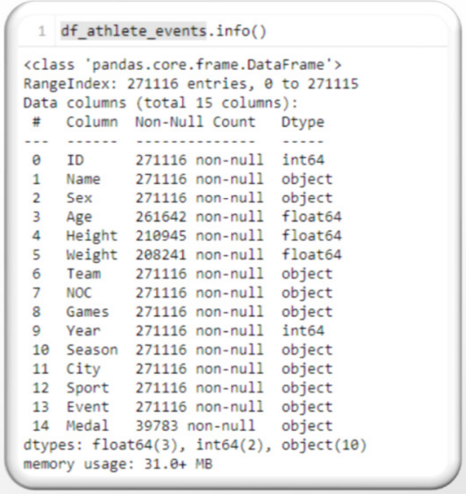
SPORTSSTATS

OLYMPICS DATASET - 120 YEARS OF DATA

SCHEMAS

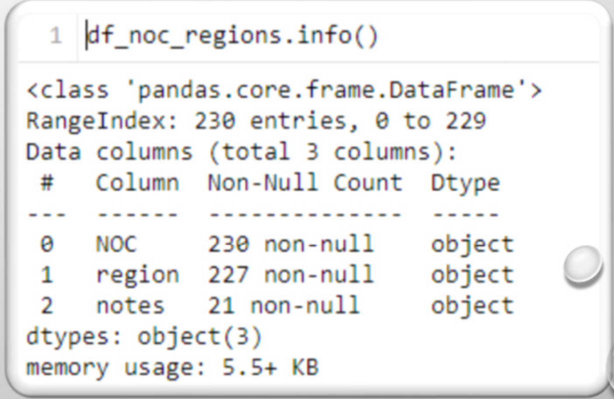
Number of records athlete_events: 271116

Number of records noc_regions: 230



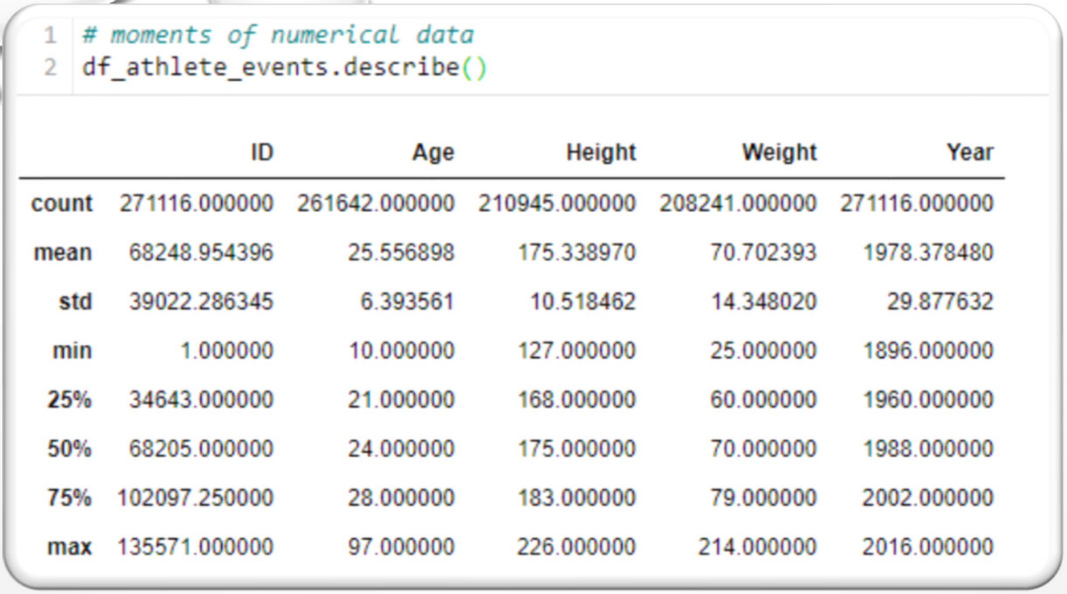
```
df_athlete_events.info()
```

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



```
df_noc_regions.info()
```

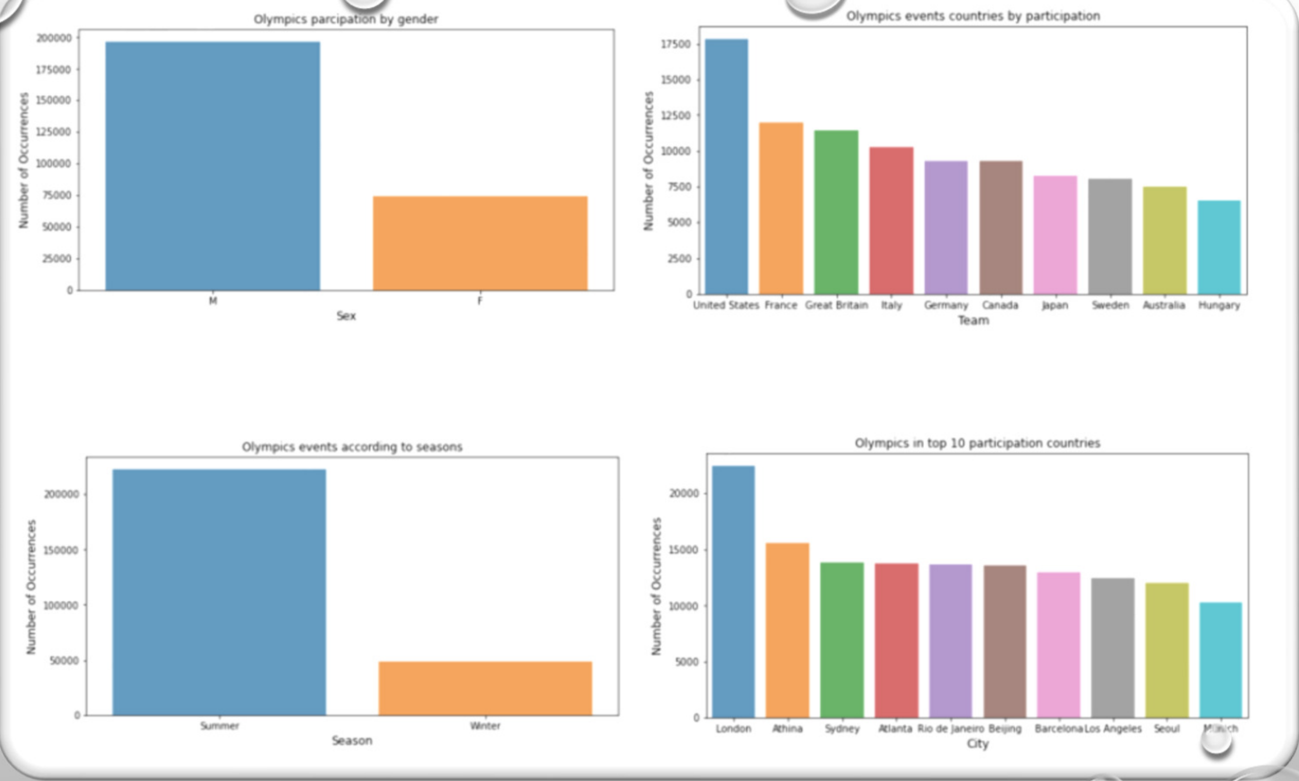
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 230 entries, 0 to 229
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   NOC     230 non-null    object
 1   region  227 non-null    object
 2   notes   21 non-null     object
dtypes: object(3)
memory usage: 5.5+ KB
```



