Data Warehousing CITS5504 Project 1

Olympic Dataset:

Data Warehouse Design and Multidimensional Analysis

Jia Min Ho (23337561)

Table of Contents

DATASET DESCRIPTION

Overview

This report leverages an extensive dataset derived from multiple reputable sources, centering around the theme of the Olympic Games. The datasets are curated to support the design and development of a comprehensive data warehouse, aimed at providing insightful analysis on various dimensions of the Olympic Games and their broader implications. The upcoming XXXIII Olympic Summer Games, hosted in Paris from 26 July to 11 August 2024, serves as a focal point for our analyses.

Datasets and Sources

Olympic Dataset: This foundational dataset comprises historical records from both the Summer and Winter Olympic Games, spanning from 1896 to 2022. It includes two primary files:

- **olympic_hosts.csv**: Details on the host cities of each Olympic Games.
- **olympic_medals.csv**: Records of medals awarded across all Olympic disciplines.

Health and Well-being Data:

- **mental-illness.csv** and **life-expectancy.csv:** These files, sourced from Our World in Data, offer insights into global health trends, including disability-adjusted life years (DALYs), providing a unique perspective on the impact of sports on mental health and longevity.

Demographic and Economic Data:

- **Global Population.csv**: Sourced from the International Monetary Fund (IMF), this dataset presents current and historical global population figures.
- **Economic data.csv**: Derived from world-development-indicators, this file includes various economic metrics across countries, offering a backdrop for analyzing the economic impact of the Olympics.
- **list-of-countries_areas-by-continent-2024.csv**: Obtained from the World Population Review, this dataset facilitates geographical analyses and demographic segmentation.

Official Website

For more detailed information about the Paris 2024 Olympic Games, please refer to the Paris 2024 Official Website.

BUSINESS QUERY

Business Queries for Client A (Australian Government):

1. How does Australia's GDP per capita correlate with its performance in the Summer Olympics across different years?

2. What is the distribution of Olympic medals won by Australia across different disciplines throughout the history of the Games, and which sport has yielded the highest number of medals?

3. Does Australia achieve better results in Olympic Games held domestically (home games) compared to those hosted internationally (away games)?

4. How does the changing life expectancy in Australia over the years influence the country's success rate in winning gold, silver, and bronze medals in swimming at the Summer Olympics?

5. Is there a correlation between population size in Australia and the total number of medals won at the Summer Olympics across the years?

Business Queries for Client B (French Government):

1. How does the GDP per capita of France correlate with its total medal count at the Summer Olympics over the years?

2. How does the burden of depression, as measured by Disability-Adjusted Life Years (DALYs), correlate with France’s Fencing Summer Olympic performance from 1988 to 2020?

3. What is the distribution of Olympic medals won by France across various sports throughout the history of the Olympic Games, and which sport has delivered the highest number of medals?

4. How does the life expectancy in France correlate with the number of medals won in Alpine Skiing at the Olympics across the years?

5. Does France achieve improved outcomes in Olympic competitions hosted domestically as opposed to those held in international venues?

Extra Query for Both Clients:

1. Identify the top 10 countries with the highest cumulative medal counts in Olympic history for a comprehensive overview of historical Olympic success.

Why the identified client(s) are important:

The identified clients, the Australian and French governments, are crucial because the queries address key performance indicators that directly relate to national sports policies, public health, and economic factors.

- For **Australia**, the questions explore correlations between GDP, population growth, life expectancy, and Olympic success, essential for strategizing future sports funding and training programs.
- Similarly, for **France**, the queries about GDP, life expectancy, mental health, and sporting performance can guide decisions on health initiatives and athlete support systems, reflecting on both nations' investments in sports and well-being, and their impact on international sporting success.

STARNET

This StarNet diagram illustrates the data model's dimensions and their hierarchies, essential for analyzing a business's data warehouse. Each dimension—Countries, Years, Games, and Disciplines—is represented by a radial line stemming from the central fact table, where each line details the hierarchy within the dimension. For example:

- **Countries Dimension:** Starts with the "ALL" aggregation level, breaking down into "region" and then further into "country".
- **Years Dimension:** Displays a straightforward hierarchy from "ALL" to individual "year".
- **Games Dimension:** Details a path from "ALL" to "game_season", and then to "game_name".
- **Disciplines Dimension:** Follows from "ALL" to "discipline_title" and then "event_title".

This structured approach allows for powerful data analysis, enabling drill-down and roll-up operations across different granularities of the business data, from broad to specific views. It has been refined to include additional concept hierarchies, enhancing its ability to address more precise business questions.

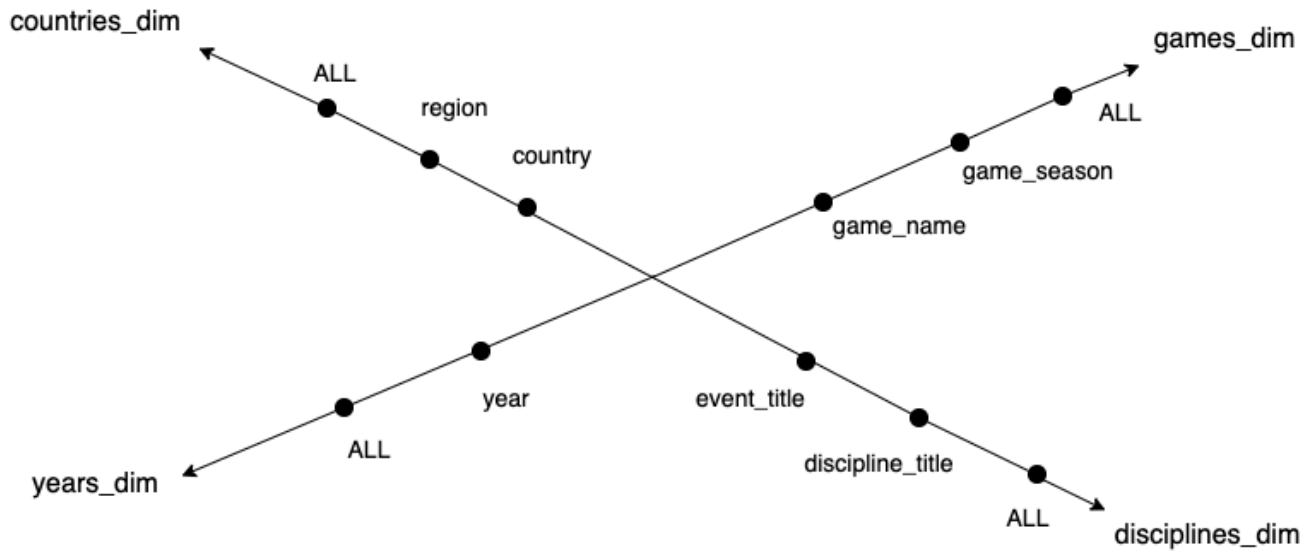


Figure 1: StarNet of business queries.

StarNet Footprints

StarNet footprints are utilized to demonstrate the application of our data warehouse design in answering business queries.

StarNet footprints of business queries for Client A (Australian Government):

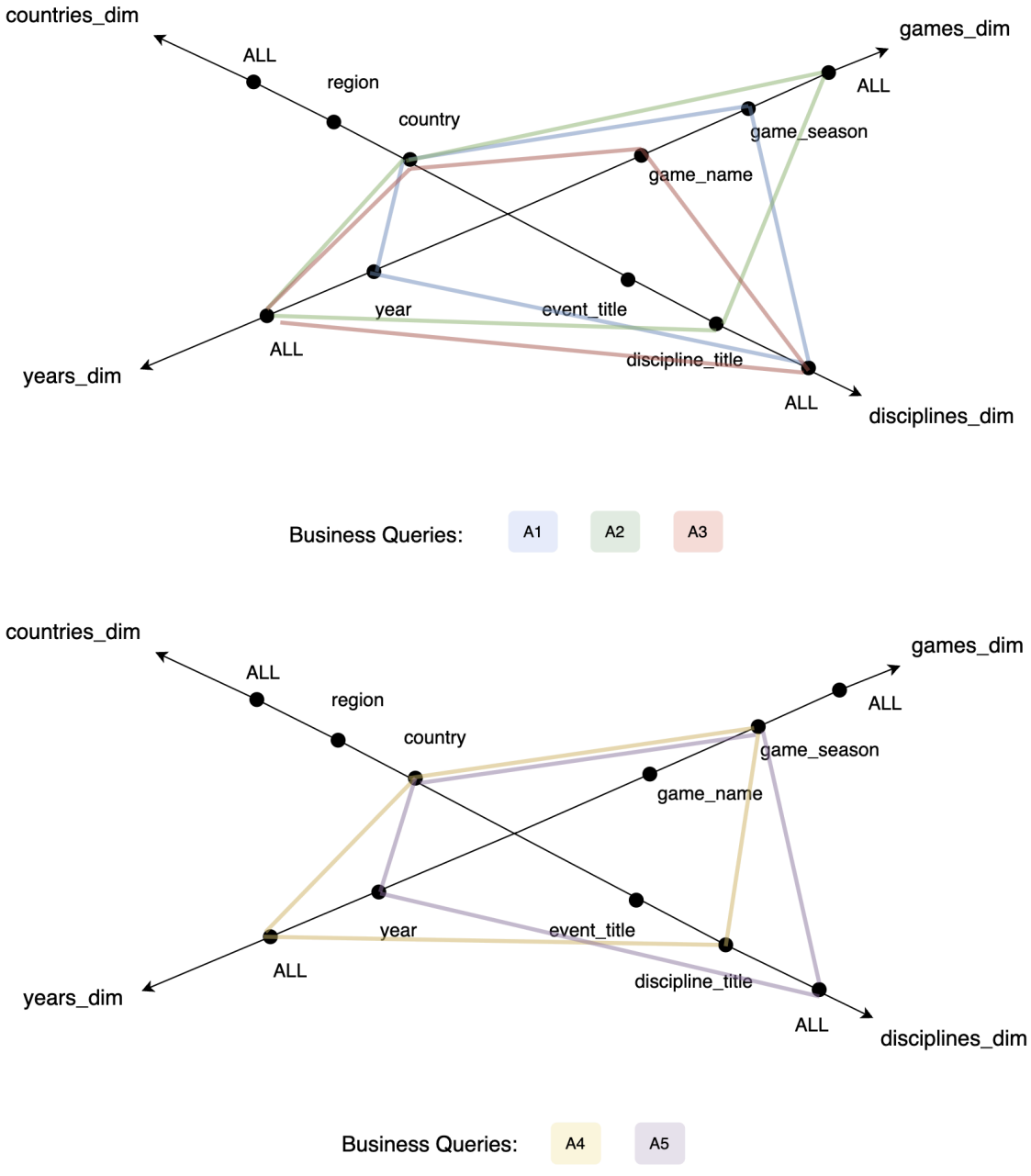


Figure 2: Questions A1 to A5 StarNet Footprints.

StarNet footprints of business queries for Client B (French Government):

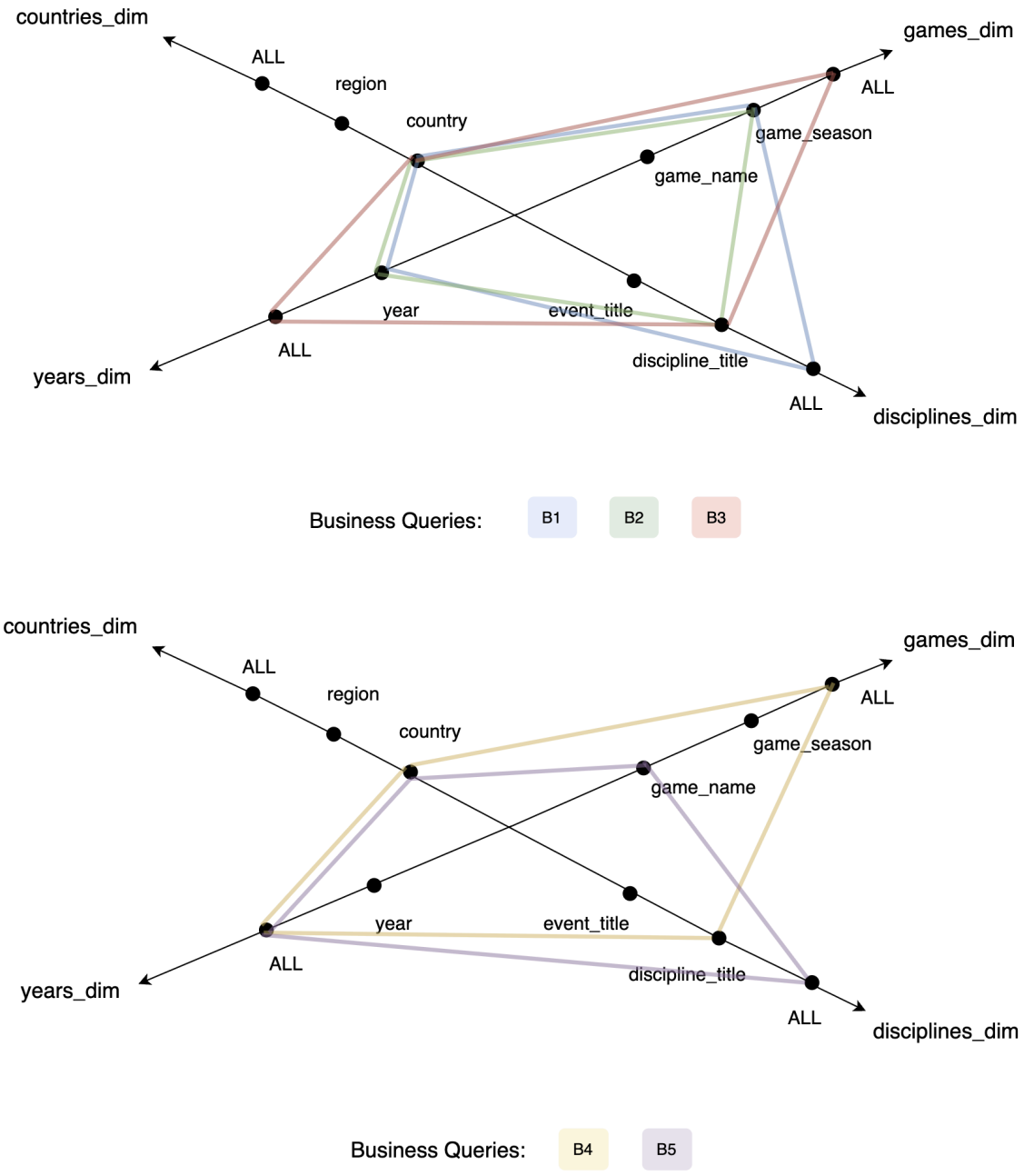


Figure 3: Questions B1 to B5 StarNet Footprints.

STAR SCHEMA

A Star Schema is a database schema design used primarily for data warehousing and business intelligence applications. It is called a "star schema" because its structure resembles a star, with a central fact table connected to multiple dimension tables. These dimension tables store descriptive attributes related to data in the fact table, facilitating efficient data retrieval for analysis.

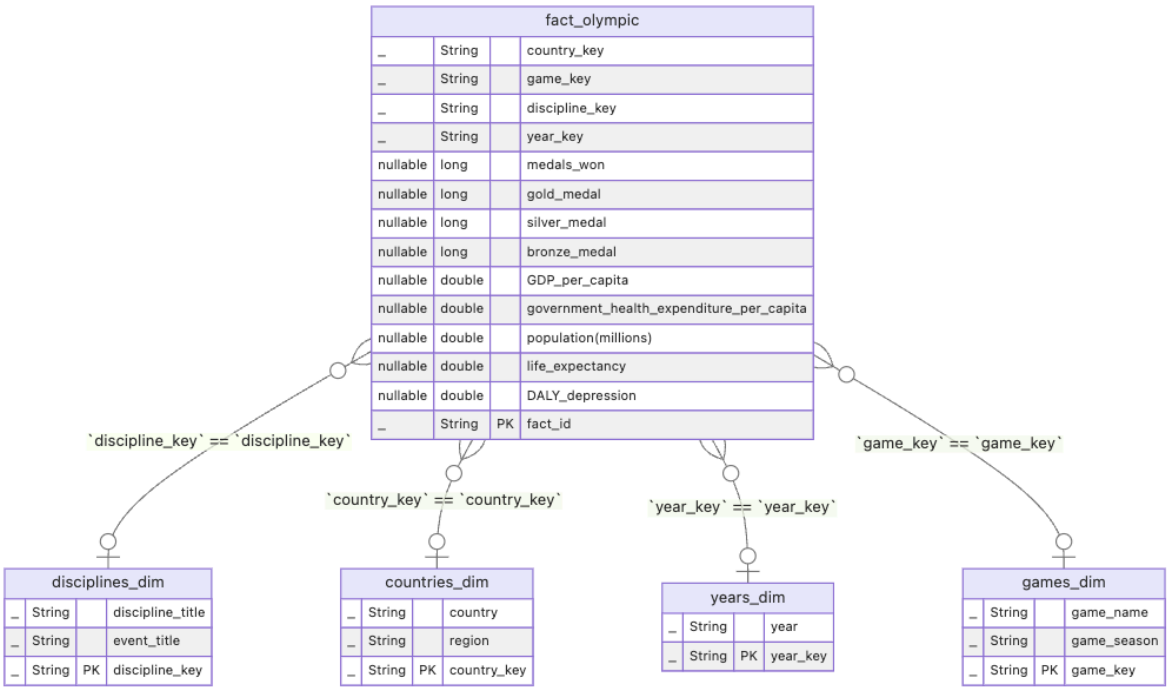


Figure 4: STAR Schema from Atoti session.

Fact Table

The fact table is the core of the star schema, containing the quantitative data (or "measures"). It links to multiple dimension tables through foreign keys.

- **Primary Key**: fact_id - uniquely identifies each record.
- **Foreign Keys**: Connects to dimension tables via country_key, game_key, discipline_key, year_key.
- **Measures**: These are the actual data points being analyzed, such as medals_won, gold_medals, silver_medals, bronze_medals, GDP_per_capita, government_health_expenditure_per_capita, life_expectancy, DALY_depression, population_millions. These metrics provide a comprehensive view of Olympic performance and contextual socio-economic and health variables.

Dimension Tables

The dimension tables contain contextual information that provides more depth to the data captured in the fact table. Each table is linked to the fact table by a primary key.

- **Country**: Holds information about countries with fields like country_key (PK), country_name, region.
- **Year**: Contains year-related information with fields such as year_key (PK), year.
- **Game**: Details about each Olympic game are stored here, including game_key (PK), game_name, game_season.
- **Discipline**: Stores information about sports disciplines with discipline_key (PK), discipline_title, event_title.

This structured approach allows for quick and efficient querying across various dimensions, making it ideal for analyzing how different factors affect Olympic outcomes.

ER DIAGRAM

An ER diagram (Entity-Relationship diagram) is a visual tool used in database design to illustrate the relationships between data entities and establish the database's logical structure.

The ER diagram below is a Star Schema design for a data warehouse in PostgreSQL. It includes four dimension tables (countries_dim, years_dim, games_dim, and disciplines_dim) and one central fact table (fact_olympic). Each dimension table contains attributes relevant to its category, such as 'country' and 'region' in the countries_dim table, and connects to the fact table via a key, such as 'country_key'.

The fact table, at the center, contains the measures of interest (like 'medals_won', 'GDP_per_capita', etc.) along with foreign keys from each dimension table. This setup allows for complex analytical queries across multiple dimensions of the data, enabling in-depth reporting and analysis of the Olympic data, such as the performance of countries, trends over the years, details of games, and disciplines of the events.

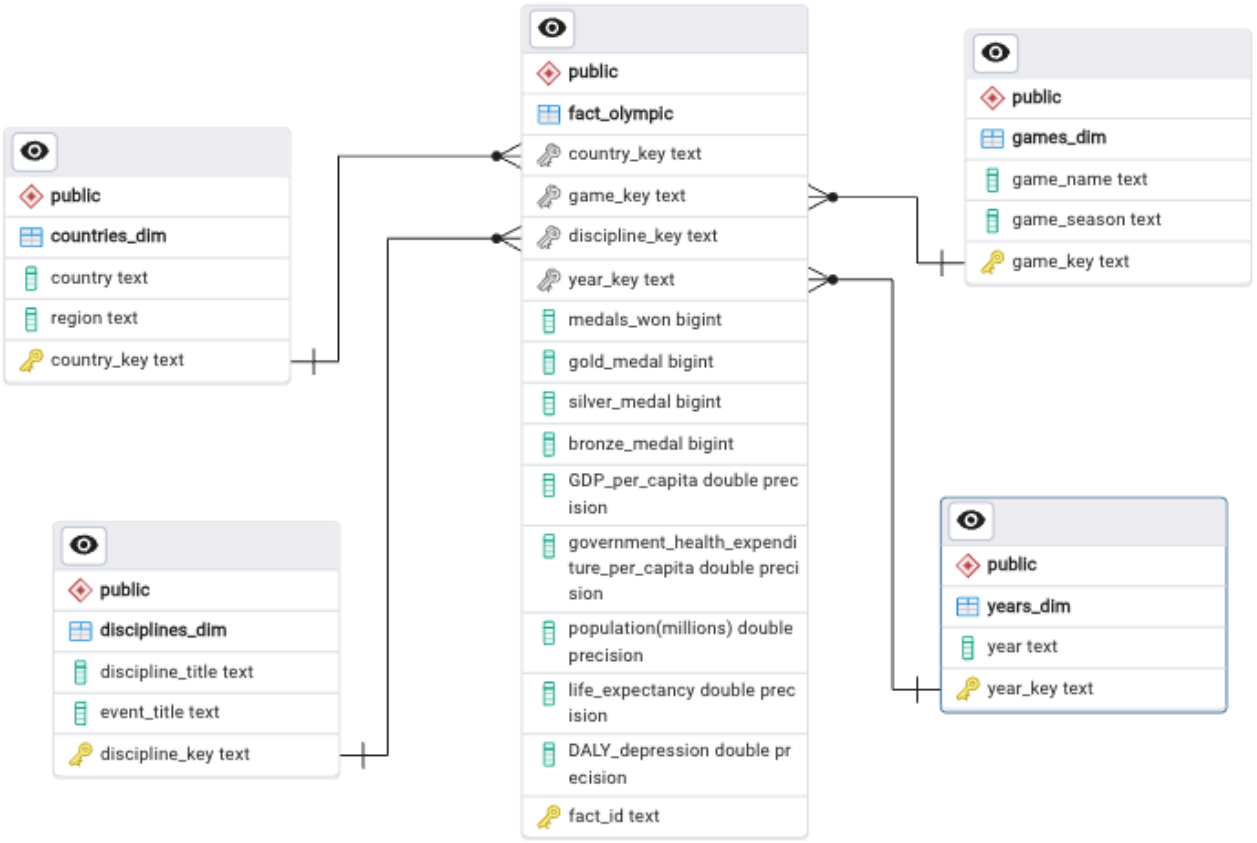


Figure 5: ER Diagram from pgadmin4.

SCHEMA HIERARCHY AND CONCEPT HIERARCHY FOR EACH DIMENSION

A concept hierarchy for each dimension has been developed, defining a sequence of mappings from detailed, low-level concepts to more generalized, higher-level ones. Diagrams for each dimension display these hierarchies; however, some nodes are omitted due to spatial constraints and are represented by ellipses instead.

1. Games Dimension

The hierarchy categorizes games by season (either Summer or Winter) at a higher level, with individual games identified by their names at the base level.

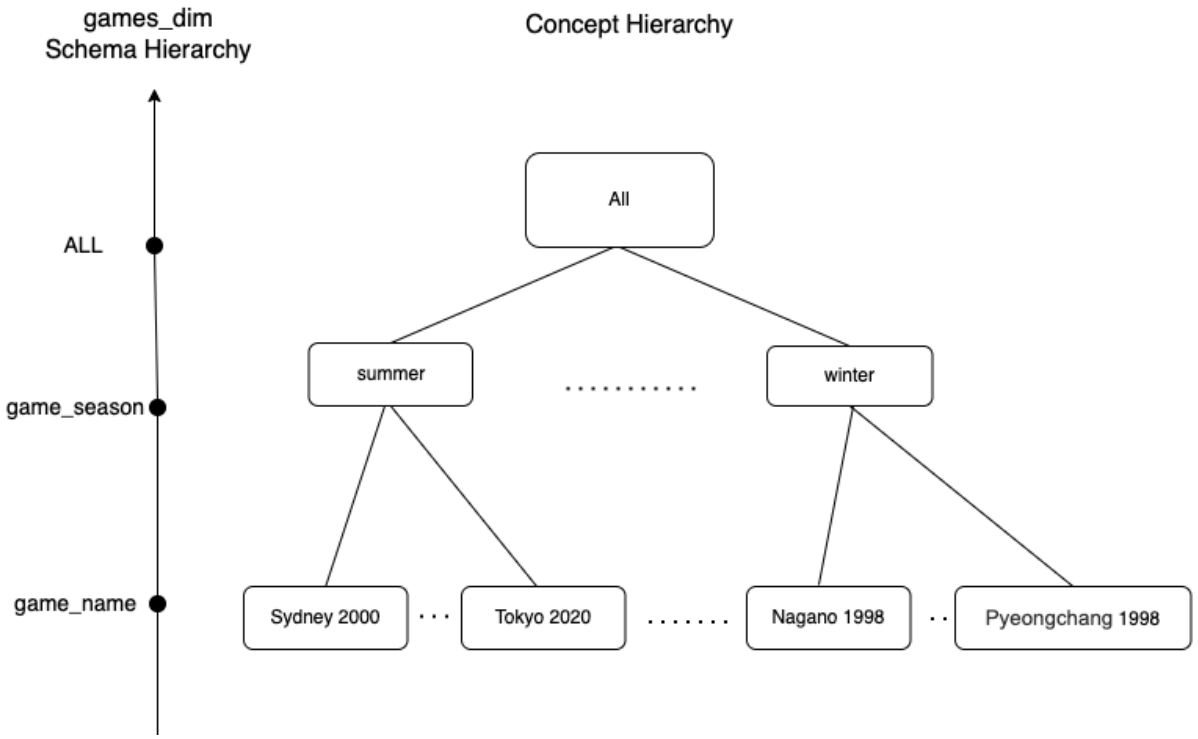


Figure 6: Games dimension schema hierarchy and concept hierarchy.

2. Countries Dimension

The country hierarchy categorizes countries by regions at a higher level, with individual country names at the base level.

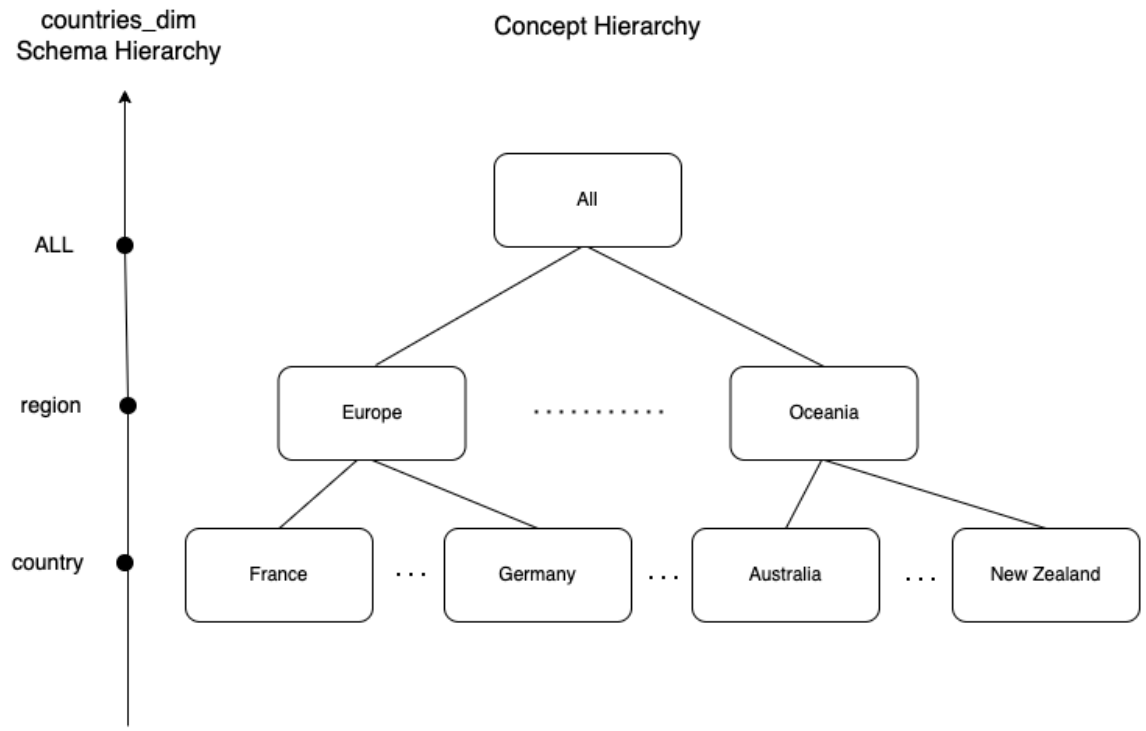


Figure 7: Countries dimension schema hierarchy and concept hierarchy.

3. Years Dimension

The hierarchy here is linear, based on chronological progression. It solely encompasses individual years as discrete entities.

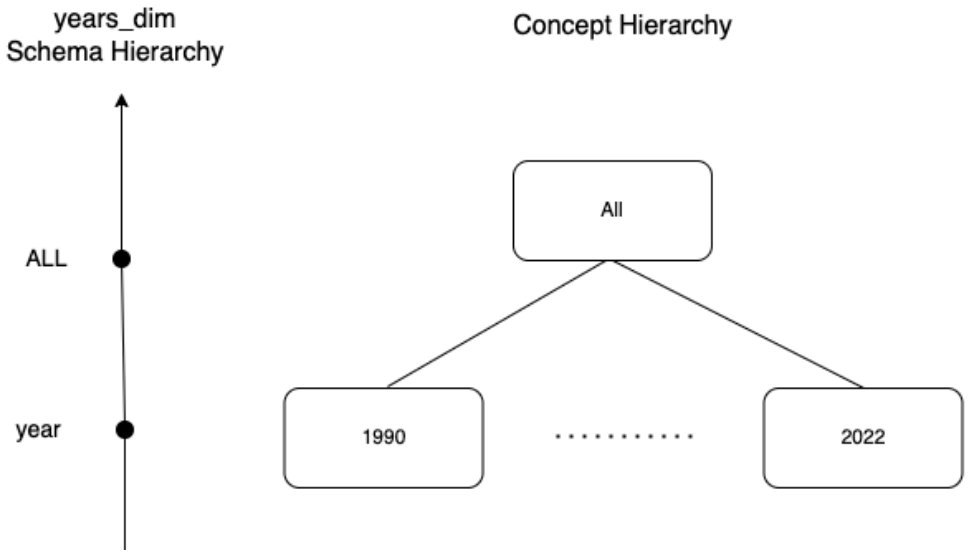


Figure 8: Years dimension schema hierarchy and concept hierarchy.

4. Disciplines Dimension

The discipline hierarchy categorizes games by disciplines at a higher level, with individual event_title identified by their names at the base level.

