

# GLOBAL ECONOMIC TRENDS ANALYSIS USING WORLD BANK DATA

- CSIT 553 Exploratory Data Analysis and Visualization
- Joana Dervishaj
- Vinayak Vemula
- Vishwanath Bhupathi
- Lam Nguyen

# PROJECT DESCRIPTION

- This project analyzes the Special Data Dissemination Standard (SDDS) dataset from the International Monetary Fund (IMF), focusing on reporting delays across various countries.
- SDDS is a framework ensuring countries report economic and financial data in a timely and transparent manner.

## PROJECT OBJECTIVE



Analyze global economic trends using **World Bank Data**.



Perform Exploratory Data Analysis (EDA).



Utilize Data Wrangling and Data Aggregation.



Design and implement data visualizations.



Derive data-driven conclusions.

## DATASET DESCRIPTION

The dataset tracks the availability and timeliness of data reported by countries under the SDDS framework.

The SDDS is an initiative aimed at enhancing the availability of timely and comprehensive economic data for countries around the world.

The dataset measures how often and how promptly these countries report their data to the IMF

Links:  
<https://datacatalog.worldbank.org/search/dataset/0037740/Quarterly-External-Debt-Statistics-SDDS>

# DATASET CONTENT

The first ten rows of dataset:

SDDS/QEDS Information on Data Availability																		
Last updated: 1/31/2025	Country	Conversion Period		Actual Reporting		2013 EDS Guide/BPM6 Tables converted or provided												
		1st QTR	Last QTR	1st QTR reported	Last QTR reported	T1	T2	T3	T4	T1.1	T1.2	T1.3	T1.4	T1.5	T1.6	T2.1	T3.1	T3.2
1	Argentina	2003Q3	2018Q1	2018Q2	2024Q3	x	x	x					x			x		
2	Armenia			1998Q4	2024Q3	x					x	x		x	x			
3	Australia	2003Q4	2024Q3			x				x	x	x						
4	Austria	2003Q2	2013Q2	2013Q3	2024Q3	x					x			x				
5	Belarus			1998Q1	2024Q3	x	x	x	x		x	x	x		x	x	x	x
6	Belgium	2003Q4	2012Q1	2013Q1	2024Q3	x												
7	Brazil	2003Q3	2014Q1	2014Q2	2024Q3	x	x		x		x	x				x		
8	Bulgaria	2005Q3	2013Q3	2013Q4	2024Q3	x	x				x					x	x	
9	Canada			1998Q1	2024Q3	x	x		x	x	x							
10	Chile			2003Q1	2024Q3	x	x	x			x	x			x		x	

## DATASET CONTENT

### Contents (Columns):

- **Index:** Numerical identifier for each entry.
- **Country:** The name of the country that the data pertains to.
- **Ist\_QTR:** The starting quarter of the period for which the data is available **Last\_QTR:** The ending quarter of the period for which the data is available.
- **First\_QTR\_Reported:** The actual first quarter in which the data was reported to the IMF.
- **Last\_QTR\_Reported:** The actual last quarter in which the data was reported.
- **T1,T2,T3,T4,...:** Columns representing different tables of data that countries report under the SDDS. A value of x means that data was reported for that specific table during the quarter, while NaN means it was not.

KEY FOCUS OF  
PROJECT

---

Data Cleaning & Preprocessing

---

Descriptive Statistics & Trend Analysis

---

Comparative Analysis

---

Data Wrangling

---

Data Aggregation

---

Data Visualization

---

Insight Generation

# EXPLORATORY DATA ANALYSIS

The exploratory data analysis

(EDA) involved :

- Data Loading : loading a dataset from the World Bank,
- Data Cleaning : steps such as renaming columns, removing empty columns, and converting data types for consistency., Handle Missing dates were filled with the earliest available date, and 'x' indicators were transformed into Boolean values for easier analysis.
- Data Analysis using descriptive statistics: Descriptive statistics were calculated to summarize numeric data, and categorical analysis revealed the number of unique countries in the dataset.
- Data wrangling techniques were applied to calculate reporting delays, and data aggregation summarized the number of reported tables by country.

```
# Load Excel file
```

```
file_path = "https://datacatalogfiles.worldbank.org/ddh-published-v2/0037740/31/DR0092145/SDDS_QEDS_Data_Availability_2024Q3_v2.xlsx"
```

```
xls = pd.ExcelFile(file_path)
```

```
# Load and clean data
```

```
df = pd.read_excel(xls, sheet_name="SDDS", skiprows=4)
```

```
df.columns = df.iloc[0] # Set first row as header
```

```
df = df[1:].reset_index(drop=True) # Remove redundant header row
```

```
df = df.dropna(axis=1, how="all") # Drop fully empty columns
```

```
# Convert columns to appropriate types
```

```
df["Index"] = pd.to_numeric(df["Index"], errors="coerce")
```

```
df["1st_QTR"] = pd.to_datetime(df["1st_QTR"], errors="coerce")
```

```
df["Last_QTR"] = pd.to_datetime(df["Last_QTR"], errors="coerce")
```

```
df["1st_QTR_reported"] = pd.to_datetime(df["1st_QTR_reported"], errors="coerce")
```

```
df["Last_QTR_reported"] = pd.to_datetime(df["Last_QTR_reported"], errors="coerce")
```

```
# Data Wrangling
```

```
# Fill missing values in date columns with the earliest available date
```

```
date_columns = ["1st_QTR", "Last_QTR", "1st_QTR_reported", "Last_QTR_reported"]
```

```
df[date_columns] = df[date_columns].fillna(df[date_columns].min())
```

```
# Data Aggregation
```

```
# Group by Country and count the number of reported tables
```

```
df_aggregated = df.iloc[:, 6:].groupby(df["Country"]).sum()
```

```
# Descriptive statistics
```

```
numeric_summary = df.describe()
```

```
categorical_summary = df["Country"].nunique()
```

```
table_availability = df.iloc[:, 6:].sum()
```

# OUTPUT

## Numeric Summary:

	Index	1st_QTR		Last_QTR	
count	80.0000	84		84	
mean	40.5000	2001-09-21 07:08:34.285714304	2008-08-19 00:00:00		
min	1.0000	1998-01-01 00:00:00	1999-10-01 00:00:00		
25%	20.7500	1998-01-01 00:00:00	1999-10-01 00:00:00		
50%	40.5000	2002-10-01 00:00:00	2012-02-15 12:00:00		
75%	60.2500	2003-07-01 00:00:00	2013-04-01 00:00:00		
max	80.0000	2013-04-01 00:00:00	2024-07-01 00:00:00		
std	23.2379	NaN		NaN	

	1st_QTR_reported		Last_QTR_reported	
count	84		84	
mean	2008-10-25 18:51:25.714285824	2024-01-27 13:25:42.857142784		
min	1998-01-01 00:00:00	2021-04-01 00:00:00		
25%	2002-01-01 00:00:00	2024-07-01 00:00:00		
50%	2012-08-16 00:00:00	2024-07-01 00:00:00		
75%	2013-07-01 00:00:00	2024-07-01 00:00:00		
max	2020-01-01 00:00:00	2024-07-01 00:00:00		
std	NaN		NaN	

## Aggregated Table Report Count by Country:

Country	T1	T2	T3	T4	T1.1	T1.2	T1.3	T1.4	T1.5	T1.6	T2.1
Argentina	1	1	1	0	0	0	0	1	0	0	1
Armenia	1	0	0	0	0	1	1	0	1	1	0
Australia	1	0	0	0	1	1	1	0	0	0	0
Austria	1	0	0	0	0	1	0	0	1	0	0
Belarus	1	1	1	1	0	1	1	1	0	1	1
...	..	..	..	..	...	...	...	...	...	...	...
United Kingdom	1	0	0	0	0	0	0	0	0	0	0
United States	1	1	1	1	0	0	0	0	0	0	0
Uruguay	1	1	0	0	0	1	0	0	1	0	1
Uzbekistan	1	1	1	1	1	1	1	1	1	1	1
West Bank and Gaza	1	1	1	1	0	1	1	1	0	0	1

## Reporting\_Delay\_Days

count	84.000000
mean	5639.559524
min	-1187.000000
25%	4109.000000
50%	4519.500000
75%	9040.000000
max	9040.000000
std	2867.398823

## Top Countries with Highest Reporting Delays:

	Country	Reporting_Delay_Days
42	Malaysia	9040
37	Korea	9040
1	Armenia	9040
27	Hungary	9040
26	Hong Kong SAR, China	9040
45	Mexico	9040
47	Mongolia	9040
49	Namibia	9040
60	Saudi Arabia	9040
61	Seychelles	9040