```
# moments of numerical data
df_athlete_events.describe()
```

	ID	Age	Height	Weight	Year
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

DESCRIPTIVE STATS FOR AGE, HEIGHT AND WEIGHT.



1. SHOWS THE MALE VS FEMALE PARTICIPATION

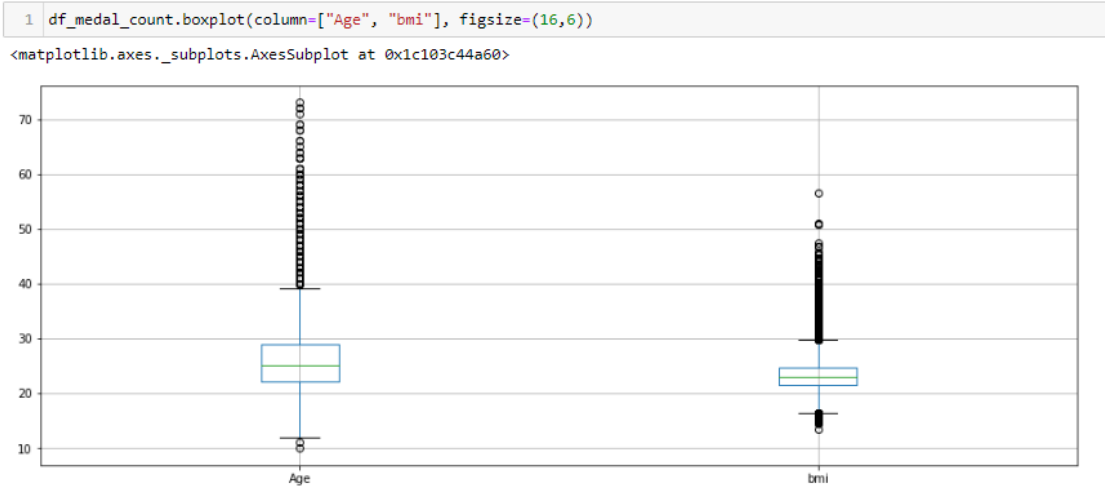
2. SPORTS SEASON DISTRIBUTION

3. CITY IN WHICH THE SPORTS HELD

HYPOTHESIS

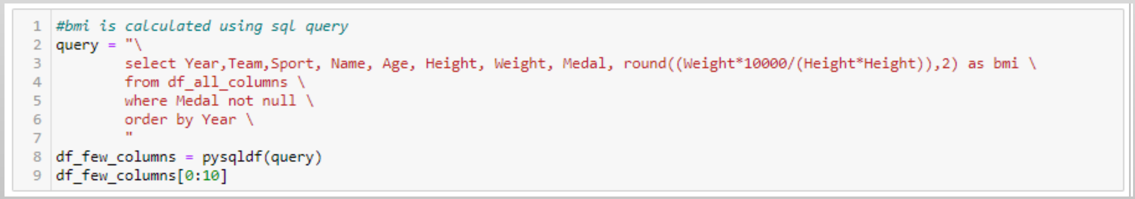
1. THE TEAMS WHO PARTICIPATE MORE OLYMPIC GAMES WINS MORE MEDALS IN LATER GAMES.

2. THE MOTIVATION OUTWEIGHS PHYSICAL CHARACTERISTICS TO WIN THE GAME IS THE HYPOTHESIS.



```
df_medal_count.boxplot(column=["Age", "bmi"], figsize=(16,6))
```

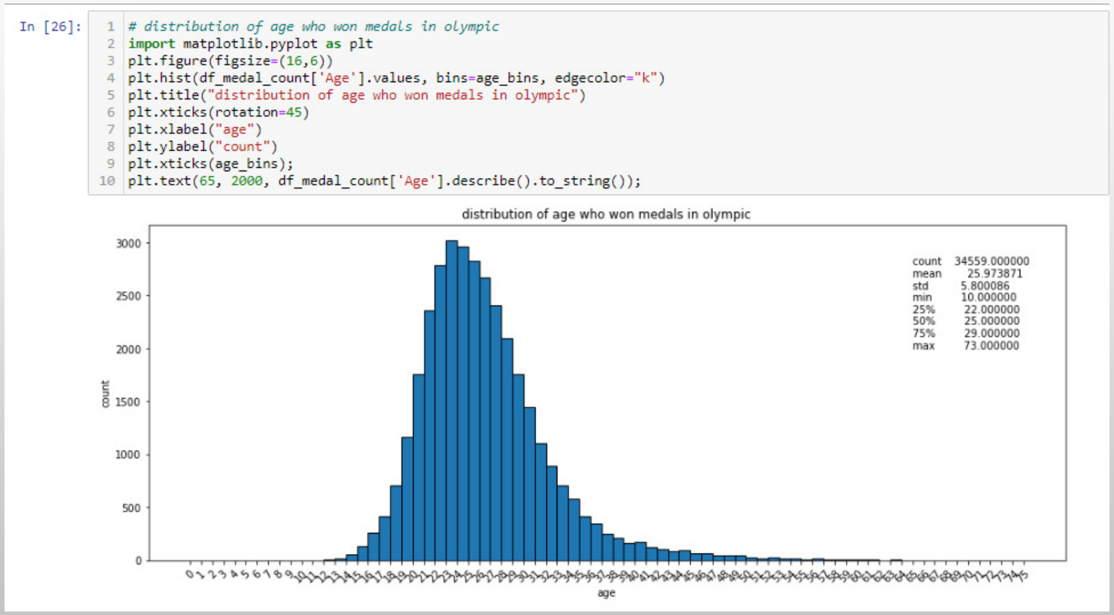
```
<matplotlib.axes._subplots.AxesSubplot at 0x1c103c44a60>
```



```
#bmi is calculated using sql query
query = "\
        select Year,Team,Sport, Name, Age, Height, Weight, Medal, round((Weight*10000/(Height*Height)),2) as bmi \
        from df_all_columns \
        where Medal not null \
        order by Year \
        "
df_few_columns = pysqldf(query)
df_few_columns[0:10]
```

- THE BOX PLOT FOR AGE AND BMI
- THE BMI FIELD IS CALCULATED USING STANDARD FORMULA WITHIN SQI QUERY.
- BMI = WEIGHT/HEIGHT**2

DISTRIBUTION OF AGE AND WINNING A MEDAL, THE DISTRIBUTION IS RIGHT TAILED.



In [26]:

```
# distribution of age who won medals in olympic
import matplotlib.pyplot as plt
plt.figure(figsize=(16,6))
plt.hist(df_medal_count['Age'].values, bins=age_bins, edgecolor="k")
plt.title("distribution of age who won medals in olympic")
plt.xticks(rotation=45)
plt.xlabel("age")
plt.ylabel("count")
plt.xticks(age_bins);
plt.text(65, 2000, df_medal_count['Age'].describe().to_string());
```

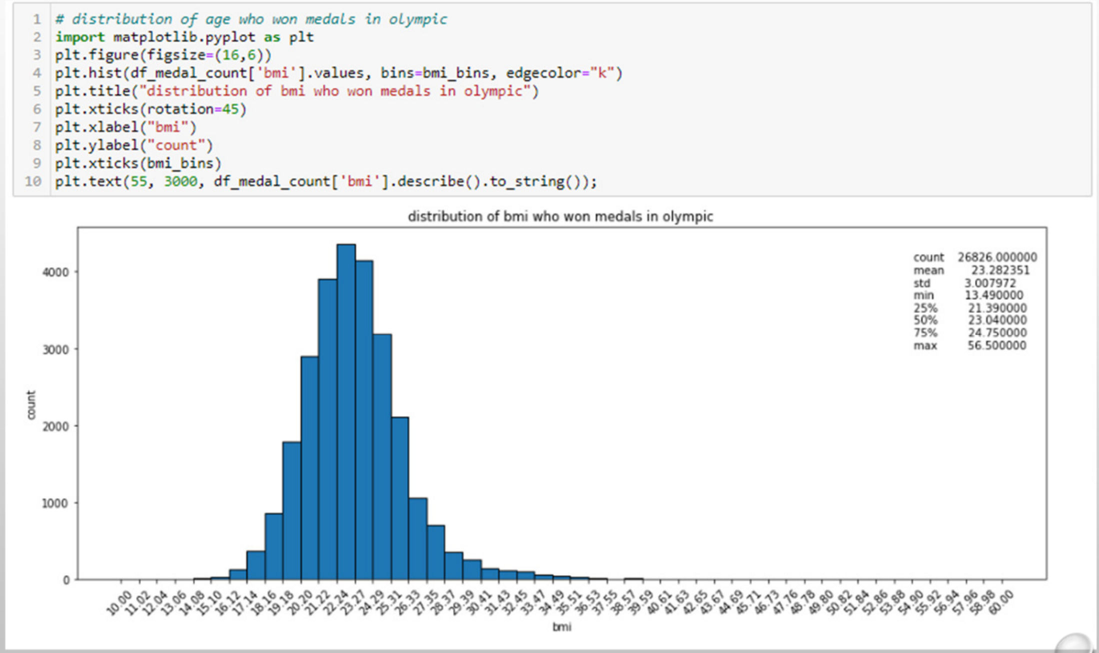
distribution of age who won medals in olympic

```
count    34559.000000
mean        25.973871
std          5.800086
min         10.000000
25%         22.000000
50%         25.000000
75%         29.000000
max         73.000000
```

- INTERESTING TO NOTE THAT THE YOUNGEST WINNER IS 10 YEAR OLD AND OLDEST WINNER IS 73 YEAR OLD

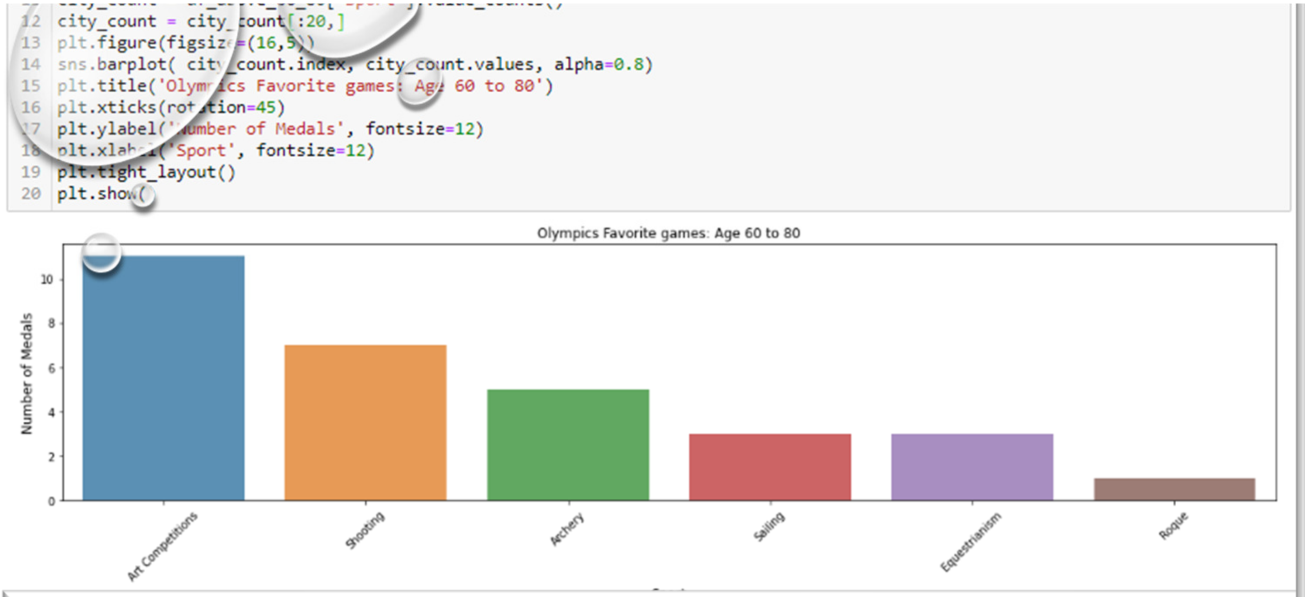
DISTRIBUTION OF BMI AND WINNING A MEDAL, THIS DISTRIBUTION ALSO RIGHT TAILED.

- THERE IS A GAME FOR EVERYONE I.E, UNDERWEIGHT, NORMAL, OVERWEIGHT, AND OBESE.
- FROM BMI 13 UNDERWEIGHT TO BMI 56 OBESE HAS GAMES TO WIN

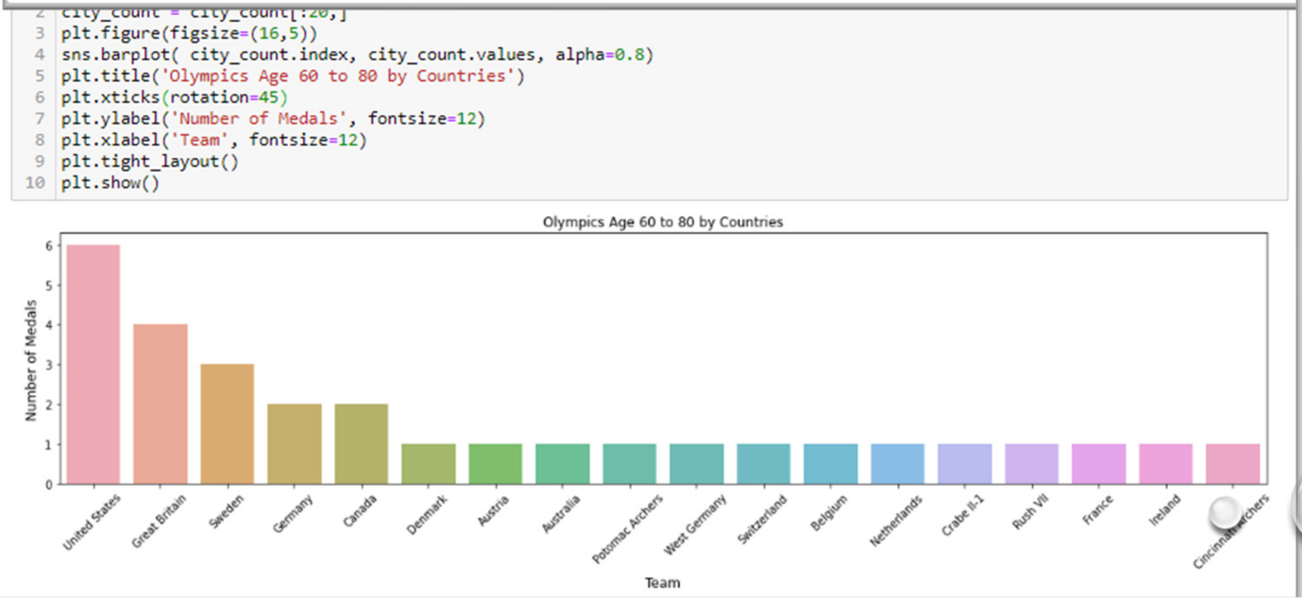


```python
# distribution of age who won medals in olympic
import matplotlib.pyplot as plt
plt.figure(figsize=(16,6))
plt.hist(df_medal_count['bmi'].values, bins=bmi_bins, edgecolor="k")
plt.title("distribution of bmi who won medals in olympic")
plt.xticks(rotation=45)
plt.xlabel("bmi")
plt.ylabel("count")
plt.xticks(bmi_bins)
plt.text(55, 3000, df_medal_count['bmi'].describe().to_string());
```

```
count    26826.000000
mean        23.282351
std          3.007972
min         13.490000
25%         21.390000
50%         23.040000
75%         24.750000
max         56.500000
```



```
city_count = city_count[:20,]
plt.figure(figsize=(16,5))
sns.barplot( city_count.index, city_count.values, alpha=0.8)
plt.title('Olympics Favorite games: Age 60 to 80')
plt.xticks(rotation=45)
plt.ylabel('Number of Medals', fontsize=12)
plt.xlabel('Sport', fontsize=12)
plt.tight_layout()
plt.show()
```

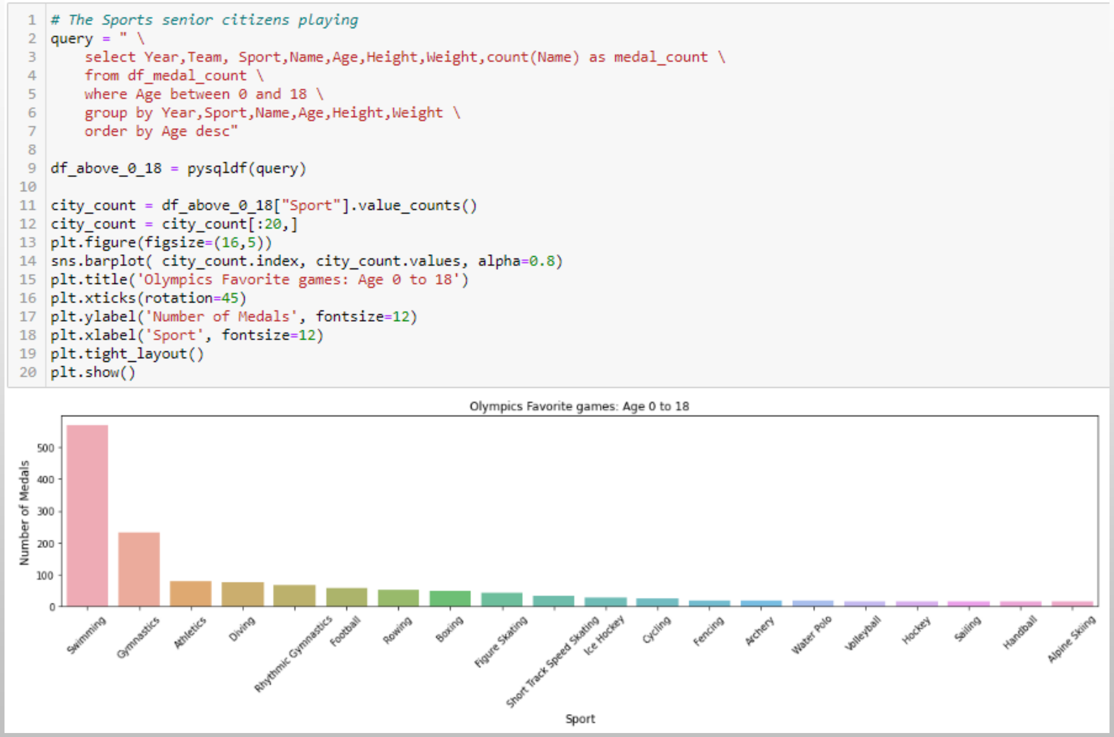


```
city_count = city_count[:20,]
plt.figure(figsize=(16,5))
sns.barplot( city_count.index, city_count.values, alpha=0.8)
plt.title('Olympics Age 60 to 80 by Countries')
plt.xticks(rotation=45)
plt.ylabel('Number of Medals', fontsize=12)
plt.xlabel('Team', fontsize=12)
plt.tight_layout()
plt.show()
```

FROM AGE 60-80 YEARS OLD PARTICIPATED IN GAMES AND WON THE MEDALS

- THE COUNTRIES THEY REPRESENTED ALSO SHOWN.
- THE OLDEST IS 73 YEAR OLD

TEENAGER’S ALSO PLAYED IN FOLLOWING GAMES

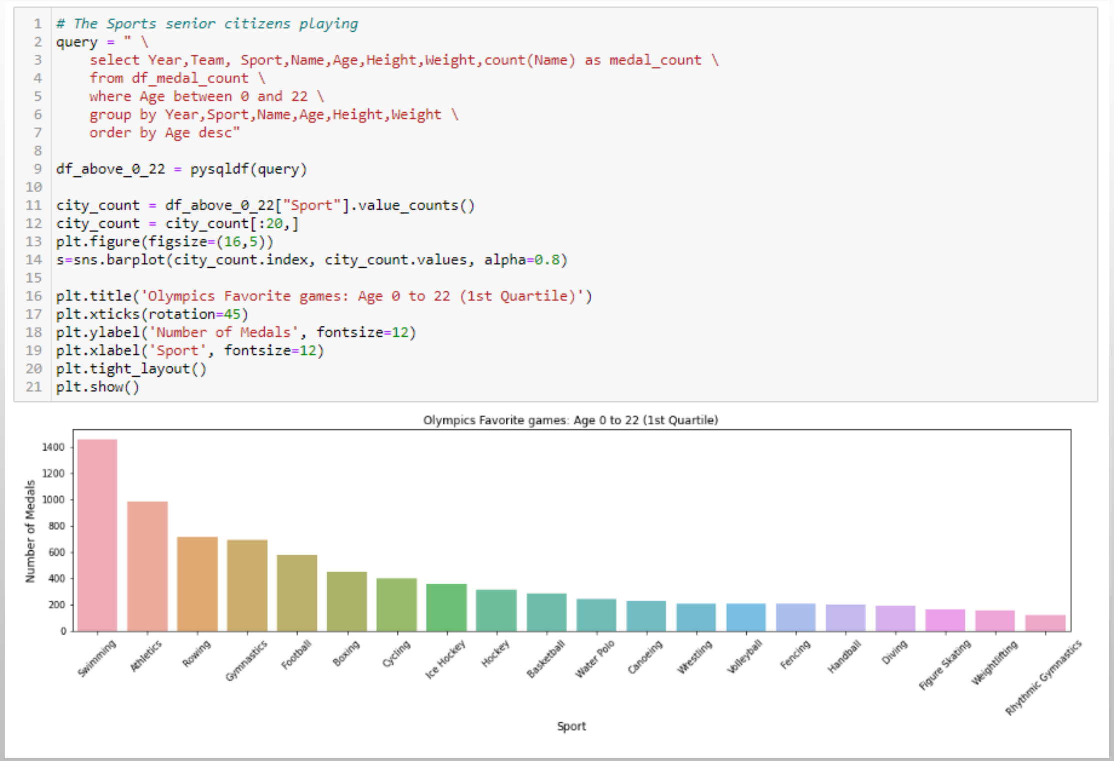


```
# The Sports senior citizens playing
query = " \
    select Year,Team, Sport,Name,Age,Height,Weight,count(Name) as medal_count \
    from df_medal_count \
    where Age between 0 and 18 \
    group by Year,Sport,Name,Age,Height,Weight \
    order by Age desc"

df_above_0_18 = pysqldf(query)

city_count = df_above_0_18["Sport"].value_counts()
city_count = city_count[:20,]
plt.figure(figsize=(16,5))
sns.barplot( city_count.index, city_count.values, alpha=0.8)
plt.title('Olympics Favorite games: Age 0 to 18')
plt.xticks(rotation=45)
plt.ylabel('Number of Medals', fontsize=12)
plt.xlabel('Sport', fontsize=12)
plt.tight_layout()
plt.show()
```

- LESS THAN 18 YEARS OLD.
- THE YOUNGEST IS 10 YEAR OLD.
- THE SPORTS THEY PARTICIPATED AS SEEN I DIAGRAM.

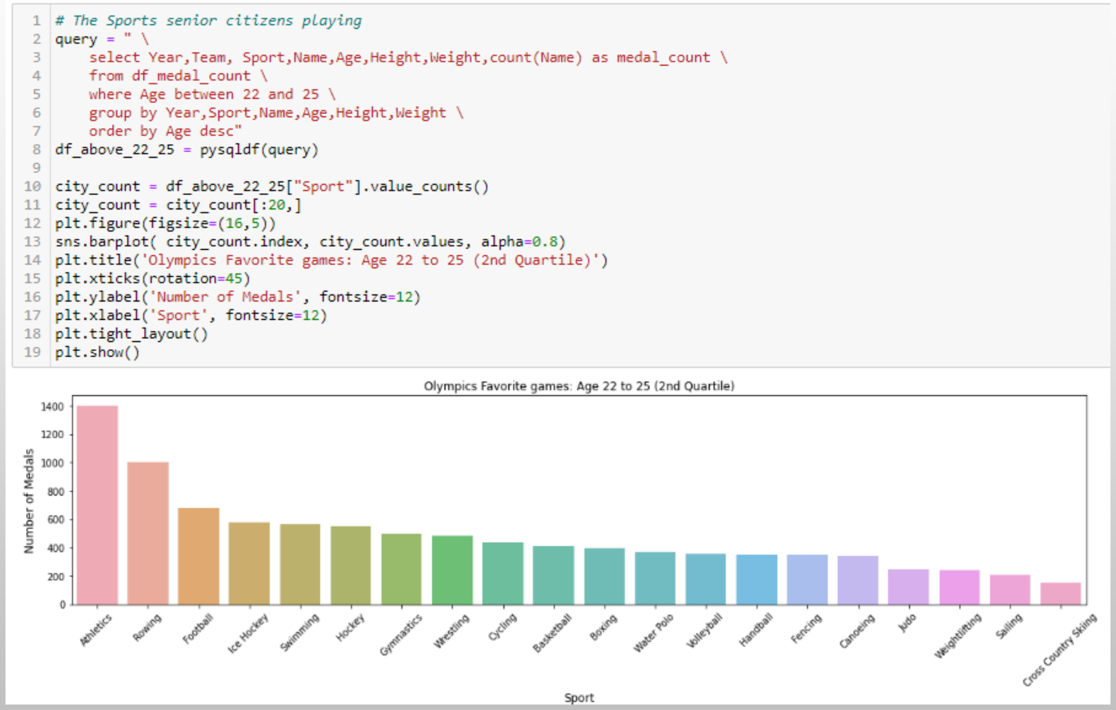
AGE DISTRIBUTION OF WINNING

THE 1ST QUARTILE OF AGE GROUP WON IN FOLLOWING GAMES



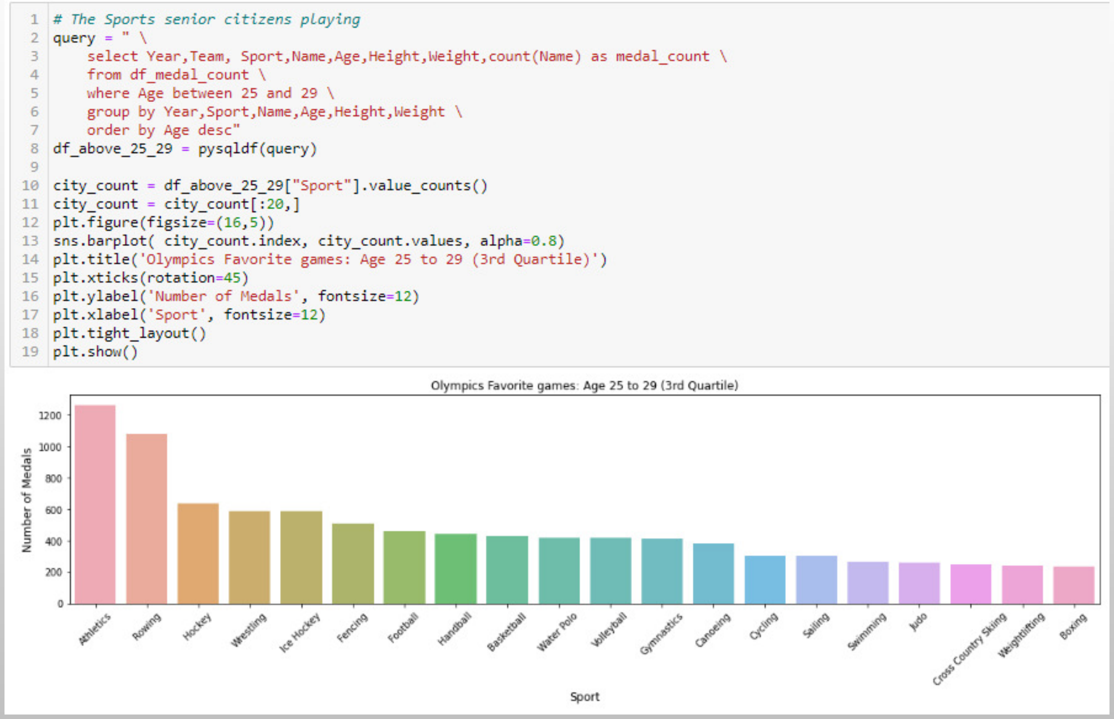
```
# The Sports senior citizens playing
query = " \
    select Year,Team, Sport,Name,Age,Height,Weight,count(Name) as medal_count \
    from df_medal_count \
    where Age between 0 and 22 \
    group by Year,Sport,Name,Age,Height,Weight \
    order by Age desc"

