Stats 436 - HW2

Dataset Description

The dataset is from the NYC government website. It contains records of yellow and green taxi trips, capturing fields such as pick-up and drop-off dates/times, locations, trip distances, itemized fares, rate types, payment methods, and driver-reported passenger counts.

For this assignment, I focused on the **green taxi** dataset for the year 2015. The required data files (12 files) can be found in the `data` folder of this GitHub repository. According to the data dictionary, the dataset includes 18 columns. However, the key columns used in this analysis are:

- `lpep_pickup_datetime`: The date and time when the meter was engaged.
- `lpep_dropoff_datetime`: The date and time when the meter was disengaged.
- `trip_distance`: The distance traveled during the trip, reported by the taximeter in miles.
- `passenger_count`: The number of passengers in the vehicle.
- `total_amount`: The total amount charged to passengers, including tips.

Requirements

The R code for this assignment can be found in the root directory of the repository, named hw2.rmd. To run the script, ensure that the necessary directories can be created with write permissions (for downloading the dataset) and that R's working directory is set to the location of the `hw2.rmd` file. The project structure matches that of the repository on GitHub, so you can also clone the entire repository to run the code.

Questions and Answers

- **Question 1**: What are some interesting facts that you learned through the visualization. Provide at least one unexpected finding.

    1. (Unexpected) The linear relation ship between trip distance and trip duration becomes weaker as the trip distance increases.
    2. (Unexpected) The data seems not clean enough, certain drop-off times appear to be truncated at midnight, probabily because of some issues with the data collection process.
    3. The charges of the trip is related to the distance, longer distance means higher charges.
    4. Of all different months, the distribution of pick-up time in a day is similar. Less trips happen from 3 to 7 am, and more trips happen from 3pm to 11pm. I didn't see a significant difference of this pattern between months.

- **Question 2**: How did you create the interface? Were there any data preparation steps? What guided the style customizations and interface layout that you used.

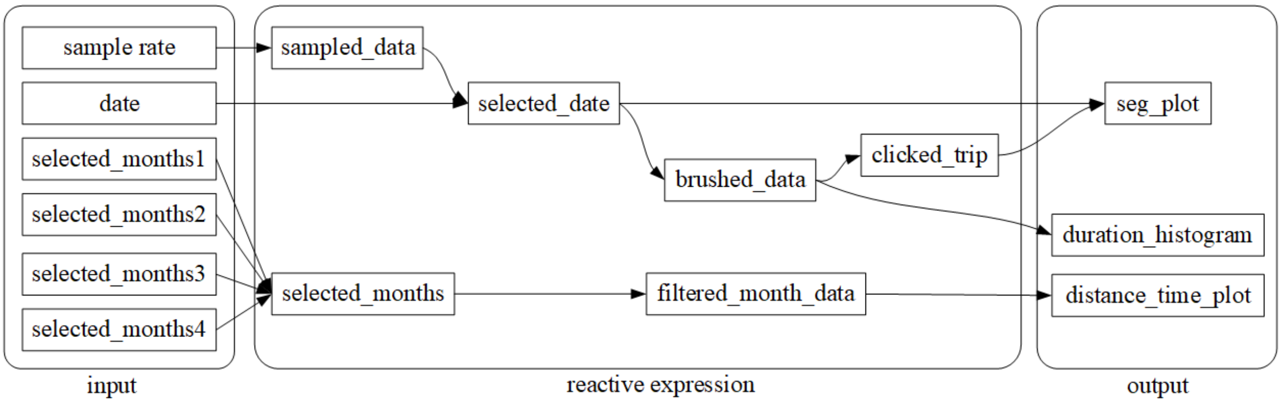
    1. **Data Preparation**: The dataset is divided into multiple Parquet files, each representing a portion of NYC Green Taxi trip data from 2015. I used the arrow package to read each file into a list, selecting key columns like pickup time, dropoff time, trip distance, and fare amount. All these individual datasets were then merged into a single data frame (combined_green_data).
    2. **Interface Design**:

- `Sample Rate Slider`. Since the dataset is large, I added a sample rate adjustment slider based on a logarithmic scale. This allows the user to control the sample size, balancing interactivity with data size. This reduces server load and improves app responsiveness.
- `Date Range Selection`. I included a date range input to allow users to select a start date and a end date, to filter trips based on pickup and dropoff times. The filtered data should meet the requirement: pick-up time ≥ start date ∧ drop-off time ≤ end date.
- `Month Selection Checkboxes`. It offers the option to select specific months of data via checkboxes. Users can explore the distribution of trip distances and pickup times for the chosen months, displayed on a scatter plot with an overlaid histogram.

3. **Data Processing for each Plot**: There are also data preparation steps before ploting each graph, such as filtering data points of interests. More details can be found in the R script.
4. **Style Customizations and Layout**:

- The `theme_minimal()` from `ggplot2` was applied to ensure a clean and modern look across all visualizations.
- I manually set the font size of the axis labels and legend items to improve readability.
- I used `fluidRow` and `column` functions to organize the interface layout.
- I use `scale_color_brewer(palette = "Set1")` and `scale_fill_brewer(palette = "Set1")` to set the color scheme of the "distance_time_plot". I use `scale_color_viridis_c(option = "viridis", direction = -1)` to set the color when visualizing the trip charges.

- **Question 3**: What is the reactive graph structure of your application?
  - I used multiple types of dynamic queries, including through UI or graphical inputs. The reactive graph structure is shown below:


sample rate
date
selected_months1
selected_months2
selected_months3
selected_months4
input
sampled_data
selected_date
brushed_data
clicked_trip
selected_months
filtered_month_data
reactive expression
seg_plot
duration_histogram
distance_time_plot
output