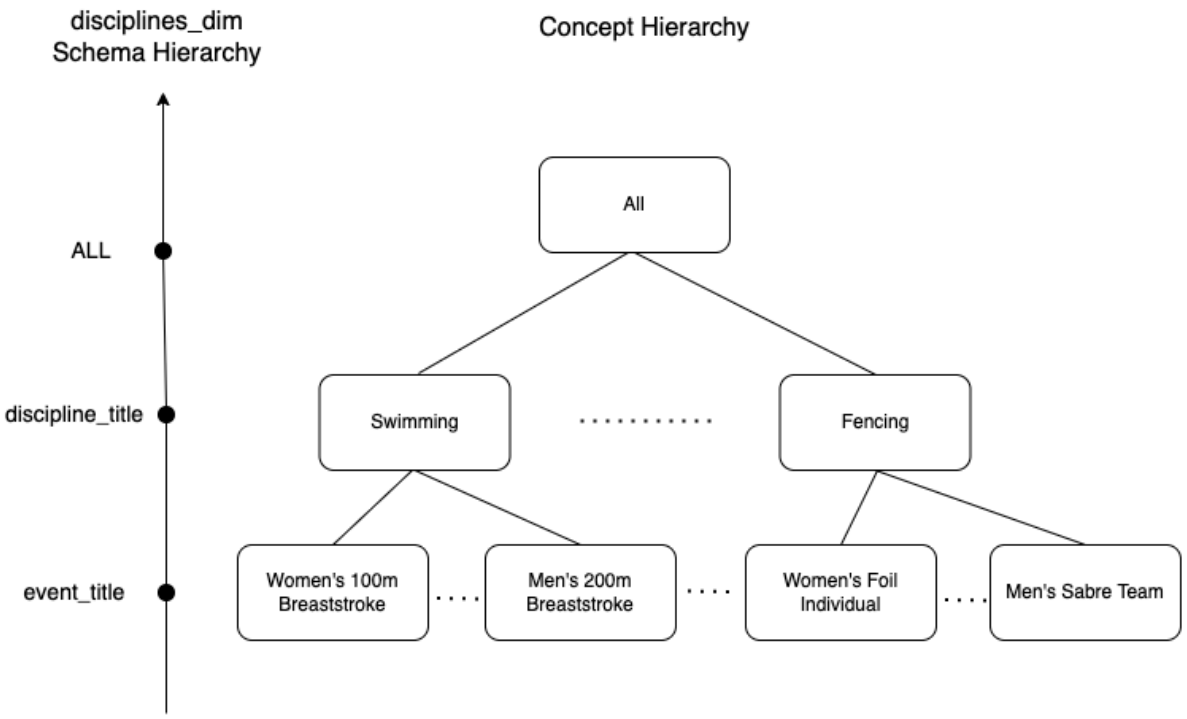


Figure 9: Disciplines dimension schema hierarchy and concept hierarchy.

EXTRACT, TRANSFORM, LOAD (ETL)

The ETL process is conducted within JupyterLab. Python and SQL are used to extract data from sources, transform (clean and reconcile) the data, and load it into a structured format suitable for analysis. All the Python scripts and SQL queries are encapsulated within a single .ipynb file which has been submitted.

Detailed Steps in the ETL Process:

1. **Data Loading:**
    - Load data from CSV files using pandas.
    - Establish a PostgreSQL connection using a JDBC URL.
    - Create tables and insert raw data into the "Olympic" database.

Constructing JDBC URL for PostgreSQL Connection

```
# Connect to the PostgreSQL database
db_name = "Olympic"
db_user = "postgres"
db_password = "postgres"
db_host = "pgdb"
db_port = "5432"

jdbc_url = f"jdbc:postgresql://{db_host}:{db_port}/{db_name}?user={db_user}&password={db_password}"

conn = psycopg2.connect(
    dbname=db_name,
    user=db_user,
    password=db_password,
    host=db_host,
    port=db_port
)

# Create a cursor object using the connection
cur = conn.cursor()
```

```
from sqlalchemy import create_engine

# Create an SQLAlchemy engine
engine = create_engine(f'postgresql://{db_user}:{db_password}@{db_host}:{db_port}/{db_name}')
```

Load Data From CSV Files

```
economic_data = pd.read_csv('olympic_data/Economic data.csv')
global_population = pd.read_csv('olympic_data/Global Population.csv', encoding='iso-8859-1')
life_expectancy = pd.read_csv('olympic_data/life-expectancy.csv')
countries_continent = pd.read_csv('olympic_data/list-of-countries_areas-by-continent-2024.csv')
mental_illness = pd.read_csv('olympic_data/mental-illness.csv')
olympic_hosts = pd.read_csv('olympic_data/olympic_hosts.csv')
olympic_medals = pd.read_csv('olympic_data/olympic_medals.csv')
```

Create Tables and Insert Raw Data in PostgreSQL (Database: Olympic)

```
# Insert data into the database from a pandas DataFrame
economic_data.to_sql('economic_data', engine, if_exists='append', index=False)
global_population.to_sql('global_population', engine, if_exists='append', index=False)
life_expectancy.to_sql('life_expectancy', engine, if_exists='append', index=False)
countries_continent.to_sql('countries_continent', engine, if_exists='append', index=False)
mental_illness.to_sql('mental_illness', engine, if_exists='append', index=False)
olympic_hosts.to_sql('olympic_hosts', engine, if_exists='append', index=False)
olympic_medals.to_sql('olympic_medals', engine, if_exists='append', index=False)
```

2. **Data Extraction:**
    - Extract raw tables from PostgreSQL into pandas DataFrames for transformation.

Extract PostgreSQL Tables into Pandas DataFrames

```
# Extract PostgreSQL tables into pandas DataFrames
economic_data_table = pd.read_sql("SELECT * FROM economic_data", engine)
global_population_table = pd.read_sql("SELECT * FROM global_population", engine)
life_expectancy_table = pd.read_sql("SELECT * FROM life_expectancy", engine)
countries_continent_table = pd.read_sql("SELECT * FROM countries_continent", engine)
mental_illness_table = pd.read_sql("SELECT * FROM mental_illness", engine)
olympic_hosts_table = pd.read_sql("SELECT * FROM olympic_hosts", engine)
olympic_medals_table = pd.read_sql("SELECT * FROM olympic_medals", engine)
```

3. **Data Cleaning and Transformation:**
    - Each dataset undergoes specific cleaning steps to ensure data quality and consistency:

3.1) countries_continent

The cleaning process of countries_continent table involves enhancing the dataset with accurate ISO country codes using the pycountry library. It first addresses mismatches and special cases through a predefined dictionary. If a country's name does not match directly, the script attempts to fetch its ISO code using alternative names (common and official). After mapping all country names to their respective codes, the process checks for any missing values. Lastly, add a new row for historical or special cases, integrating them into the main dataset for completeness.

3.2) olympic_hosts

The cleaning process for the Olympic hosts dataset involves converting dates to a standardized format using pd.to_datetime for consistency. It identifies and handles missing values, standardizes country names via a mapping dictionary to ensure alignment with countries_continent dataset, and converts the 'year' column to string for uniform data handling.

3.3) life_expectancy

Initially, the life_expectancy column names are renamed for clarity and consistency. The 'Code' column, filled with many missing values, is removed for streamlined data. Country names are standardized using a mapping dictionary to correct variations and ensure uniformity. Non-country entities are identified and removed to focus on actual countries. Country codes are then mapped to the dataset using a dictionary created from countries_continent table. For historical and special country names like the Soviet Union, 'SPECIAL' code was used to identify them. Finally, the 'year' column is converted to string format to standardize data types.

3.4) global_population

The cleaning process for the global population dataset begins by trimming irrelevant initial and final rows, replacing "no data" entries with NaN, and removing rows entirely populated by NaNs. It proceeds to address inconsistencies in country names by applying a mapping of alternate names to standardized ones from countries_continent table, removing non-country entities, and transforming the dataset into a long format suitable for time series analysis. The 'year' and 'population(millions)' columns are converted to appropriate data types. The population(millions) column was then subjected to **linear interpolation** to estimate the missing population values by connecting the dots between available data points with straight lines and also **forward/backward filling** to handle missing values effectively. Lastly, country codes are mapped and missing codes are filled with 'SPECIAL'.

Example python codes for dealing with inconsistent country names:

```python
# Mapping for country name corrections
name_correction_mapping = {
    "Bahamas, The": "Bahamas",
    "Brunei Darussalam": "Brunei",
    "Cabo Verde": "Cape Verde",
    "China, People's Republic of": "China",
    "Congo, Dem. Rep. of the": "DR Congo",
    "Congo, Republic of ": "Republic of the Congo",
    "Côte d'Ivoire": "Ivory Coast",
    "Gambia, The": "Gambia",
    "Hong Kong SAR": "Hong Kong",
    "Korea, Republic of": "South Korea",
    "Kyrgyz Republic": "Kyrgyzstan",
    "Lao P.D.R.": "Laos",
    "Macao SAR": "Macau",
    "Micronesia, Fed. States of": "Micronesia",
    "North Macedonia ": "North Macedonia",
    "Russian Federation": "Russia",
    "Slovak Republic": "Slovakia",
    "South Sudan, Republic of": "South Sudan",
    "São Tomé and Príncipe": "Sao Tome and Principe",
    "Taiwan Province of China": "Taiwan",
    "Türkiye, Republic of": "Turkey",
    "West Bank and Gaza": "Palestine",
}

# Apply the name corrections
global_population_clean['Population (Millions of people)'] = global_population_clean['Population (Millions of people)'].replace(name_correction_mapping)

# List of non-country entities to be removed.
non_country_entities = [
    'Africa (Region)', 'Asia and Pacific', 'Australia and New Zealand', 'Caribbean',
    'Central America', 'Central Asia and the Caucasus', 'East Asia', 'Eastern Europe',
    'Europe', 'Middle East (Region)', 'North Africa', 'North America', 'Pacific Islands',
    'South America', 'South Asia', 'Southeast Asia', 'Sub-Saharan Africa (Region)',
    'Western Europe', 'Western Hemisphere (Region)', 'ASEAN-5', 'Advanced economies',
    'Emerging and Developing Asia', 'Emerging and Developing Europe',
    'Emerging market and developing economies', 'Euro area', 'European Union',
    'Latin America and the Caribbean', 'Major advanced economies (G7)',
    'Middle East and Central Asia', 'Other advanced economies', 'Sub-Saharan Africa', 'World',
]

# Remove non-country entities
global_population_clean = global_population_clean[~global_population_clean['Population (Millions of people)'].isin(non_country_entities)]
```

Example python codes for performing linear interpolation and forward/backward fill:

Perform linear interpolation to estimate the missing population values by connecting the dots between available data points with straight lines.

```
# Ensure 'year' is of type int for sorting and interpolation purposes
global_population_clean['year'] = global_population_clean['year'].astype(int)

# Filter out rows with 'year' greater than 2024
global_population_clean = global_population_clean[global_population_clean['year'] <= 2024]

# Convert 'population(millions)' to a numeric type, errors='coerce' will convert non-convertible values to NaN, which can be useful if there are any non-n
global_population_clean['population(millions)'] = pd.to_numeric(global_population_clean['population(millions)'], errors='coerce')

# Sort the data by country and year to ensure interpolation works correctly
global_population_clean.sort_values(by=['country', 'year'], inplace=True)

# Perform linear interpolation on the 'population(millions)' column, grouped by country
global_population_clean['population(millions)'] = global_population_clean.groupby('country')['population(millions)'].transform(lambda x: x.interpolate())

# Convert 'year' back to string
global_population_clean['year'] = global_population_clean['year'].astype(str)

# Check if there are still missing values after interpolation
missing_after_interpolation = global_population_clean['population(millions)'].isnull().sum()

print(f"Missing values after interpolation: {missing_after_interpolation}")
```

```
Missing values after interpolation: 772
```

Forward Fill and Backward Fill to fill missing values by propagating the last known value forward or the next known value backward, respectively.

```
# Forward fill
global_population_clean['population(millions)'] = global_population_clean.groupby('country')['population(millions)'].ffill()

# Backward fill
global_population_clean['population(millions)'] = global_population_clean.groupby('country')['population(millions)'].bfill()

# Check if there are still missing values
missing_after_fill = global_population_clean['population(millions)'].isnull().sum()

print(f"Missing values after fill: {missing_after_fill}")
```

```
Missing values after fill: 0
```

3.5) mental_illness

Initially, the 'Entity' column is renamed to 'country' for clarity. A check for mismatches between this dataset and a reference country dataset identifies non-matching entities, which are then corrected using a mapping of common alternative names in countries_continent table. Non-country entities, including regional groupings and specific parts of the UK, are removed to focus solely on recognized countries. The dataset is further streamlined by selecting relevant columns, renaming them for consistency, and mapping country codes to each entry. Missing country codes are filled with 'SPECIAL', and the 'year' column is converted to string format, preparing the dataset for further analysis.

3.6) economic_data

As the economic data given only includes year 2020 data, external data was downloaded (https://databank.worldbank.org/source/world-development-indicators) to include the year from 1960 to 2022. The cleaning process for the economic dataset starts by eliminating irrelevant end rows and converting placeholders ("..") to NaN for better missing value management. Columns that do not contribute to analysis, like 'Country Code' and 'Time Code', are removed. Country names are standardized

using a predefined mapping to ensure consistency with other datasets. Non-country entities such as regions and income classifications are identified and removed to focus solely on national data. The dataset is then refined to retain only relevant economic indicators, and country codes are mapped and filled where missing. Missing values are handled through forward and backward filling to enhance data completeness. Finally, the dataset is prepared for analysis by adjusting column names and ensuring data types are appropriate, particularly converting 'year' to a string format.

3.7) olympic_medals

The cleaning process for the Olympic medals dataset starts by assessing the data shape and types, and identifying missing values. It corrects mismatches in country names using a predefined mapping, which aligns names with the reference countries_continent dataset. Non-standard entries like historical names or specific conditions are preserved for accuracy. After applying corrections, the dataset's structure is refined by selecting essential columns and renaming them for consistency. Country codes are then mapped to each country entry; missing codes are designated as 'SPECIAL'. Additional enhancements include merging with the Olympic hosts dataset to incorporate game years and adding columns to track the number of each type of medal won. The process concludes with data type conversions for year columns, ensuring uniformity and readiness for analysis.

4. Creating Fact and Dimension tables

Dimension tables are created for **countries, olympic games, disciplines,** and **years** by extracting unique entries and assigning unique keys to each dimension. The merged dataset is enriched by mapping these dimension keys back to it.

Finally, a comprehensive **fact table** is created, encapsulating key **measures** like **medal counts, GDP, health expenditure, population, life expectancy,** and **depression rates**. Each entry in the fact table is uniquely identified by a fact_id. This structured approach allows for efficient querying and analysis, linking various aspects of Olympic performance with economic and demographic indicators.

** Please note that not all transformation and cleaning processes are displayed in this report due to their extensive length. Detailed explanations are provided here for clarity. For the complete Python code and in-depth details on how the data was cleaned, please refer to the accompanying JupyterLab Notebook (ipynb file).*

5. **Loading Transformed Data into OlympicDW (PostgreSQL)**

The cleaned fact and dimension tables are loaded into a new PostgreSQL database named OlympicDW to be stored, accessed, and queried effectively. This is the final step in the ETL process, ensuring that data is available for business intelligence, reporting, and analysis purposes.

Add Cleaned Data into New Database in PostgreSQL - OlympicDW

```
# Connect to the PostgreSQL database
db_name = "OlympicDW"
db_user = "postgres"
db_password = "postgres"
db_host = "pgdb"
db_port = "5432"

jdbc_url = f"jdbc:postgresql://{db_host}:{db_port}/{db_name}?user={db_user}&password={db_password}"

conn = psycopg2.connect(
    dbname=db_name,
    user=db_user,
    password=db_password,
    host=db_host,
    port=db_port
)

# Create a cursor object using the connection
cur = conn.cursor()
```

```
from sqlalchemy import create_engine

# Create an SQLAlchemy engine
engine = create_engine(f'postgresql://{db_user}:{db_password}@{db_host}:{db_port}/{db_name}')
```

```
# Insert data into the database from a pandas DataFrame
fact_olympic.to_sql('fact_olympic', engine, if_exists='append', index=False)
years_dim.to_sql('years_dim', engine, if_exists='append', index=False)
disciplines_dim.to_sql('disciplines_dim', engine, if_exists='append', index=False)
games_dim.to_sql('games_dim', engine, if_exists='append', index=False)
countries_dim.to_sql('countries_dim', engine, if_exists='append', index=False)
```

IMPLEMENTING MULTI-DIMENSIONAL **CUBE** WITH ATOTI

Atoti will be used to construct a multi-dimensional cube for analyzing Olympic data and answering business queries for both clients. This cube allows for advanced analytical operations such as roll-up, drill-down, and pivoting, which are essential for uncovering strategic insights from complex datasets.

Setting Up an Atoti Session

```
session = tt.Session(
    user_content_storage=".content",
    port=9092,
    java_options=["-Xms1G", "-Xmx10G"]
)
```

Load PostgreSQL Data to Atoti

```
# load data from database to Atoti

fact_olympic = session.read_sql(
    "SELECT * FROM fact_olympic",
    keys=["fact_id"],
    table_name="fact_olympic",
    url=jdbc_url,
)
```

… load other data.

Implement a Star Schema in Atoti

```
fact_olympic.join(years_dim, fact_olympic["year_key"] == years_dim["year_key"])

fact_olympic.join(disciplines_dim, fact_olympic["discipline_key"] == disciplines_dim["discipline_key"])

fact_olympic.join(games_dim, fact_olympic["game_key"] == games_dim["game_key"])

fact_olympic.join(countries_dim, fact_olympic["country_key"] == countries_dim["country_key"])
```

Create a Cube

```
cube = session.create_cube(fact_olympic)
cube
```

▾ **fact_olympic**
  ▸ **Dimensions**
  ▸ **Measures**

After creating the cube, the hierarchies and measures within the cube were cleaned up and organized to match the StarNet design.

More detailed drill down and roll up analyses of the cube are shown in the visualization section using Atoti and Tableau.

```
cube
```


- **fact_olympic**
  - **Dimensions**
    - **countries_dim**
      - **countries_dim** [] *2 items*
        - **0** "region"
        - **1** "country"
      - **country** [] *1 item*
        - **0** "country"
      - **region** [] *1 item*
        - **0** "region"
    - **disciplines_dim**
      - **discipline_title** [] *1 item*
        - **0** "discipline_title"
      - **disciplines_dim** [] *2 items*
        - **0** "discipline_title"
        - **1** "event_title"
      - **event_title** [] *1 item*
        - **0** "event_title"
    - **fact_olympic**
      - **country_key** [] *1 item*
        - **0** "country_key"
      - **discipline_key** [] *1 item*
        - **0** "discipline_key"
      - **fact_id** [] *1 item*
        - **0** "fact_id"
      - **game_key** [] *1 item*
        - **0** "game_key"
      - **year_key** [] *1 item*
        - **0** "year_key"
    - **games_dim**
      - **game_name** [] *1 item*
        - **0** "game_name"
      - **game_season** [] *1 item*
        - **0** "game_season"
      - **games_dim** [] *2 items*
        - **0** "game_season"
        - **1** "game_name"
    - **years_dim**
      - **year** [] *1 item*
        - **0** "year"
      - **years_dim** [] *1 item*
        - **0** "year"
  - **Measures**
    - **DALY_depression.MEAN**
    - **DALY_depression.SUM**
    - **GDP_per_capita.MEAN**
    - **GDP_per_capita.SUM**
    - **bronze_medal.MEAN**
    - **bronze_medal.SUM**
    - **contributors.COUNT**
    - **gold_medal.MEAN**
    - **gold_medal.SUM**
    - **government_health_expenditure_per_capita.MEAN**
    - **government_health_expenditure_per_capita.SUM**
    - **life_expectancy.MEAN**
    - **life_expectancy.SUM**
    - **medals_won.MEAN**
    - **medals_won.SUM**
    - **population(millions).MEAN**
    - **population(millions).SUM**
    - **silver_medal.MEAN**
    - **silver_medal.SUM**

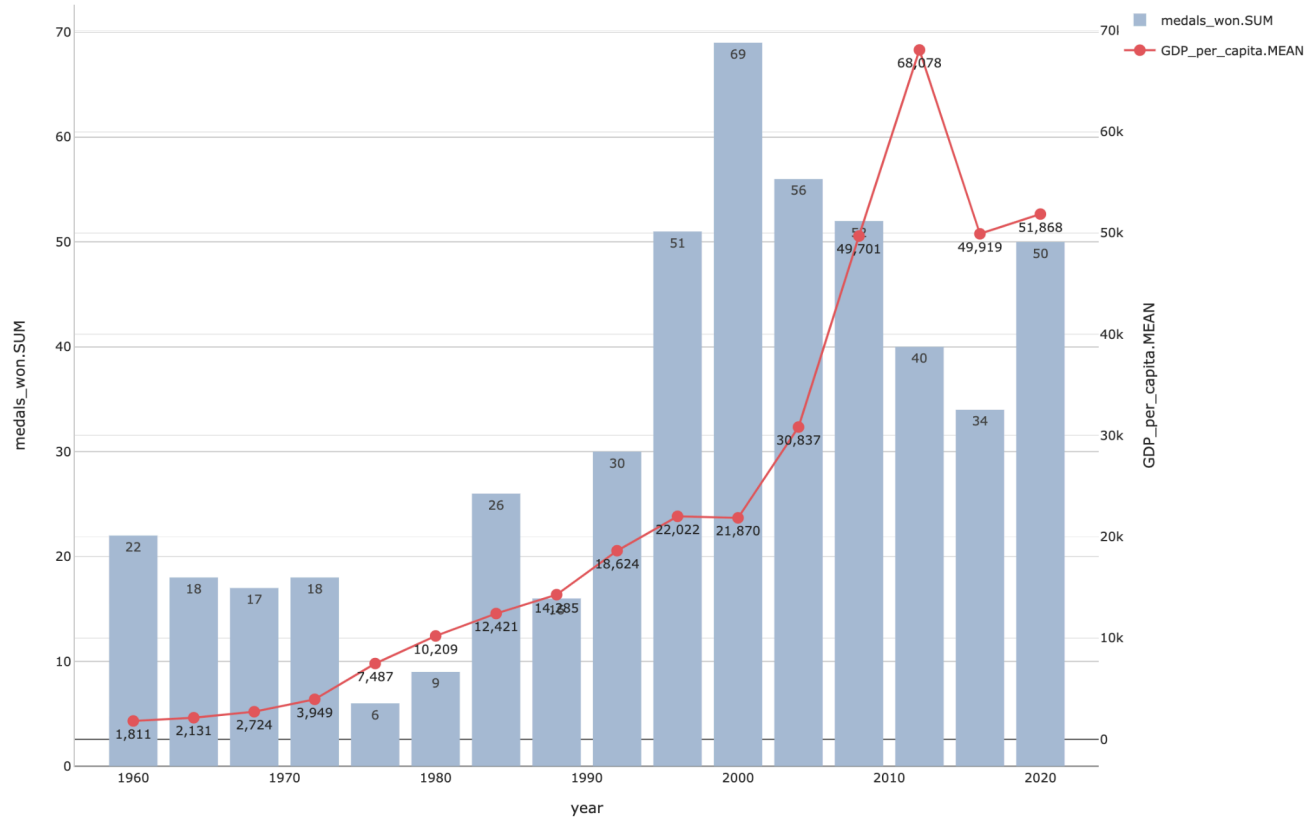
TABLEAU AND ATOTI VISUALISATION FOR BUSINESS QUERIES

Both Tableau and Atoti are used for visualizing query results. Tableau offers interactive dashboards, extensive visualization options, and real-time data integration, facilitating collaborative decision-making. Atoti excels in multidimensional analysis, Python integration, and scenario simulation, ideal for in-depth analytical tasks and interactive data exploration within Jupyter notebooks.

Business Queries for Client A (Australian Government):

1. How does Australia's GDP per capita correlate with its performance in the Summer Olympics across different years?

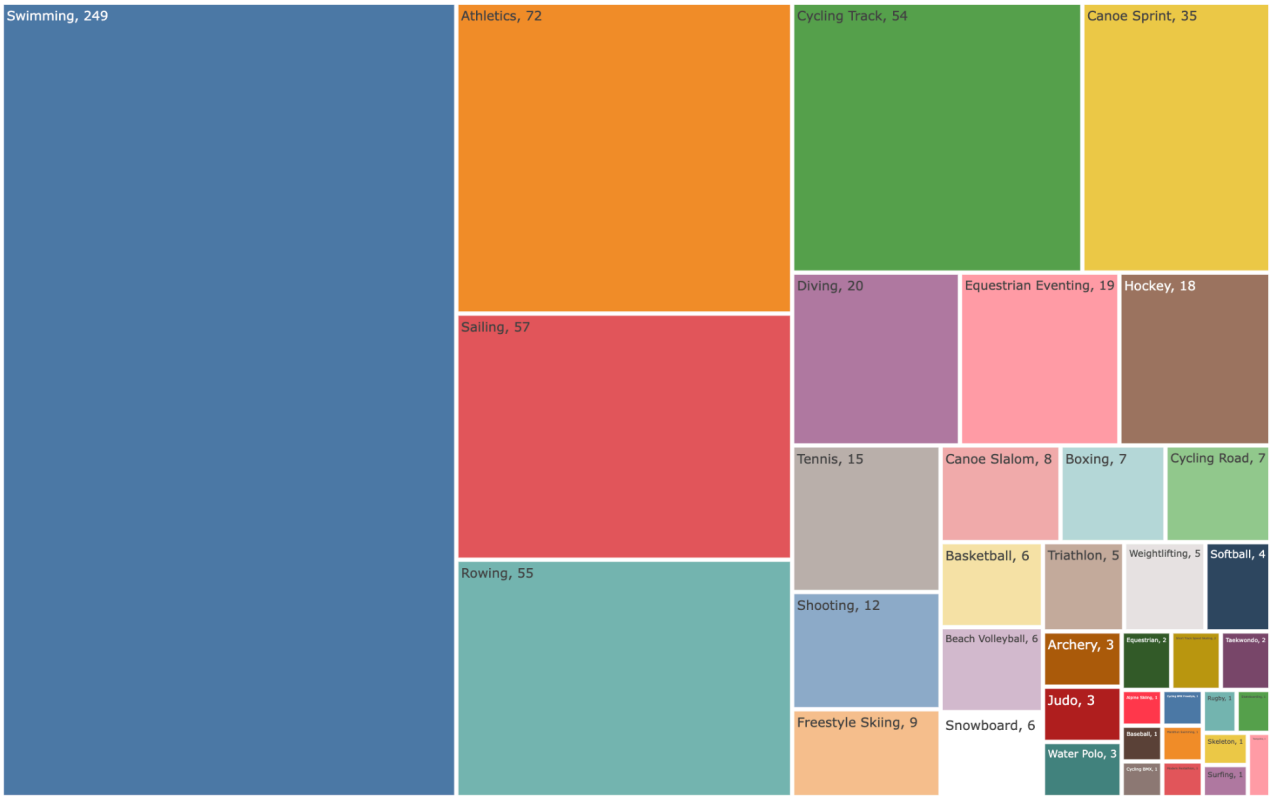
The relationship between Australia's GDP per capita and its total medal won in the Summer Olympics over the years.



The general trend indicates that as Australia's GDP per capita increased, so did the number of medals won, especially noticeable in years when Australia performed exceedingly well like 2000 and 2016. This suggests that higher economic capacity may contribute to better resources for sports training and facilities, potentially aiding Olympic success.

2. **What is the distribution of Olympic medals won by Australia across different disciplines throughout the history of the Games, and which sport has yielded the highest number of medals?**

Total Olympic medals won by Australia across different disciplines, visualized in a treemap.

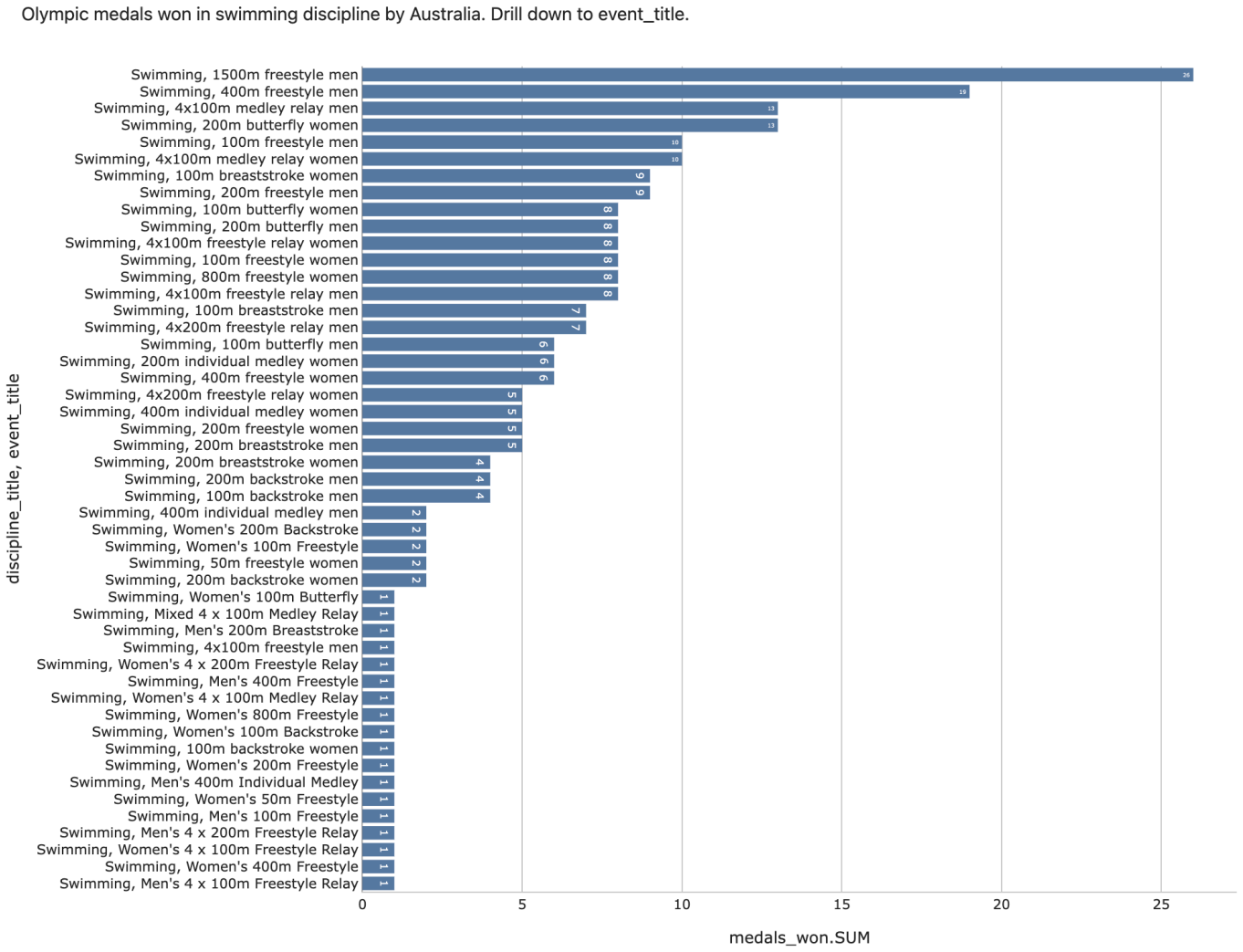


- **Swimming** emerges as the standout discipline for Australia, accounting for a significant proportion of the total medals (249 out of 695), making it the sport that has yielded the highest number of medals for Australia.
- **Athletics** and **Cycling Track** also show strong performances, with substantial contributions to the total medal count.
- Other water-related sports like **Rowing** and **Sailing** have also been successful for Australia, indicating a strong tradition and investment in these areas.

