LAB REPORT – 1

CSE-564 (VISUALIZATION)
NYC Taxi Trip & Weather Data
By: **Kartik Kirankumar Vadhawana (116740869)**

Aim

I aim to visualize a combined dataset of **NYC Yellow Taxi Trips** and **NYC Weather** data using various charts (bar charts, histograms, scatter plots, and time-series charts) in a Dash application. These visualizations help me understand how time of day, distance, fares, and weather conditions (especially precipitation) influence taxi usage.

Data Set

I used two main datasets:

1. **NYC Yellow Taxi Trip Data**
   - **Source**: Kaggle – https://www.kaggle.com/datasets/elemento/nyc-yellow-taxi-trip-data
   - Contains information about trip distances, fare amounts, pickup and drop-off times, passenger counts, and more. The data includes both numerical and categorical features, making it suitable for a variety of visualizations.
2. **NYC Weather Data (1869–2022)**
   - **Source**: Kaggle – https://www.kaggle.com/datasets/danbraswell/new-york-city-weather-18692022/data
   - Provides daily weather observations, such as precipitation (PRCP), temperature, and other meteorological variables.

These two datasets were merged based on matching dates to associate each taxi trip with the corresponding daily precipitation level. This integration allowed me to explore potential relationships between weather conditions and taxi trip characteristics.

Processing of Raw Data and Features/Attribute Selection

I used a Python script to:

1. Read both datasets.
2. Clean the data, removing or filtering out any incomplete records.

3. Convert date/time columns (e.g., `tpep_pickup_datetime`) to proper datetime objects.
4. Merge weather information (specifically precipitation, `PRCP`) with the taxi data on the matching date fields.

From the merged dataset, I created derived columns to facilitate analysis:

- **pickup_hour**: Extracted from the pickup timestamp (`tpep_pickup_datetime`) to categorize trips by the hour of day (0–23).
- **pickup_date**: Extracted as the date component (YYYY-MM-DD) for time-series analysis.

Below is a brief description of the key attributes used:

Categorical Attributes

1. **VendorID**: The ID of the vendor providing the trip record.
2. **RatecodeID**: The fare rate code (e.g., standard rate, JFK, Newark, etc.).
3. **payment_type**: How the passenger paid for the trip (credit card, cash, etc.).

Numerical Attributes

1. **passenger_count**: Number of passengers in the taxi.
2. **trip_distance**: Distance of the trip in miles.
3. **fare_amount**: Base fare charged (not including tips and surcharges).
4. **extra, mta_tax, tip_amount, tolls_amount**: Various additional charges.
5. **total_amount**: Final total fare including taxes, tolls, and tip (if paid by card).
6. **PRCP**: Precipitation (in inches), merged from the weather dataset.
7. **pickup_hour** (derived): Hour of day for the pickup time.
8. **pickup_date** (derived): Date of the pickup for day-by-day grouping.

Why These Data Are Interesting & Their Value

NYC sees a **high volume** of daily taxi trips, making it an ideal case study for urban transportation analysis. **Weather conditions** (like precipitation) can affect passenger demand, trip distances, and fare amounts. By merging taxi trip data with weather observations, I can examine:

- **Peak Hour Analysis**: Identifying the busiest hours aids driver shift scheduling and traffic management.
- **Weather Influence**: Observing how precipitation correlates with fares or distances highlights operational challenges during bad weather.
- **Fare Distribution**: Understanding how fares vary by time can guide ride-hailing platforms and city regulators.

Deploying the Dash Application

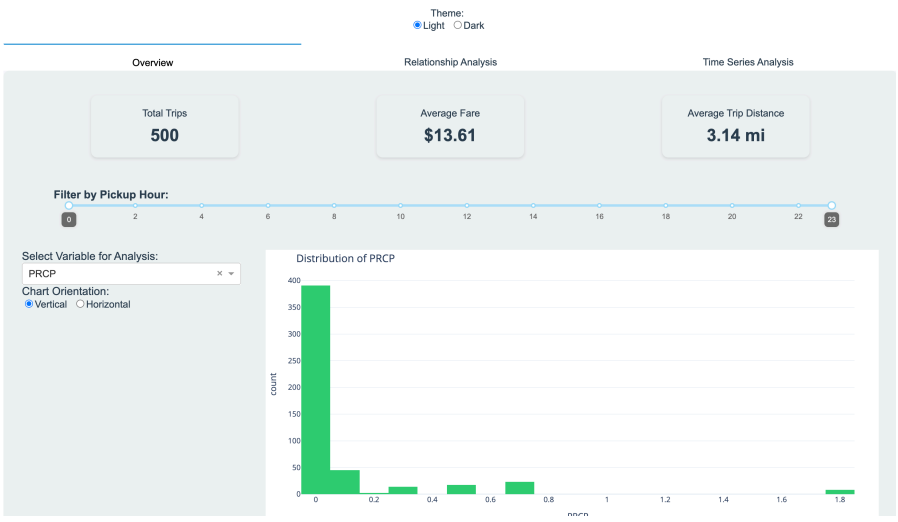
1. **Install Dependencies**: Ensure the required Python packages (e.g., `dash`, `plotly`, `pandas`) are installed.
2. **Run the Script**: Execute `python app.py` (or the appropriate filename).
3. **Access the App**: Open a web browser at the local URL (e.g., `http://127.0.0.1:8050`).