## T3.1 T3.2 Reporting\_Delay\_Days

Country	T3.1	T3.2	Reporting_Delay_Days
Argentina	0	0	2373
Armenia	0	0	9040
Austria	0	0	4109
Belarus	1	1	9040
Belgium	0	0	4565
...	...	...	...
United Kingdom	0	0	9040
United States	0	1	4109
Uruguay	0	0	3653
Uzbekistan	0	1	9040
West Bank and Gaza	1	1	4109



# DATA VISUALIZATION

- **Bar Chart**
- **Area Chart**
- **Diverging Heat map**

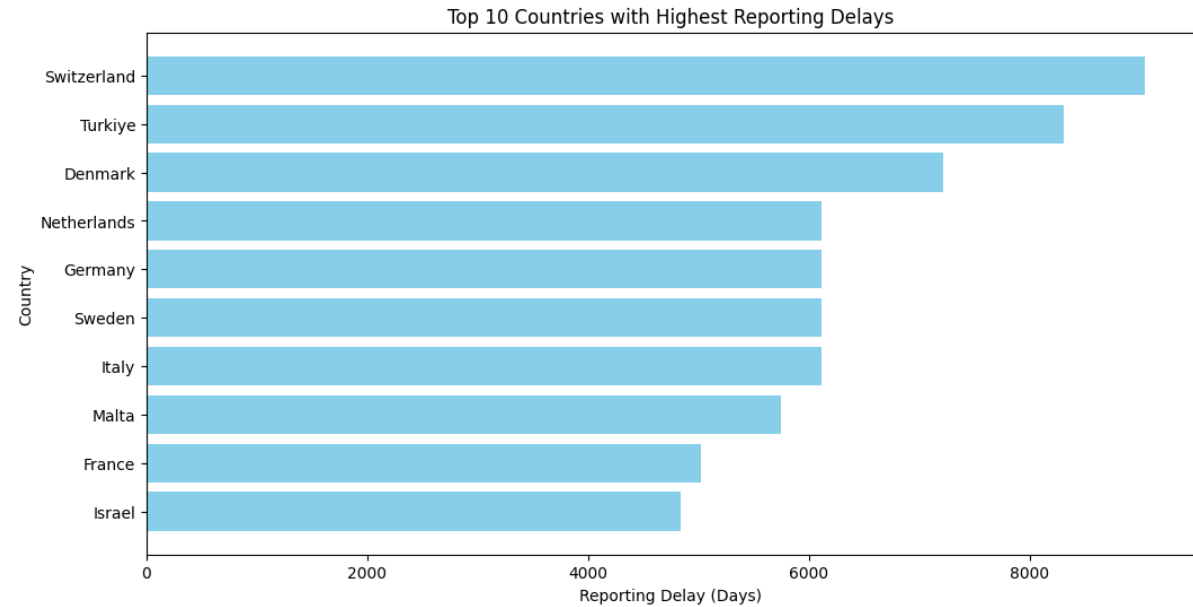
# DATA VISUALIZATION BY BAR CHART

- **Bar Chart:** Visualizes the top 10 countries with the highest reporting delays

```
[ ] plt.figure(figsize=(12, 6))
    top_10_countries = df.sort_values(by="Reporting_Delay_Days", ascending=False).head(10)
    plt.barh(top_10_countries["Country"], top_10_countries["Reporting_Delay_Days"], color='skyblue')
    plt.xlabel("Reporting Delay (Days)")
    plt.ylabel("Country")
    plt.title("Top 10 Countries with Highest Reporting Delays")
    plt.gca().invert_yaxis()
    plt.show()
```

