Data Visualisation Project 1 Report

Samantha Emma Morris, Le Gia Duc and Vo Khoi Thanh Lam

April 10, 2025

Introduction

We chose Olympic and global GDP datasets to analyse the history of people and countries who have achieved highly in worldwide sports and compare this with GDP. We aimed to uncover trends between the economies of different countries and their impact on success in sports. The first dataset our project uses is olympics.csv, downloaded from TidyTuesdays but was scraped and wrangled from www.sports-reference.com in May 2018 (Harmon, 2024). The dataset contains 15 columns and 271,116 rows detailing athlete data and medal results from summer and winter Olympic events from Athens 1896 to Rio 2016. The column names include athlete ID number, sex (M/F), age, height (cm), weight (kg), team, National Olympic Committee (three-letter code for the country), games (Olympic game athlete competed in), year (of Olympic game), season (summer/winter), city (Olympic host city), sport, event (specific sports event), and medals (gold, silver or bronze if achieved). Overall, the dataset was detailed and comprehensive, but it required cleaning to remove duplicate rows and non-medallists, conversion of team names (e.g. United States, United States-1, and United States-3 all converted to United States), and host cities mapped to host countries.

The second dataset used is Countries GDP 1960-2020.csv, which is obtained from the World Bank national accounts and OECD National Accounts data files. It contains 120 rows and 63 columns, detailing country/region names, country codes (3 letter code), and Gross Domestic Product (GDP) data for every year from 1960 until 2016 (60-year period). Gross Domestic Product (GDP) represents the total value added by all resident producers in an economy, measured at purchaser's prices. It is calculated as the sum of gross value added by all producers, including product taxes while subtracting any products subsidies.

Question 1: What are the medal-winning trends of selected countries across different Olympic Games?

Introduction

The Olympic Games globally allows athletes and countries to demonstrate and celebrate their athletic success and engage an excited global audience. Understanding the trends in medal-winning achievements offers valuable insights into a country's focus on sports, investment in athlete training, and potential physiological or demographic advantages. With Question 1, we aim to explore how selected countries have performed across different Olympic Games by analysing their medal counts over the years. This analysis will not distinguish between Summer and Winter Olympics but will instead focus on overall performance trends.

To answer this question, focused on the following columns from the olympics.csv dataset: ['name', 'sex', 'height', 'weight', 'team', 'medal', 'year', 'city', 'games', 'country']. These attributes help track the number and type of medals won by athletes from selected countries over time, identify the top-performing athletes from each nation, and determine how consistent a country's performance has been. This exploration identifies high and low-performing nations and investigates the fluctuations in performance across different Olympic events. These insights could be valuable for sports analysts aiming to assess national athletic development, historians studying geopolitical influences on sports, and policymakers seeking to improve or benchmark national performance in international competitions.

Approach

This question was answered by being broken down into three graphs. The first is a stacked bar chart where you can select the countries where you want to see the total number of Olympic medals they have won and order the chart in either ascending or descending order. Hovering over parts of the stacked barplot breaks the data down further e.g. hovering over the yellow section will show the country's sum of gold medals. Additionally, the stacked bar plot can be ordered in either ascending or descending order depending on the user's viewing preference.

The user will then click on the medals from one country they want to investigate further. A stacked barplot was chosen for this Olympics data visualisation because it effectively displays the total number of medals won by each country and the breakdown by medal type (gold, silver, and bronze) in a single intuitive view. A line plot would not be effective here because it is more useful for showing trends over time rather than categorical comparisons. Similarly, a boxplot highlights data spread like medians and outliers, which is not the focus of this visualisation. The goal here is to compare quantities and compositions of medals across countries, which the stacked barplot does clearly and efficiently.

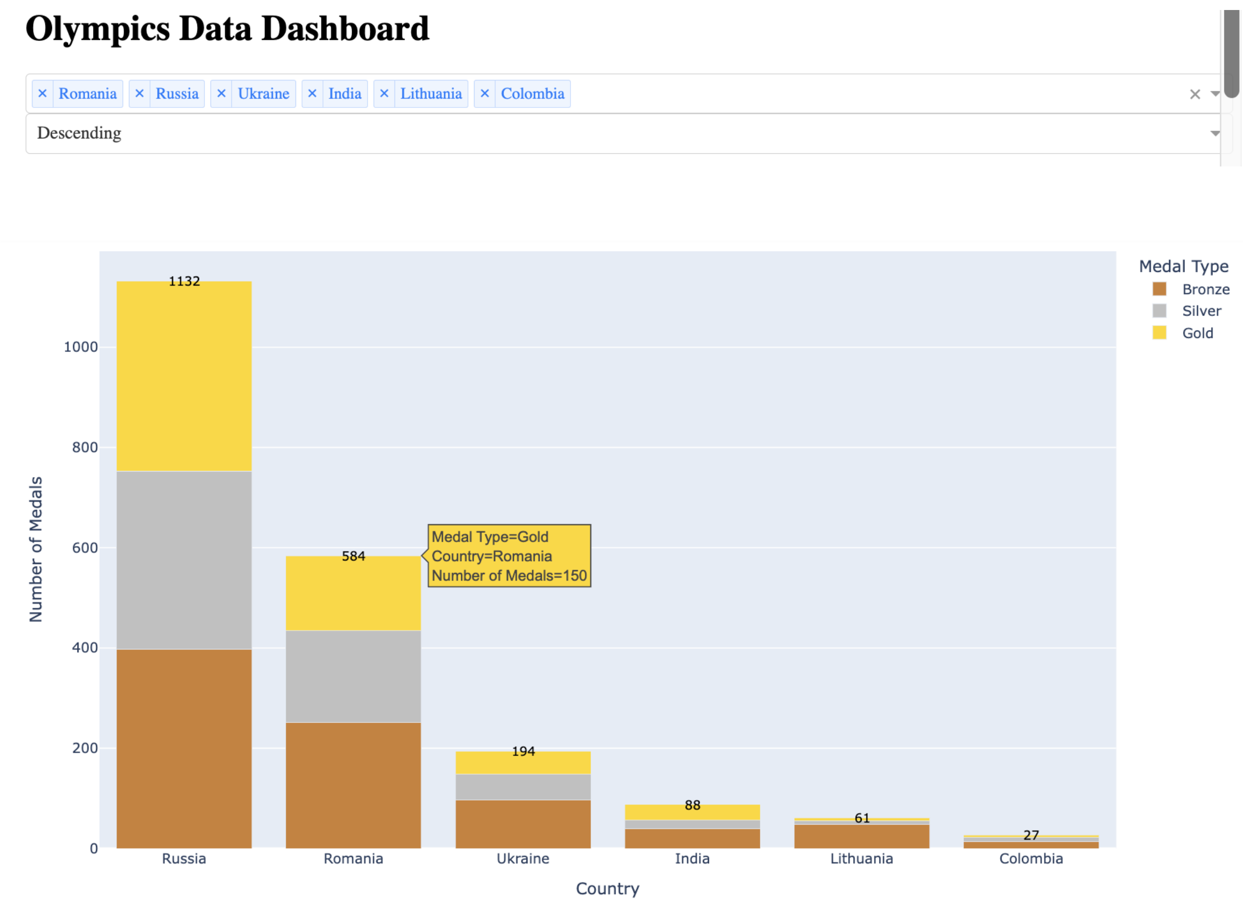


Figure 1: Medals won by each country

The second graph is a line plot that uses this input to show the total medals won by that one country over the Summer and Winter Olympics from 1960-2016. By hovering over one of the dots, the user can see further details including the Olympic game edition, the number of medals won during that game, the host country, and city. The country Romania was selected for the following example of graph 2. The user will then click on one Olympic game for that country (Romania in this example) where they would like to investigate the medal winners further. A line plot was the type of graph chosen because it effectively visualises trends and patterns in continuous data. It helps us see the rises and falls of Romania's performance in the Olympics, helping identify long-term performance trends including peaks and declines.

