VISUALIZING US NATURAL DIASTER DECLARATION – TRENDS AND PATTERNS

Week 4 Documentation

Live Data, ETL Pipeline, and GitHub Project Organization

Live Data, ETL Pipeline, and GitHub Project Organization

1. Introduction

Week 4 focused on advancing the FEMA Disaster Declarations project from static analysis to concepts of **live data integration**, **ETL pipeline design**, and **project organization for GitHub deployment**. While the FEMA dataset itself is not updated daily, understanding how live data works in Power BI and how to structure projects for collaboration is essential for building scalable, professional data visualization solutions.

2. Live Data in Power BI

2.1 Power BI Server Connections

- Power BI can connect to organizational servers such as **SQL Server** or **Oracle databases**.
- These connections allow dashboards to refresh automatically when new records are added to the source system.

2.2 Refresh Limitations

- **Normal Power BI usage**: Limited to **8 refreshes per day**, and refreshes must be scheduled manually.
- **Premium subscription**: Allows **24–25 refreshes per day**, with **automatic refreshes** enabled.
- This distinction is critical for organizations that require near real-time reporting.

2.3 Government and Sensitive Data

- Power BI cannot directly connect to official government websites due to **data sensitivity and access restrictions**.
- Instead, organizations use **approved APIs** or secure data services.
- Some websites allow access, while others restrict it to protect sensitive information.

2.4 FEMA Dataset Context

- For FEMA disaster declarations, **live data integration is not practical** because disasters do not occur daily.
- Instead, the dataset is downloaded periodically and imported into Power BI for analysis.
- This ensures consistency while avoiding unnecessary refresh cycles.

2.5 Real-World Examples

- **CRM Systems**: Customer tickets are stored in databases; dashboards refresh as new tickets are raised and resolved.

- **Banking Systems**: Transactions update continuously, requiring live dashboards for fraud detection or compliance.
- **Dummy/Synthetic Data**: Often used in training or testing environments to simulate live data streams.

3. ETL Pipeline

3.1 Definition

- **ETL (Extract–Transform–Load)** is a structured process to:
  - **Extract** raw data from source systems.
  - **Transform** it through cleaning, formatting, and enrichment.
  - **Load** the processed data into Power BI to create visualizations.

3.2 Purpose

- Automates repetitive cleaning tasks.
- Ensures consistency across datasets.
- Provides scalability for larger or multiple data sources.
- Optimizes the data preparation process for analytics.

3.3 ETL in the FEMA Project

- **Extract**: Download FEMA disaster declarations dataset.
- **Transform**: Apply cleaning steps (remove duplicates, handle missing values, derive fiscal year, status flags).
- **Load**: Import cleaned dataset into Power BI for visualization.

3.4 Benefits

- Reduces manual effort in data preparation.
- Ensures reproducibility of cleaning steps.
- Provides a template that can be reused for other datasets.
- Aligns with professional standards in data analytics.

4. GitHub Repository Organization

4.1 Folder Structure

To ensure clarity and collaboration, the project repository was organized into the following folders:

1. **data/**
    - Contains raw FEMA dataset and processed cleaned dataset.
    - Ensures reproducibility by keeping both versions available.
2. **Power BI/**
    - Stores .pbix files for dashboards.
    - Allows others to open and explore the visualizations directly.
3. **Data Cleaning/**
    - Documents the cleaning process in Power Query and Python.
    - Includes scripts and notes for reproducibility.
4. **Screenshot/**
    - Contains labeled images of dashboards and workflow steps.
    - Provides quick visual references for readers.
5. **Documentation/**
    - Week-by-week documentation.
    - Explains what was done, analyzed, and learned.

4.2 README.md Structure

The repository includes a **README.md** file to guide users:

- **Project Title**: Clear and descriptive.
- **Problem Statement**: 2–3 lines explaining the purpose of the project.
- **Dataset Description**: Rows, columns, and data types.
- **KPIs Used**: Key metrics for quick understanding.
- **Dashboard Pages**: Breakdown of each dashboard page with brief explanations.
- **Key Insights**: Summarized findings from analysis.
- **Recommendations**: Suggestions for improving dashboards or project scope.
- **Tools Used**: List of all tools (Power BI, Python, GitHub, etc.) with one-line descriptions of usage.

5. Outcome of Week 4

By the end of Week 4:

- Gained understanding of **live data integration** in Power BI and its limitations.
- Understood the concept of **ETL pipelines** to automate data cleaning and loading.
- Established a **GitHub repository structure** with clear folders and README documentation.
- Prepared the project for collaborative sharing and professional presentation.

6. Next Steps (Week 5 Preview)