     Windows-1252
data_dictionary.csv
  summerOly_medal_counts.csv                                    Unnamed: 1  \
0                  variables                                   explanation
1                       Rank      Rank of country based on total medals won
2                        NOC  Name of country as recorded for that Olympics
3                       Gold       Number of Gold medals the country earned
4                     Silver     Number of Silver medals the country earned

      Unnamed: 2
0        example
1           1, 2
2  China, France
3        0, 1, 2
4        0, 1, 2
   2025_Problem_C_Data\summerOly_medal_counts.csv     utf-8

summerOly_medal_counts.csv
   Rank            NOC  Gold  Silver  Bronze  Total  Year
0     1  United States    11       7       2     20  1896
1     2         Greece    10      18      19     47  1896
2     3        Germany     6       5       2     13  1896
3     4         France     5       4       2     11  1896
4     5  Great Britain     2       3       2      7  1896
   2025_Problem_C_Data\summerOly_hosts.csv     UTF-8-SIG

summerOly_hosts.csv
   Year                         Host
0  1896               Athens, Greece
1  1900                Paris, France
2  1904     St. Louis, United States
3  1908       London, United Kingdom
4  1912            Stockholm, Sweden
```

```
   2025_Problem_C_Data\summerOly_programs.csv     Windows-1252

summerOly_programs.csv
      Sport         Discipline Code Sports Governing Body 1896 1900 1904  \
0  Aquatics  Artistic Swimming  SWA        World Aquatics    0    0    0
1  Aquatics             Diving  DIV        World Aquatics    0    0    2
2  Aquatics  Marathon Swimming  OWS        World Aquatics    0    0    0
3  Aquatics           Swimming  SWM        World Aquatics    4    7    9
4  Aquatics         Water Polo  WPO        World Aquatics    0    1    1

   1906*  1908 1912  …  1988 1992  1996  2000  2004  2008  2012  2016  2020  \
0      0     0    0  …     2    2   1.0   2.0   2.0   2.0   2.0   2.0   2.0
1      1     2    4  …     4    4   4.0   8.0   8.0   8.0   8.0   8.0   8.0
2      0     0    0  …     0    0   0.0   0.0   0.0   2.0   2.0   2.0   2.0
3      4     6    9  …    31   31  32.0  32.0  32.0  32.0  32.0  32.0  35.0
4      0     1    1  …     1    1   1.0   2.0   2.0   2.0   2.0   2.0   2.0

   2024
0   2.0
1   8.0
2   2.0
3  35.0
4   2.0

[5 rows x 35 columns]
   2025_Problem_C_Data\summerOly_athletes.csv     utf-8

summerOly_athletes.csv
                    Name Sex           Team  NOC  Year         City  \
0              A Dijiang   M          China  CHN  1992    Barcelona
1               A Lamusi   M          China  CHN  2012       London
2            Gunnar Aaby   M        Denmark  DEN  1920    Antwerpen
3            Edgar Aabye   M  Denmark/Sweden  DEN  1900        Paris
4  Cornelia (-strannood)   F     Netherlands  NED  1932  Los Angeles

        Sport                          Event     Medal
0  Basketball   Basketball Men's Basketball  No medal
1        Judo  Judo Men's Extra-Lightweight  No medal
2    Football        Football Men's Football  No medal
3  Tug-Of-War   Tug-Of-War Men's Tug-Of-War      Gold
4   Athletics  Athletics Women's 100 metres  No medal
```

2

2.0.1

```
[109]: # 1.
def check_missing_values(file_path):
    """
      CSV

     :
        file_path (str): CSV

     :
        None
    """
    try:
        #    CSV
        data = pd.read_csv(file_path, encoding='utf-8')
    except UnicodeDecodeError:
        data = pd.read_csv(file_path, encoding='ISO-8859-1')

    print(file_path)