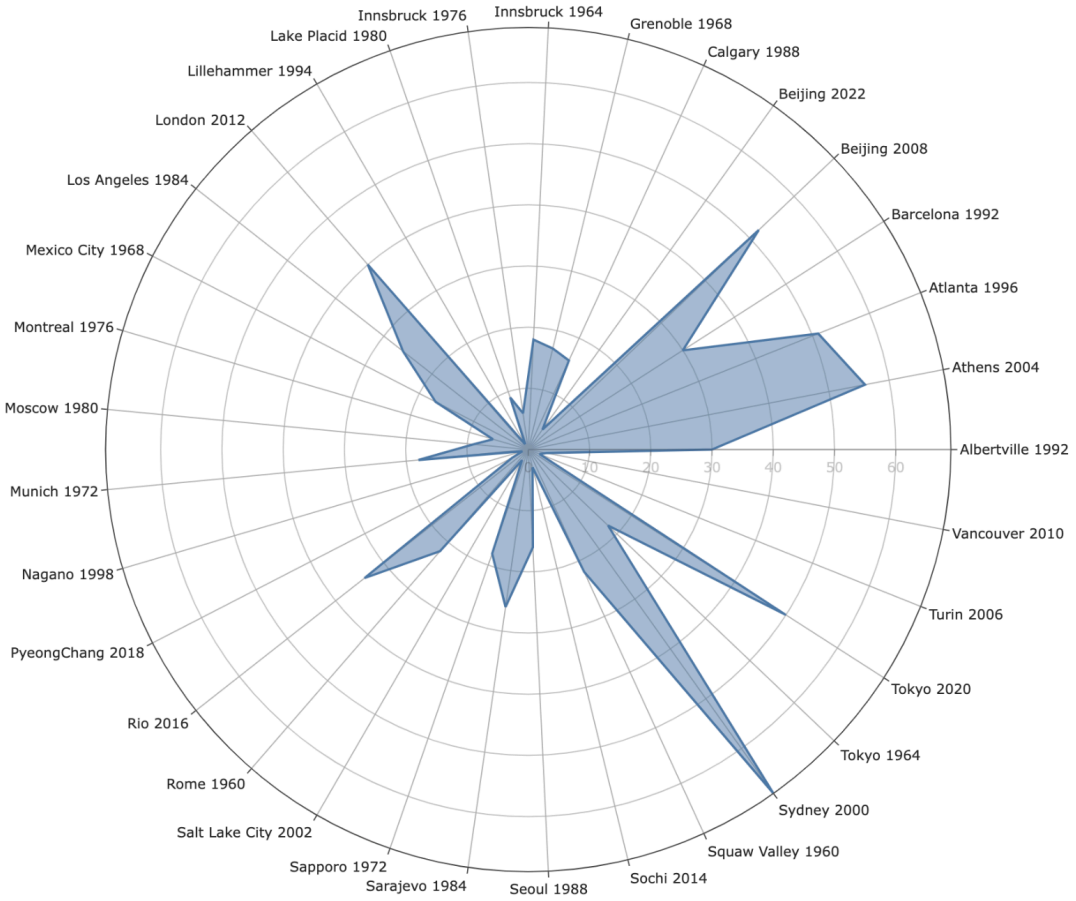
The distribution of medals across disciplines highlights Australia's strong sporting diversity and prowess, particularly in water sports and athletics. This data can provide insights into areas where Australia excels and where it may focus future athletic training and development.

Drill-Down Analysis:

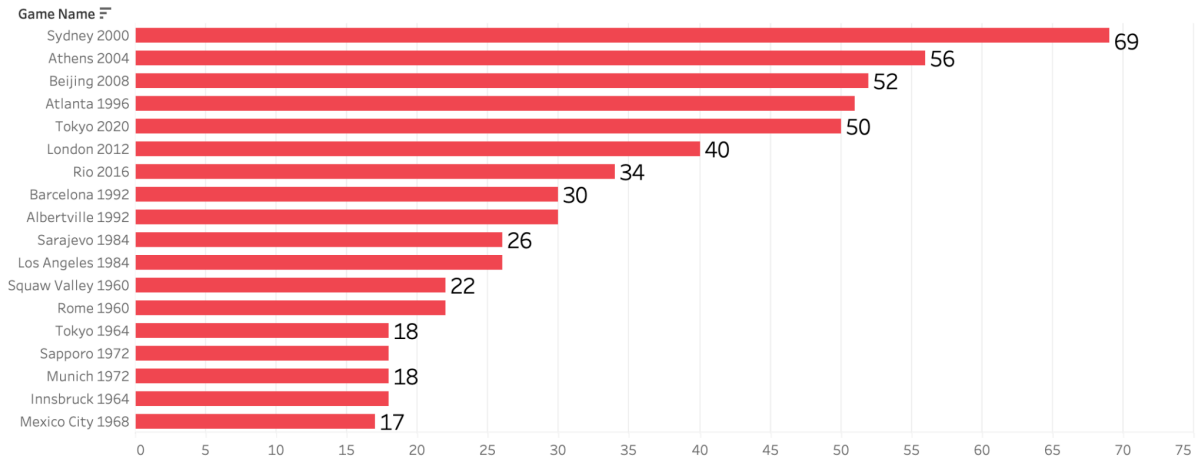
Upon drilling down into the swimming category, it was found that the men's 1500m freestyle event yielded the highest number of medals, totaling 26. This analysis aids in pinpointing specific areas where Australia has excelled historically and could inform strategic enhancements in sports training programs for future competitions.



3. **Does Australia achieve better results in Olympic Games held domestically (home games) compared to those hosted internationally (away games)?**

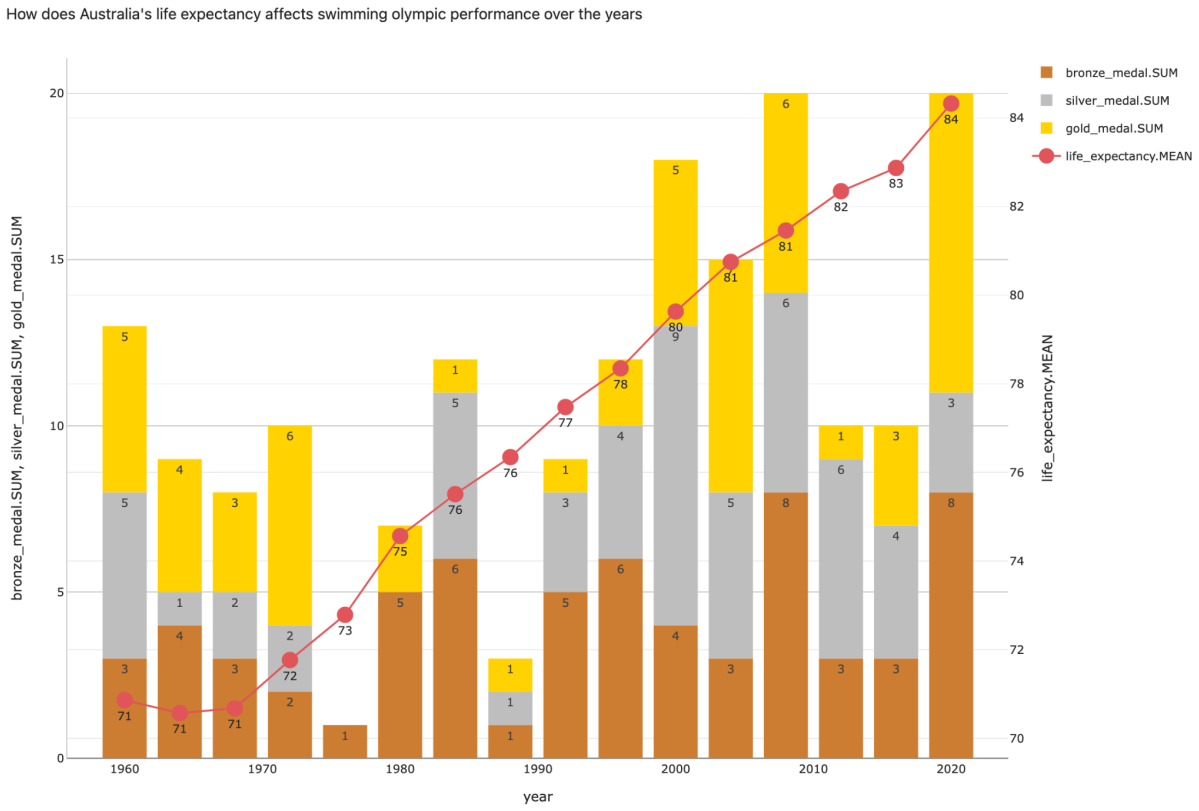


Australia's Olympic Performance: Home vs. Away Games



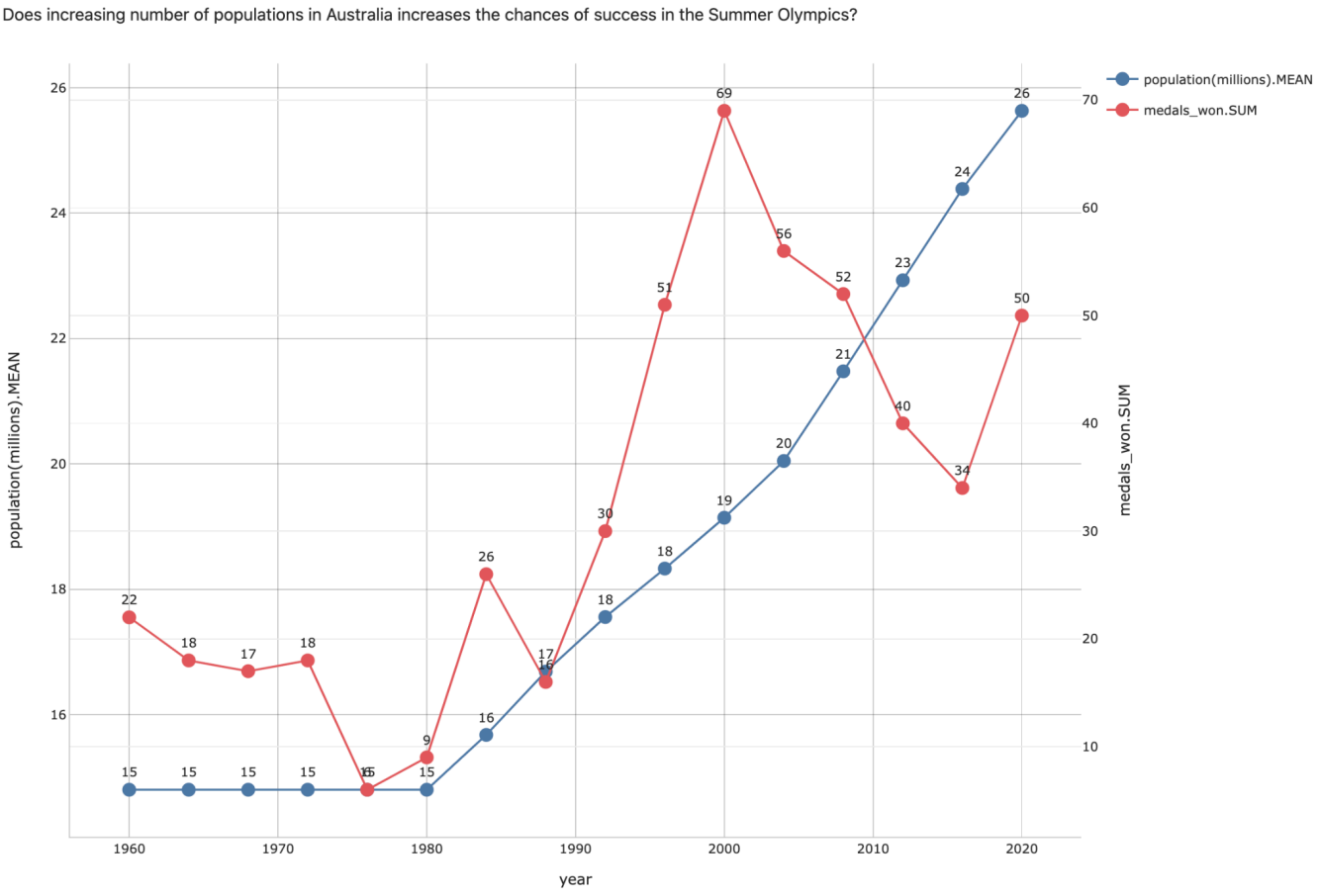
Sydney 2000: Australia won 69 medals, which is significantly higher than its performance in most other editions of the Games. This is the highest medal count in the dataset for Australia and supports the notion that hosting the Olympics may have a positive impact on the host nation's medal tally.

4. **How does the changing life expectancy in Australia over the years influence the country's success rate in winning gold, silver, and bronze medals in swimming at the Summer Olympics?**



The data suggests a correlation between Australia's increasing life expectancy and its Olympic success in swimming, particularly noticeable in the number of gold and silver medals won. Higher life expectancy could indicate better overall health and wellness in the population, which might contribute to enhanced athletic performance. Years with higher life expectancies align with Australia's most successful years in Olympic swimming, suggesting that improvements in health and longevity could be contributing factors to athletic excellence.

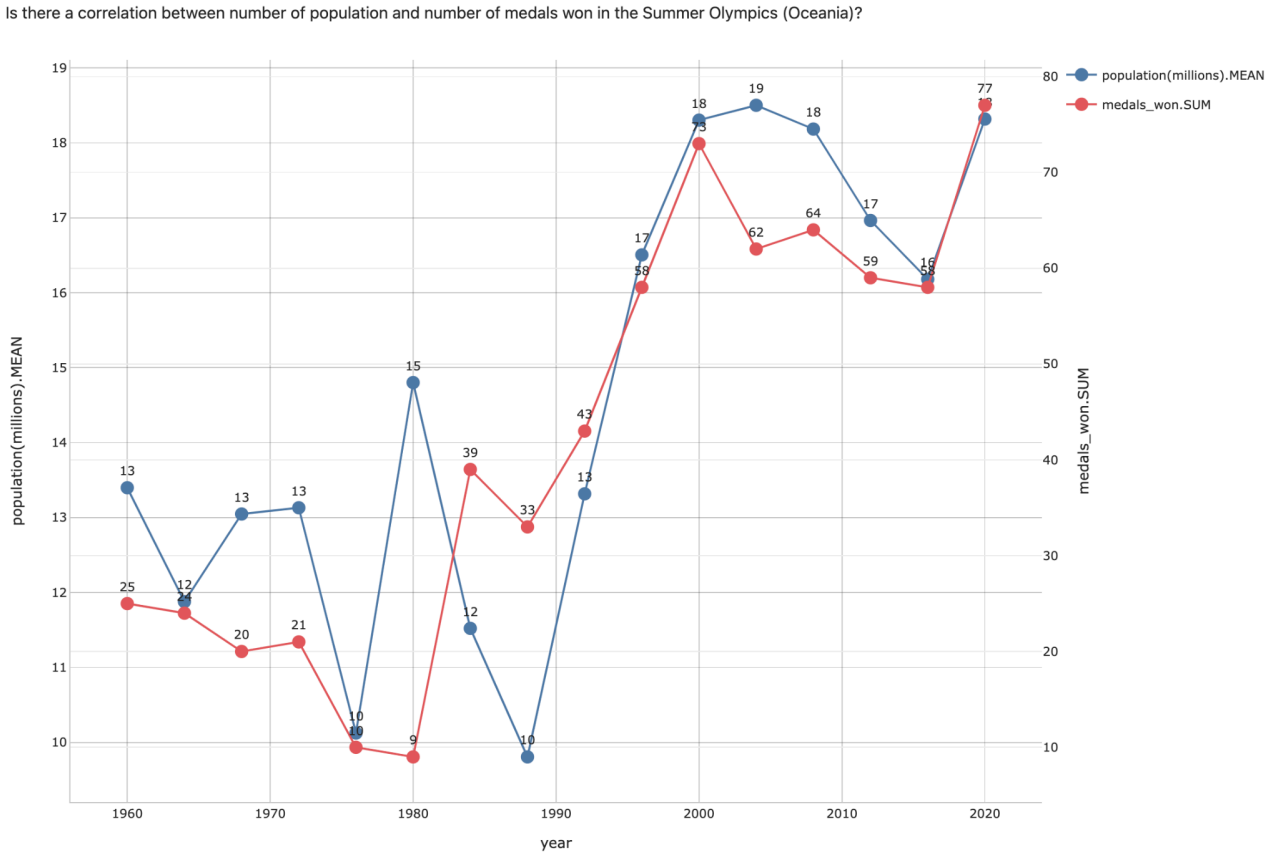
5. Is there a correlation between population size in Australia and the total number of medals won at the Summer Olympics across the years?



The data indicates an upward trend in medal counts as the population increases, particularly noticeable around the Sydney 2000 Olympics. However, while there is an apparent correlation in some years, the relationship is not consistently linear, as seen in the fluctuations post-2000 despite continuous population growth.

Roll-Up Analysis:

Is there a correlation between population size and the total number of medals won at the Summer Olympics when analyzed across the Oceania region?



The roll-up analysis examines whether there's a correlation between the population size and the total number of medals won at the Summer Olympics, focusing on the Oceania region.

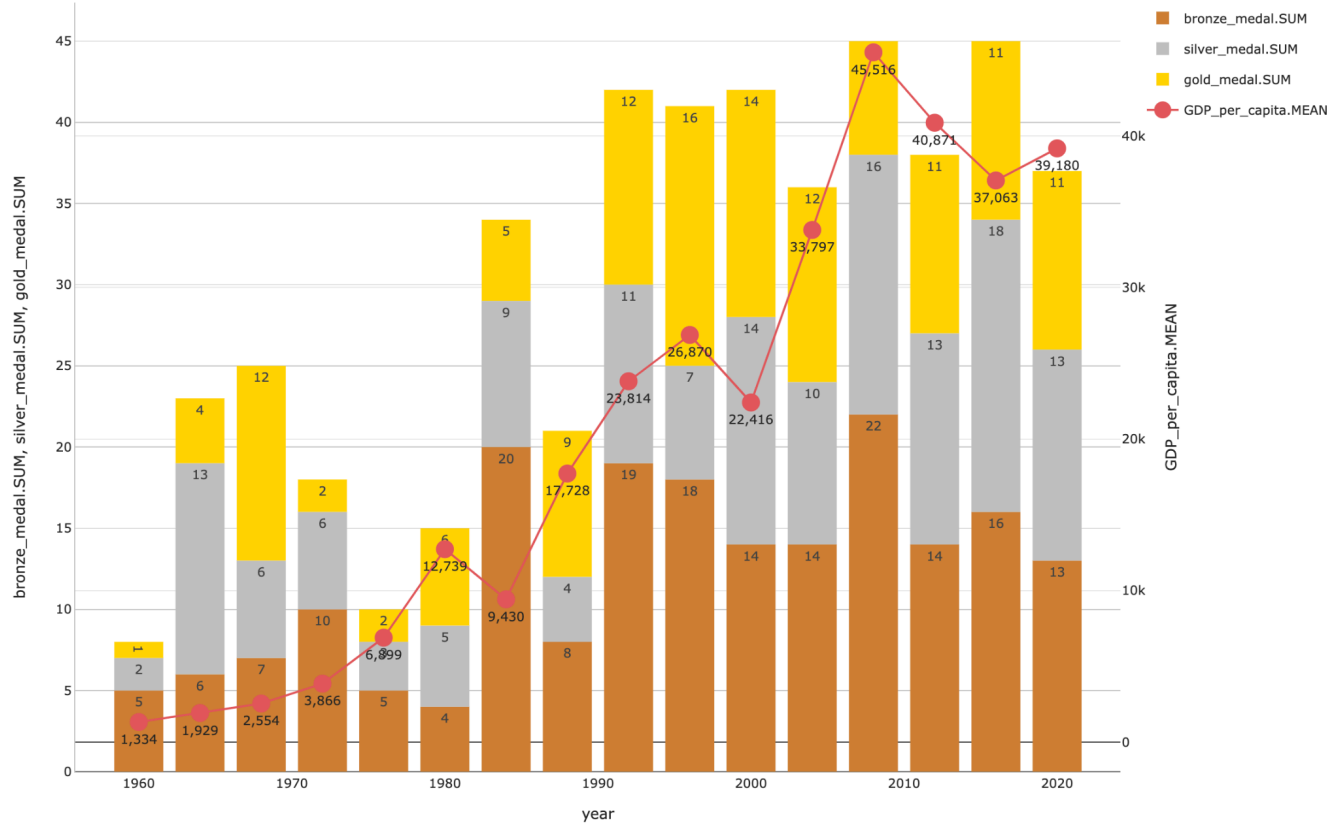
The year 2000 shows a significant peak in medals (73 medals), which correlates with the population peak and the Olympic Games being hosted in Sydney. This supports the notion that hosting the Olympics likely provided a substantial boost to the region's medal count. Following 2000, despite slight fluctuations in population, there's a noticeable decline in the number of medals won. This decline in medals could be attributed to the "post-hosting" effect, where the initial surge in infrastructure and investment in sports may not have been sustained.

Population vs. Medal Count Correlation: The data after 2000 suggests that while the population size in Oceania slightly decreases, the total medals won also decrease, which could imply a correlation.

Business Queries for Client B (French Government):

1. How does the GDP per capita of France correlate with its total medal count at the Summer Olympics over the years?

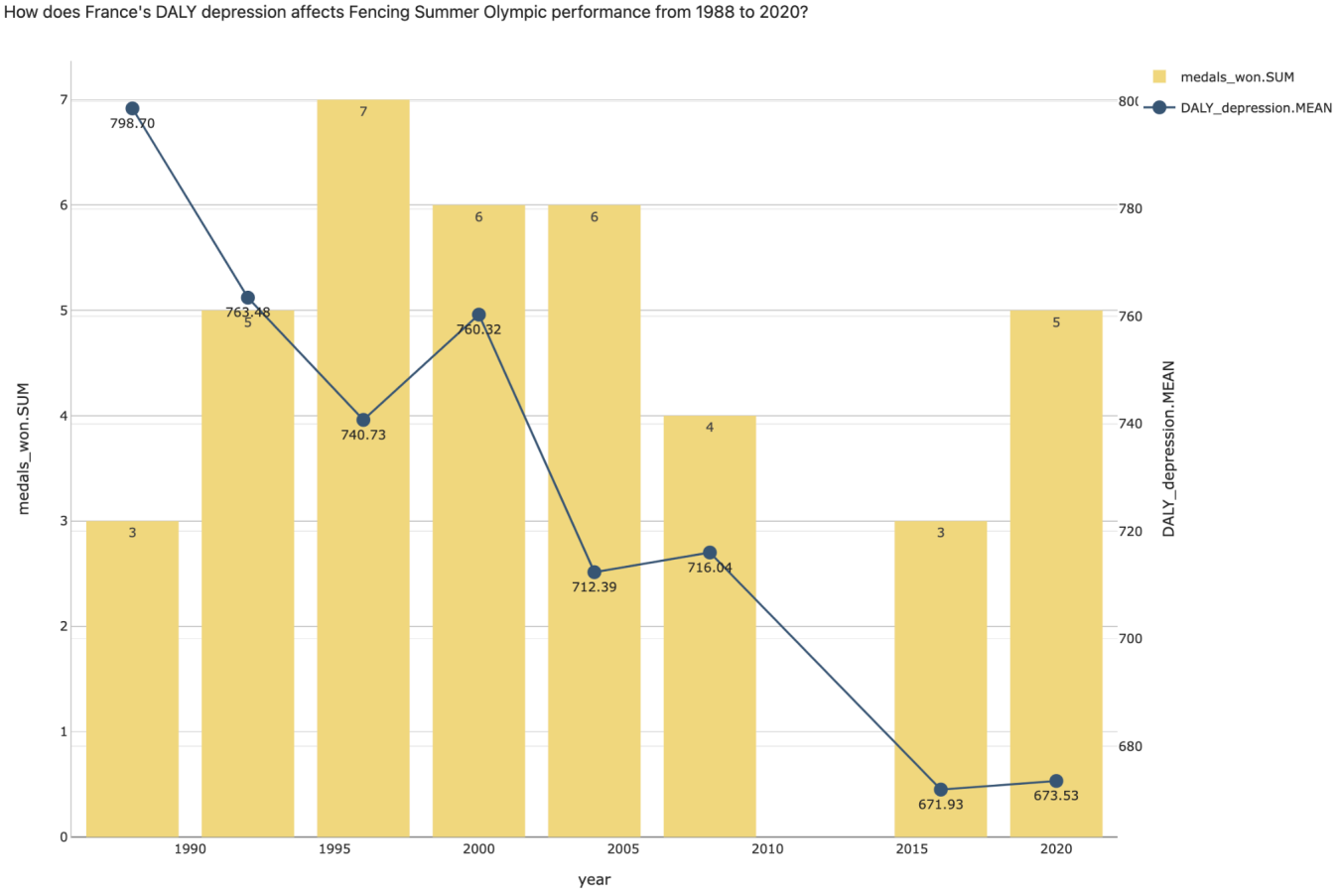
The relationship between France's GDP per capita and its total medal won in the Summer Olympics over the years.



While there is a general trend of increasing GDP correlating with higher medal counts, it is not strictly linear. For example, in 1980, despite a lower GDP per capita of $12,739, France achieved a relatively high total medal count (15), and in 2008, with the highest GDP per capita listed ($45,516), France secured the most medals (45).

The data suggests a possible correlation between France's GDP per capita and its Olympic success, with higher GDP per capita years often aligning with higher medal counts. However, the relationship is not directly proportional, as other factors such as investment in sports infrastructure, athlete training programs, and overall national focus on Olympic preparation play significant roles.

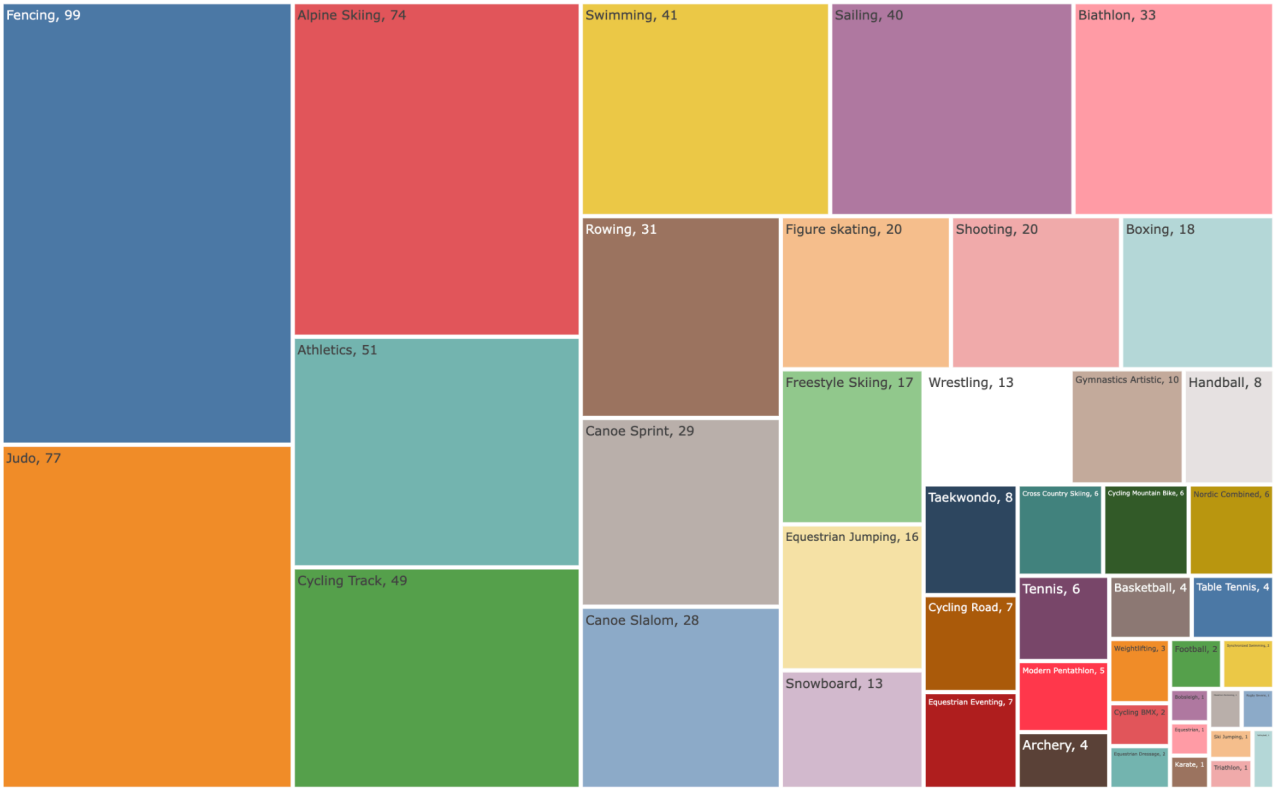
2. **How does the burden of depression, as measured by Disability-Adjusted Life Years (DALYs), correlate with France's Fencing Summer Olympic performance from 1988 to 2020?**



Based on the result, there does not seem to be a direct or clear correlation between DALY rates for depression and France's fencing performance in the Summer Olympics. While one might expect that a lower burden of depression would correlate with better athletic performance, the data does not show a consistent pattern that supports this hypothesis. This indicates that the success of France in fencing is likely influenced by a multitude of factors beyond the mental health indicator of depression DALY rates.

3. **What is the distribution of Olympic medals won by France across various sports throughout the history of the Olympic Games, and which sport has delivered the highest number of medals?**

Total Olympic medals won by France across different disciplines, visualized in a treemap.

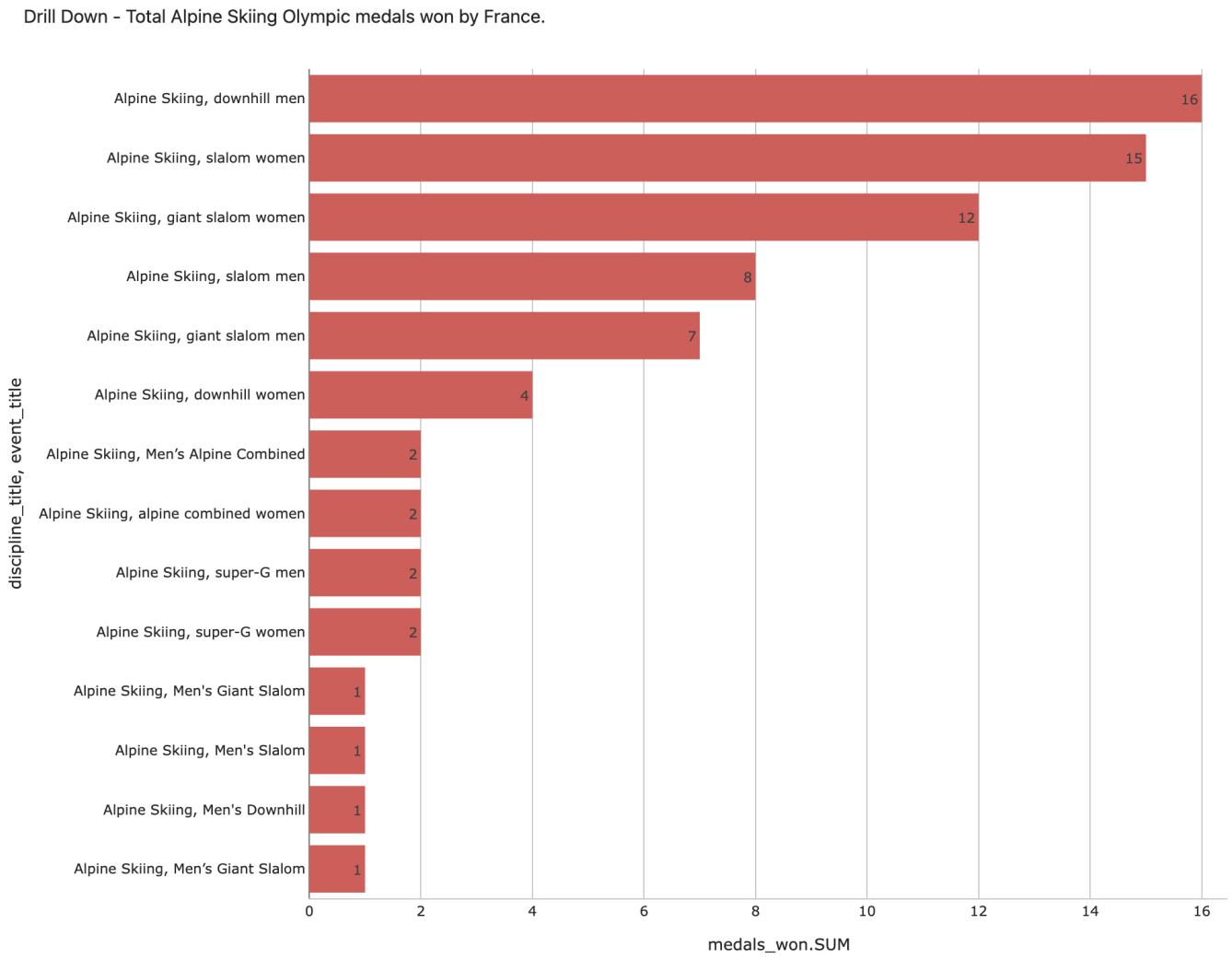


France has demonstrated exceptional performance in fencing, leading to a significant number of Olympic medals (99 medals) and highlighting the country's historical strength in this sport. The success in judo (77 medals) and alpine skiing (74 medals) underscores France's versatility and ability to cultivate Olympic-level talent across diverse sports.

