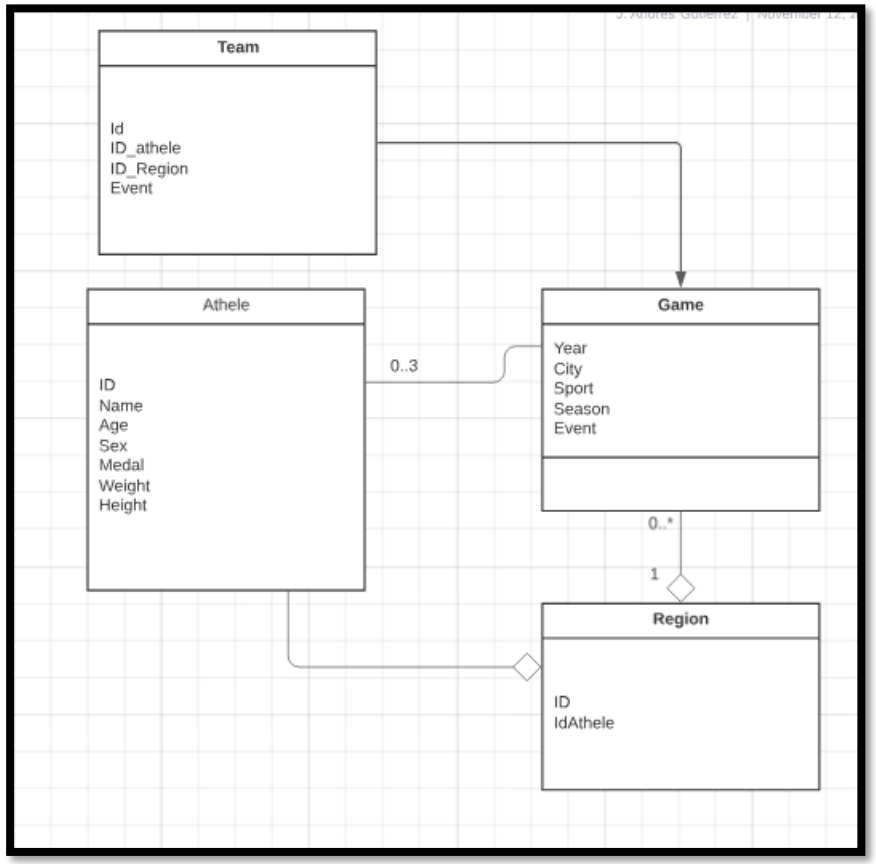
Sql For Data Science Capstone

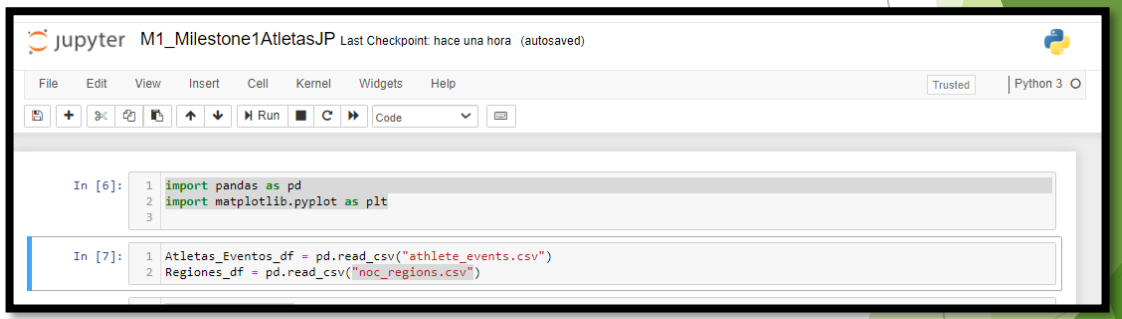
Athletes Dataset

Proposal

- CUSTOMER:
  - The aim of this project is to incorporate denotations specifically for client 3: Sportstats. Select your client/dataset - (Olympic Games Dataset - 120 years of data). The database in question will explore the influence of event management attention based on gender, nationality, and other grouping scenarios. Additionally, it will delve into the potential advantages and gains associated with the implementation of a scheduling system. The next step involves categorizing the database into two segments, distinguishing between Summer and Winter Olympic Events. The goal is to examine denotations among athletes, focusing on age, weight, and height, while observing stability in Olympic medal attainment and its impact on athletes.
- HYPOTHESES:
  - Developing a database for a novel client pertaining to participation trends over the last 120 years. We aim to establish a direct correlation through the creation of a sandbox browser programmed in Python, introducing new search criteria to facilitate the analysis of extensive information.
- APPROACH:
  - Conduct an exploratory analysis on athletes from records spanning all Olympic events from the past 120 years to the present. Employ a multifaceted approach in examining athletes within the significant themes of this time period. Identify correlations between athlete gender and similar groups across both Summer and Winter Olympics.

Devolop Sanbox: Develop an Entity Relationship Diagram (ERD)





DataBase: Atletas_Eventos Regiones

In [9]:
```
Atletas_Eventos_df
```

Out[9]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	Sport	Event	Medal
0	1	A Dijiang	M	24.0	180.0	80.0	China	CHN	1992 Summer	1992	Summer	Barcelona	Basketball	Basketball Men's Basketball	NaN
1	2	A Lamusi	M	23.0	170.0	60.0	China	CHN	2012 Summer	2012	Summer	London	Judo	Judo Men's Extra-Lightweight	NaN
2	3	Gunnar Nielsen Aaby	M	24.0	NaN	NaN	Denmark	DEN	1920 Summer	1920	Summer	Antwerpen	Football	Football Men's Football	NaN
3	4	Edgar Lindenau Aabye	M	34.0	NaN	NaN	Denmark/Sweden	DEN	1900 Summer	1900	Summer	Paris	Tug-Of-War	Tug-Of-War Men's Tug-Of-War	Gold
4	5	Christine Jacoba Aaftink	F	21.0	185.0	82.0	Netherlands	NED	1988 Winter	1988	Winter	Calgary	Speed Skating	Speed Skating Women's 500 metres	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
271111	135569	Andrzej ya	M	29.0	179.0	89.0	Poland-1	POL	1976 Winter	1976	Winter	Innsbruck	Luge	Luge Mixed (Men)'s Doubles	NaN
271112	135570	Piotr ya	M	27.0	176.0	59.0	Poland	POL	2014 Winter	2014	Winter	Sochi	Ski Jumping	Ski Jumping Men's Large Hill, Individual	NaN
271113	135570	Piotr ya	M	27.0	176.0	59.0	Poland	POL	2014 Winter	2014	Winter	Sochi	Ski Jumping	Ski Jumping Men's Large Hill, Team	NaN
271114	135571	Tomasz Ireneusz ya	M	30.0	185.0	96.0	Poland	POL	1998 Winter	1998	Winter	Nagano	Bobsleigh	Bobsleigh Men's Four	NaN
271115	135571	Tomasz Ireneusz ya	M	34.0	185.0	96.0	Poland	POL	2002 Winter	2002	Winter	Salt Lake City	Bobsleigh	Bobsleigh Men's Four	NaN

271116 rows × 15 columns

In [10]:
```
Regiones_df
```

Out[10]:

	NOC	region	notes
0	AFG	Afghanistan	NaN
1	AHO	Curacao	Netherlands Antilles
2	ALB	Albania	NaN
3	ALG	Algeria	NaN
4	AND	Andorra	NaN
...	...	...	...
225	YEM	Yemen	NaN
226	YMD	Yemen	South Yemen
227	YUG	Serbia	Yugoslavia
228	ZAM	Zambia	NaN
229	ZIM	Zimbabwe	NaN