    #
    missing_values_per_column = data.isnull().sum()
    print("     ")
    print(missing_values_per_column)

    #
    total_missing_values = missing_values_per_column.sum()
    print("       ", total_missing_values)

    #
    has_missing_values = data.isnull().values.any()
    print("       ", has_missing_values)
    print("\n")

    #
    if has_missing_values:
        print("\n     ")
        print(data[data.isnull().any(axis=1)])

content_name = ['2025_Problem_C_Data\\summerOly_medal_counts.csv',␣
 ↪'2025_Problem_C_Data\\summerOly_hosts.csv',␣
 ↪'2025_Problem_C_Data\\summerOly_programs.csv',␣
 ↪'2025_Problem_C_Data\\summerOly_athletes.csv']
for i in content_name:
    check_missing_values(i)
```

```
2025_Problem_C_Data\summerOly_medal_counts.csv

Rank      0
NOC       0
Gold      0
Silver    0
Bronze    0
Total     0
Year      0
dtype: int64
        0
       False


2025_Problem_C_Data\summerOly_hosts.csv

Year    0
Host    0
dtype: int64
        0
       False


2025_Problem_C_Data\summerOly_programs.csv

Sport                    0
Discipline               2
Code                     0
Sports Governing Body    0
1896                     0
1900                     0
1904                     0
1906*                    0
1908                     0
1912                     0
1920                     0
1924                     0
1928                     2
1932                     2
1936                     2
1948                     2
1952                     2
1956                     2
1960                     2
1964                     2
1968                     2
1972                     2
1976                     2
```

```
1980                    2
1984                    2
1988                    3
1992                    2
1996                    2
2000                    2
2004                    2
2008                    2
2012                    2
2016                    2
2020                    2
2024                    2
dtype: int64
        49
       True
```

```
                Sport     Discipline Code Sports Governing Body 1896 1900  \
12      Basque Pelota  Basque Pelota  PEL                  FIPV    0    1   
44  Modern Pentathlon            NaN  MPN                  UIPM    0    0   
65  Water Motorsports            NaN  PBT                   UIM    0        
69            Skating         Figure  FSK                   ISU    0    0   
70         Ice Hockey     Ice Hockey  IHO                  IIHF    0    0   

   1904  1906*  1908 1912  …  1988 1992 1996 2000 2004 2008  2012 2016  \
12    0      0     0    0  …   NaN       0.0  0.0  0.0  0.0   0.0  0.0   
44    0      0     0    1  …     2    2  1.0  2.0  2.0  2.0   2.0  2.0   
65    0      0     3    0  …     0    0  0.0  0.0  0.0  0.0   0.0  0.0   
69    0      0     4    0  …   NaN  NaN  NaN  NaN  NaN  NaN   NaN  NaN   
70    0      0     0    0  …   NaN  NaN  NaN  NaN  NaN  NaN   NaN  NaN   

    2020 2024  
12   0.0  0.0  
44   2.0  2.0  
65   0.0  0.0  
69   NaN  NaN  
70   NaN  NaN  

[5 rows x 35 columns]
2025_Problem_C_Data\summerOly_athletes.csv

Name     0
Sex      0
Team     0
NOC      0
Year     0
```

```
City     0
Sport    0
Event    0
Medal    0
dtype: int64
       0
       False
```

2.0.2 summerOly_programs.csv

[110]:
```
import pandas as pd
import numpy as np
from sklearn.ensemble import RandomForestRegressor
from sklearn.linear_model import LinearRegression
from sklearn.neighbors import KNeighborsRegressor
import re
import os

#
os.makedirs('Generated', exist_ok=True)

data = olympic_programs.copy()

# 3.
#print("     ")
#print(data.isnull().sum())

# 4.   Discipline
data['Discipline'] = data['Discipline'].fillna(data['Sport'])

# 5.
years = [col for col in data.columns if col.isdigit() or col.endswith('*')]

# 6.
data_long = data.melt(id_vars=['Sport', 'Discipline', 'Code', 'Sports Governing␣
 ↪Body'],
                      value_vars=years,
                      var_name='Year',
                      value_name='Events')

# 7.
data_long['Year'] = data_long['Year'].str.replace('*', '').astype(int)

# 8.   Events
def clean_events(value):
    if isinstance(value, str):
        #
        cleaned_value = re.sub(r'[^0-9]', '', value)
        return float(cleaned_value) if cleaned_value.isdigit() else np.nan
    return value

data_long['Events'] = data_long['Events'].apply(clean_events)

