CY-Truck Project: Advanced Data Analysis System for Logistics Management

N.Guihot
E.Szpoper
N.Rayée

Contents

Coding the Future of Transport

1 Introduction

The **CY-Truck** project is designed to revolutionize data processing in logistics. The team includes N. Guihot, E. Szpoper, and N. Rayée, each with distinct roles contributing to the project's success.

1.1 About the project

The CY Truck project, led by Eva Ansermin and Romuald Grignon, involves creating a program to manage the logistics of a national road transport company. This project addresses the challenge of manually processing large and varied logistics data. The program is designed to analyze a data file and generate graphs summarizing the company's activities.

1.1.1 Development of a Shell Script and a C Program

The Shell script, along with a C program for performance operations, analyzes the data file and creates graphs.

1.1.2 Data Management and Processing

The program manages a CSV file containing road trip data, including 'RouteID', 'StepID', 'Departures', 'Arrivals', 'Distance', and 'DriverNames'.

1.1.3 Graph Creation using GnuPlot

After processing the data, the Shell script uses GnuPlot to create graphs illustrating various aspects of the data, such as drivers with the most trips, longest distances traveled, longest trips, and most traversed cities.

1.1.4 Organization and Storage of Data and Results

Input data, programs, graphs, and intermediate files are organized into specific folders for better management.

1.1.5 Flexibility and Scalability

The project is designed to be scalable, with options for enhancement and adaptation to specific needs.

2 Getting Started

This project requires several tools and packages to be installed on your system. Ensure you have the following prerequisites installed before proceeding with the project setup.

2.1 Prerequisites

- GnuPlot is used for generating graphs from data. Install it via your system's package manager. For example, on Ubuntu:

```
sudo apt-get install gnuplot
```

- Make is a build automation tool that automatically builds executable programs and libraries. Install it as follows:

```
sudo apt-get install make
```

- ImageMagick is a software suite to create, edit, compose, or convert bitmap images. It can be installed with:

```
sudo apt-get install imagemagick
```

Ensure that all these tools are correctly installed on your system. They are essential for running and managing various aspects of the project, from data processing to image manipulation.

2.2 Installation

- Clone the repo

```
git clone https://github.com/guinat/Projet_CY_Truck_preIng2_2023_2024.git
```

- After performing a git clone of the repository, navigate to the project directory with the following command:

```
cd Projet_CY_Truck_preIng2_2023_2024
```

3 Usage

This section guides you through the steps to execute and use the CY Truck program. The program is executed via a Shell script that processes a CSV file and generates various outputs based on the options provided.

3.1 Preparing the Data File

Ensure that you have the CSV file containing the road trip data prepared and ready for use. This file should include the following details, listed in order: 'RouteID', 'StepID', 'Departures', 'Arrivals', 'Distance', and 'Driver Names'.

3.2 Running the Script

- Before running the script, ensure that your main script has the necessary permissions to be executed. You can grant execute permission using the following command in your terminal

```
chmod +x main.sh
```

- To execute the program, use the following command format in your terminal. Replace [path_to_csv_file] with the actual path to your CSV file and [option] with the desired operation option.

```
./main.sh [path_to_csv_file] [option]
```