230 rows × 3 columns

Milestone 1

The initial traceability identification focuses on information within athletes, distinguishing them by gender, grouping their attributes, and analyzing the medals they have earned.

```
In [11]:
grouped = Atletas_Eventos_df.groupby(Atletas_Eventos_df["Sex"])
print("Male athletes")
print(grouped.get_group("M").nunique())
print("---")
print("Female athletes")
print(grouped.get_group("F").nunique())
```

```
Male athletes
ID        101590
Name      100979
Sex            1
Age           74
Height        92
Weight       206
Team        1154
NOC          230
Games         51
Year          35
Season         2
City          42
Sport         63
Event        554
Medal          3
dtype: int64
---
Female athletes
ID        33981
Name      33808
Sex           1
Age          62
Height       77
Weight      140
Team        374
NOC         222
Games        50
Year         34
Season        2
City         42
Sport        53
Event       269
Medal         3
dtype: int64
```

```
In [12]:
Atletas_Eventos_df.isna().sum().sum()
```

```
Out[12]: 363853
```

```
In [13]:
gender_df = Atletas_Eventos_df.drop(
    ["ID", "Name", "Age", "Height", "Weight", "Team", "NOC", "Games", "Year", "Season", "City", "Sport", "Event"], axis='col
gender_df = gender_df.dropna()
gender_df
```

Out[13]:

	Sex	Medal
3	M	Gold
37	M	Bronze
38	M	Bronze
40	M	Bronze
41	M	Bronze
...	...	...
271078	F	Silver
271080	F	Bronze
271082	M	Bronze
271102	F	Bronze
271103	F	Silver

39783 rows × 2 columns

```
In [14]:
# I want to know how many unique athletes we have of each gender.
gender_df.groupby("Sex").count()
```

Out[14]:

Sex	Medal
F	11253
M	28530

```
In [15]:
gender_df.groupby("Medal").count()
```

Out[15]:

Medal	Sex
Bronze	13295
Gold	13372
Silver	13116

Milestone 2

Before embarking on the second theorem, it is crucial to perform an analysis of the Athletes database, examining the events documented within the "Summer" and "Winter" seasons.

```
In [16]:
grouped = Atletas_Eventos_df.groupby(athelete_events_df["Season"])
print("Summer games")
print(grouped.get_group("Summer").nunique())
print("---")
print("Winter games")
print(grouped.get_group("Winter").nunique())
```

```
Summer games
ID        116776
Name      116122
Sex            2
Age           74
Height        95
Weight       219
Team        1157
NOC          230
Games         29
Year          29
Season         1
City          23
Sport         52
Event        651
Medal          3
dtype: int64
---
Winter games
ID         18958
Name       18923
Sex            2
Age           47
Height        64
Weight       125
Team         221
NOC          119
Games         22
Year          22
Season         1
City          19
Sport         17
Event        119
Medal          3
dtype: int64
```

```
In [17]:
Atletas_Eventos_df.groupby("Season").count()
```

Out[17]:

Season	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	City	Sport	Event	Medal
Summer	222552	222552	222552	213363	170695	168698	222552	222552	222552	222552	222552	222552	222552	34088
Winter	48564	48564	48564	48279	40250	39543	48564	48564	48564	48564	48564	48564	48564	5695

```
In [18]:
Atletas_Eventos_df.groupby("Team").count()
```

Out[18]:

Team	ID	Name	Sex	Age	Height	Weight	NOC	Games	Year	Season	City	Sport	Event	Medal
30. Februar	2	2	2	2	2	1	2	2	2	2	2	2	2	0
A North American Team	4	4	4	3	0	0	4	4	4	4	4	4	4	4
Acipactli	3	3	3	3	3	3	3	3	3	3	3	3	3	0
Acturus	2	2	2	1	0	0	2	2	2	2	2	2	2	0
Afghanistan	126	126	126	78	54	61	126	126	126	126	126	126	126	2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
Zambia	183	183	183	154	128	139	183	183	183	183	183	183	183	2
Zefyros	2	2	2	2	2	2	2	2	2	2	2	2	2	0
Zimbabwe	309	309	309	307	286	287	309	309	309	309	309	309	309	22
Zut	3	3	3	3	0	0	3	3	3	3	3	3	3	3
rn-2	5	5	5	5	1	1	5	5	5	5	5	5	5	0

1184 rows × 14 columns

```
In [ ]:
```

Exploratory Data Analysis: Athletes – Summer and Winter Olympics

```
In [48]: Atletas_Eventos_df = pd.read_csv("athlete_events.csv")
         Regiones_df = pd.read_csv("noc_regions.csv")

In [49]: Summer_Games=Atletas_Eventos_df[Atletas_Eventos_df['Season']=='Summer']
```

```
In [52]: Winter_Games=Atletas_Eventos_df[Atletas_Eventos_df['Season']=='Winter']
```