## DATA VISUALIZATION BY BAR CHART

**Bar Chart:** Visualizes  
the top 10 countries with  
the highest reporting  
delays



## DATA VISUALIZATION BY AREA CHART

**Area Chart:** Displays the trend of average reporting delays over time, highlighting patterns and variations.

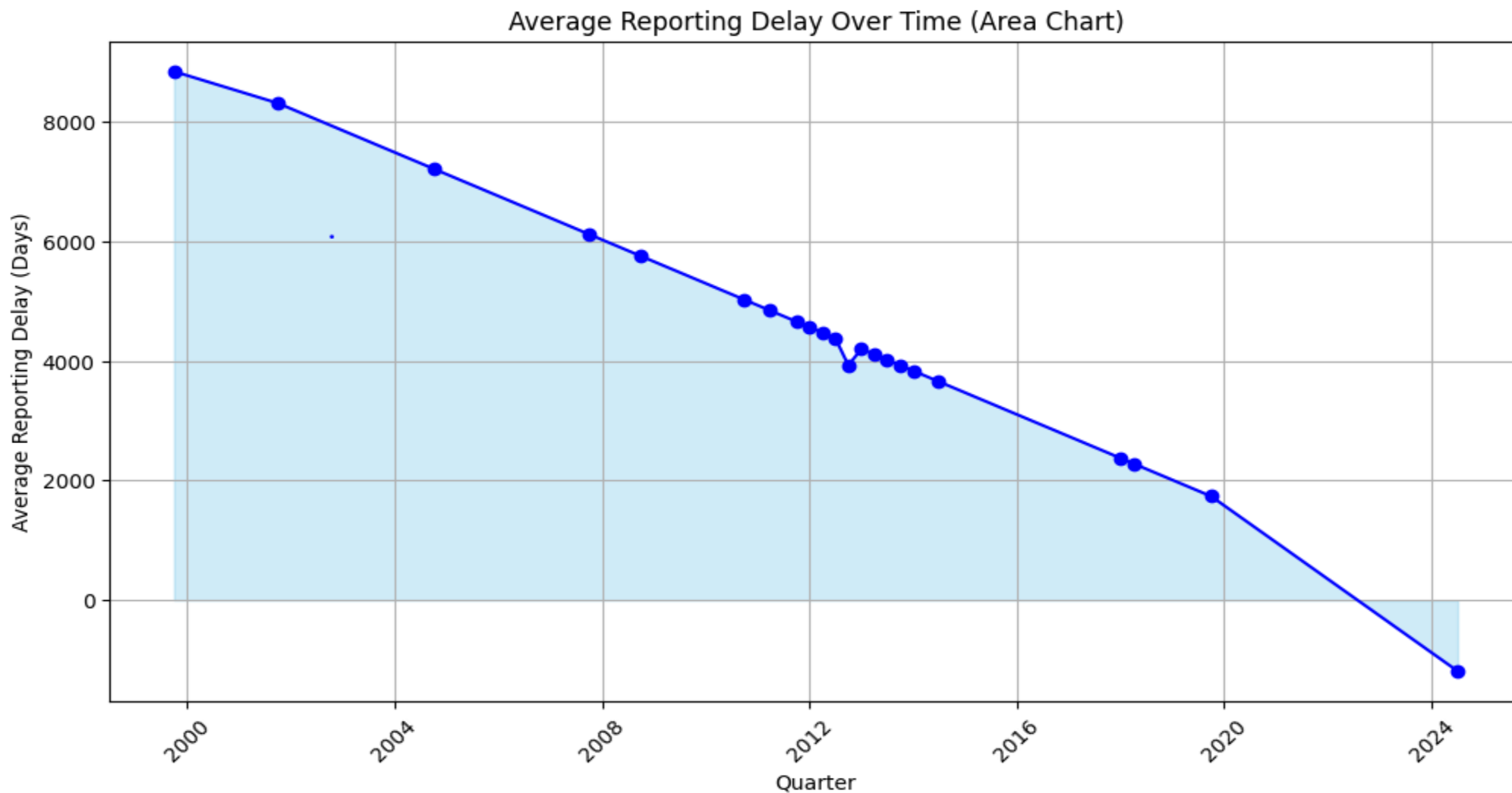
The **average reporting delay over time** is visualized using an area chart.

The x-axis represents time (quarterly), and the y-axis represents the **average reporting delay**.

The area chart fills the space under the line plot with a light blue color.

The title of the chart: "Trend of Reporting Delays Over Time".

- **Area Chart:** Displays the trend of average reporting delays over time, highlighting patterns and variations.