3.3 Understanding the Options

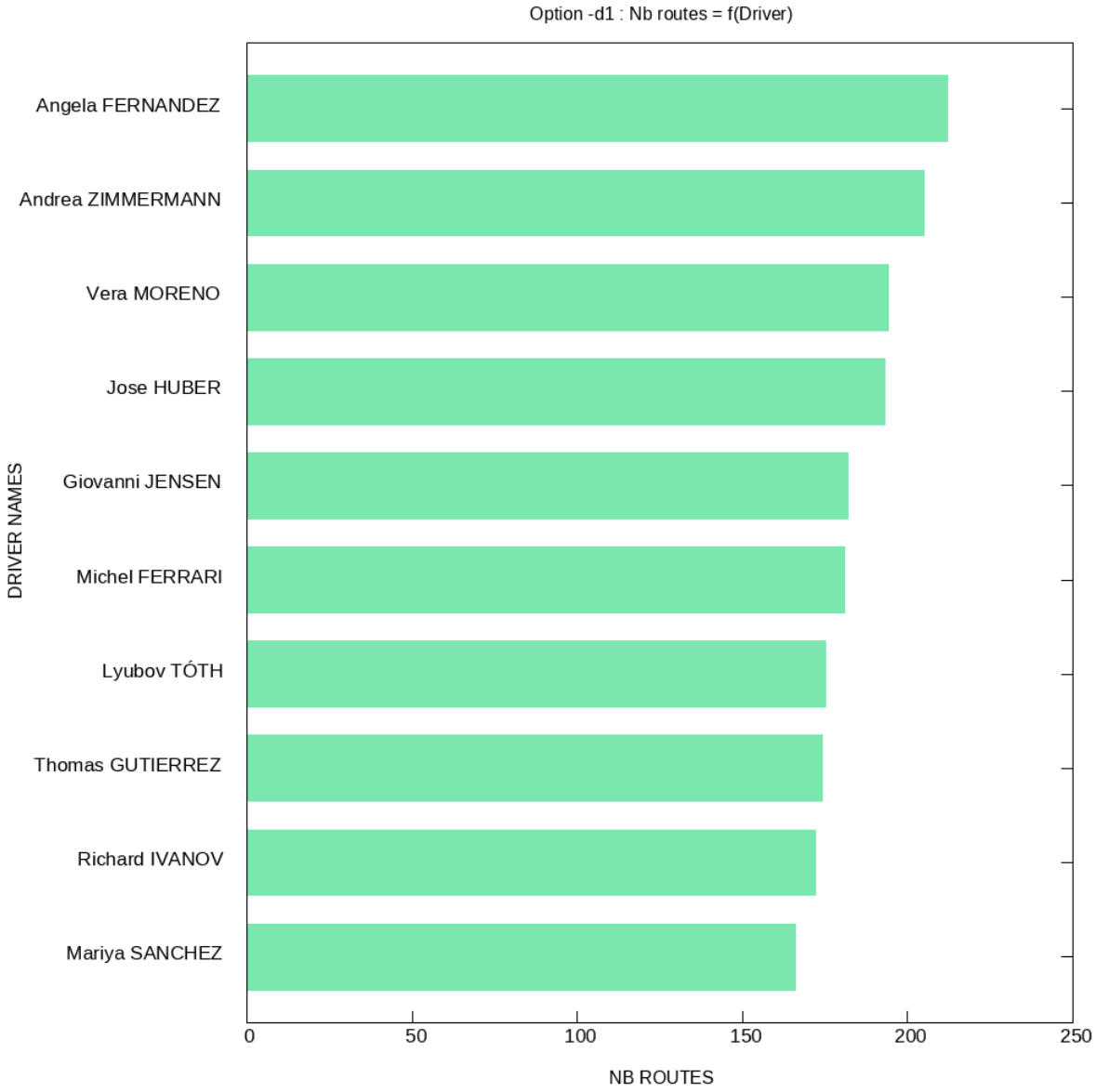
The [option] in the command represents various operations you can perform with the script. Here is the list of available options:

Option: -h

This option displays a help message explaining the available options for the script. It's useful for users who need guidance on how to use the script and what parameters to input.

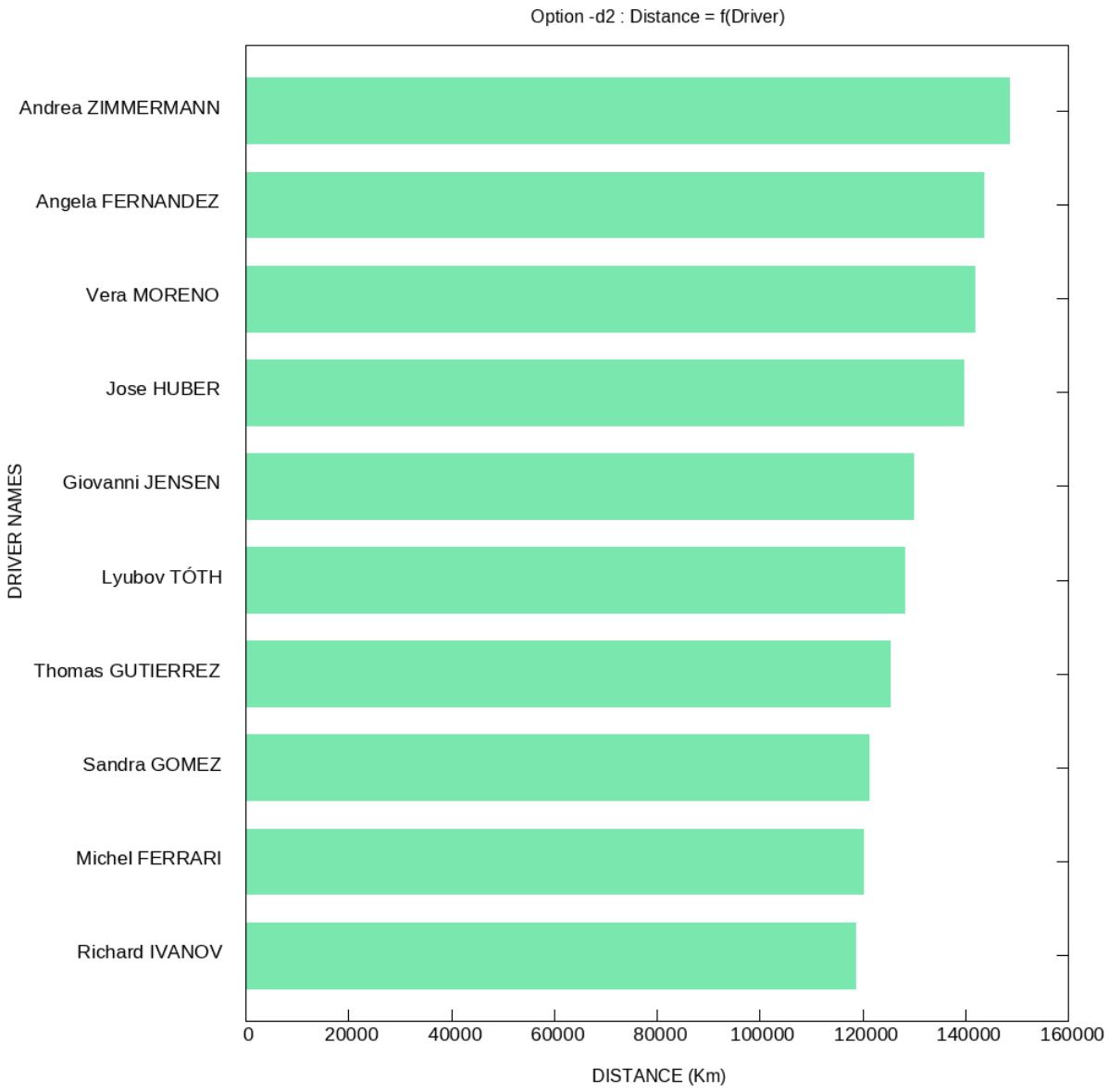
Option: -d1

This option is for processing and generating a horizontal histogram graph showing the top 10 drivers with the most trips. It sorts the drivers by the number of trips they have made in descending order and displays this information graphically.