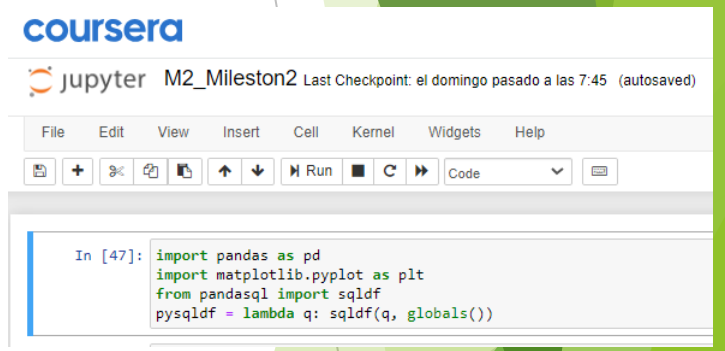
```
In [50]: Summer_Games
```

Out[50]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	Sport	Event	Medal
0	1	A Dijiang	M	24.0	180.0	80.0	China	CHN	1992 Summer	1992	Summer	Barcelona	Basketball	Basketball Men's Basketball	NaN
1	2	A Lamusi	M	23.0	170.0	60.0	China	CHN	2012 Summer	2012	Summer	London	Judo	Judo Men's Extra-Lightweight	NaN
2	3	Gunnar Nielsen Aaby	M	24.0	NaN	NaN	Denmark	DEN	1920 Summer	1920	Summer	Antwerpen	Football	Football Men's Football	NaN
3	4	Edgar Lindenau Aabye	M	34.0	NaN	NaN	Denmark/Sweden	DEN	1900 Summer	1900	Summer	Paris	Tug-Of-War	Tug-Of-War Men's Tug-Of-War	Gold
26	8	Cornelia "Cor" Aalten (-Strannood)	F	18.0	168.0	NaN	Netherlands	NED	1932 Summer	1932	Summer	Los Angeles	Athletics	Athletics Women's 100 metres	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
271106	135565	Fernando scar Zylberberg	M	27.0	168.0	76.0	Argentina	ARG	2004 Summer	2004	Summer	Athina	Hockey	Hockey Men's Hockey	NaN
271107	135566	James Francis "Jim" Zylker	M	21.0	175.0	75.0	United States	USA	1972 Summer	1972	Summer	Munich	Football	Football Men's Football	NaN
271108	135567	Aleksandr Viktorovich Zyuzin	M	24.0	183.0	72.0	Russia	RUS	2000 Summer	2000	Summer	Sydney	Rowing	Rowing Men's Lightweight Coxless Fours	NaN
271109	135567	Aleksandr Viktorovich Zyuzin	M	28.0	183.0	72.0	Russia	RUS	2004 Summer	2004	Summer	Athina	Rowing	Rowing Men's Lightweight Coxless Fours	NaN
271110	135568	Olga Igorevna Zyuzkova	F	33.0	171.0	69.0	Belarus	BLR	2016 Summer	2016	Summer	Rio de Janeiro	Basketball	Basketball Women's Basketball	NaN

222552 rows × 15 columns

Now starting the new laboratory survey, it is necessary to define the information contained in a simplified structure in relation to the "Summer" and "Winter" seasons.



Which will allow us to obtain the requested records.

Milestone 2 Which will allow us to obtain the requested records.

```
In [53]: Winter_Games
```