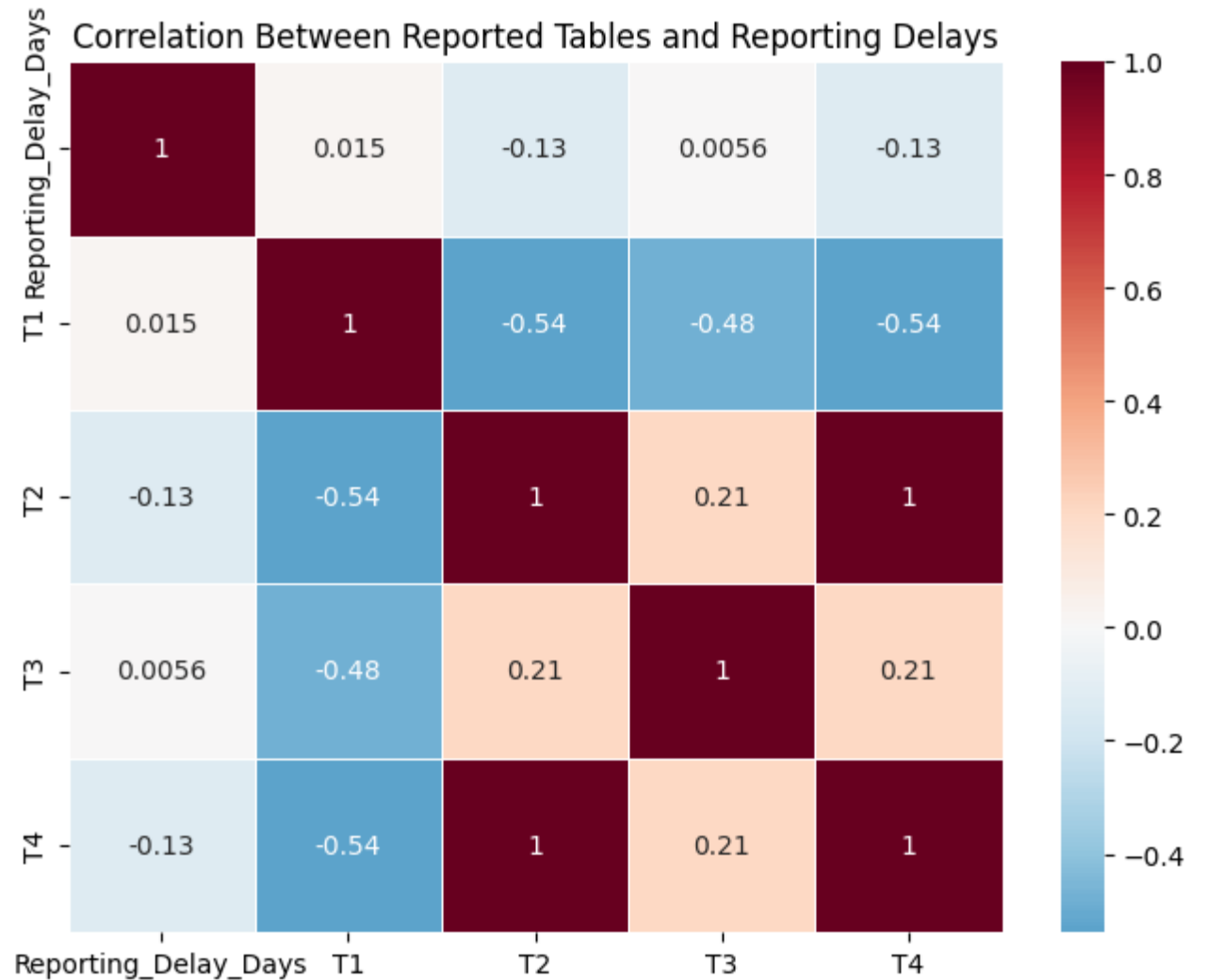


# DATA VISUALIZATION BY HEAT MAP

- The Heat map will display the **correlation** between each of the reported tables (T1,T2, etc.) and the **reporting delays**.
- If there's a **positive correlation**, it means that as the number of tables reported increases, the delay tends to also increase. Conversely, a **negative correlation** indicates that reporting more tables could be associated with less delay.
- The values on the Heat map range from **-1 to 1**, where:
  - **1**: Perfect positive correlation.
  - **-1**: Perfect negative correlation.
  - **0**: No correlation.

## DATA VISUALIZATION BY HEAT MAP

**Diverging Heatmap:**  
Correlation between  
reported tables and  
reporting delays





## SNAPSHOTS OF CODE

- Python code link:  
[https://colab.research.google.com/drive/1iB6kreDfuQN36AHMxjaA\\_12tMdcBlm32#scrollTo=df1feb0f](https://colab.research.google.com/drive/1iB6kreDfuQN36AHMxjaA_12tMdcBlm32#scrollTo=df1feb0f)

# CONCLUSION

- The analysis revealed significant reporting delays across multiple countries, with some nations experiencing notably higher delays.
- The trend analysis indicated fluctuations in average reporting delays over time, suggesting potential seasonal or systemic patterns.
- The correlation heatmap showed limited correlation between the number of reported tables and reporting delays, indicating delays may be influenced by other factors.
- Overall, the findings emphasize the need for improved reporting processes to ensure timely data availability.