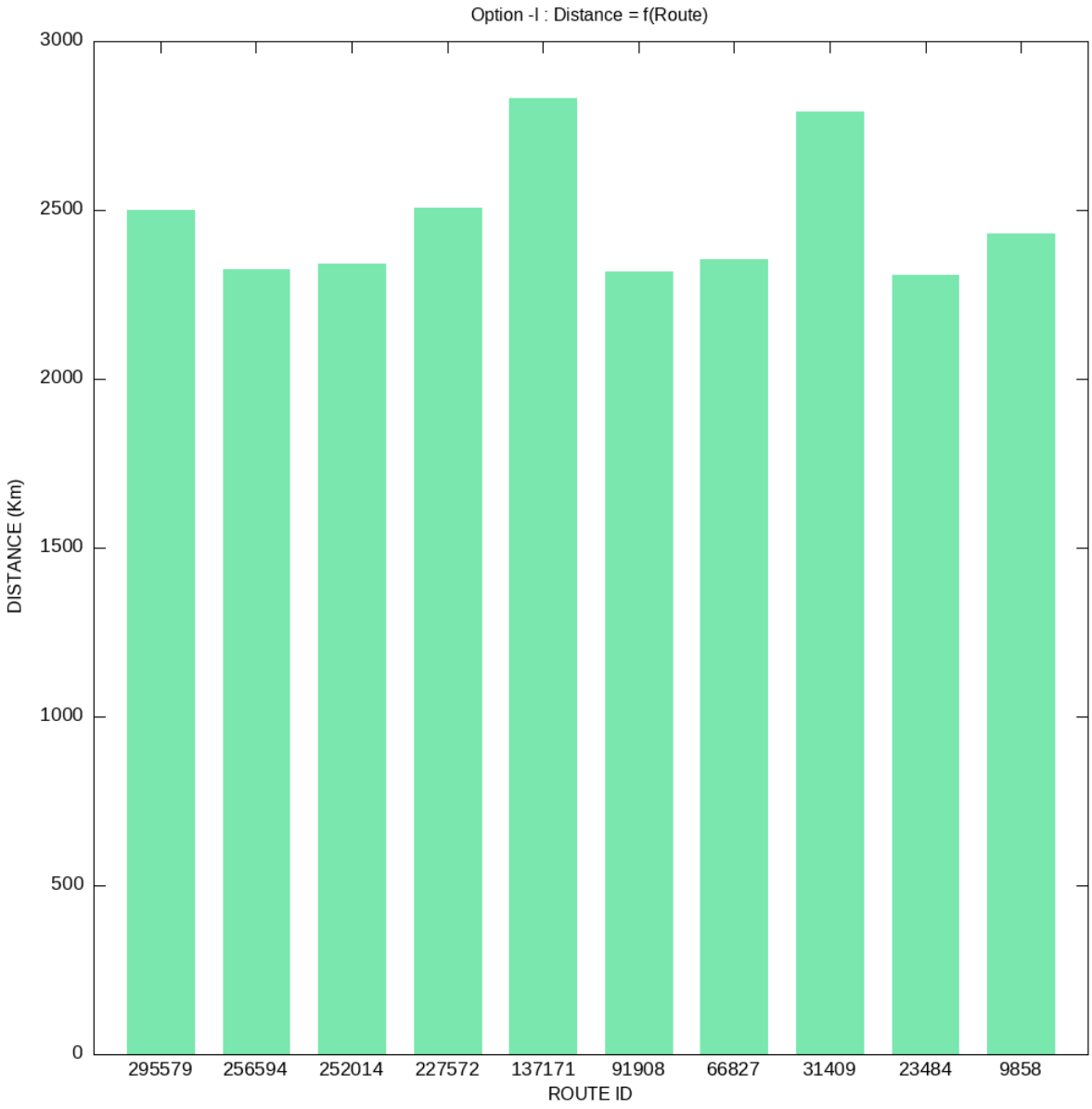
Option: -d2

When using this option, the script calculates the total distance traveled by each driver and generates a horizontal histogram graph. It focuses on the top 10 drivers who have covered the longest distances, sorted in descending order of distance traveled.



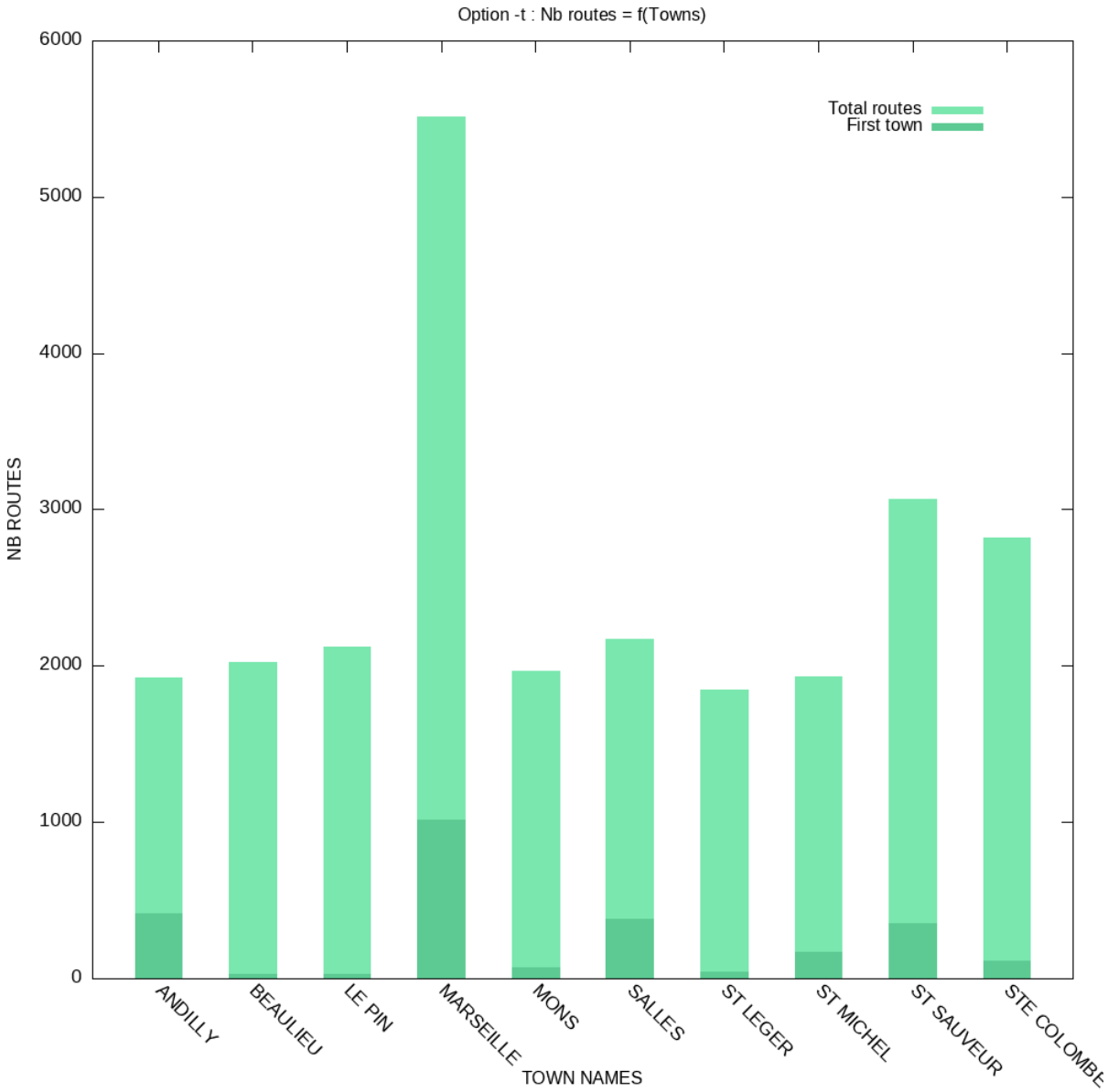
Option: -l

This option identifies the 10 longest trips based on total distance. It processes the data to sum up the distance for each trip and then generates a vertical histogram graph, sorting the results by trip identifier in ascending order.



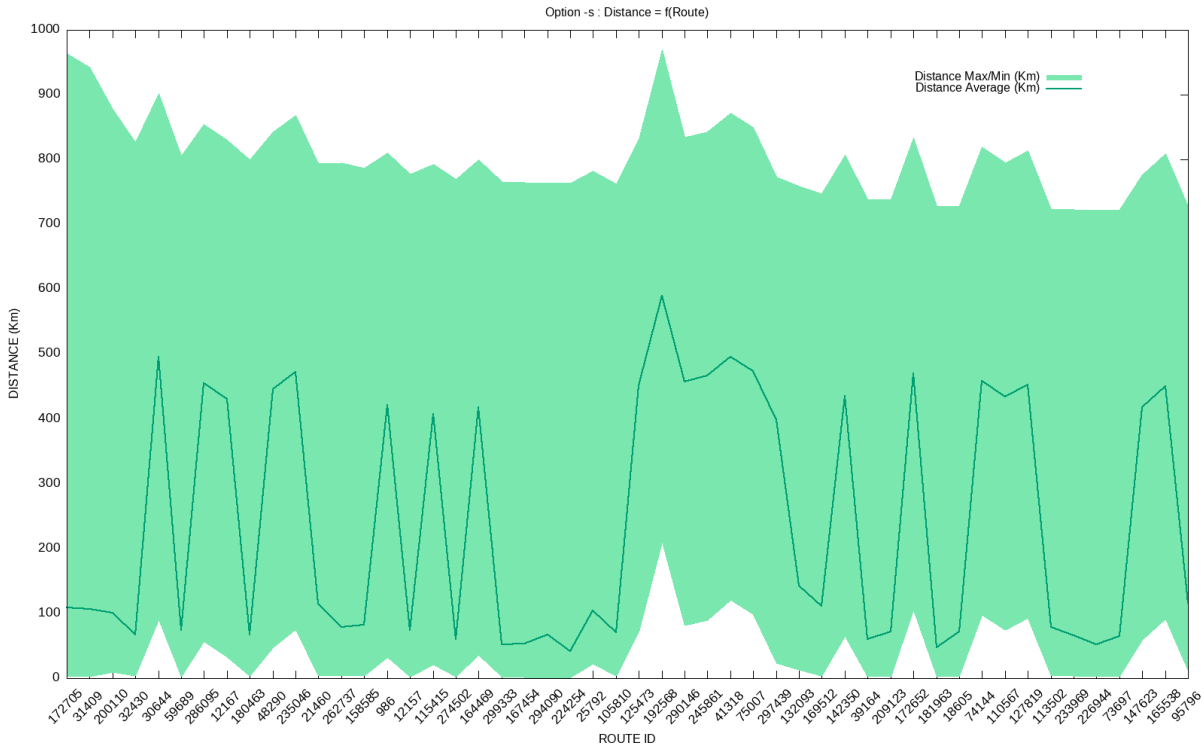
Option: -t

When selected, this option counts the number of trips passing through each city and the number of times these cities are starting points of trips. It then creates a grouped histogram graph for the top 10 cities, sorted alphabetically, showing the total number of trips through each city and the number of times they are trip starting points.