# 9.  1924     Skating   Ice Hockey      0
mask = (data_long['Year'] >= 1924) & (data_long['Sport'].isin(['Skating', 'Ice␣
 ↪Hockey']))
data_long.loc[mask, 'Events'] = 0

# 10.
for sport, group in data_long.groupby('Sport'):
    #
    known_data = group.dropna(subset=['Events'])
    missing_data = group[group['Events'].isna()]

    if not known_data.empty and not missing_data.empty:
        #
        X_known = known_data[['Year']]
        y_known = known_data['Events']

        #
        if len(y_known) < 5:
            print(f"     '{sport}'        KNN      ")

            #
            if len(y_known) >= 3:  #   3
                model = LinearRegression()
                model.fit(X_known, y_known)
                predicted_events = model.predict(missing_data[['Year']])
            else:  #   KNN K=1
                model = KNeighborsRegressor(n_neighbors=1)
                model.fit(X_known, y_known)
                predicted_events = model.predict(missing_data[['Year']])

            #
            predicted_events = np.round(predicted_events).astype(int)

            #     Pandas Series
            predicted_series = pd.Series(predicted_events, index=missing_data.
 ↪index)

            #
            data_long.loc[data_long['Sport'] == sport, 'Events'] = data_long.
↪loc[data_long['Sport'] == sport, 'Events'].fillna(predicted_series)
        else:
            #
            model = RandomForestRegressor(n_estimators=100, random_state=42)
            model.fit(X_known, y_known)

            #
            X_missing = missing_data[['Year']]
            predicted_events = model.predict(X_missing)

            #
            predicted_events = np.round(predicted_events).astype(int)

            #      Pandas Series
            predicted_series = pd.Series(predicted_events, index=missing_data.
↪index)

            #
            data_long.loc[data_long['Sport'] == sport, 'Events'] = data_long.
↪loc[data_long['Sport'] == sport, 'Events'].fillna(predicted_series)

            #
            print(f"   '{sport}'        {len(predicted_events)}    ")
    else:
        print(f"   '{sport}'          ")

# 11.
data_filled = data_long.pivot_table(index=['Sport', 'Discipline', 'Code',␣
↪'Sports Governing Body'],
                                    columns='Year',
                                    values='Events',
                                    aggfunc='first').reset_index()

# 12.
print("\n    ")
print(data_filled.head())

# 13.      CSV
output_path = 'Generated\\summerOly_programs_filled.csv'
data_filled.to_csv(output_path, index=False, encoding='utf-8')  #
print(f"      {output_path}")
```

```
  'Aquatics'
  'Archery'
  'Athletics'
  'Badminton'       2
```

```
'Baseball and Softball'        8
'Basketball'        2
'Basque Pelota'        4
'Boxing'
'Breaking'
'Canoeing'        1
'Cricket'
'Croquet'
'Cycling'
'Equestrian'
'Fencing'
'Field hockey'
'Flag football'
'Football'
'Golf'
'Gymnastics'
'Handball'        1
'Ice Hockey'
'Jeu de Paume'
'Judo'
'Karate'
'Lacrosse'        3
'Modern Pentathlon'
'Polo'
'Rackets'
'Roque'
'Rowing'
'Rugby'
'Sailing'
'Shooting'
'Skateboarding'
'Skating'
'Sport Climbing'
'Squash'
'Surfing'
'Table Tennis'
'Taekwondo'        2
'Tennis'        2
'Total disciplines'
'Total events'
'Total sports'
'Triathlon'
'Tug of War'
'Volleyball'        1
'Water Motorsports'        1
'Weightlifting'
'Wrestling'
```

```
Year     Sport         Discipline Code Sports Governing Body  1896  1900  \
0     Aquatics  Artistic Swimming  SWA        World Aquatics   0.0   0.0
1     Aquatics             Diving  DIV        World Aquatics   0.0   0.0
2     Aquatics  Marathon Swimming  OWS        World Aquatics   0.0   0.0
3     Aquatics           Swimming  SWM        World Aquatics   4.0   7.0
4     Aquatics         Water Polo  WPO        World Aquatics   0.0   1.0