df_above_0_22 = pysqldf(query)

city_count = df_above_0_22["Sport"].value_counts()
city_count = city_count[:20,]
plt.figure(figsize=(16,5))
s=sns.barplot(city_count.index, city_count.values, alpha=0.8)

plt.title('Olympics Favorite games: Age 0 to 22 (1st Quartile)')
plt.xticks(rotation=45)
plt.ylabel('Number of Medals', fontsize=12)
plt.xlabel('Sport', fontsize=12)
plt.tight_layout()
plt.show()
```

2ND QUARTILE OF AGE GROUP WON FOLLOWING GAMES



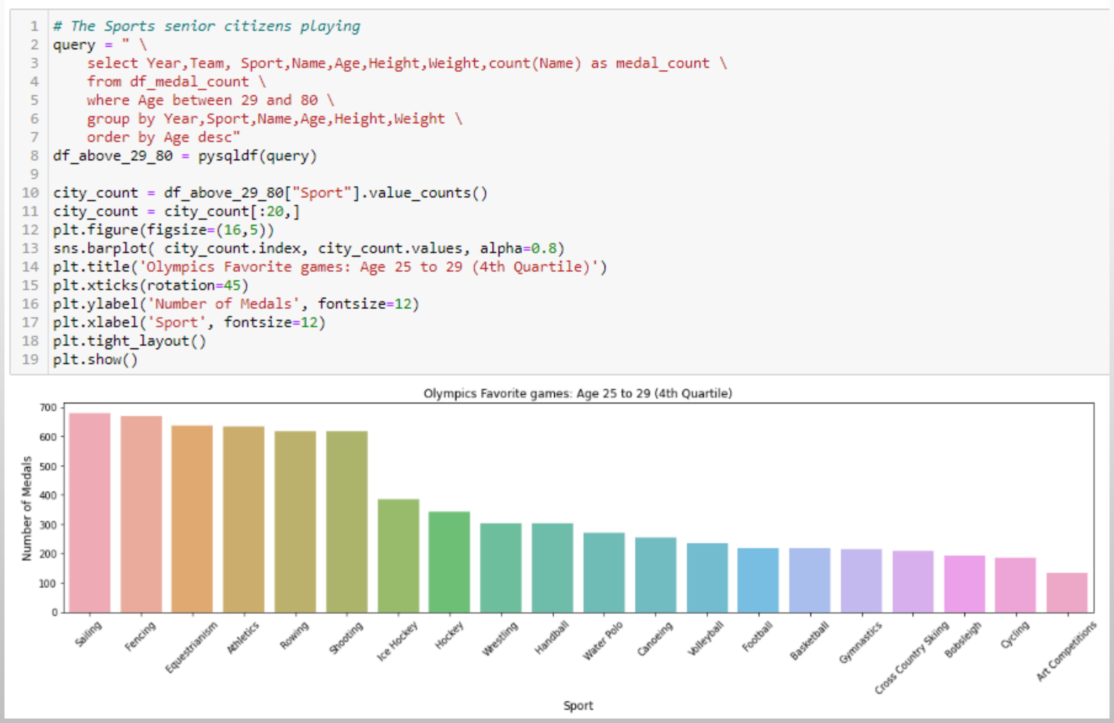
```
# The Sports senior citizens playing
query = " \
    select Year,Team, Sport,Name,Age,Height,Weight,count(Name) as medal_count \
    from df_medal_count \
    where Age between 22 and 25 \
    group by Year,Sport,Name,Age,Height,Weight \
    order by Age desc"
df_above_22_25 = pysqldf(query)

city_count = df_above_22_25["Sport"].value_counts()
city_count = city_count[:20,]
plt.figure(figsize=(16,5))
sns.barplot( city_count.index, city_count.values, alpha=0.8)
plt.title('Olympics Favorite games: Age 22 to 25 (2nd Quartile)')
plt.xticks(rotation=45)
plt.ylabel('Number of Medals', fontsize=12)
plt.xlabel('Sport', fontsize=12)
plt.tight_layout()
plt.show()
```

3RD QUARTILE OF AGE GROUP WON FOLLOWING GAMES



```
# The Sports senior citizens playing
query = " \
    select Year,Team, Sport,Name,Age,Height,Weight,count(Name) as medal_count \
    from df_medal_count \
    where Age between 25 and 29 \
    group by Year,Sport,Name,Age,Height,Weight \
    order by Age desc"
df_above_25_29 = pysqldf(query)

city_count = df_above_25_29["Sport"].value_counts()
city_count = city_count[:20,]
plt.figure(figsize=(16,5))
sns.barplot( city_count.index, city_count.values, alpha=0.8)
plt.title('Olympics Favorite games: Age 25 to 29 (3rd Quartile)')
plt.xticks(rotation=45)
plt.ylabel('Number of Medals', fontsize=12)
plt.xlabel('Sport', fontsize=12)
plt.tight_layout()
plt.show()
```

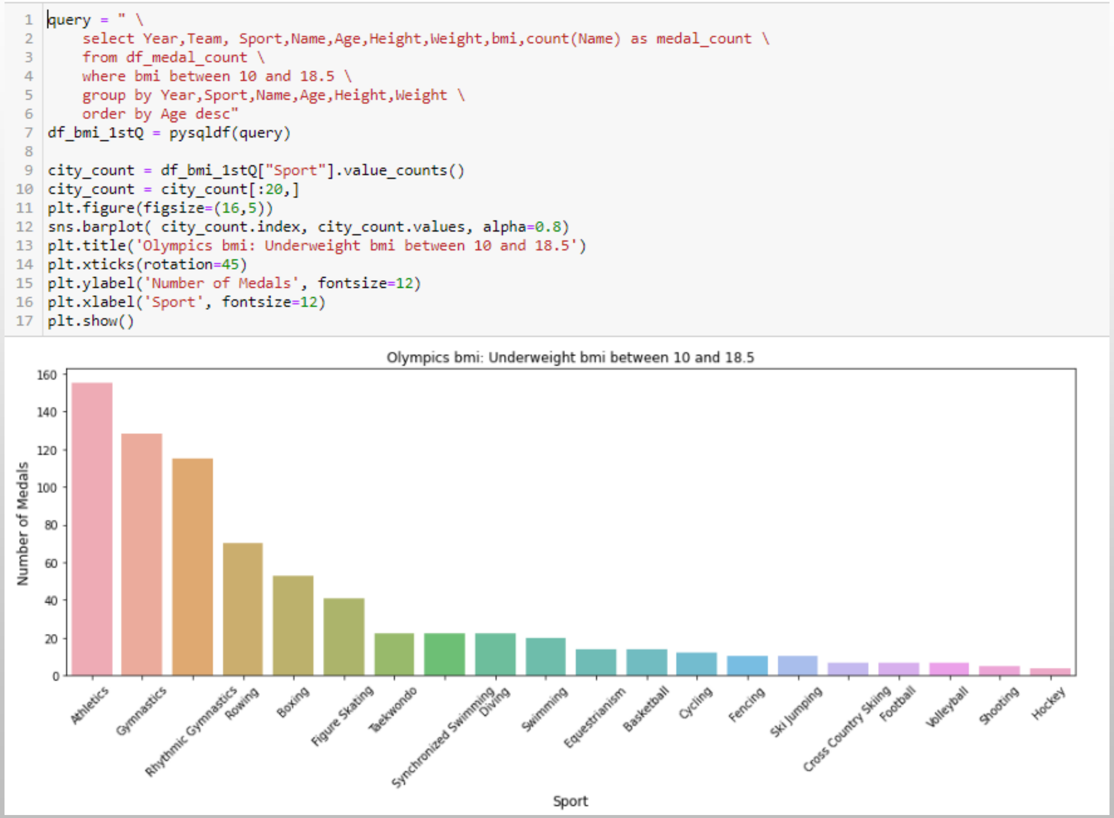
THE 4TH QUARTILE WON IN FOLLOWING GAMES