Please note that an **active internet connection** may be required if your environment needs to download any additional libraries.

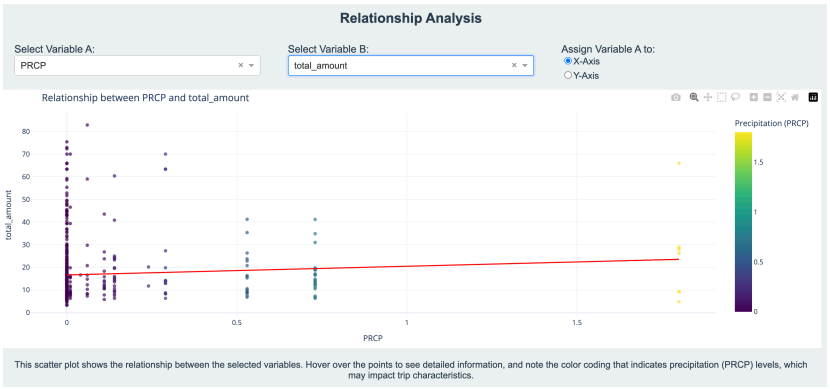
Features of the Application

The dashboard comprises multiple **tabs** and **interactive elements**:

1. **Overview Tab**
    - **Key Metrics**:
        - Total Trips
        - Average Fare
        - Average Trip Distance
    - **Hour Slider**: Filters data by `pickup_hour`, allowing users to explore how trips vary throughout the day.
    - **Single Variable Analysis**:
        - A **dropdown** to select a variable (categorical or numerical).
        - A **radio button** to switch the orientation of the chart (vertical/horizontal).
        - If the variable is **categorical**, a **bar chart** displays frequency counts.
        - If the variable is **numerical**, a **histogram** shows the distribution in fixed bins.



2. **Relationship Analysis Tab**
    - **Scatter Plot**:
        - Two dropdowns let users select **Variable A** and **Variable B**.
        - A radio button determines which variable is assigned to the **X-axis** vs. **Y-axis**.
        - The points are color-coded by **PRCP** (precipitation), with an optional trendline overlay to highlight correlations.



3. **Time Series Analysis Tab**
    - A dropdown allows users to choose one of three metrics:
        1. Total Trips
        2. Average Fare
        3. Average Trip Distance
    - A line chart plots the chosen metric over the `pickup_date`, revealing daily trends or anomalies.

4. **Dark/Light Theme Toggle**
    - A radio button at the top lets users switch between **light mode** and **dark mode**, dynamically updating the dashboard's background, text colors, and chart templates.

Noteworthy Points About the Implementation

1. **Data Integration**
   Merging daily weather data with individual taxi trips required carefully aligning dates and handling any missing or partial records.
2. **User-Friendly Interactivity**
   - **Hour Slider**: Real-time filtering of the dataset by pickup hour.
   - **Orientation Toggle**: Instantly switch bar/histogram orientation.
   - **Scatter Plot Axis Assignment**: A flexible approach that lets users decide which variable goes on X or Y.
3. **Responsive Layout & Theming**
   - Each tab is structured for clarity, ensuring that key metrics, filters, and graphs are neatly organized.
   - Dark mode uses `plotly_dark` templates for the charts, while light mode uses `plotly_white`, offering better readability.
4. **Performance Considerations**
   For large datasets, additional optimizations or server-side callbacks might be needed. However, for moderate data sizes, the client-side interactivity is generally smooth.

Conclusion

By integrating **NYC Taxi Trip Data** with **NYC Weather** observations, I created a **rich, interactive dashboard** that reveals how factors such as time of day, trip distance, fare, and precipitation come together to shape taxi usage. The multi-tab layout, real-time filters, and theming options make the application accessible to both technical and non-technical audiences. This approach highlights the importance of **data-driven decision-making** in urban transportation planning and provides a robust foundation for further analysis or expansion (e.g., adding temperature, wind speed, or holiday data).