Out[53]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	Sport	Event	Medal
4	5	Christine Jacoba Aaftink	F	21.0	185.0	82.0	Netherlands	NED	1988 Winter	1988	Winter	Calgary	Speed Skating	Speed Skating Women's 500 metres	NaN
5	5	Christine Jacoba Aaftink	F	21.0	185.0	82.0	Netherlands	NED	1988 Winter	1988	Winter	Calgary	Speed Skating	Speed Skating Women's 1,000 metres	NaN
6	5	Christine Jacoba Aaftink	F	25.0	185.0	82.0	Netherlands	NED	1992 Winter	1992	Winter	Albertville	Speed Skating	Speed Skating Women's 500 metres	NaN
7	5	Christine Jacoba Aaftink	F	25.0	185.0	82.0	Netherlands	NED	1992 Winter	1992	Winter	Albertville	Speed Skating	Speed Skating Women's 1,000 metres	NaN
8	5	Christine Jacoba Aaftink	F	27.0	185.0	82.0	Netherlands	NED	1994 Winter	1994	Winter	Lillehammer	Speed Skating	Speed Skating Women's 500 metres	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
271111	135569	Andrzej ya	M	29.0	179.0	89.0	Poland-1	POL	1976 Winter	1976	Winter	Innsbruck	Luge	Luge Mixed (Men)'s Doubles	NaN
271112	135570	Piotr ya	M	27.0	176.0	59.0	Poland	POL	2014 Winter	2014	Winter	Sochi	Ski Jumping	Ski Jumping Men's Large Hill, Individual	NaN
271113	135570	Piotr ya	M	27.0	176.0	59.0	Poland	POL	2014 Winter	2014	Winter	Sochi	Ski Jumping	Ski Jumping Men's Large Hill, Team	NaN
271114	135571	Tomasz Ireneusz ya	M	30.0	185.0	96.0	Poland	POL	1998 Winter	1998	Winter	Nagano	Bobsleigh	Bobsleigh Men's Four	NaN
271115	135571	Tomasz Ireneusz ya	M	34.0	185.0	96.0	Poland	POL	2002 Winter	2002	Winter	Salt Lake City	Bobsleigh	Bobsleigh Men's Four	NaN

48564 rows × 15 columns

```
In [54]: print(pysqldf('''SELECT Sex,
                            COUNT(*),
                            COUNT(*) * 100.0 / SUM(COUNT(*)) over () AS ratio
                        FROM Winter_Games
                        GROUP BY Sex'''))
```

```
  Sex  COUNT(*)      ratio
0   F     15079  31.049749
1   M     33485  68.950251
```

```
In [55]: print(pysqldf('''SELECT Sex,
                            AVG(Age),
                            AVG(Height),
                            AVG(Weight)
                        FROM Summer_Games
                        GROUP BY Sex
                        '''))
```

```
  Sex   AVG(Age)  AVG(Height)  AVG(Weight)
0   F  23.660997   168.169025    60.087644
1   M  26.443944   178.901874    75.604195
```

```
In [56]: print(pysqldf('''SELECT Sex,
                            AVG(Age),
                            AVG(Height),
                            AVG(Weight)
                        FROM Winter_Games
                        GROUP BY Sex
                        '''))
```

```
  Sex   AVG(Age)  AVG(Height)  AVG(Weight)
0   F  24.014398   166.528250    59.755156
1   M  25.504261   178.668699    76.357058
```

A forthcoming study will entail a comprehensive tally and systematic grouping of athletes' attributes, providing a preliminary overview before implementing measurements that will be visualized through graphical structures.

Exploratory Data Analysis: Athletes – Summer and Winter Olympics

```
In [62]: #Summer Olympics:
summer_medals = pysqldf('''
                    SELECT
                      Year,
                      CAST(medal_count AS FLOAT) / total_count AS medal_ratio,
                      CAST(gold_count AS FLOAT) / medal_count AS gold_ratio,
                      CAST(silver_count AS FLOAT) / medal_count AS silver_ratio,
                      CAST(bronze_count AS FLOAT) / medal_count AS bronze_ratio
                    FROM
                    (
                        SELECT
                            Year,
                            COUNT(*) AS total_count,
                            SUM(CASE
                                  WHEN Medal IS NOT NULL THEN 1 ELSE 0
                                END) AS medal_count,
                            SUM(CASE
                                  WHEN Medal = "Gold" THEN 1 ELSE 0
                                END) AS gold_count,
                            SUM(CASE
                                  WHEN Medal = "Silver" THEN 1 ELSE 0
                                END) AS silver_count,
                            SUM(CASE
                                  WHEN Medal = "Bronze" THEN 1 ELSE 0
                                END) AS bronze_count
                            FROM
                              Summer_Games
                            GROUP BY
                              Year
                       )new_table
                     ''')
```