Year  1904  1906  1908  1912  …  1988  1992  1996  2000  2004  2008  2012  \
0      0.0   0.0   0.0   0.0  …   2.0   2.0   1.0   2.0   2.0   2.0   2.0
1      2.0   1.0   2.0   4.0  …   4.0   4.0   4.0   8.0   8.0   8.0   8.0
2      0.0   0.0   0.0   0.0  …   0.0   0.0   0.0   0.0   0.0   2.0   2.0
3      9.0   4.0   6.0   9.0  …  31.0  31.0  32.0  32.0  32.0  32.0  32.0
4      1.0   0.0   1.0   1.0  …   1.0   1.0   1.0   2.0   2.0   2.0   2.0

Year  2016  2020  2024
0      2.0   2.0   2.0
1      8.0   8.0   8.0
2      2.0   2.0   2.0
3     32.0  35.0  35.0
4      2.0   2.0   2.0

[5 rows x 35 columns]
      Generated\summerOly_programs_filled.csv
```

2.0.3 Medal_counts

```
[111]: # 2.
       #
       data = medal_counts[['Year', 'NOC', 'Gold', 'Silver', 'Bronze', 'Total']]

       # 3.
       years = data['Year'].unique()
       noc = data['NOC'].unique()

       # 4.
       def generate_table(data, column_name):
           #     DataFrame
           table = pd.DataFrame(index=noc, columns=years)

           #
           for index, row in data.iterrows():
               year = row['Year']
               country = row['NOC']
               value = row[column_name]
               table.at[country, year] = value
```

```
    #        0
    table = table.infer_objects(copy=False).fillna(0).astype(int)

    return table

# 5.
gold_table = generate_table(data, 'Gold')
silver_table = generate_table(data, 'Silver')
bronze_table = generate_table(data, 'Bronze')
total_table = generate_table(data, 'Total')

# 6.     CSV
gold_table.to_csv('Generated\\summerOly_gold_summary.csv')
silver_table.to_csv('Generated\\summerOly_silver_summary.csv')
bronze_table.to_csv('Generated\\summerOly_bronze_summary.csv')
total_table.to_csv('Generated\\summerOly_total_summary.csv')
```

[112]:
```
# 7.
print("   ")
print(gold_table)
```

```
                      1896  1900  1904  1908  1912  1920  1924  1928  1932  \
United States           11    19    76    23    26    41    45    22     0
Greece                  10     0     1     0     1     0     0     0     0
Germany                  6     4     4     3     5     0     0    10     0
France                   5    27     0     5     7     9    13     6     0
Great Britain            2    15     1    56    10    14     9     3     0
…                      …   …   …   …   …   …   …   …   …
Saint Lucia              0     0     0     0     0     0     0     0     0
Dominica                 0     0     0     0     0     0     0     0     0
Albania                  0     0     0     0     0     0     0     0     0
Cabo Verde               0     0     0     0     0     0     0     0     0
Refugee Olympic Team     0     0     0     0     0     0     0     0     0

                      1936  …  1988  1992  1996  2000  2004  2008  2012  \
United States           24  …    36    37    44    37    36    36    48
Greece                   0  …     0     2     4     4     6     0     0
Germany                 38  …     0    33    20    13    13    16    11
France                   7  …     6     8    15    13    11     7    11
Great Britain            4  …     5     5     1    11     9    19    29
…                      …  …   …   …   …   …   …   …   …
Saint Lucia              0  …     0     0     0     0     0     0     0
Dominica                 0  …     0     0     0     0     0     0     0
Albania                  0  …     0     0     0     0     0     0     0
Cabo Verde               0  …     0     0     0     0     0     0     0
Refugee Olympic Team     0  …     0     0     0     0     0     0     0
```

```
                      2016  2020  2024
United States           46    39    40
Greece                   3     2     1
Germany                 17    10    12
France                  10    10    16
Great Britain           27    22    14
…                     …    …    …
Saint Lucia              0     0     1
Dominica                 0     0     1
Albania                  0     0     0
Cabo Verde               0     0     0
Refugee Olympic Team     0     0     0

