

**Note: To run the program, follow the steps on README.md.**

### **Data Preparation and Setup**

1. Which database system(s) and version(s) are you using? How do we install it/them? (providing a link to official documentation will be enough)
  - a. Install MongoDB Compass (version 1.26) from <https://www.mongodb.com/try/download/compass>
  - b. