

🏅 Paris 2024 Olympic Games Dashboard

An interactive Streamlit dashboard for exploring Paris 2024 Olympic data. Built as a coursework project for the Software Engineering for Data Science module.

📋 What This Project Does

We built a multi-page dashboard that lets you explore the Paris 2024 Olympics from different angles. You can see which countries dominated, explore athlete demographics, check out the event schedule, and see where everything happened on interactive maps.

The dashboard has 6 pages:

- **Overview** - The landing page with key stats and medal standings
- **Global Analysis** - Maps and charts breaking down medals by continent and country
- **Head-to-Head** - Compare any two countries side-by-side
- **Athlete Performance** - Search individual athletes and explore demographics
- **Sports & Events** - Event schedules and venue information
- **Daily Highlights** - Day-by-day breakdown of what happened during the Games

🚀 Getting Started

What You Need

- Python 3.8+
- The packages in `requirements.txt` (streamlit, plotly, pandas)

Setup

```
# Install the dependencies
pip install -r requirements.txt

# Make sure the dataset folder exists at:
# paris-2024-olympic-summer-games/versions/27/
# (Download from Kaggle if you don't have it)

# Run it!
streamlit run 1_🏠_Overview.py
```

Opens at <http://localhost:8501>

📁 How It's Organized

```
├── 1_🏠_Overview.py          # Entry point - main dashboard page
├── pages/
│   ├── 2_🗺_Global_Analysis.py # World maps, sunburst charts
```

```

    ├── 3_vs_Head_to_Head.py      # Country comparison tool
    ├── 3_athlete_Performance.py # Athlete search & demographics
    ├── 4_sports_and_Events.py   # Schedule timeline, venue maps
    └── 4_dailystats.py         # Day-by-day medal breakdown

    └── utils/
        ├── data_loader.py       # Loads CSVs with caching
        ├── filters.py           # Sidebar filter widgets
        ├── ioc_iso_mapping.py   # Country code conversion
        └── venue_coordinates.py # Lat/lon for venue markers

    └── paris-2024-olympic-summer-games/ # The Kaggle dataset
    └── requirements.txt

```

The emoji prefixes in filenames aren't just for fun - Streamlit uses them to order pages in the sidebar and display nice icons.

⌚ Why We Built It This Way

The Filtering Problem

One thing we realized early on: Olympic data has a lot of dimensions. Country, continent, sport, medal type, date... users want to slice it different ways. Instead of building separate filter controls on each page, we put everything in a shared sidebar that persists across pages.

The tricky part was continents. The dataset only has NOC codes (like "USA", "GER", "FRA"), not continent info. So we built a manual mapping in [data_loader.py](#) - about 200 countries mapped to their continents. It's not elegant, but it works, and now you can filter Europe vs Asia vs whatever.

Choosing Chart Types

We tried to match chart types to what makes sense for the data:

- **Choropleth map** for the world view - this is the obvious choice when you want to show "value by country" on a geographic layout. Plotly handles the projections and country shapes automatically, you just need ISO-3 country codes (another reason we have that mapping file).
- **Sunburst and Treemap** for the hierarchy view (Continent → Country → Sport). These show part-to-whole relationships nicely. We put both side-by-side because some people prefer circular layouts and others prefer rectangular. Sunburst is better for seeing the hierarchy, treemap is better for comparing sizes.
- **Box and Violin plots** for age distributions - box plots show the quartiles clearly, violin plots show the actual distribution shape. Athletes in shooting sports tend to be older, swimmers tend to be young. These charts make that pattern visible.
- **Gantt/Timeline chart** for the schedule - when you're showing things with start and end times, a timeline is the natural representation. We limited it to 50 events per view because otherwise it gets unreadable.
- **Scatter mapbox** for venue locations - we manually looked up lat/lon coordinates for all the Paris venues (and the Tahiti surfing location!) because the dataset didn't include geographic data.

Performance Decisions

Streamlit reruns the entire script on every interaction. That would be painfully slow if we re-read all the CSVs each time. So every data loading function uses `@st.cache_data` - the first load takes a second or two, then it's instant.

We also made a decision to load all data upfront via `load_all_data()` rather than lazy-loading per page. The dataset is small enough (~10 CSV files, largest is maybe 15MB) that it doesn't matter, and it simplifies the code.

Things We'd Do Differently

If we had more time:

- The IOC-to-ISO code mapping is incomplete. Some countries show up as blanks on the choropleth because we missed them.
- The venue coordinate lookup is brittle - if the schedule data uses a slightly different venue name (like "South Paris Arena 1" vs "South Paris Arena"), the map won't find it. We added some aliases but probably missed some.
- We hardcoded "2024" for age calculations. If someone runs this in 2025, the ages will be off by a year.

About the Data

We're using the [Paris 2024 Olympic Summer Games dataset from Kaggle](#) by piterfm. It includes:

File	What's In It
<code>athletes.csv</code>	~11,000 athletes with name, country, birth date, height/weight, sports
<code>medals.csv</code>	Every medal awarded - who won, what event, which country
<code>medals_total.csv</code>	Aggregated counts: Gold/Silver/Bronze per country
<code>nocs.csv</code>	NOC codes to country names
<code>schedules.csv</code>	Event schedule with dates, times, venues
<code>medallists.csv</code>	Detailed medalist info
<code>venues.csv</code>	Venue names (no coordinates though, we added those ourselves)
<code>events.csv</code>	List of events by sport
<code>teams.csv</code>	Team compositions
<code>coaches.csv</code>	Coach information

The dataset is version 27 - there were updates throughout the Games as results came in.

Competition Requirements

This was built for the SEDS Streamlit Challenge. Here's how we hit the requirements:

Requirement	How We Did It
-------------	---------------

Requirement	How We Did It
Multi-page structure	6 pages via Streamlit's pages/ folder convention
Global filters	Country, Continent, Sport, Medal Type - all in sidebar, all pages
Required charts	Choropleth ✓, Bar ✓, Pie ✓, Sunburst ✓, Treemap ✓, Timeline ✓, Scatter map ✓
Cohesive layout	Consistent sidebar, consistent header style, columns for side-by-side charts
Interactivity	Hover tooltips, filter updates, drill-down in hierarchical charts

The Team

- **Khedim Youcef**
- **Bensetallah Soufiane**
- **Zitouni Ahmed**
- **Houari Mohamed**

Group 1 | Specialty: AIDS (Artificial Intelligence & Data Science)

Course: Software Engineering for Data Science

Instructor: Dr. Belkacem KHALDI

If Something Breaks

"No module named streamlit"

→ Run `pip install -r requirements.txt`

"FileNotFoundException" for CSV files

→ Check that [paris-2024-olympic-summer-games/versions/27/](#) exists and has the CSVs

Map shows blank/missing countries

→ That country might be missing from our IOC-to-ISO mapping. Check [utils/ioc_iso_mapping.py](#)

Venue markers not showing

→ The venue name might not match our coordinates lookup. Check [utils/venue_coordinates.py](#)

Deprecation warnings about use_container_width

→ Streamlit is updating their API. The dashboard still works, just with warnings in the terminal.

License

Educational project for university coursework. Dataset is from Kaggle (check their terms for usage).

Built with Streamlit, Plotly, and Pandas. Data from [piterfm's Kaggle dataset](#).