[210 rows x 30 columns]
```

[113]:
```
print("\n   ")
print(silver_table)
```

```
                      1896  1900  1904  1908  1912  1920  1924  1928  1932  \
United States            7    14    78    12    19    27    27    18     0
Greece                  18     0     0     3     0     1     0     0     0
Germany                  5     3     5     5    13     0     0     7     0
France                   4    39     1     5     4    19    15    10     0
Great Britain            3     7     1    51    15    15    13    10     0
…                     …    …    …    …    …    …    …    …    …
Saint Lucia              0     0     0     0     0     0     0     0     0
Dominica                 0     0     0     0     0     0     0     0     0
Albania                  0     0     0     0     0     0     0     0     0
Cabo Verde               0     0     0     0     0     0     0     0     0
Refugee Olympic Team     0     0     0     0     0     0     0     0     0

                      1936  …  1988  1992  1996  2000  2004  2008  2012  \
United States           21  …    31    34    32    24    39    39    26
Greece                   0  …     0     0     4     6     6     2     0
Germany                 31  …     0    21    18    17    16    11    20
France                   6  …     4     5     7    14     9    16    11
Great Britain            7  …    10     3     8    10     9    13    18
…                     …  …   …    …    …    …    …    …    …
Saint Lucia              0  …     0     0     0     0     0     0     0
Dominica                 0  …     0     0     0     0     0     0     0
Albania                  0  …     0     0     0     0     0     0     0
Cabo Verde               0  …     0     0     0     0     0     0     0
Refugee Olympic Team     0  …     0     0     0     0     0     0     0

                      2016  2020  2024
United States           37    41    44
```

```
Greece                       1      1      1
Germany                     10     11     13
France                      18     12     26
Great Britain               23     20     22
…                     …    …    …
Saint Lucia                  0      0      1
Dominica                     0      0      0
Albania                      0      0      0
Cabo Verde                   0      0      0
Refugee Olympic Team         0      0      0

[210 rows x 30 columns]
```

[114]:
```
print("\n   ")
print(bronze_table)
```

```
                      1896  1900  1904  1908  1912  1920  1924  1928  1932  \
United States            2    15    77    12    19    27    27    16     0
Greece                  19     0     1     1     1     0     0     0     0
Germany                  2     2     6     5     7     0     0    14     0
France                   2    37     0     9     3    13    10     5     0
Great Britain            2     9     0    39    16    13    12     7     0
…                     …    …    …    …    …    …    …    …    …
Saint Lucia              0     0     0     0     0     0     0     0     0
Dominica                 0     0     0     0     0     0     0     0     0
Albania                  0     0     0     0     0     0     0     0     0
Cabo Verde               0     0     0     0     0     0     0     0     0
Refugee Olympic Team     0     0     0     0     0     0     0     0     0

                      1936  …  1988  1992  1996  2000  2004  2008  2012  \
United States           12  …    27    37    25    32    26    37    30
Greece                   0  …     1     0     0     3     4     1     2
Germany                 32  …     0    28    27    26    20    14    13
France                   6  …     6    16    15    11    13    20    13
Great Britain            3  …     9    12     6     7    12    19    18
…                     …  …  …    …    …    …    …    …    …
Saint Lucia              0  …     0     0     0     0     0     0     0
Dominica                 0  …     0     0     0     0     0     0     0
Albania                  0  …     0     0     0     0     0     0     0
Cabo Verde               0  …     0     0     0     0     0     0     0
Refugee Olympic Team     0  …     0     0     0     0     0     0     0

                      2016  2020  2024
United States           38    33    42
Greece                   2     1     6
Germany                 15    16     8
```

```
France                 14   11   22
Great Britain          17   22   29
…                    …   …   …
Saint Lucia             0    0    0
Dominica                0    0    0
Albania                 0    0    2
Cabo Verde              0    0    1
Refugee Olympic Team    0    0    1

[210 rows x 30 columns]
```

[115]:
```python
print("\n   ")
print(total_table)
```