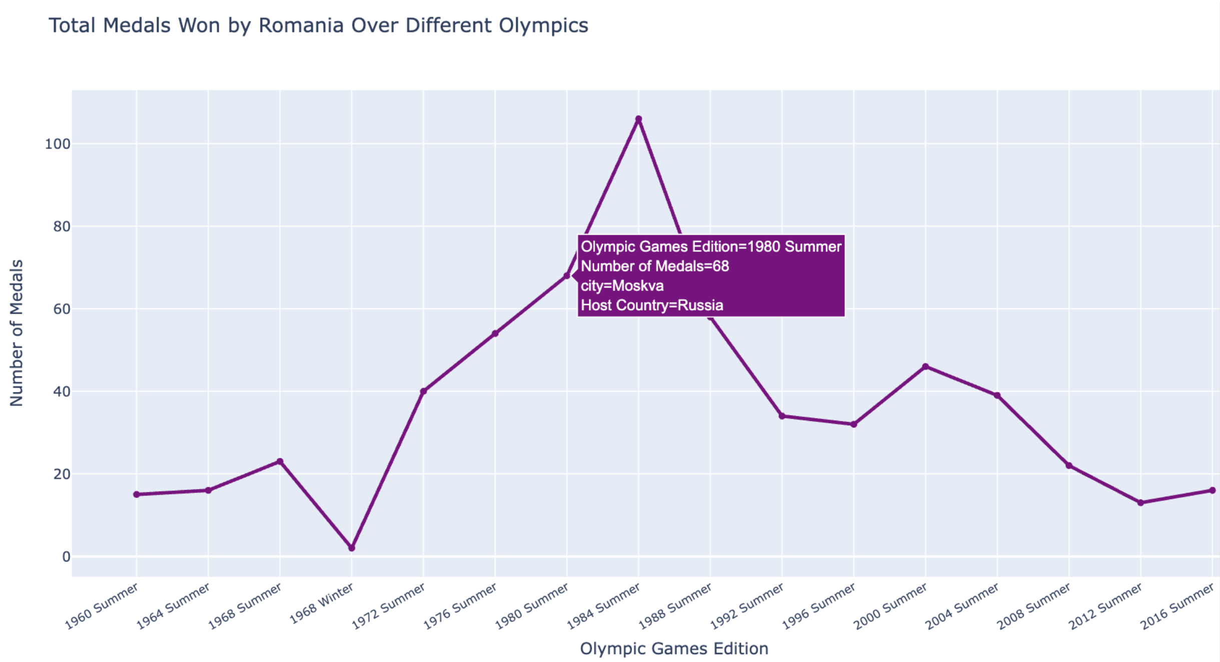


Figure 2: Medals won by selected country

By using the same country from graphs 1 and 2, and selecting a specific Olympic Games edition in graph 2, a third graph is generated in the application. This third graph highlights the top individual medal winners (up to 10 athletes) from that country during the selected Olympics. The example below shows the top 10 medal winners from Romania in the 1980 Summer Olympics. A stacked barplot is used to represent the number and types of medals (gold, silver, and bronze) each athlete won. The bars are organised by medal type and quantity, making it easy to compare the performance results of different athletes. This plot provides a more detailed and athlete-focused perspective on a country's success, extending upon the medal trends shown in the earlier graphs. A stacked barplot was chosen because one chart visualises medals by type (gold, silver, bronze) for each athlete, making it easy to compare how many medals each athlete won and what kinds of medals they earned. Other charts, like a scatterplot or boxplot, would not suitably show this information because they will not effectively show totals and comparisons, they are better for quantitative analysis.

Top 10 Athletes and Their Medals

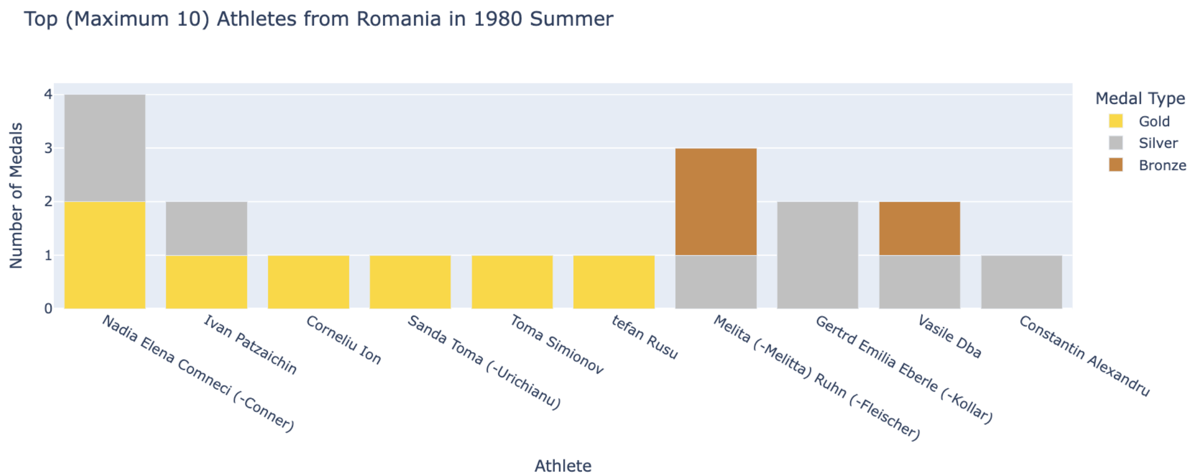


Figure 3: Athletes and their medals from the selected year

Lastly, a table was made listing all the athletes that won medals from the specific country and Olympic games to show the user more information. This table displays each athlete's name, the number of each medal type won, and the total medal count for that Olympic game. It allows users to explore the data further which is not displayed in the barplot and the tabular format offers a precise, text-based summary, instead of a visual summary.

All Athletes and Medals for Selected Country and Olympics

Athlete	Gold	Silver	Bronze	Total Medals
Nadia Elena Comneci (-Conner)	2	2	0	4
Melita (-Melitta) Ruhn (-Fleischer)	0	1	2	3
Ivan Patzaichin	1	1	0	2
Gertrd Emilia Eberle (-Kollar)	0	2	0	2
Vasile Dba	0	1	1	2
Corneliu Ion	1	0	0	1
Sanda Toma (-Urichianu)	1	0	0	1
Toma Simionov	1	0	0	1
tefan Rusu	1	0	0	1
Constantin Alexandru	0	1	0	1
Cristina Elena Grigora	0	1	0	1
Dumitria Turner	0	1	0	1
Ion Geant	0	1	0	1
Mihai Zafiu	0	1	0	1
Nicuor Eanu	0	1	0	1
Petre Capusta	0	1	0	1
Rodica Dunca (-Kszegi)	0	1	0	1
Adrian Cosma	[illegible]	[illegible]	[illegible]	[illegible]
Alexandru Flker	0	0	1	1

⚠ Errors ✂ Callbacks v3.0.0 Dash update available - v3.0.2 Server

Figure 4: View of the Dataset

Analysis

Below is the code we used to generate the plots:

```
# Code Block 1
#Preprocessing columns
df_olympics_cleaning = olympics_df_uncleaned.dropna(subset=['medal'])

# Define a mapping of Olympic host cities to their respective countries
city_to_country = {
    "Albertville": "France",
    "Amsterdam": "Netherlands",
    "Antwerp": "Belgium",
    "Antwerpen": "Belgium",  # Alternative spelling
    "Athens": "Greece",
    "Athina": "Greece",  # Alternative spelling
    "Atlanta": "USA",
    "Barcelona": "Spain",
    "Beijing": "China",
    "Berlin": "Germany",
    "Calgary": "Canada",
    "Chamonix": "France",
    "Cortina d'Ampezzo": "Italy",
    "Garmisch-Partenkirchen": "Germany",
    "Grenoble": "France",
    "Helsinki": "Finland",
    "Innsbruck": "Austria",
    "Lake Placid": "USA",
    "Lillehammer": "Norway",
    "London": "UK",
    "Los Angeles": "USA",
    "Melbourne": "Australia",
    "Mexico City": "Mexico",
    "Montreal": "Canada",
    "Moscow": "Russia",
    "Moskva": "Russia",  # Alternative spelling
    "Munich": "Germany",
    "Nagano": "Japan",
    "Oslo": "Norway",
    "Paris": "France",
    "Rio de Janeiro": "Brazil",
    "Rome": "Italy",
    "Roma": "Italy",  # Alternative spelling
    "Salt Lake City": "USA",
    "Sankt Moritz": "Switzerland",  # Alternative spelling of St. Moritz
    "Sapporo": "Japan",
    "Sarajevo": "Bosnia and Herzegovina",
    "Seoul": "South Korea",
```

```
    "Sochi": "Russia",
    "Squaw Valley": "USA",
    "St. Louis": "USA",
    "Stockholm": "Sweden",
    "Sydney": "Australia",
    "Tokyo": "Japan",
    "Torino": "Italy",
    "Vancouver": "Canada"
}


# Map the city column to the corresponding country
df_olympics_cleaning["host country"] = df_olympics_cleaning["city"].map(city_to_country)

# Dropping duplicates
df_olympics_cleaning = df_olympics_cleaning.drop_duplicates()

#select relevant years and columns for analysis
olympics_df_ = olympics_df[(olympics_df["year"] >= 1960) & (olympics_df["year"] <= 2020)]
olympics_df_q1 = olympics_df_[['name', 'sex', 'height', 'weight', 'team', 'medal', 'year', 'ci

#print the columns and a sample of the data
print(olympics_df_q1.columns)
olympics_df_q1

#clean the team names (e.g. United States, United States-1 and United States-2) and remove data
import re

def clean_team_name(team):
    # Remove numeric suffixes like '-1', '-2', etc.
    team = re.sub(r"-\d+$", "", team)
    return team

# Apply the cleaning function
olympics_df_q1["team"] = olympics_df_q1["team"].astype(str).apply
(clean_team_name)

# Remove non-country entries (manually identified)
invalid_entries = [
    "Elvis Va", "Bonaparte", "Nadine", "Sunrise", "Salinero", "Satchmo",
    "Digby", "Clearwater", "Don Schufro", "Mutafo", "Bingo", "Widgeon",
    "White Lady", "Gem", "Aphrodite", "Rush VII", "Lady C", "Barrenjoey",
    "Pandora", "Humbug V", "Glider", "Sirene", "Web II", "Tango", "Venilia",
    "Ballerina IV", "Nirefs", "Skum", "Shrew II", "Minotaur", "Macky VI",
    "Tornado", "Ma'Lindo"
]
```

```
# Keep only valid country names
olympics_df_q1 = olympics_df_q1[~olympics_df_q1["team"].
isin(invalid_entries)]


# Generate the graph
import dash
from dash import dcc, html
from dash.dependencies import Input, Output
from dash import dash_table
import plotly.express as px
import pandas as pd

#data selection
olympics_df_q1 = pd.read_csv("olympics_data.csv")
olympics_df_q1["medal"] = olympics_df_q1["medal"].astype(str)
olympics_df_q1 = olympics_df_q1.dropna(subset=["medal", "team", "year"])

# initialize app + create layout
app = dash.Dash(__name__)
app.layout = html.Div([
    html.H1("Olympics Data Dashboard"),

    dcc.Dropdown(
        id='country-dropdown',
        options=[{'label': country, 'value': country} for country in olympics_df_q1["team"].uni
        multi=True,
        placeholder="Select countries..."
    ),

    dcc.Dropdown(
        id='sort-order-dropdown',
        options=[
            {'label': 'Ascending', 'value': 'asc'},
            {'label': 'Descending', 'value': 'desc'}
        ],
        value='desc',
        clearable=False
    ),

    # Graph1: amount of medals per selected country
    dcc.Graph(id='medal-bar-chart'),

    # Graph2: medals for selected country per game
    html.H2("Medals Over Time"),
    dcc.Graph(id='medal-line-chart'),
```

```
    # Graph3: top 10 Athletes in selected country in selected game
    html.H3("Top 10 Athletes and Their Medals"),
    dcc.Graph(id='athlete-bar-chart'),

    #Table1: for selected country and Olympic, show each athlete that achieved medals and what
    html.H3("All Athletes and Medals for Selected Country and Olympics"),
    dash_table.DataTable(
        id='athlete-table',
        columns=[
            {'name': 'Athlete', 'id': 'name'},
            {'name': 'Gold', 'id': 'Gold'},
            {'name': 'Silver', 'id': 'Silver'},
            {'name': 'Bronze', 'id': 'Bronze'},
            {'name': 'Total Medals', 'id': 'total'}
        ],
        style_table={'overflowX': 'auto'},
        style_cell={'textAlign': 'left', 'padding': '5px'},
        style_header={'backgroundColor': '#f9f9f9', 'fontWeight': 'bold'},
        style_data_conditional=[
            {
                'if': {'column_id': 'Gold'},
                'backgroundColor': '#FFF8DC'
            },
            {
                'if': {'column_id': 'Silver'},
                'backgroundColor': '#F5F5F5'
            },
            {
                'if': {'column_id': 'Bronze'},
                'backgroundColor': '#FAEBD7'
            },
        ]
    )
])

# Callback 1:  bar chart for selected countries
@app.callback(
    Output('medal-bar-chart', 'figure'),
    [Input('country-dropdown', 'value'),
     Input('sort-order-dropdown', 'value')]
)
def update_country_graph(selected_countries, sort_order):
    if not selected_countries:
        return px.bar(title="Select at least one country to display medal counts")

    filtered_df = olympics_df_q1[olympics_df_q1["team"].isin(selected_countries)]
```

```
    medal_counts = (filtered_df.groupby(["team", "medal"]).size()
                    .reset_index(name="count"))
    #count total medals
    total_medals = medal_counts.groupby("team")["count"].sum().reset_index()
    total_medals = total_medals.sort_values(by="count", ascending=(sort_order == "asc"))

    #use total medal count to count gold, silver, bronze
    medal_counts = medal_counts.merge(total_medals[["team", "count"]], on="team", suffixes=(""
    medal_counts = medal_counts.sort_values(by="count_total", ascending=(sort_order == "asc"))

    #make sure gold at top, silver in middle, bronze at bottom
    medal_order = ["Bronze", "Silver", "Gold"]
    medal_counts["medal"] = pd.Categorical(medal_counts["medal"], categories=medal_order, orde
    medal_counts = medal_counts.sort_values(by=["count_total", "medal"], ascending=[(sort_orde

    #plot
    fig = px.bar(
        medal_counts, x="team", y="count", color="medal", barmode="stack",
        title="Medal Counts by Country",
        labels={"count": "Number of Medals", "team": "Country", "medal": "Medal Type"},
        color_discrete_map={"Gold": "#FFD700", "Silver": "#C0C0C0", "Bronze": "#CD7F32"}
    )

    for team in total_medals["team"]:
        total = total_medals.loc[total_medals["team"] == team, "count"].values[0]
        fig.add_annotation(
            x=team, y=total, text=str(total), showarrow=False, font=dict(size=11, color="black
        )

    #clickmode and graph size
    fig.update_layout(height=700, clickmode="event+select")
    return fig

# Callback 2: Update line chart when clicking on a country's bar
@app.callback(
    Output('medal-line-chart', 'figure'),
    [Input('medal-bar-chart', 'clickData')]
)
def update_medal_trend(clickData):
    if clickData is None:
        return px.line(title="Click on a country's bar to see medal trends")

    #based on click data from graph 1
    selected_country = clickData["points"][0]["x"]
    #find country medals for selected country from clickdata graph1
    country_medals = olympics_df_q1[olympics_df_q1["team"] == selected_country]
    medals_per_games = (country_medals.groupby(["games", "city", "country"]).size()
```

```
                    .reset_index(name="count")
                    .sort_values("games"))

    #make linegraph
    fig = px.line(
        medals_per_games, x="games", y="count", markers=True,
        title=f"Total Medals Won by {selected_country} Over Different Olympics",
        labels={"count": "Number of Medals", "games": "Olympic Games Edition", "country": "Hos
        hover_data={"city": True, "country": True}
    )

    #format graph
    fig.update_traces(line=dict(color="#800080", width=3))
    fig.update_layout(
        height=600, width=1200, clickmode="event+select",
        xaxis=dict(tickangle=-30, tickmode="array", tickfont=dict(size=10), title_standoff=10)
    )
    return fig

# Callback 3: update top 10 athletes bar chart and table
@app.callback(
    [Output('athlete-bar-chart', 'figure'),
     Output('athlete-table', 'data')],
    [Input('medal-bar-chart', 'clickData'),
     Input('medal-line-chart', 'clickData')]
)

#use click data from previous country selection and specific olympic game selection
def update_athlete_graph(clickData_country, clickData_game):
    if not clickData_country or not clickData_game:
        return px.bar(title="Click on a country's bar to see top athletes for a selected Olympi

    selected_country = clickData_country["points"][0]["x"]
    selected_game = clickData_game["points"][0]["x"]

    filtered_athletes = olympics_df_q1[
        (olympics_df_q1["team"] == selected_country)
        &
        (olympics_df_q1["games"] == selected_game)
    ]
    #find athletes that achieved medals and count
    them
    filtered_athletes = filtered_athletes[filtered_athletes["medal"].
    isin(["Gold", "Silver", "Bronze"])]
    all_athlete_counts = filtered_athletes.groupby(["name", "medal"]).size().reset_index(name="

    #create table with those results
```

```
    pivot_table = all_athlete_counts.pivot(index="name", columns="medal", values="count").filln
    pivot_table = pivot_table.reindex(columns=["Gold", "Silver", "Bronze"], fill_value=0)
    pivot_table["total"] = pivot_table.sum(axis=1)
    pivot_table = pivot_table.sort_values(by=["total", "Gold", "Silver", "Bronze"],
    ascending=[False, False, False, False]).reset_index()

    table_data = pivot_table.to_dict("records")

    top_10_names = pivot_table.head(10)["name"]
    top_athletes = all_athlete_counts[all_athlete_counts["name"].
    isin(top_10_names)]

    top_athletes_total = top_athletes.groupby("name")["count"].sum().reset_index(name="total_co
    top_athletes = top_athletes.merge(top_athletes_total, on="name")

    top_athletes["medal"] = pd.Categorical(top_athletes["medal"], categories=["Gold", "Silver"
    top_athletes = top_athletes.sort_values(by=["total_count", "medal"], ascending=[False, True

    fig = px.bar(
        top_athletes,
        x="name",
        y="count",
        color="medal",
        barmode="stack",
        title=f"Top 10 Athletes from {selected_country} in {selected_game} (Total Medals: {len
        labels={"count": "Number of Medals", "name": "Athlete", "medal": "Medal Type"},
        color_discrete_map={"Gold": "#FFD700", "Silver": "#C0C0C0", "Bronze": "#CD7F32"}
    )

    fig.update_layout(
        xaxis_title="Athlete",
        yaxis_title="Number of Medals",
        xaxis_tickangle=30,
        barmode="stack",
        coloraxis_showscale=False
    )

    return fig, table_data

# Run the app
if __name__ == '__main__':
    app.run(debug=True, port=7001)
```

As the plots are designed to be interactive, the full plots can not be shown in this report, and should be viewed as we run the code. A preview has been included in the previous section.

Discussion

First plot

Since the olympic.csv dataset includes all countries that participated in the Summer and Winter Olympics from 1960 to 2020, it contains a large volume of data. Therefore, this discussion will focus on analysing the performance trends of a range of countries: the top performer United States; high-to-mid achievers Germany, Russia, and China; and lower-performing nations such as India and Vietnam.

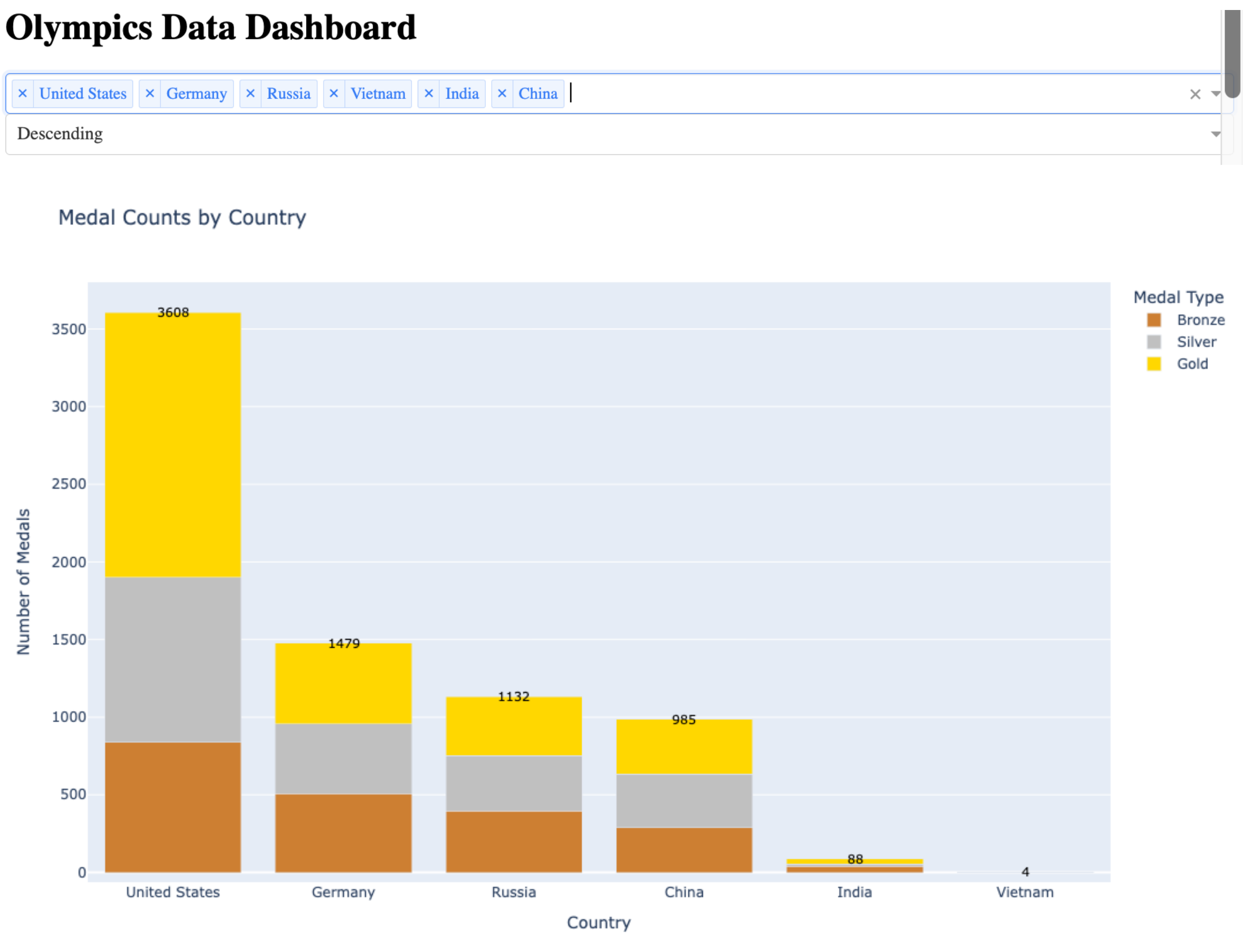
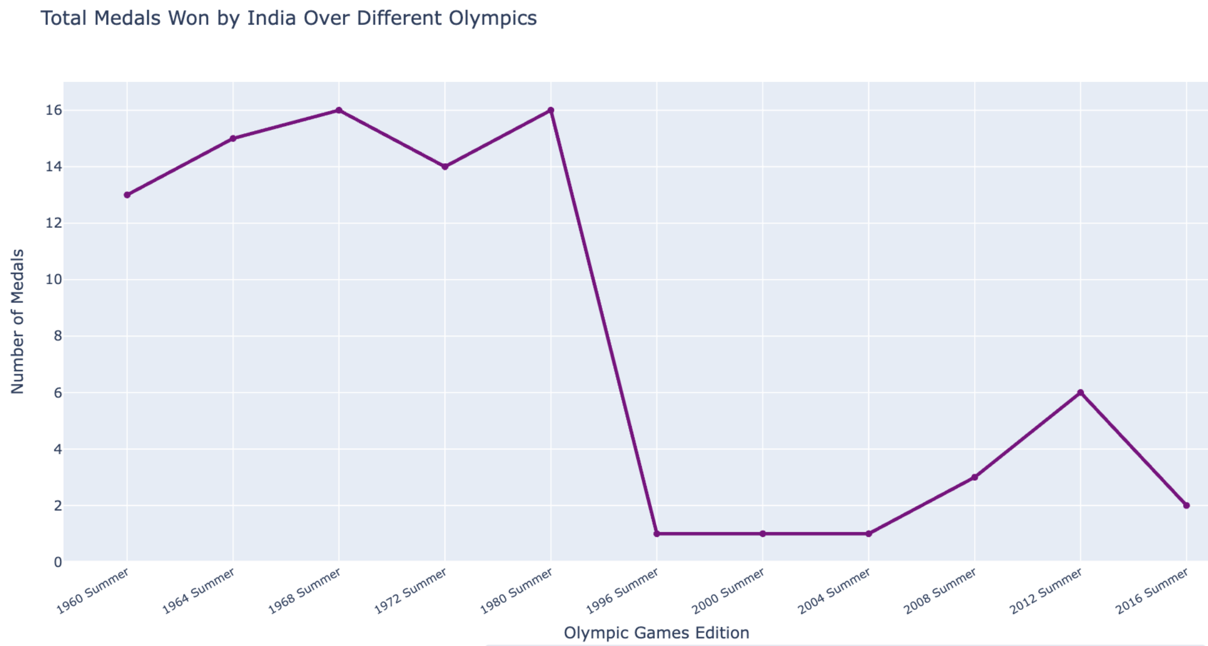


Figure 5: Medal Counts by Country for Analysis

Our graph comparing the medal counts of the different countries shows the Unites States stands out as the dominant high performer with 3,608 medals. The medal type distribution is balanced, with the highest amount being hold medals, which highlights consistent excellence across different olympics and time periods. Germany (1,479 medals), Russia (1,132 medals), and China (985 medals) form the group of high-to-mid achievers. Germany leads this group, with a strong and even spread across all medal types, while Russia and China follow with slightly lower totals. Further analysis of how these countries perform in the Winter versus Summer Olympics over time could provide deeper insight, particularly since countries like

Russia may have a natural advantage in Winter sports due to their cold climate.In contrast, India (88 medals) and Vietnam (4 medals) were selected as examples of lower-performing countries. Despite their large populations, both nations show relatively low medal counts. This may be due to limited sports infrastructure, insufficient funding, and a weaker national sports culture, which can hinder athlete development and international competitiveness, but further investigation is required to confirm this impact.