Option: -s

This option is designed to gather statistics on the stages of each trip, such as minimum, maximum, and average distances. It creates a min-max-average curve graph, displaying these values for the top 50 trips, sorted by the difference between maximum and minimum distances in descending order.



4 Task Allocation

4.1 N. Guihot - Project Structuring and C Programming

- **Resource Provision**: Identifying and providing essential tools and resources.
- **Task Distribution**: Assigning tasks based on skills and ensuring a balanced workload.
- **Performance Optimization**: Enhancing the efficiency of the C program in execution and memory usage.
- **Data Processing for T and S**: Implementing and optimizing data processing for project parts T and S.
- **Version Management and Documentation**: Setting up version control using Git and creating technical documentation.
- **Data Security**: Implementing secure practices for data handling and storage.
- **Project Report Writing**: Preparing a detailed project report on task distribution, project timeline, and application limitations.

4.2 E. Szpoper - Main Script Management

- **Data Processing for D1 and D2**: Implementing and optimizing data processing for project parts D1 and D2.
- **Script Component Functionality Assurance**: Ensuring effective operation of all script components.
- **Team Meeting Planning and Organization**: Scheduling regular meetings for project updates and planning.
- **Shared Resource Management**: Managing access to shared resources like documents and software tools.
- **Test Automation**: Developing scripts for automated testing of script components.
- **Robustness Testing**: Ensuring the program handles errors and unexpected data inputs effectively.

4.3 N. Rayée - GnuPlot and Graphical Generation

- **Graph Creation for Each Treatment**: Designing and generating graphs for each data treatment.
- **Data Visualization Using Temporary Data Files**: Creating visualizations from interim data files.
- **Documentation and Archiving**: Documenting and archiving all materials, including code and reports.
- **Time and Priority Management**: Using project management tools for tracking deadlines and prioritizing tasks.
- **Code Quality Assurance**: Establishing coding standards and conducting code reviews.
- **Code Review Organization**: Organizing sessions to review code for cleanliness, documentation, and best practices.

- **Delivery Planning and Deadlines**: Setting and adhering to a timeline for different project phases.

5 Project Timeline

5.1 Project Organization

Our project followed a rigorous and structured approach, emphasizing daily progress and continuous communication. We committed to making near-daily deposits to our Git repository, ensuring a consistent and trackable development process. Additionally, we held daily calls via Discord or WhatsApp to discuss progress, address immediate issues, and plan for upcoming tasks. This regular rhythm of coding and communication was crucial for maintaining project momentum and aligning team efforts.

Date	Activity
Day 1: Dec 19	Initial project setup - First commit.
Day 2: Dec 20	Data processing update for D1, performance optimization, and color addition.
Day 3: Dec 21	Implementation of alert management features.
Day 4: Dec 22	Enhancement of graph generation for D1.
Day 5: Dec 23	Implementation of -d1 option.
Day 6: Dec 24	Addition of new features: options processing, D2, and L treatments.
Day 7: Dec 25	Start of -t option development.
Day 9: Dec 27	Completion of -t option.
Day 10: Dec 28	"Almost finished" - Finalizing features.
Day 11: Dec 29	Final additions and adjustments.
Day 12: Jan 24	Added a data file verification function.
Day 13: Jan 25	Updating the pdf (TODO: Modify processing -t)

6 Functional Limitations

6.1 Unimplemented Features

—

6.2 Operational Issues

While the project has successfully met all the requirements of the specifications as of December 31, 2023, we are continuously working on improving the optimization of treatments D1, D2, and L. These enhancements are focused on increasing efficiency, reducing execution time, and improving memory usage. The current implementation meets the functional requirements, but we aim to achieve even better performance and reliability in these areas.

7 Conclusion

This document outlines the CY-Truck project's task distribution, development timeline, and functional limitations, providing a clear view of our project management and technical challenges.