```
# The Sports senior citizens playing
query = " \
    select Year,Team, Sport,Name,Age,Height,Weight,count(Name) as medal_count \
    from df_medal_count \
    where Age between 29 and 80 \
    group by Year,Sport,Name,Age,Height,Weight \
    order by Age desc"
df_above_29_80 = pysqldf(query)

city_count = df_above_29_80["Sport"].value_counts()
city_count = city_count[:20,]
plt.figure(figsize=(16,5))
sns.barplot( city_count.index, city_count.values, alpha=0.8)
plt.title('Olympics Favorite games: Age 25 to 29 (4th Quartile)')
plt.xticks(rotation=45)
plt.ylabel('Number of Medals', fontsize=12)
plt.xlabel('Sport', fontsize=12)
plt.tight_layout()
plt.show()
```

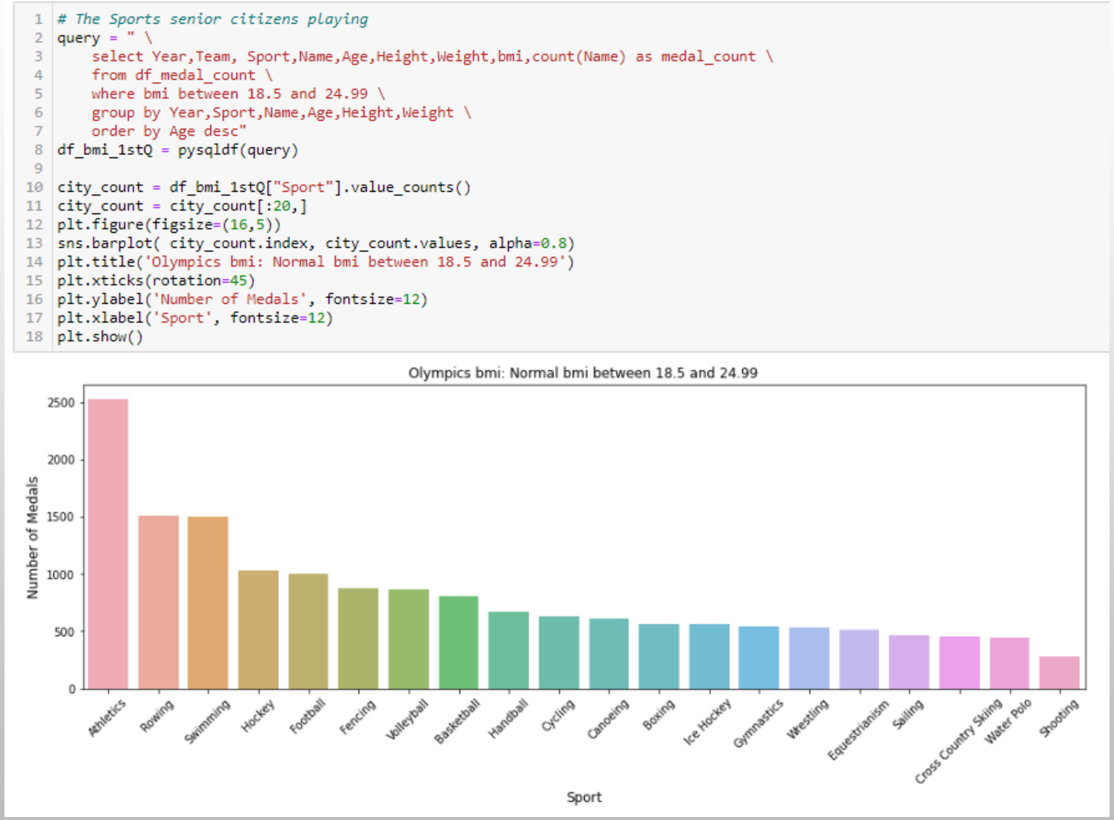
BMI DISTRIBUTION OF WINNING

THE 1ST QUARTILE OF BMI GROUP WON IN FOLLOWING GAMES



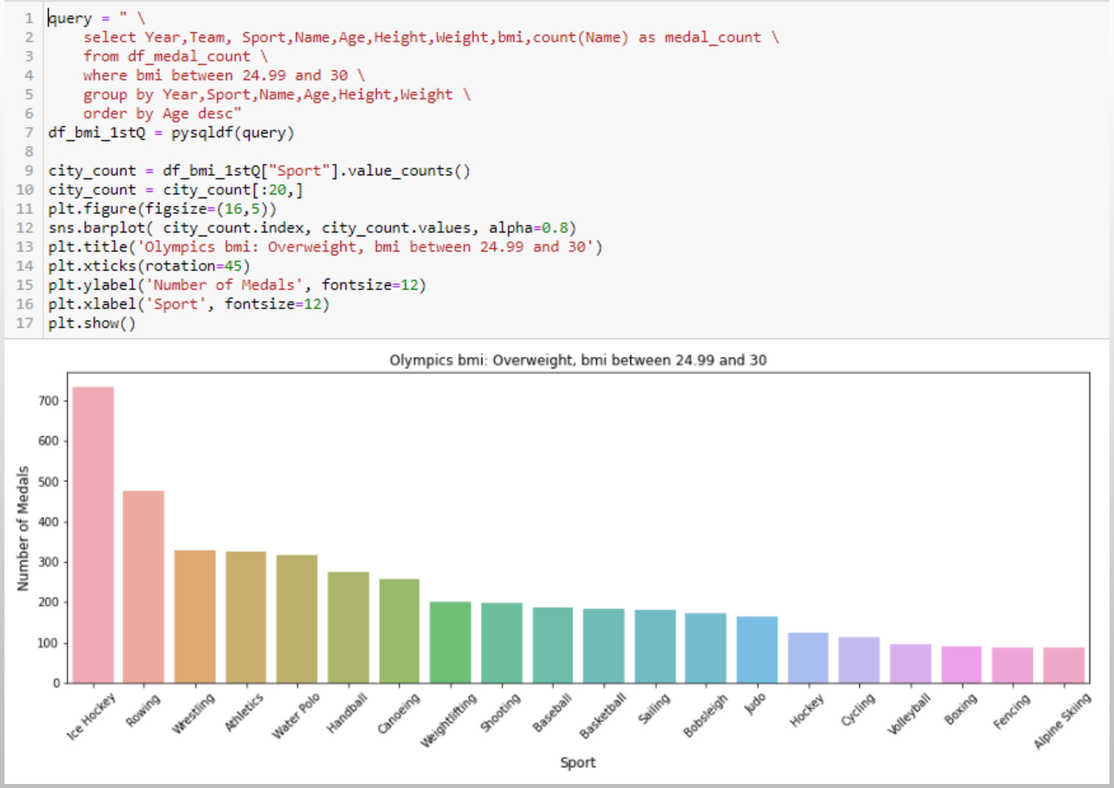
```
query = " \
    select Year,Team, Sport,Name,Age,Height,Weight,bmi,count(Name) as medal_count \
    from df_medal_count \
    where bmi between 10 and 18.5 \
    group by Year,Sport,Name,Age,Height,Weight \
    order by Age desc"
df_bmi_1stQ = pysqldf(query)

city_count = df_bmi_1stQ["Sport"].value_counts()
city_count = city_count[:20,]
plt.figure(figsize=(16,5))
sns.barplot( city_count.index, city_count.values, alpha=0.8)
plt.title('Olympics bmi: Underweight bmi between 10 and 18.5')
plt.xticks(rotation=45)
plt.ylabel('Number of Medals', fontsize=12)
plt.xlabel('Sport', fontsize=12)
plt.show()
```

THE 2ND QUARTILE OF BMI GROUP WON IN FOLLOWING GAMES



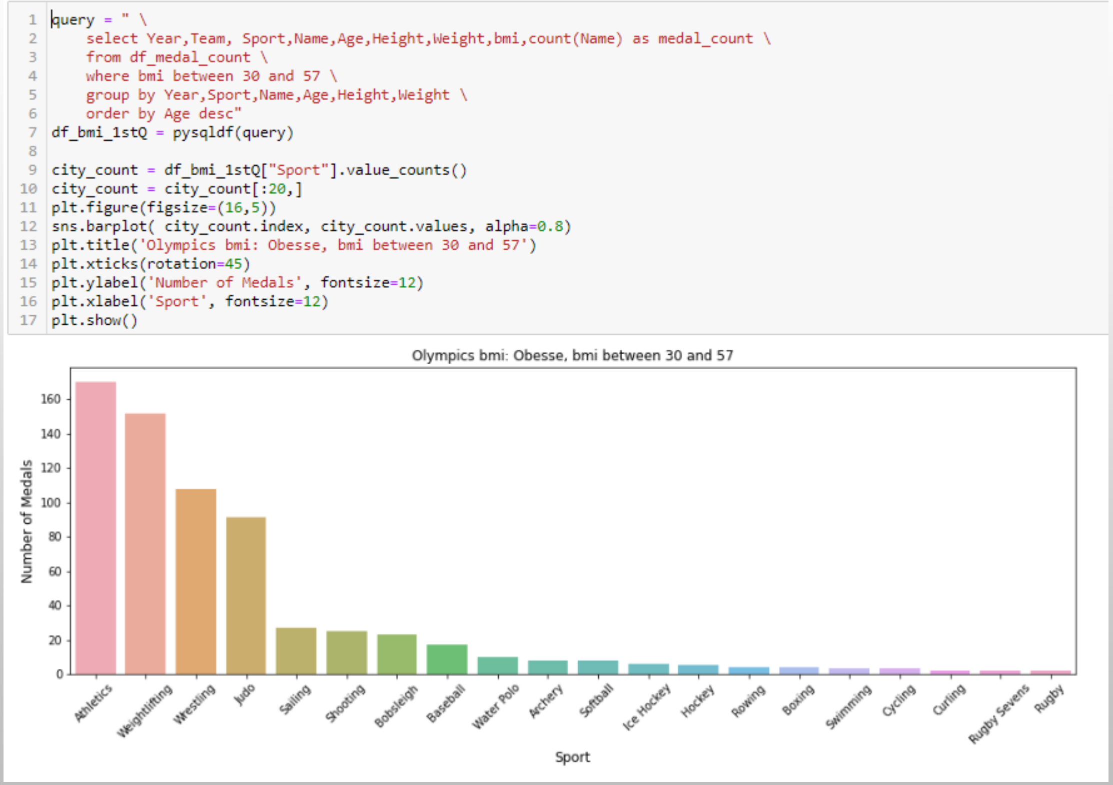
```python
# The Sports senior citizens playing
query = " \
    select Year,Team, Sport,Name,Age,Height,Weight,bmi,count(Name) as medal_count \
    from df_medal_count \
    where bmi between 18.5 and 24.99 \
    group by Year,Sport,Name,Age,Height,Weight \
    order by Age desc"
df_bmi_1stQ = pysqldf(query)