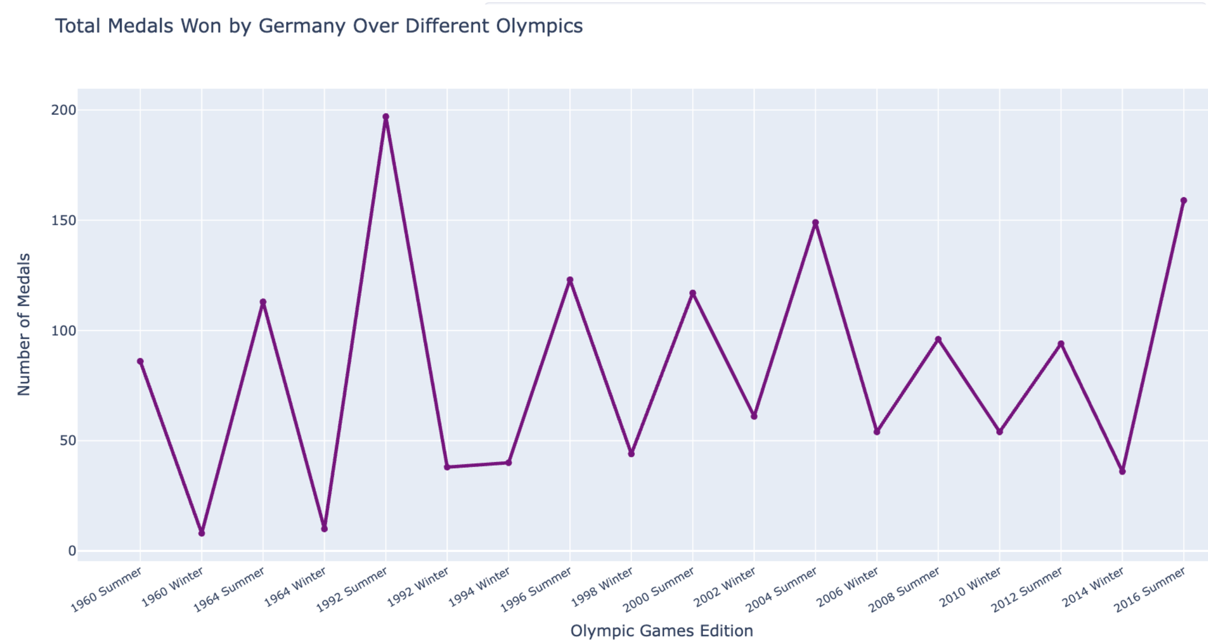


Figure 6: Lineplots of Olympic Medal Trends for Germany and India

The second graph analyses the total medals won over the different Olympics, and for this example we decided to compare Germany and India. The Summer Olympics usually awards approximately 1080 medals per year, whereas the Winter Olympics awards approximately 327 medals per year, causing the trend to go up and down significantly for every country due to there being less likelihood of winning medals during the Winter Olympics. The Summer Olympics hosts more athletic events than Winter Olympics, awarding approximately 1080 medals per Olympic Games compared to 327 for the Winter Olympics. These results cause the line graphs for every country to fluctuate up and down significantly, instead of showing a smooth trend. In the examples below, Germany peaked in medal winnings during the 1992 Summer Olympics, 2004 Summer Olympics and 2016 Summer Olympics. After Germany's reunification in 1990, West and East Germany competed as a unified team for the first time in the 1992 Summer Olympics. In the post-Cold War era, sport helped foster national identity and unity within Germany, influencing their increased performance success. While India has participated in the Winter Olympics, they have never won any medals in those events, which is why no Winter Olympic data appears on the x-axis of the graph. Historically, India won between 13 to 16 medals per Games during the 1960–1980 period, largely driven by dominance in field hockey. However, post-1980, medal counts declined sharply to just 1–6 per Olympic Games. This drop can be attributed to several factors, including a cultural shift in national sports preference toward cricket, along with reduced investment in sports infrastructure and elite athlete development. These examples underscore how political, economic, and cultural factors deeply influence Olympic success over time and should be investigated further.

Top 10 Athletes and Their Medals

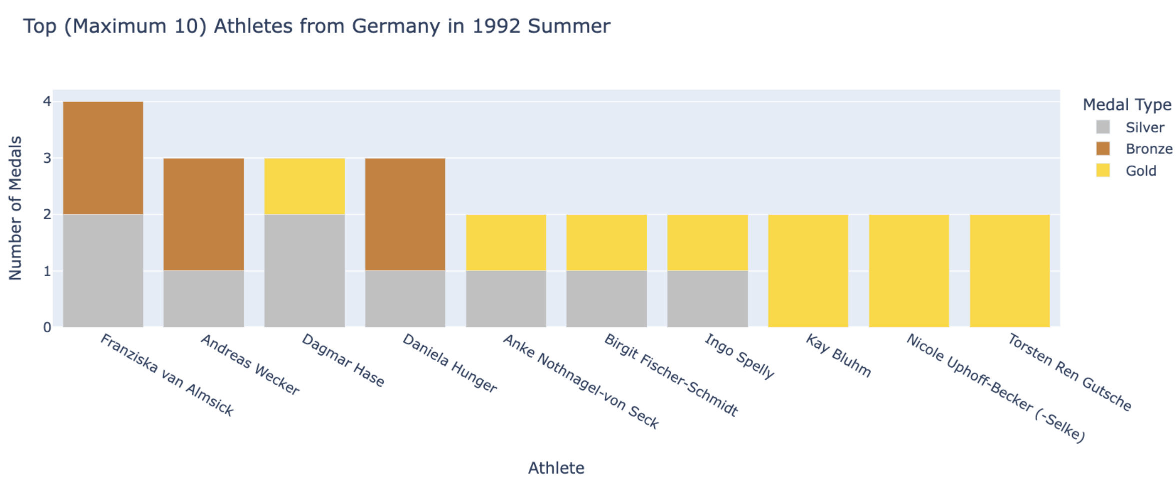


Figure 7: Stacked barplot of top 10 Athletes for Germany during 1992 Summer Olympic Games

Focusing on Germany's wins during the 1992 Summer Olympic games, graph 3 displays the top 10 athletes who competed. The stacked bar plot presents Franziska van Almsick as the winner of the highest amount of medals - two silver and two bronze. Andreas Wecker, Dagmar Hase, and Daniela Hunger each won three medals each with a combination of different medal types. The remaining athletes earned two medals, with gold being the most common. Kay Bluhm, Nicole Uphoff-Becker and Torsten Ren Gutsche each won two gold medals, which is a lower quantity than other athletes but higher achievement. The success of athletes can be compared overtime by selecting different Olympic games from graph 2 and viewing their medal achievements in graph 3. In conclusion, the graphs created for question 1 offer a variety of insights that could be further explored by comparing athletes and countries, and contextualized with physiological, climate, economic, political, and policy data over time to better understand trends in athletic success.

Question 2: How has hosting the Olympics affected the host country's GDP?

Introduction

Hosting the Olympic Games is one of the most visible and ambitious undertakings for any country. It is often promoted as a catalyst for national development, international visibility, tourism, and economic growth. However, beyond the rhetoric and spectacle, questions remain about whether hosting the Olympics has a measurable economic impact—specifically, whether it influences a country's GDP.

This analysis aims to examine the GDP trends of Olympic host nations and explore whether any relationship exists between Olympic years and economic growth. We focus on a data-driven, visual approach using GDP figures from 1960 to 2020 and a curated list of Olympic hosts. Our interest in this question stems from its relevance to policy, planning, and public investment debates, as countries continue to bid for future Games with hopes of both symbolic and tangible returns.

Approach

To investigate the relationship between Olympic hosting and GDP, we used two core visual tools in an interactive dashboard:

- **World Map Visualization:** A choropleth-style interactive map displays all countries for which GDP data is available. This allows users to easily explore global economic coverage and select countries of interest. It also provides an intuitive gateway into a more detailed exploration of individual countries' economic trends.
- **Dynamic Line Graph:** When a user selects a country from the map, a line graph is generated that shows the country's GDP from 1960 to 2020. Vertical dashed lines indicate the years the Olympic Games were held. This plot provides temporal context and allows users to visually assess whether hosting the Games coincides with noticeable changes in GDP.

We specifically highlight Olympic years because they serve as natural intervention points in a country's historical timeline. By observing the GDP trajectory surrounding those years, we can begin to assess whether hosting may be associated with economic shifts—or whether the changes observed are part of broader, unrelated trends.

Unlike traditional static visualizations, this interactive setup encourages user-driven exploration and hypothesis testing. It accommodates a wide range of perspectives: a policymaker can explore economic returns on investment, a historian can track development milestones, and a casual user can explore trends by curiosity.