The data reflects the country's investment in sports development and the success of its national sports programs in nurturing athletes who can compete on the world stage. The achievements in technical and individual sports like fencing, judo, and alpine skiing suggest a focus on specialized training programs that have translated into Olympic success.

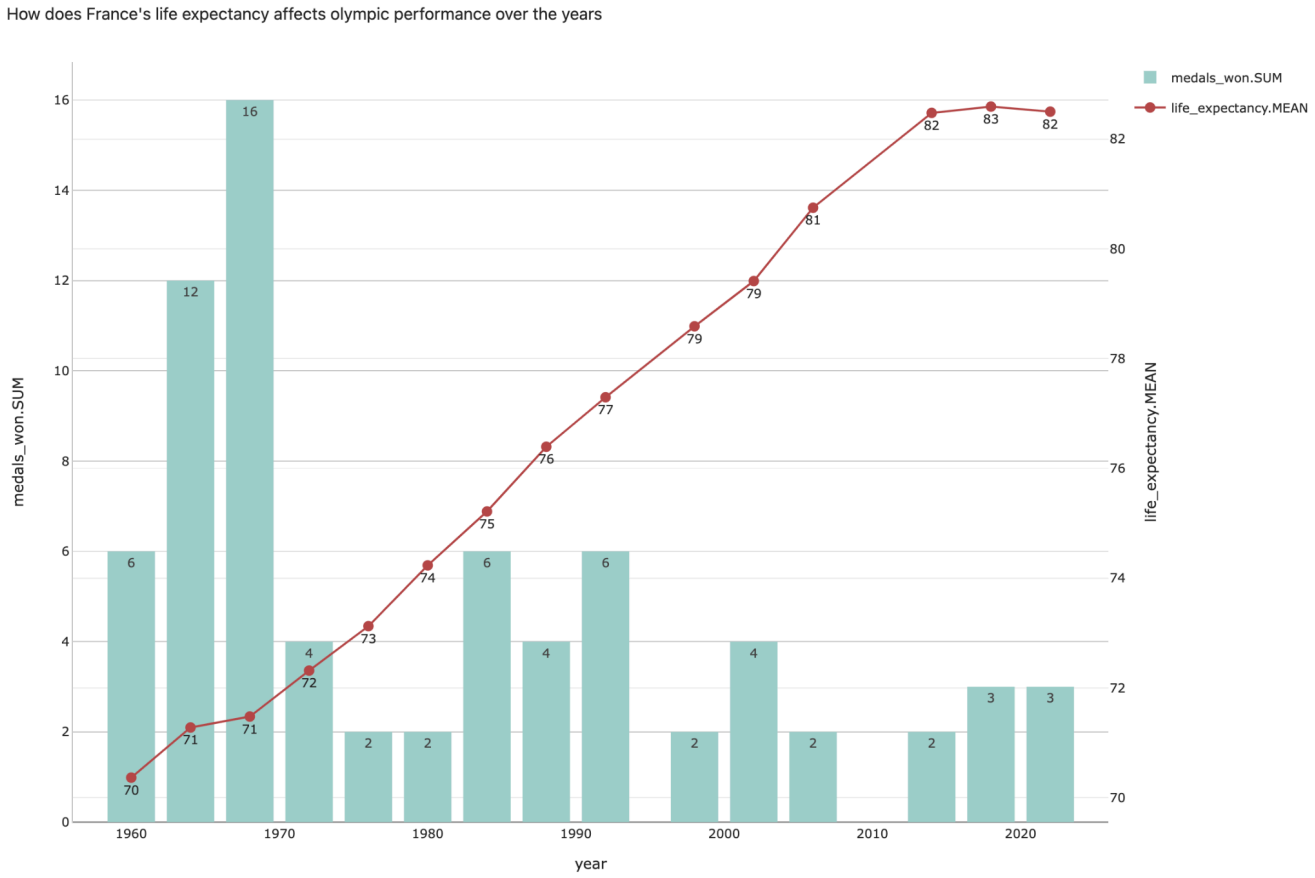
Drill-Down Analysis:

Breakdown of the total number of Olympic medals won by France in the discipline of Alpine Skiing, divided by specific events within the discipline.



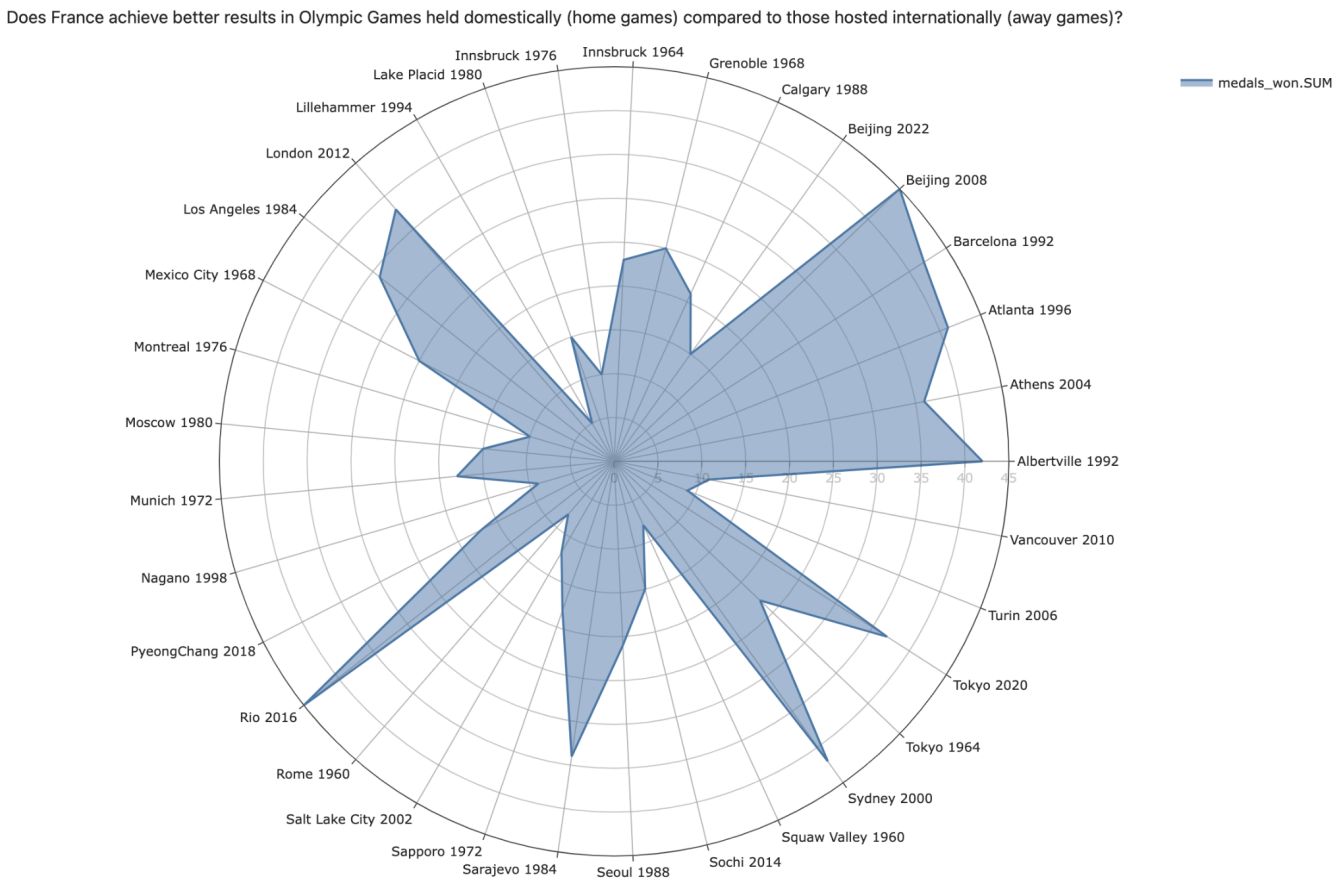
- France has demonstrated strong performances in technical skiing events, particularly in men's downhill (16 medals) and women's slalom (15 medals).
- There is notable success in both men's and women's events, indicating a well-rounded national skiing program.
- The medal counts suggest consistent excellence and competitive presence in Alpine Skiing events across multiple Olympic Games.

4. How does the life expectancy in France correlate with the number of medals won in Alpine Skiing at the Olympics across the years?



There doesn't appear to be a direct correlation between life expectancy and the number of medals won. While life expectancy has steadily increased, the Alpine Skiing medal count varies significantly across the years, not showing a pattern that correlates with the improvements in life expectancy. While increased life expectancy often reflects better overall health conditions, which could contribute to sports performance, the data does not indicate that higher life expectancy directly leads to more Olympic medals in Alpine Skiing for France. The success in Alpine Skiing likely depends more on specific factors such as athlete training, skill, and conditions during the Olympic Games, rather than general health indicators like life expectancy.

5. Does France achieve improved outcomes in Olympic competitions hosted domestically as opposed to those held in international venues?



Domestic Performance: The 1968 Grenoble Winter Games, a home Olympics for France, resulted in 25 medals for the country, and in Albertville 1992 they won 42 medals.

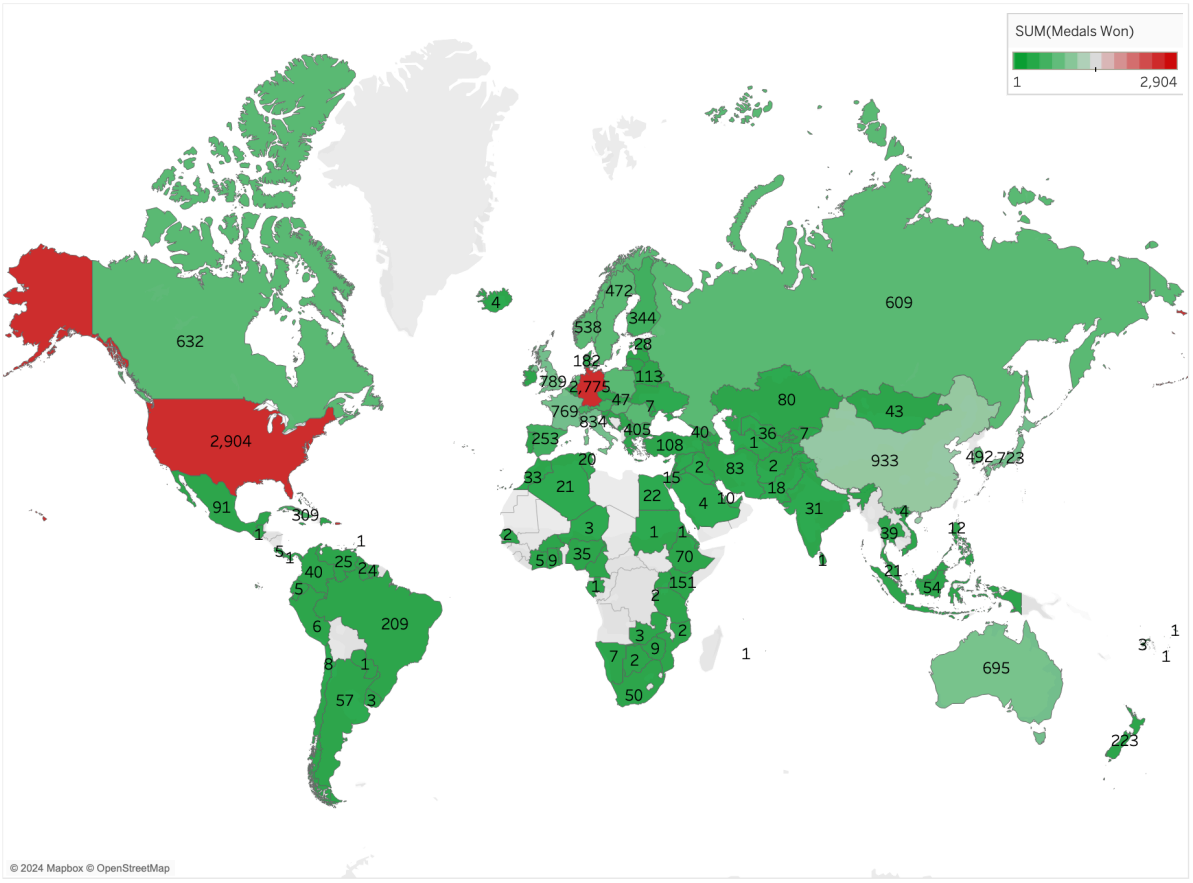
International Performance: Comparing France's domestic performance to international ones, France has achieved more significant success in away games, such as in Beijing 2008 and Rio 2016, where they won 45 medals each, Barcelona 1992, and Sydney 2000, where they won 42 medals each.

This suggests that France does not necessarily perform better in home games compared to away games. In fact, some of the best Olympic performances by France have occurred internationally.

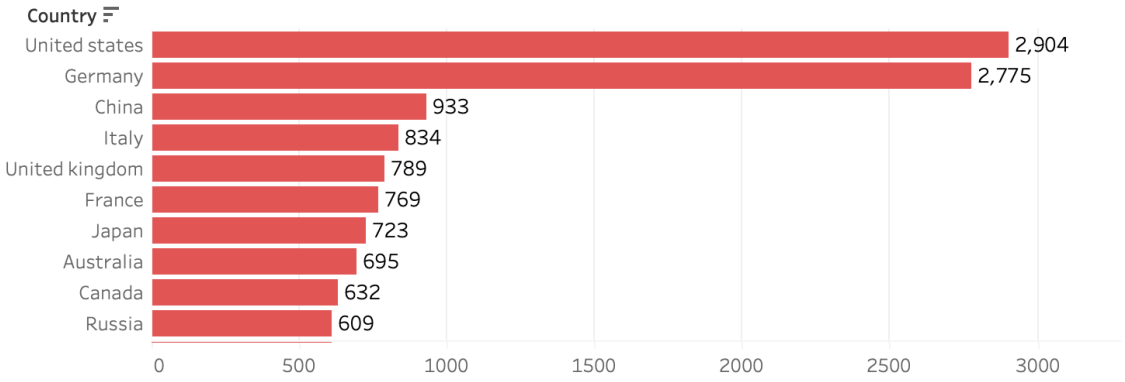
Extra Business Query for Both Clients:

Identify the top 10 countries with the highest cumulative medal counts in Olympic history for a comprehensive overview of historical Olympic success.

Total Number of Olympic Medals Won in History - World Map View



Top 10 countries with the highest cumulative medal counts in Olympic history



The charts above illustrate the top 10 countries with the most cumulative medals in Olympic history, with both Australia and France featuring prominently on the list. Australia has amassed a total of 695 medals, while France has achieved 769, placing them both within the elite group of the most successful countries in the Olympic Games.

ASSOCIATION RULE MINING

The association rule mining process involves analyzing data to uncover hidden patterns and relationships.

- First, key Python libraries such as pandas, numpy, and mlxtend are imported.
- Data is then extracted from PostgreSQL into pandas DataFrames, grouped by key attributes, and aggregated to summarize features like medal counts and average GDP per capita.
- The data is cleaned, removing unnecessary columns and converting numerical data into categorized ranges, such as discretizing GDP into quartiles.
- This transformed data is structured into a format suitable for mining—transactions are encoded using mlxtend's TransactionEncoder.
- Apriori algorithm is applied to find frequent itemsets, and from these, association rules are generated.
- These rules are sorted by confidence and lift to identify the strongest relationships, with the top 10 rules presented for review.