city_count = df_bmi_1stQ["Sport"].value_counts()
city_count = city_count[:20,]
plt.figure(figsize=(16,5))
sns.barplot( city_count.index, city_count.values, alpha=0.8)
plt.title('Olympics bmi: Normal bmi between 18.5 and 24.99')
plt.xticks(rotation=45)
plt.ylabel('Number of Medals', fontsize=12)
plt.xlabel('Sport', fontsize=12)
plt.show()
```

THE 3RD QUARTILE OF BMI GROUP WON IN FOLLOWING GAMES



```
query = " \
    select Year,Team, Sport,Name,Age,Height,Weight,bmi,count(Name) as medal_count \
    from df_medal_count \
    where bmi between 24.99 and 30 \
    group by Year,Sport,Name,Age,Height,Weight \
    order by Age desc"
df_bmi_1stQ = pysqldf(query)

city_count = df_bmi_1stQ["Sport"].value_counts()
city_count = city_count[:20,]
plt.figure(figsize=(16,5))
sns.barplot( city_count.index, city_count.values, alpha=0.8)
plt.title('Olympics bmi: Overweight, bmi between 24.99 and 30')
plt.xticks(rotation=45)
plt.ylabel('Number of Medals', fontsize=12)
plt.xlabel('Sport', fontsize=12)
plt.show()
```

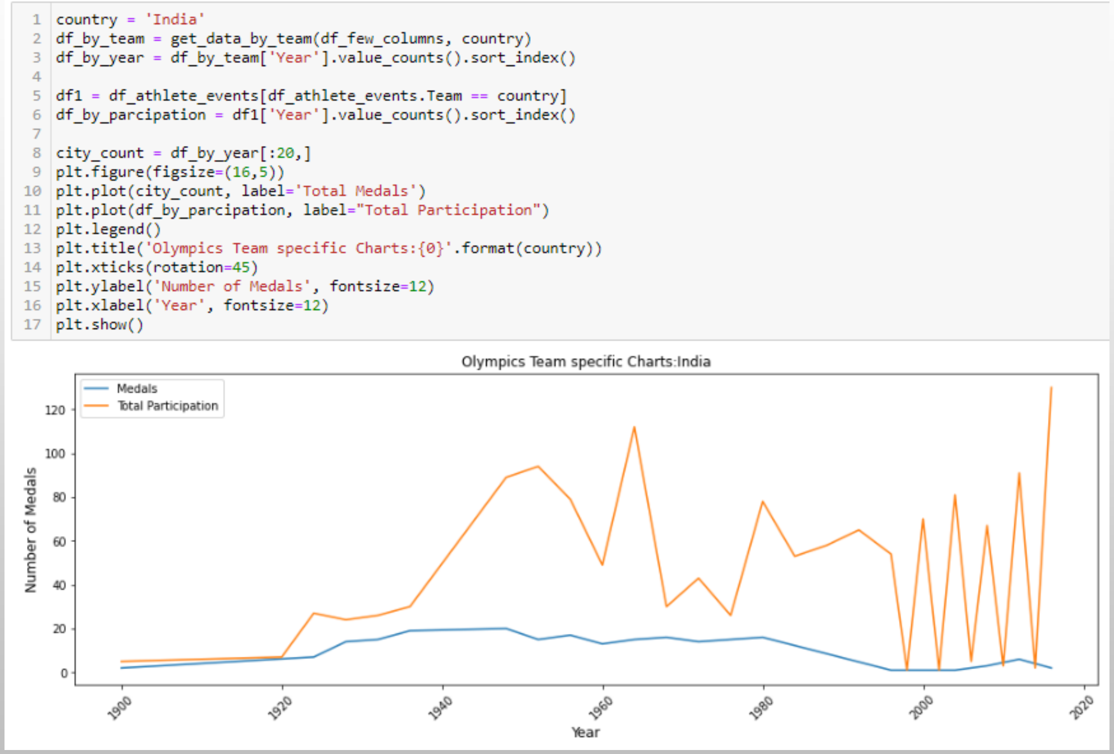
THE 4TH QUARTILE OF BMI GROUP WON IN FOLLOWING GAMES



```
query = " \
    select Year,Team, Sport,Name,Age,Height,Weight,bmi,count(Name) as medal_count \
    from df_medal_count \
    where bmi between 30 and 57 \
    group by Year,Sport,Name,Age,Height,Weight \
    order by Age desc"
df_bmi_1stQ = pysqldf(query)

city_count = df_bmi_1stQ["Sport"].value_counts()
city_count = city_count[:20,]
plt.figure(figsize=(16,5))
sns.barplot( city_count.index, city_count.values, alpha=0.8)
plt.title('Olympics bmi: Obesse, bmi between 30 and 57')
plt.xticks(rotation=45)
plt.ylabel('Number of Medals', fontsize=12)
plt.xlabel('Sport', fontsize=12)
plt.show()
```

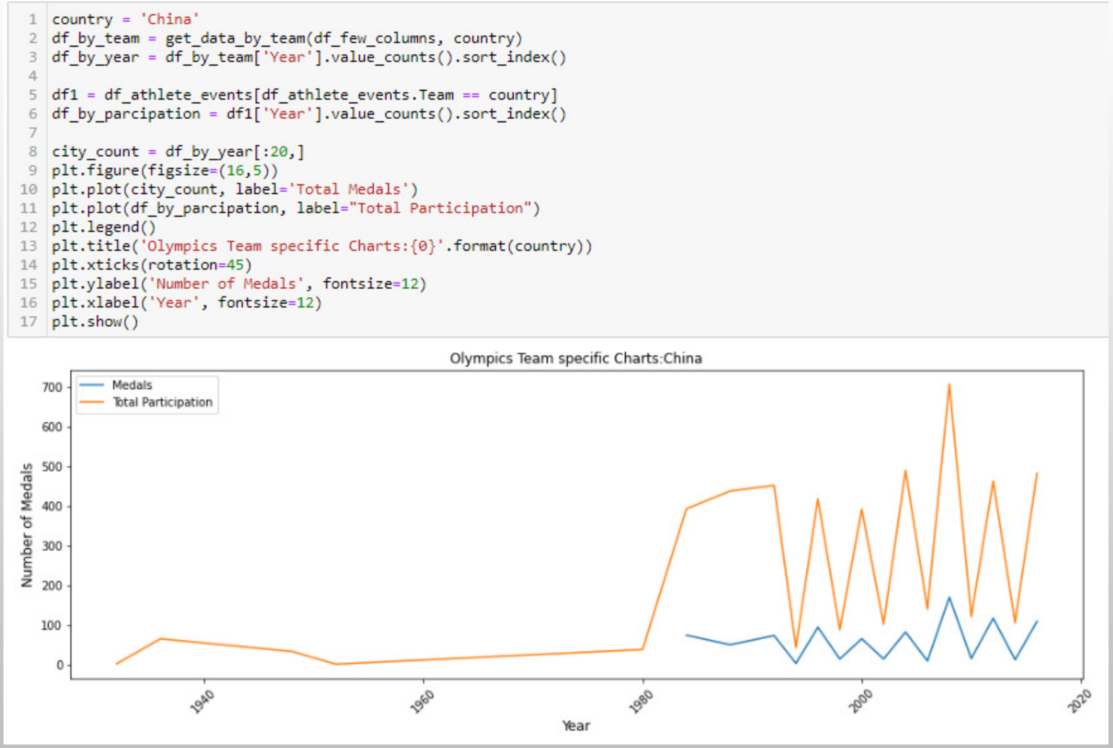
PARTICIPATION VS WINNING GAME

INDIA



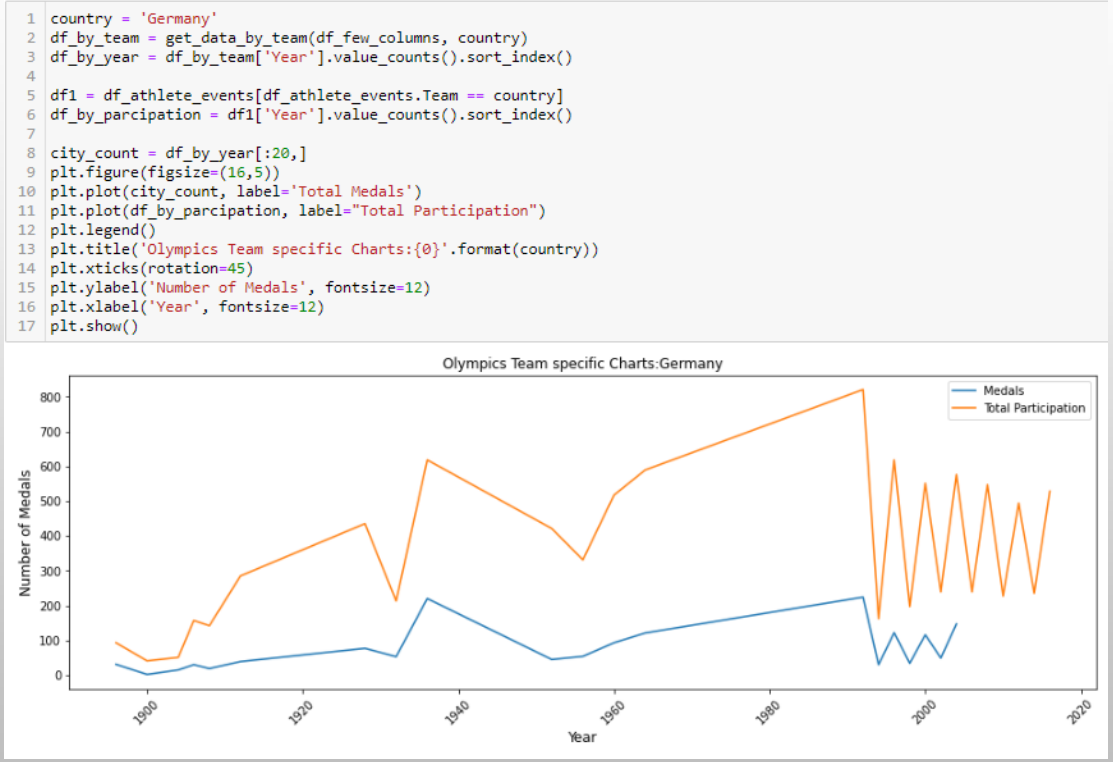
```
country = 'India'
df_by_team = get_data_by_team(df_few_columns, country)
df_by_year = df_by_team['Year'].value_counts().sort_index()