```

                      1896  1900  1904  1908  1912  1920  1924  1928  1932  \
United States           20    48   231    47    64    95    99    56     0
Greece                  47     0     2     4     2     1     0     0     0
Germany                 13     9    15    13    25     0     0    31     0
France                  11   103     1    19    14    41    38    21     0
Great Britain            7    31     2   146    41    42    34    20     0
…                    …   …   …   …   …   …   …   …   …
Saint Lucia              0     0     0     0     0     0     0     0     0
Dominica                 0     0     0     0     0     0     0     0     0
Albania                  0     0     0     0     0     0     0     0     0
Cabo Verde               0     0     0     0     0     0     0     0     0
Refugee Olympic Team     0     0     0     0     0     0     0     0     0

                      1936  …  1988  1992  1996  2000  2004  2008  2012  \
United States           57  …    94   108   101    93   101   112   104
Greece                   0  …     1     2     8    13    16     3     2
Germany                101  …     0    82    65    56    49    41    44
France                  19  …    16    29    37    38    33    43    35
Great Britain           14  …    24    20    15    28    30    51    65
…                    …  …  …   …   …   …   …   …   …
Saint Lucia              0  …     0     0     0     0     0     0     0
Dominica                 0  …     0     0     0     0     0     0     0
Albania                  0  …     0     0     0     0     0     0     0
Cabo Verde               0  …     0     0     0     0     0     0     0
Refugee Olympic Team     0  …     0     0     0     0     0     0     0

                      2016  2020  2024
United States          121   113   126
Greece                   6     4     8
Germany                 42    37    33
France                  42    33    64
Great Britain           67    64    65
```

```
…                        …    …    …
Saint Lucia                0     0     2
Dominica                   0     0     1
Albania                    0     0     2
Cabo Verde                 0     0     1
Refugee Olympic Team       0     0     1

[210 rows x 30 columns]
```

2.0.4 athletes.csv

[116]:
```python
#   summerOly_athletes.csv
data = olympic_athletes.copy()

#
pivot_df = data.pivot_table(index=['Name', 'Sex', 'Team', 'NOC', 'City',␣
  ↪'Sport', 'Event'],
                                  columns='Year',
                                  values='Medal',
                                  aggfunc='first').reset_index()

#    0
pivot_df = pivot_df.fillna(0)

#
print("      ")
print(pivot_df.head())

#     CSV
output_path = 'Generated\\summerOly_athletes_wide_format.csv'
pivot_df.to_csv(output_path, index=False, encoding='utf-8')
print(f"      {output_path}")
```

```
Year           Name Sex          Team  NOC   City        Sport  \
0      (jr) Larocca   M     Argentina  ARG  Paris   Equestrian
1     . Chadalavada   F         India  IND  Tokyo      Fencing
2            . Deni   M     Indonesia  INA  Tokyo Weightlifting
3               671   F         China  CHN  Paris     Breaking
4         A Alayed   F  Saudi Arabia  KSA  Paris     Swimming

Year                   Event 1896 1900 1904  … 1988 1992 1996 2000 2004  \
0        Jumping Individual    0    0    0  …    0    0    0    0    0
1     Women's Sabre Individual    0    0    0  …    0    0    0    0    0
2                 Men's 67kg    0    0    0  …    0    0    0    0    0
3                    B-Girls    0    0    0  …    0    0    0    0    0
4      Women's 200m Freestyle    0    0    0  …    0    0    0    0    0
```

```
Year 2008 2012 2016      2020      2024
0       0    0    0         0  No medal
1       0    0    0  No medal         0
2       0    0    0  No medal         0
3       0    0    0         0    Bronze
4       0    0    0         0  No medal

[5 rows x 38 columns]
      Generated\summerOly_athletes_wide_format.csv
```

3

3.0.1

[117]:
```python
#   summerOly_athletes.csv
data = olympic_athletes.copy()

