The German International University
Faculty of Informatics and Computer Science
Dr. Nada Sharaf
TA. Mariam Ali
TA. May Magdy
TA. Mohamed Abdelsatar

Data Engineering and Visualization, Winter Semester 2025
Project Milestone 1
Submission Date: Friday 21/11/2025

Project Overview:

The goal of this project is to navigate the complete data engineering process using a real-world dataset on motor vehicle collisions in New York City. You will explore, clean, integrate, and then build an interactive website to visualize insights in a dynamic report. The datasets are:

- **NYC Motor Vehicle Collisions - Crashes**: Available at `https://data.cityofnewyork.us/Public-Safety/Motor-Vehicle-Collisions-Crashes/h9gi-nx95/data_preview`. This dataset contains over 2 million crash records (2012 to 2025), with columns like crash date/time, location (latitude/longitude, borough, zip code), injuries, fatalities, contributing factors, and vehicle types. It is raw, with missing values, outliers, and inconsistencies requiring robust cleaning.
- Integrate with a related dataset called **Motor Vehicle Collisions - Person** or **Motor Vehicle Collisions - Vehicles** using the common column `COLLISION_ID`:
`https://data.cityofnewyork.us/Public-Safety/Motor-Vehicle-Collisions-Person/f55k-p6yu/data_preview` or
`https://data.cityofnewyork.us/Public-Safety/Motor-Vehicle-Collisions-Vehicles/bm4k-52h4/data_preview`.

Project Team (from 4 to 5 people per group)

Each team member must propose at least 2 research questions (e.g., Which borough has the highest crash rate per capita); a 4-member team needs 8, and a 5-member team needs 10. Clearly indicate each members contributions (e.g., research questions, cleaning tasks, website components).

No extensions will be granted for milestone due dates. Late submissions will not be graded. If you anticipate issues (e.g., travel), email the instructor at least one week in advance.

Getting Started: Loading the Datasets

Add the following code to your Jupyter/Colab notebook to load the datasets directly from NYC Open Data:

```python
import pandas as pd

# Load crashes dataset
crashes_url = 'https://data.cityofnewyork.us/api/views/h9gi-nx95/rows.csv?accessType=download'
df_crashes = pd.read_csv(crashes_url, low_memory=False)

# Load persons dataset
```

```python
persons_url = 'https://data.cityofnewyork.us/api/views/f55k-p6yu/rows.csv?accessType=
    download'
df_persons = pd.read_csv(persons_url, low_memory=False)

# Quick preview
df_crashes.head()
df_persons.head()
```

Listing 1: Loading Crashes and Persons Datasets

Milestone 1: Cleaning, Integration, and Visualization

In this milestone, you will:

- **Explore the Data**: Use descriptive statistics and initial plots to understand the datasets structure, issues, and patterns.
- **Clean the Data (Pre-Integration)**:
  - Handle missing values (justify drop vs. impute).
  - Detect and address outliers (e.g., IQR, domain rules).
  - Standardize formats (dates, strings, categories).
  - Remove duplicates.
- **Integrate Additional Data: Join with related NYC Open Data table (Person via COLLISION_ID). Document integration steps and justify choices.**
- **Post-Integration Cleaning** *(Required)*: After joining, resolve:
  - New missing values from joins.
  - Inconsistent or redundant columns.
  - Data type mismatches.
- **Build an Interactive Website:** Create a fully interactive data visualization website using **Dash (Python/Plotly)** or any web framework such as **React with Plotly.js, Flask, or Django**. The website should provide an engaging and dynamic way for users to explore and generate insights from the integrated dataset.

  The website must include the following structure and functionality:
  - Multiple **dropdown filters** (e.g., Borough, Year, Vehicle Type, Contributing Factor, Injury Type) allowing users to dynamically filter data.
  - A **search mode**, where users can type queries (e.g., “Brooklyn 2022 pedestrian crashes”) to automatically apply filters.
  - A central **“Generate Report” button** that, when clicked, dynamically updates all visualizations based on selected filters or search terms.
  - Visualizations should include a variety of chart types, such as bar charts, line charts, heatmaps, maps, or pie charts, and must offer interactivity (hover, zoom, or filter updates).
  - Ensure that all components respond in real time to user interactions and that the website layout is user-friendly and visually consistent.
  - Host and test the website using a free deployment platform such as **Vercel**, **Render**, or **Heroku**, and confirm full functionality before submission.

The Grades Will Be Given For:

- **Implementation**:
  - Thorough exploratory data analysis (EDA) with relevant statistics and visualizations.
  - Effective **pre- and post-integration cleaning** (e.g., appropriate handling of missing values, outliers, and inconsistencies).
  - Quality of integration: clear joins, justified data sources, and added value.
  - Relevance and interactivity of website visualizations (e.g., do dropdowns + **Generate Report button** drive meaningful updates?).
  - Complexity and originality of research questions (e.g., avoid simplistic counts; aim for insights like predictive patterns or spatial trends).
- **Descriptive Markdown Cells**: Document exploration, cleaning, and integration steps. Justify decisions (e.g., why drop vs. impute nulls?), discuss alternatives considered, and explain how insights were reached.
- **Clean Code**: Use descriptive variable names, modular functions, and clear structure.
- **Code Comments**: Comment on complex logic (e.g., outlier detection algorithms, integration joins).
- **Deductions** for:
  - Poor contribution documentation.
  - Inappropriate plot choices or unclear labels.
  - Messy or uncommented code.
  - Incorrect or trivial conclusions.
  - Missing **"Generate Report" button** or post-integration cleaning.

Deliverables

- A **Jupyter/Colab notebook** containing:
  - Dataset overview (e.g., size, columns, known issues).
  - Detailed steps for **EDA**, **pre-integration cleaning**, **integration**, and **post-integration cleaning**.
  - Visualizations supporting data understanding and validation of cleaning steps.
- A **functional website** demonstrating interactive exploration and reporting, including:
  - Dropdown-based filtering, search mode, and Generate Report interactivity.
  - Deployed version (e.g., on Vercel) and full source code in a GitHub repository.
  - A **README file** that includes setup steps, deployment instructions, and a short description of each team members contribution.

Submission Instructions

All project code, notebooks, and website files must be hosted on a **GitHub repository**. Each team member must be added to the same repository and have visible contributions (commits or pull requests). Submit your GitHub repository link through the submission form.

Ensure your submission includes all required files in the GitHub repository before the deadline. Changes after the deadline (21/11/2025) will not be considered. Further submission details will be announced later.