Analysis

The world map provides a clear global overview of countries with historical GDP data, highlighting a concentration of coverage in high- and middle-income regions. Users can

World GDP Map with Olympic Hosting Highlights

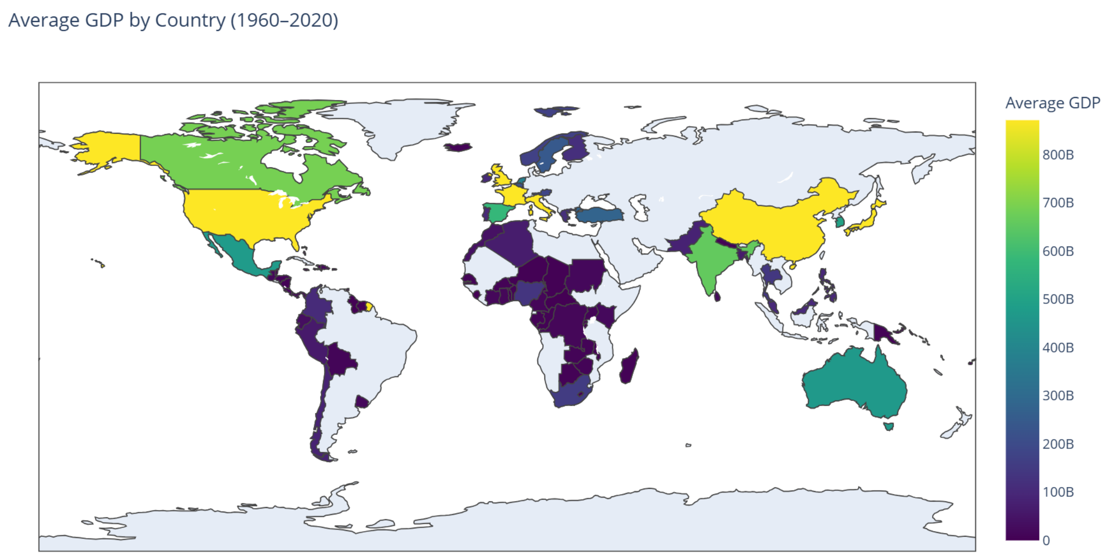


Figure 1: Interactive world map showing countries with available GDP data. Users can click on a country to view its economic trend.

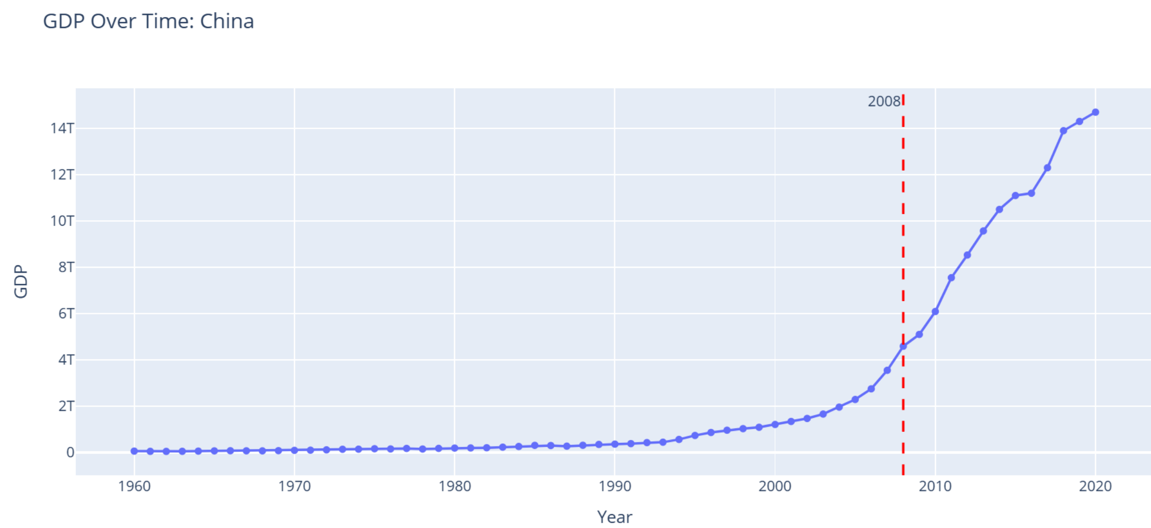
explore any available country, not just Olympic hosts, allowing for comparative study across different regions and economic profiles.

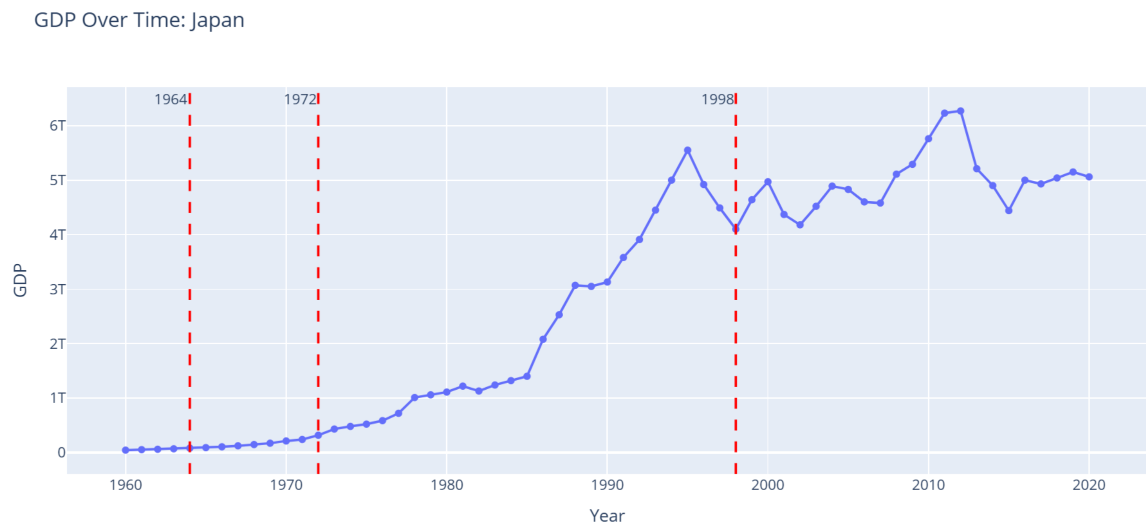
When a user selects a country—such as Japan, China, or Australia—the system generates a line graph plotting GDP against time. Each Olympic year is clearly marked, allowing the viewer to visually identify whether any substantial changes occur around those points.

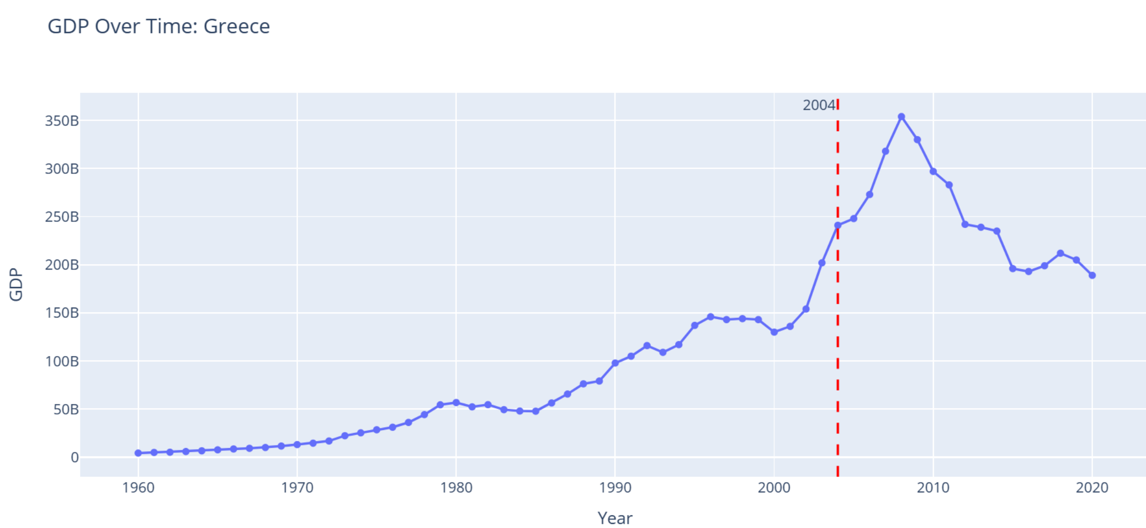
For example:

- **China (2008):** The graph shows an already steep upward GDP trend in the early 2000s, which continues after the Beijing Olympics. While 2008 coincides with a global recession, China's growth trajectory remains relatively unbroken.
- **Japan (1964, 1972, 2020):** Following the 1964 Tokyo Olympics, Japan experienced sustained economic expansion—though this aligns with its post-war industrial boom. The 2020 Games, delayed and limited by COVID-19, coincide with a dip in GDP due to the pandemic.
- **Greece (2004):** Greece hosted the Olympics at a time of moderate GDP levels. However, the years following 2004 saw severe economic decline, which many scholars have attributed to broader fiscal mismanagement rather than the Games themselves.