#
athlete_years = olympic_athletes[['Name', 'Sex', 'NOC', 'Year']].
 ↪drop_duplicates()

#
athlete_years = athlete_years.sort_values(by=['Name', 'Sex', 'NOC', 'Year'])

#
def count_consecutive_years(group):
    years = group['Year'].values
    consecutive_years = []
    current_count = 1
    for i in range(1, len(years)):
        if years[i] == years[i - 1] + 4:
            current_count += 1
        else:
            consecutive_years.append(current_count)
            current_count = 1
    consecutive_years.append(current_count)
    return pd.Series(consecutive_years)

#
consecutive_years = athlete_years.groupby('Name').
 ↪apply(count_consecutive_years).explode().reset_index()
consecutive_years.columns = ['level_0', 'Name', 'Consecutive_Years']  #
consecutive_years = consecutive_years.drop(columns=['level_0'])  #

#
consecutive_years_count = consecutive_years['Consecutive_Years'].value_counts().
 ↪reset_index()
```

```
consecutive_years_count.columns = ['Consecutive_Years', 'Count']

#
print("          ")
print(consecutive_years_count)

#    CSV
output_path = 'Generated\\consecutive_years_count.csv'
consecutive_years_count.to_csv(output_path, index=False, encoding='utf-8')
print(f"     {output_path}")
```

```
   Consecutive_Years   Count
0                  1  106963
1                  2   23393
2                  3    6086
3                  4    1627
4                  5     391
5                  6      80
6                  7      23
7                  8       5
8                 10       1
9                  9       1
     Generated\consecutive_years_count.csv
```

```
C:\Users\Ziqi\AppData\Local\Temp\ipykernel_23036\2472914887.py:25:
DeprecationWarning: DataFrameGroupBy.apply operated on the grouping columns.
This behavior is deprecated, and in a future version of pandas the grouping
columns will be excluded from the operation. Either pass `include_groups=False`
to exclude the groupings or explicitly select the grouping columns after groupby
to silence this warning.
  consecutive_years = athlete_years.groupby('Name').apply(count_consecutive_year
s).explode().reset_index()
```

[118]:
```
#
def calculate_year_gap(group):
    years = group['Year'].sort_values().values
    if len(years) > 1:
        return years[-1] - years[0] + 1
    else:
        return 1  #       1

#
athlete_gaps = athlete_years.groupby('Name').apply(calculate_year_gap).
 ↪reset_index()
athlete_gaps.columns = ['Name', 'Year_Gap']

#
```

```
gap_counts = athlete_gaps['Year_Gap'].value_counts().reset_index()
gap_counts.columns = ['Year_Gap', 'Count']

#   Year_Gap
gap_counts = gap_counts.sort_values(by='Year_Gap')

#
print("                ")
print(gap_counts)

#     CSV
output_path = 'Generated\\athlete_year_gaps.csv'
gap_counts.to_csv(output_path, index=False, encoding='utf-8')
print(f"     {output_path}")
```

```
    Year_Gap  Count
0          1  94044
11         3     97
1          5  21426
15         7     69
2          9   8160
30        11     11
3         13   3095
28        15     16
4         17   1082
31        19      9
5         21    476
33        23      4
6         25    270
40        27      1
7         29    213
39        31      2
8         33    133
9         37    113
10        41    101
46        43      1
12        45     95
44        47      1
13        49     74
43        51      1
17        53     59
14        57     69
45        59      1
16        61     64
18        65     58
37        67      2
19        69     51
```

```
20        73     27
21        77     23
24        81     19
23        85     19
42        87      1
22        89     20
47        91      1
27        93     16
25        97     18
29       101     14
41       103      1
26       105     17
38       107      2
34       109      3
32       113      8
36       117      2
35       121      3
     Generated\athlete_year_gaps.csv
```

```
C:\Users\Ziqi\AppData\Local\Temp\ipykernel_23036\3557634001.py:10:
DeprecationWarning: DataFrameGroupBy.apply operated on the grouping columns.
This behavior is deprecated, and in a future version of pandas the grouping
columns will be excluded from the operation. Either pass `include_groups=False`
to exclude the groupings or explicitly select the grouping columns after groupby
to silence this warning.
  athlete_gaps =
athlete_years.groupby('Name').apply(calculate_year_gap).reset_index()
```