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
7	(Very Low GDP)	(Few Medals)	0.250219	0.576822	0.185250	0.740351	1.283500	0.040918	1.629808	0.294594
3	(Low GDP)	(Few Medals)	0.250219	0.576822	0.153644	0.614035	1.064514	0.009311	1.096416	0.080830
8	(High GDP)	(Many Medals)	0.250219	0.423178	0.151010	0.603509	1.426134	0.045122	1.454816	0.398521
4	(Medium GDP)	(Few Medals)	0.249342	0.576822	0.138718	0.556338	0.964489	-0.005107	0.953830	-0.046755
12	(Medium GDP)	(Many Medals)	0.249342	0.423178	0.110623	0.443662	1.048405	0.005107	1.036819	0.061506
0	(High GDP)	(Few Medals)	0.250219	0.576822	0.099210	0.396491	0.687372	-0.045122	0.701196	-0.377567
11	(Low GDP)	(Many Medals)	0.250219	0.423178	0.096576	0.385965	0.912062	-0.009311	0.939395	-0.113941
9	(Many Medals)	(High GDP)	0.423178	0.250219	0.151010	0.356846	1.426134	0.045122	1.165788	0.518017
6	(Few Medals)	(Very Low GDP)	0.576822	0.250219	0.185250	0.321157	1.283500	0.040918	1.104497	0.521956
2	(Few Medals)	(Low GDP)	0.576822	0.250219	0.153644	0.266362	1.064514	0.009311	1.022004	0.143213

Top k Rules Explanation:

1. **Rule:** Countries with 'Very Low GDP' have a 74% chance of winning 'Few Medals'.

2. **Rule:** Countries with 'High GDP' are likely to win 'Many Medals' with a confidence of about 60%.

3. **Rule:** Countries with 'Medium GDP' have a 56% chance of winning 'Few Medals'.

Meaning of the Rules in Plain English:

Very Low GDP and Few Medals: Countries with significantly low economic output consistently show a trend of winning only a few medals. This suggests that the economic constraints are limiting their potential to excel in the Olympics, potentially due to insufficient investment in sports infrastructure and athlete development.

High GDP and Many Medals: Nations with robust economies are more likely to secure a larger number of medals. This high probability (60%) indicates that economic prosperity plays a crucial role in sports success, likely due to better resources, facilities, and access to professional training.

Medium GDP and Few Medals: Countries that fall in the middle economic bracket tend to win few medals, although the confidence (56%) is not as high as with 'Very Low GDP'. This implies a mixed outcome, where some countries might perform better than others, influenced by factors other than just economic output.

Insights Derived from the Mining Results:

Economic Barriers to Sports Excellence: The strong link between 'Very Low GDP' and winning few medals underlines the economic barriers that hinder sports excellence. This association points to a need for strategic investment in sports within these countries.

Economic Prosperity as a Catalyst for Sports Success: The correlation between 'High GDP' and winning 'Many Medals' showcases economic prosperity as a significant catalyst for Olympic success. Wealthier countries likely have the means to invest heavily in sports, from infrastructure to athlete welfare, enhancing their competitive edge.

Uncertain Impact of Medium Economic Status: The results for 'Medium GDP' countries suggest that while economic capability provides a platform for sports, other factors such as governance, policy, public interest, and cultural support for sports might play crucial roles in translating economic capability into tangible results.

Suggestions for Commerce Based on the Results:

1. **Encourage Global Sports Investment:** International sports bodies and wealthier nations could look to support Olympic programs in lower GDP countries through funding, infrastructure projects, and expertise exchange, leveling the playing field and promoting global sports participation.

2. **Policy Reforms in Medium GDP Countries:** Countries with medium GDP could benefit from reforms and targeted investments in sports to leverage their existing economic capabilities. Enhancing community sports programs and improving access to sports education and facilities could bridge the gap to higher Olympic success.

3. **Foster Corporate Partnerships and Sponsorships:** Corporate entities can be encouraged to sponsor athletes and sports programs in economically disadvantaged regions. These partnerships can provide necessary financial support and resources, offering mutual benefits: enhanced sports performance and corporate social responsibility fulfillment, alongside marketing visibility.

ASSOCIATION RULE MINING - Implement without using Python packages

Step 1: Preparing Data
Use aggregated_data, categorized GDP and medal counts, and handled missing data. Then, use this DataFrame to find association rules.

Step 2: Create Transaction-like Data
Transactions are typically expected to be in a list of lists, where each inner list is a transaction. Here, each country and year pair is a transaction, and items in transactions are the discretized GDP and medal categories.

Step 3: Compute Support Values
Support is computed as the fraction of transactions that contain a given itemset.

Step 4: Find Frequent Itemsets
Iteratively generate itemsets starting with 1-itemsets, 2-itemsets, etc., filtered by a minimum support threshold.

Step 5: Generate Association Rules
For those frequent itemsets, generate association rules and then calculate the confidence and lift for these rules.

Python codes showing implementation of Association Rule Mining from scratch:

```
import pandas as pd
import numpy as np
from itertools import combinations, chain

# Convert DataFrame into a list of lists, treating each row as a transaction
transactions = aggregated_data[['GDP_per_capita', 'medals_won']].apply(lambda row: [str(item) for item in row], axis=1).tolist()

# Compute Support for all items and item combinations
def get_support(itemset, transactions):
    return sum([set(itemset).issubset(transaction) for transaction in transactions]) / len(transactions)

# Find all Frequent Itemsets
def apriori(transactions, min_support):
    # Single item support
    items = set(chain.from_iterable(transactions))
    itemsets = [{item} for item in items]
    support_dict = {frozenset(itemset): get_support(itemset, transactions) for itemset in itemsets if get_support(itemset, transactions) >= min_support

    current_itemsets = [itemset for itemset, sup in support_dict.items() if sup >= min_support]
    k = 2
    while current_itemsets:
        # Generate new candidates from last frequent itemsets
        new_candidates = [frozenset(x.union(y)) for x in current_itemsets for y in current_itemsets if len(x.union(y)) == k]
        new_candidates = list(set(new_candidates))  # Remove duplicates
        # Filter new candidates by support
        valid_candidates = {frozenset(itemset): get_support(itemset, transactions) for itemset in new_candidates if get_support(itemset, transactions)
        support_dict.update(valid_candidates)
        current_itemsets = [itemset for itemset, sup in valid_candidates.items() if sup >= min_support]
        k += 1
    return support_dict
```

```
# Generate frequent itemsets
frequent_itemsets_support = apriori(transactions, min_support=0.01)

# Generate Association Rules from Frequent Itemsets
def generate_rules(frequent_itemsets_support, confidence_threshold):
    rules = []
    for itemset in frequent_itemsets_support.keys():
        if len(itemset) > 1:
            for consequence in combinations(itemset, 1):
                antecedent = itemset.difference(consequence)
                antecedent_support = frequent_itemsets_support[frozenset(antecedent)]
                consequence_support = frequent_itemsets_support[frozenset(consequence)]
                rule_support = frequent_itemsets_support[itemset]
                confidence = rule_support / antecedent_support
                lift = confidence / consequence_support
                if confidence >= confidence_threshold:
                    rules.append((antecedent, consequence, rule_support, confidence, lift))
    return rules

# Using a threshold for demonstration
rules = generate_rules(frequent_itemsets_support, confidence_threshold=0.1)

# Sort rules by confidence and lift
sorted_rules = sorted(rules, key=lambda x: (x[3], x[4]), reverse=True)
sorted_rules[:10]  # Display top 10 rules
```

Output:

```
[(frozenset({'Very Low GDP'}),
  ('Few Medals',),
  0.18525021949078138,
  0.7403508771929824,
  1.2835002269753533),
 (frozenset({'Low GDP'}),
  ('Few Medals',),
  0.15364354697102722,
  0.6140350877192983,
  1.0645144062591794),
 (frozenset({'High GDP'}),
  ('Many Medals',),
  0.15100965759438104,
  0.6035087719298246,
  1.4261337992283614),
 (frozenset({'Medium GDP'}),
  ('Few Medals',),
  0.13871817383669885,
  0.5563380281690141,
  0.9644886059124919),
 (frozenset({'Medium GDP'}),
  ('Many Medals',),
  0.1106233538191396,
  0.44366197183098594,
  1.0484045350943838),
 (frozenset({'High GDP'}),
  ('Few Medals',),
  0.09920983318700614,
  0.39649122807017545,
  0.6873721594702129),
 (frozenset({'Low GDP'}),
  ('Many Medals',),
  0.09657594381035997,
  0.3859649122807018,
  0.9120623134599987),
 (frozenset({'Many Medals'}),
  ('High GDP',),
  0.15100965759438104,
  0.35684647302904565,
  1.4261337992283616),
 (frozenset({'Few Medals'}),
  ('Very Low GDP',),
  0.18525021949078138,
  0.32115677321156777,
  1.2835002269753535),
 (frozenset({'Few Medals'}),
  ('Low GDP',),
  0.15364354697102722,
  0.26636225266362257,
  1.0645144062591794)]
```

Association Rules Explanation

Very Low GDP Leads to Few Medals
Support: 18.53% Confidence: 74.04% Lift: 1.284
Interpretation: There's a strong and positive relationship indicating that countries with a very low GDP are likely to win few medals.

Low GDP Leads to Few Medals
Support: 15.36% Confidence: 61.40% Lift: 1.065
Interpretation: Countries with low GDP also tend to win few medals, although the relationship is weaker than with very low GDP.

High GDP Leads to Many Medals
Support: 15.10% Confidence: 60.35% Lift: 1.426
Interpretation: A strong and positive association suggests that countries with high GDP tend to win many medals.

Medium GDP Leads to Few Medals
Support: 13.87% Confidence: 55.63% Lift: 0.964
Interpretation: This inverse relationship implies that countries with medium GDP do not necessarily translate their economic status into a high number of medals.

WHAT IF ANALYSIS

What-if analysis is a strategic process used to forecast outcomes based on changes in variables. Here, we examine how increases in government health expenditure might enhance France's Olympic performance.

The process begins with data extraction from a PostgreSQL database into a pandas DataFrame. We then manipulate this data, assuming a positive impact of higher health funding on athletes' performance, creating various scenarios with increased health expenditures.

For each scenario, we calculate the predicted total medals won, observing a direct relationship between health investment and Olympic success. Similarly, we assess economic scenarios, noting that an improved GDP per capita—indicative of economic health—may also correlate with better Olympic outcomes. This meticulous method, which we've demonstrated without the use of specialized Python packages, enables us to draw insightful conclusions that can inform future investment strategies in sports development.

What If Analysis

Sensitivity Analysis

Government Health Expenditure Impact: We assume that better-funded health systems contribute to better athlete performance, so we could analyze how changes in government health expenditure impact the mean number of medals won.

```
# Extract PostgreSQL tables into pandas DataFrames
fact_olympic = pd.read_sql("SELECT * FROM fact_olympic", engine)
```

```
# Assuming that higher government health expenditure positively affects athletes' performance
# Create a sensitivity range for health expenditure increases
health_exp_increases = [1.05, 1.10, 1.15]  # 5%, 10%, 15% increases

# Iterate over increases and estimate impact on medals won
for increase in health_exp_increases:
    # Replace 'government_health_expenditure_column' with the actual column name
    fact_olympic['adjusted_health_expenditure_per_capita'] = fact_olympic['government_health_expenditure_per_capita'] * increase

    # Replace 'medals_won_count_column' with the actual column name for medal count
    fact_olympic['predicted_medals_won'] = fact_olympic['medals_won'] * (1 + (increase - 1))

    # Summarize the impact
    total_predicted_medals = fact_olympic['predicted_medals_won'].sum()
    print(f"Total predicted medals with a {increase - 1:.0%} increase in health expenditure: {total_predicted_medals}")
```

```
Total predicted medals with a 5% increase in health expenditure: 21540.750000000007
Total predicted medals with a 10% increase in health expenditure: 22566.499999999996
Total predicted medals with a 15% increase in health expenditure: 23592.25
```

Scenario Analysis

Economic Impact Scenario: Analyze how fluctuations in GDP per capita, which reflects the country's economic situation, might affect Olympic performance.

```
# Define scenarios: Economic Boom and Economic Downturn
gdp_growth_rates = {'Economic Boom': 1.10, 'Economic Downturn': 0.90}

# Apply scenarios to GDP per capita and estimate impact on medals won
for scenario, growth_rate in gdp_growth_rates.items():
    fact_olympic['adjusted_gdp_per_capita'] = fact_olympic['GDP_per_capita'] * growth_rate
    # Assuming that a higher GDP per capita means better funding for athletes
    # A simple model might just scale medals won by the GDP growth rate
    fact_olympic['predicted_medals_won'] = fact_olympic['medals_won'] * growth_rate
    total_predicted_medals = fact_olympic['predicted_medals_won'].sum()
    print(f"Total predicted medals in {scenario}: {total_predicted_medals}")
```

```
Total predicted medals in Economic Boom: 22566.499999999996
Total predicted medals in Economic Downturn: 18463.500000000004
```

Interpretation of Results from the What-If Analysis:

1. **Health Expenditure Increase:**

- **With a 5% increase in health expenditure**, the total predicted medals won is 21,540.75. This substantial increase suggests that even a slight boost in health funding could significantly enhance Olympic performance, presumably due to improved healthcare and support systems for athletes.
- **With a 10% increase**, the total predicted medals won rises to 22,566.5, indicating a continued improvement in performance with higher investment in health. This trend reaffirms the potential benefits of investing in athlete health and wellness.
- **With a 15% increase**, the predicted total reaches 23,592.25. The consistent increase in predicted medals with higher health expenditure underscores a potential direct correlation between health investment and Olympic success.

2. **GDP Scenarios:**

- Under the **Economic Boom' scenario**, which assumes a 10% increase in GDP per capita, the total predicted medals won matches the outcome of a 10% increase in health expenditure, at 22,566.5. This alignment suggests that economic prosperity, reflected by GDP growth, significantly contributes to Olympic success, likely through enhanced funding and resources for sports programs.
- Conversely, in the **Economic Downturn' scenario**, assuming a 10% decrease in GDP per capita, the total predicted medals won drops to 18,463.5. This decline indicates that economic hardships can detrimentally impact Olympic performance, likely due to cutbacks in sports funding and suboptimal conditions for athlete training.

Interpreting the Findings for the Client:

- **Investment in Athlete Health**: Increased health expenditure is predicted to lead to better athlete performance. This could be due to enhanced healthcare facilities, improved injury management, and overall superior athlete welfare, which are crucial for top-tier performance.
- **Economic Impact:** The prosperity of a country, as measured by GDP per capita, appears closely linked to its Olympic success. Strategic investments in both the economy and sports infrastructure during times of economic growth could yield better results. Conversely, during economic downturns, maintaining investment in sports could help mitigate potential declines in performance.

Strategy Suggestions:

- **Invest in Health Programs and Facilities:** Encourage incremental increases in health expenditure for athletes, as this has been shown to potentially yield substantial improvements in Olympic results.
- **Establish Economic Safety Nets:** Develop financial safety nets or reserve funds to support athletes and sports programs during economic downturns, helping to stabilize performance levels despite fiscal challenges.

- **Foster Public and Private Partnerships:** Promote partnerships between government bodies and private sectors to diversify funding sources. This can reduce dependency on fluctuating government budgets and provide a more stable financial environment for athletes.

Each interpretation assumes a direct causal relationship between the adjusted variables (health expenditure and GDP per capita) and Olympic success. It's critical to clarify to the client that while the analysis suggests correlations, causation cannot be definitively established without further detailed study. Additional research and possibly controlled experiments would be necessary to verify these causal links.

AI ASSISTANCE REFERENCE

In this project, ChatGPT was instrumental in generating ideas, assisting with debugging, clarifying complex processes, and refining the writing to ensure clarity and professionalism in the final report.