These case studies suggest that while there are occasional visual alignments between GDP growth and Olympic hosting, they are not universal or consistent. The line graph, with its emphasis on context, helps reveal that Olympic years are rarely turning points in a country's economic narrative.


GDP Over Time: China
2008
14T
12T
10T
8T
6T
4T
2T
0
GDP
1960
1970
1980
1990
2000
2010
2020
Year


GDP Over Time: Japan
1964
1972
1998
6T
5T
4T
3T
2T
1T
0
GDP
1960
1970
1980
1990
2000
2010
2020
Year


GDP Over Time: Greece
2004
350B
300B
250B
200B
150B
100B
50B
0
GDP
1960
1970
1980
1990
2000
2010
2020
Year

Here is the code block used to generate the plots (utilizing the same preprocessing steps as shown in the previous section)

```
import pandas as pd
import plotly.express as px
import plotly.graph_objects as go
from dash import Dash, dcc, html, Input, Output, State

# Load and prepare data
gdp_df = pd.read_csv("data/Countries GDP 1960-2020.csv")
olympics_df = pd.read_csv("data/olympics_cleaned.csv")

# Melt GDP into long format
gdp_long = gdp_df.melt(id_vars=["Country Name", "Country Code"],
                       var_name="Year", value_name="GDP")
gdp_long["Year"] = pd.to_numeric(gdp_long["Year"])

# Standardize country names for Olympic data
country_name_mapping = {
    "USA": "United States",
    "UK": "United Kingdom",
    "South Korea": "Korea, Rep.",
}
olympics_df["country"] = olympics_df["country"].replace(country_name_mapping)

# Average GDP per country
avg_gdp = gdp_long.groupby("Country Name")["GDP"].mean().reset_index()
avg_gdp.columns = ["Country", "Average GDP"]

# Olympic host years
olympic_hosts = olympics_df[["country", "year"]].drop_duplicates()
olympic_hosts = olympic_hosts[olympic_hosts['year'] >= 1960]

# Determine custom range for color scale
quantile = avg_gdp["Average GDP"].quantile(0.8)
color_range_max = quantile

# Dash App
app = Dash(__name__)

app.layout = html.Div([
    html.H1("World GDP Map with Olympic Hosting Highlights"),
    dcc.Graph(id='gdp-map', style={'height': '600px'}),
    html.Div(id='selected-country', style={'display': 'none'}),
    dcc.Graph(id='gdp-line')
])

@app.callback(
```

```
    Output('gdp-map', 'figure'),
    Input('selected-country', 'children')
)
def update_map(_):
    fig = px.choropleth(avg_gdp,
                        locations="Country",
                        locationmode="country names",
                        color="Average GDP",
                        hover_name="Country",
                        color_continuous_scale="Viridis",
                        range_color=(0, color_range_max),
                        title="Average GDP by Country (1960{2020)")
    return fig

@app.callback(
    Output('selected-country', 'children'),
    Input('gdp-map', 'clickData')
)
def store_click_data(clickData):
    if clickData:
        return clickData['points'][0]['location']
    return "United States"

@app.callback(
    Output('gdp-line', 'figure'),
    Input('selected-country', 'children')
)
def update_line_chart(country):
    country_gdp = gdp_long[gdp_long["Country Name"] == country]
    host_years = olympic_hosts[olympic_hosts["country"] == country]["year"].values

    fig = go.Figure()
    fig.add_trace(go.Scatter(x=country_gdp["Year"],
                             y=country_gdp["GDP"],
                             mode='lines+markers',
                             name='GDP'))

    for year in host_years:
        fig.add_vline(x=year, line=dict(color='red', dash='dash'),
                      annotation_text=f"{year}", annotation_position="top left")

    fig.update_layout(title=f"GDP Over Time: {country}",
                      xaxis_title="Year", yaxis_title="GDP",
                      height=500)
    return fig

if __name__ == '__main__':
```

```
app.run(debug=False)
```

Discussion

The visual evidence compiled from the interactive map and GDP timelines reveals a complex picture: hosting the Olympics does not guarantee an immediate or lasting economic benefit, at least not in the form of GDP growth. In most cases, the GDP trajectory of host countries appears to follow long-term trends, unaffected by the Olympic year itself.

This does not mean the Olympics are economically insignificant. Hosting may impact tourism, employment, national branding, or urban development in ways not fully captured by GDP alone. Additionally, the economic "return" from hosting is likely uneven, benefiting certain sectors or regions more than others.

More importantly, countries that host the Olympics tend to already be among the world's economic leaders—or at least possess the infrastructure and political capacity to organize such an event. This introduces selection bias: rather than the Olympics boosting a country's economy, it may be that only economically capable countries host the Olympics in the first place.

The results of this project align with academic literature that questions the economic rationale behind Olympic bids. While hosting may bring symbolic and strategic advantages, the short-term GDP impact appears minimal. Future analysis could enrich this project by incorporating data on debt, employment, tourism revenue, or even public opinion before and after the Games.

Conclusion

This project set out to explore the connections between Olympic success and economic development through the lens of interactive data visualizations. We developed two main analytical paths: one investigating medal trends across countries and Olympic Games, and another evaluating the GDP trajectories of Olympic host nations from 1960 to 2020.

From the medal data, we observed that top-performing countries like the United States, Germany, and China consistently dominate Olympic competitions, supported by strong investments in sports infrastructure, cultural emphasis on athletic excellence, and long-term development strategies. In contrast, lower-performing nations such as India and Vietnam revealed the influence of structural and cultural challenges in achieving consistent international sports success. The ability to drill down from country-level trends to individual athletes and specific Games editions provided layered insight into how national Olympic performance is shaped.

In our economic analysis, we found no strong evidence that hosting the Olympics results in immediate GDP increases. Instead, GDP patterns appear to follow pre-existing national trends, often unrelated to the Olympic event itself. The cases of China, Japan, and Greece illustrate how the economic outcomes of hosting vary depending on broader political and financial contexts. Olympic host status tends to reflect a country's existing economic capability rather than serve as a catalyst for new growth.

Moving forward, there are several opportunities for extending this analysis:

- **Broaden the metrics:** Incorporating indicators such as employment, debt, inequality, or tourism revenue would provide a more holistic view of Olympic impact beyond GDP.
- **Examine pre- and post-Olympic windows:** Comparing economic data across specific intervals before and after hosting may help isolate event-related effects.
- **Include societal impacts:** Analysis of public sentiment, media coverage, or displacement data could capture the lived experience of hosting.
- **Compare non-hosts:** Including countries that applied to host but were not selected could shed light on economic or policy factors that differentiate successful and unsuccessful bids.

Ultimately, this project demonstrates the power of interactive visualization in making large-scale historical data accessible, engaging, and interpretable. Our findings suggest that while Olympic hosting and success are often framed as national triumphs, their economic consequences are far from guaranteed—and deserve ongoing critical evaluation.