Data Pipeline Report

Question

What is the impact of various factors on jail deaths across different states in the US?

Data Sources

Source: The datasets I used are sourced from Thomson Reuters, available at this link, Link to Dataset which is part of the Reuters special report on US jails Reuters Page

Reason for Choice: I chose these datasets (all_jails, all_deaths, jail_deaths) because they are provided in CSV format, making them examples of structured data. Each dataset has a clear schema with defined columns, such as state, cause_of_death, and num_deaths. I found these structured characteristics ideal for easy querying and analysis using traditional data science methods. However, I noticed that the datasets did not include any semi-structured or unstructured components, such as text descriptions, which could have provided additional context for the analysis.

Data Structure and Quality: The datasets include columns such as state, date, cause_of_death, num_deaths, and other relevant information. Initially, the data quality was not perfect; there were missing values and potential duplicates. Addressing these issues was a key part of my data cleaning process to ensure the reliability of my analysis.

Licenses: The datasets are publicly available and can be used for analysis. I made sure to check the specific license terms provided by Thomson Reuters and comply with any attribution requirements. This ensures that my use of the data is both legal and ethical.

Data Pipeline

Technologies Used: I implemented the pipeline in Python, utilizing libraries such as pandas for data manipulation, requests for downloading data, zip file for extracting ZIP files, sqlite3 for database operations, and logging for tracking the pipeline's execution.

Steps and Transformations:

1. **Download and Extraction**:
    1. The first step in my pipeline was to download the datasets from the provided URL and extract them from a ZIP file. This ensured that I had the latest data available for analysis.
2. **Data Cleaning**:
    1. **Remove Duplicates**: I removed duplicate rows to ensure data integrity. Duplicates can skew the analysis and lead to incorrect conclusions, so this step was crucial.
    2. **Handle Missing Values**: I checked critical columns like cause_of_death and state for missing values. For rows with missing critical information, I decided to drop them to maintain the quality of the dataset.
    3. **Date Conversion**: The date column was converted to a datetime format. This step was important for any time-series analysis and to ensure consistency in date-related operations. I handled invalid dates by coercing errors, which helped in identifying and managing incorrect date entries.
    4. **Fill Missing Numerical Values**: For numerical columns like num_deaths, I filled missing values with 0. This approach ensured that the absence of data did not affect the numerical analysis.
3. **Data Storage**:
    1. **CSV Files**: I saved the cleaned data as CSV files for easy access and analysis. CSV is a widely used format that is compatible with many tools and platforms.
    2. **SQLite Database**: Additionally, I stored the data in an SQLite database. This allowed for more complex queries and analysis, which can be beneficial for in-depth studies.

Challenges and Solutions:

- **Missing Values**: One of the main challenges was dealing with missing values in critical columns. I addressed this by dropping rows with missing critical information and filling in missing numerical values with 0.
- **Invalid Dates**: Converting dates to a standard format was another challenge. I handled invalid dates by coercing errors, which helped in maintaining the consistency of the date data.

Meta-Quality Measures: To ensure the robustness of my pipeline, I included logging at each step. This allowed me to monitor the process and handle errors effectively.

Logging is crucial for traceability and debugging, making it easier to identify and fix issues that may arise during the pipeline’s execution.

Result and Limitations

Output Data: The cleaned datasets are stored in both CSV and SQLite formats. The CSV files are suitable for quick analysis, while the SQLite database allows for more complex queries. This dual storage approach provides flexibility depending on the analysis needs.

Data Structure and Quality: The structured format of my output datasets allows me to perform efficient analysis and create clear visualizations. However, I recognize that relying solely on structured data limits the depth of insights. For example, unstructured or semi-structured data, such as detailed incident reports, could provide richer contextual understanding. In the future, I plan to explore ways to integrate semi-structured data for a more holistic analysis.

Format Choice: I chose CSV for its simplicity and ease of use. It is a common format that can be easily shared and used with various tools. SQLite was chosen for its ability to handle more complex data operations and queries, making it suitable for in-depth analysis.

Critical Reflection: While the data cleaning process significantly improved data quality, there may still be limitations due to the inherent quality of the source data. For example, there might be inaccuracies or biases in the original data collection process that I cannot correct. Future work could include more sophisticated data validation and enrichment techniques to further enhance data quality.