Milestone 2

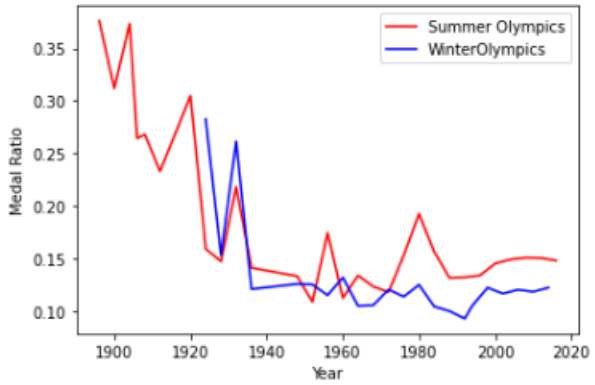
To enhance the understanding for our client and align with actual values across the various Olympic events spanning its inception and evolutionary timeline, we will generate graphs embedded within a new table code.

```
In [63]: #Winter Olympics:
winter_medals = pysqldf('''
                    SELECT
                      Year,
                      CAST(medal_count AS FLOAT) / total_count AS medal_ratio,
                      CAST(gold_count AS FLOAT) / medal_count AS gold_ratio,
                      CAST(silver_count AS FLOAT) / medal_count AS silver_ratio,
                      CAST(bronze_count AS FLOAT) / medal_count AS bronze_ratio
                    FROM
                    (
                        SELECT
                            Year,
                            COUNT(*) AS total_count,
                            SUM(CASE
                                  WHEN Medal IS NOT NULL THEN 1 ELSE 0
                                END) AS medal_count,
                            SUM(CASE
                                  WHEN Medal = "Gold" THEN 1 ELSE 0
                                END) AS gold_count,
                            SUM(CASE
                                  WHEN Medal = "Silver" THEN 1 ELSE 0
                                END) AS silver_count,
                            SUM(CASE
                                  WHEN Medal = "Bronze" THEN 1 ELSE 0
                                END) AS bronze_count
                            FROM
                              Winter_Games
                            GROUP BY
                              Year
                       )new_table
                     ''')
```

Graphics – Summer and Winter Olympics

```
In [64]: plt.plot(summer_medals.Year, summer_medals.medal_ratio, color = "red", label = "Summer Olympics")
         plt.plot(winter_medals.Year, winter_medals.medal_ratio, color = "blue", label = "WinterOlympics")
         plt.xlabel("Year")
         plt.ylabel("Medal Ratio")
         plt.legend()

Out[64]: <matplotlib.legend.Legend at 0x7f2bc24c98d0>
```

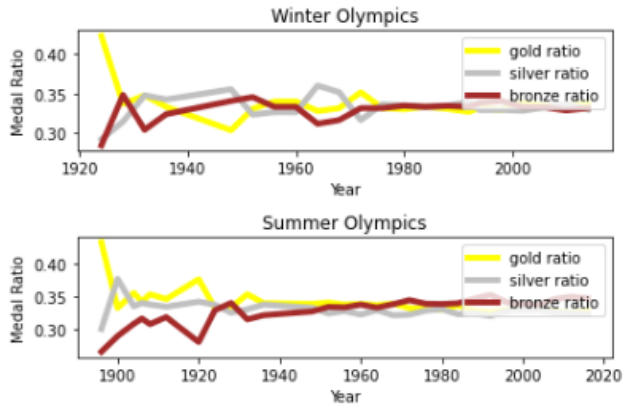


Milestone 2

in order to provide our client with a more comprehensive perspective and align with authentic values across the numerous Olympic events conducted from its establishment to the present, we will generate graphs incorporated into a new table code.