df1 = df_athlete_events[df_athlete_events.Team == country]
df_by_parcipation = df1['Year'].value_counts().sort_index()

city_count = df_by_year[:20,]
plt.figure(figsize=(16,5))
plt.plot(city_count, label='Total Medals')
plt.plot(df_by_parcipation, label="Total Participation")
plt.legend()
plt.title('Olympics Team specific Charts:{0}'.format(country))
plt.xticks(rotation=45)
plt.ylabel('Number of Medals', fontsize=12)
plt.xlabel('Year', fontsize=12)
plt.show()
```

CHINA



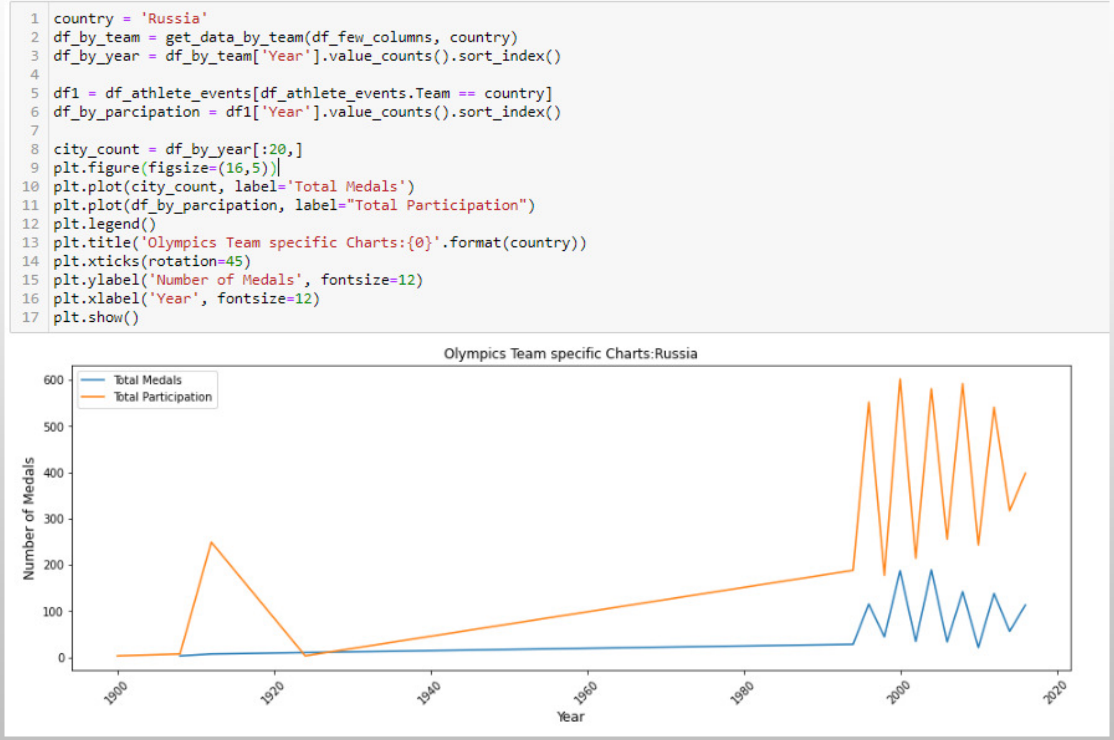
```
country = 'China'
df_by_team = get_data_by_team(df_few_columns, country)
df_by_year = df_by_team['Year'].value_counts().sort_index()

df1 = df_athlete_events[df_athlete_events.Team == country]
df_by_parcipation = df1['Year'].value_counts().sort_index()

city_count = df_by_year[:20,]
plt.figure(figsize=(16,5))
plt.plot(city_count, label='Total Medals')
plt.plot(df_by_parcipation, label="Total Participation")
plt.legend()
plt.title('Olympics Team specific Charts:{0}'.format(country))
plt.xticks(rotation=45)
plt.ylabel('Number of Medals', fontsize=12)
plt.xlabel('Year', fontsize=12)
plt.show()
```

GERMANY



```
country = 'Germany'
df_by_team = get_data_by_team(df_few_columns, country)
df_by_year = df_by_team['Year'].value_counts().sort_index()

df1 = df_athlete_events[df_athlete_events.Team == country]
df_by_parcipation = df1['Year'].value_counts().sort_index()

city_count = df_by_year[:20,]
plt.figure(figsize=(16,5))
plt.plot(city_count, label='Total Medals')
plt.plot(df_by_parcipation, label="Total Participation")
plt.legend()
plt.title('Olympics Team specific Charts:{0}'.format(country))
plt.xticks(rotation=45)
plt.ylabel('Number of Medals', fontsize=12)
plt.xlabel('Year', fontsize=12)
plt.show()
```

RUSSIA



```
country = 'Russia'
df_by_team = get_data_by_team(df_few_columns, country)
df_by_year = df_by_team['Year'].value_counts().sort_index()

df1 = df_athlete_events[df_athlete_events.Team == country]
df_by_parcipation = df1['Year'].value_counts().sort_index()

city_count = df_by_year[:20,]
plt.figure(figsize=(16,5))
plt.plot(city_count, label='Total Medals')
plt.plot(df_by_parcipation, label="Total Participation")
plt.legend()
plt.title('Olympics Team specific Charts:{0}'.format(country))
plt.xticks(rotation=45)
plt.ylabel('Number of Medals', fontsize=12)
plt.xlabel('Year', fontsize=12)
plt.show()
```

INFERENCE OF INITIAL HYPOTHESIS

1. THE TEAMS WHO PARTICIPATE MORE OLYMPIC GAMES WINS MORE MEDALS IN LATER GAMES.

THE WINNING OF GAMES IS NOT INCREASING PATTERN. SO MORE EXPERIENCE IN OLYMPIC PARTICIPATION IS NOT LINEARLY PROPORTIONAL TO WINNING.

2. THE MOTIVATION OUTWEIGHS PHYSICAL CHARACTERISTICS TO WIN THE GAME IS THE HYPOTHESIS.

THE BMI ANALYSIS SHOWS THAT THERE ARE GAMES FOR EVERYONE.

SINCE THE PARTICIPATION EXPERIENCE IS FLUCTUATING, THE BMI INDEX ALSO SHOWS WINNING, THE INDIVIDUAL MOTIVATION PLAYS AN IMPORTANT ROLE IN WINNING OF OLYMPIC GAMES.

THANK YOU