Graphics – Medals

In [65]:
```
fig, ax = plt.subplots(2)
ax[0].plot(winter_medals.Year, winter_medals.gold_ratio, marker='', color='yellow', linewidth=4, label = "gold ratio")
ax[0].plot(winter_medals.Year, winter_medals.silver_ratio, marker='', color='silver', linewidth=4, label = "silver ratio")
ax[0].plot(winter_medals.Year, winter_medals.bronze_ratio, marker='', color='brown', linewidth=4, label = "bronze ratio")
ax[0].legend(loc=1)
ax[0].set_xlabel("Year")
ax[0].set_ylabel("Medal Ratio")
ax[0].set_title("Winter Olympics")
ax[1].plot(summer_medals.Year, summer_medals.gold_ratio, marker='', color='yellow', linewidth=4, label = "gold ratio")
ax[1].plot(summer_medals.Year, summer_medals.silver_ratio, marker='', color='silver', linewidth=4, label = "silver ratio")
ax[1].plot(summer_medals.Year, summer_medals.bronze_ratio, marker='', color='brown', linewidth=4, label = "bronze ratio")
plt.legend(loc=1)
ax[1].set_xlabel("Year")
ax[1].set_ylabel("Medal Ratio")
ax[1].set_title("Summer Olympics")
plt.tight_layout()
```



The relative percentages of medals have also stabilized, depending on the events of the Summer Olympics and the Winter Olympics, it is necessary to denote that athletes have different abilities, in some cases age, weight and height mark a difference.

Total Number of Medals

```
In [11]: print(summer_medal_count.head())
```

	Year	total_count	medal_count	gold_count	silver_count	bronze_count
0	1896	380	143	62	43	38
1	1900	1936	604	201	228	175
2	1904	1301	486	173	163	150
3	1906	1733	458	157	156	145
4	1908	3101	831	294	281	256

```
In [12]: print(winter_medal_count.head())
```

	Year	total_count	medal_count	gold_count	silver_count	bronze_count
0	1924	460	130	55	38	37
1	1928	582	89	30	28	31
2	1932	352	92	32	32	28
3	1936	895	108	36	37	35
4	1948	1075	135	41	48	46

```
In [14]: summer_medal_count_new = summer_medal_count[7:]
```

```
In [15]: print(summer_medal_count_new)
```

	Year	total_count	medal_count	gold_count	silver_count	bronze_count
7	1924	5233	832	277	281	274
8	1928	4992	734	245	239	250
9	1932	2969	647	229	214	204
10	1936	6506	917	312	310	295
11	1948	6405	852	289	284	279
12	1952	8270	897	306	291	300
13	1956	5127	893	302	293	298
14	1960	8119	911	309	294	308
15	1964	7702	1029	347	339	343
16	1968	8588	1057	359	340	358
17	1972	10304	1215	404	392	419
18	1976	8641	1320	438	434	448
19	1980	7191	1384	457	458	469
20	1984	9454	1476	497	477	502
21	1988	12037	1582	520	513	549
22	1992	12977	1712	559	549	604
23	1996	13780	1842	608	605	629
24	2000	13821	2004	663	661	680
25	2004	13443	2001	664	660	677
26	2008	13602	2048	671	667	710
27	2012	12920	1941	632	630	679
28	2016	13688	2023	665	655	703

```
In [13]: print(summer_medal_count)
```

	Year	total_count	medal_count	gold_count	silver_count	bronze_count
0	1896	380	143	62	43	38
1	1900	1936	604	201	228	175
2	1904	1301	486	173	163	150
3	1906	1733	458	157	156	145
4	1908	3101	831	294	281	256
5	1912	4040	941	326	315	300
6	1920	4292	1308	493	448	367
7	1924	5233	832	277	281	274
8	1928	4992	734	245	239	250
9	1932	2969	647	229	214	204
10	1936	6506	917	312	310	295
11	1948	6405	852	289	284	279
12	1952	8270	897	306	291	300
13	1956	5127	893	302	293	298
14	1960	8119	911	309	294	308
15	1964	7702	1029	347	339	343
16	1968	8588	1057	359	340	358
17	1972	10304	1215	404	392	419
18	1976	8641	1320	438	434	448
19	1980	7191	1384	457	458	469
20	1984	9454	1476	497	477	502
21	1988	12037	1582	520	513	549
22	1992	12977	1712	559	549	604
23	1996	13780	1842	608	605	629
24	2000	13821	2004	663	661	680
25	2004	13443	2001	664	660	677
26	2008	13602	2048	671	667	710
27	2012	12920	1941	632	630	679
28	2016	13688	2023	665	655	703

Hypothesis Conclusions

- There will be groups of athletes that can be aligned largely based on attributes and medals
- There will be defined graphs between these groups from which clients will be able to see the statistics by year concept.
- Some themes are representative between time periods.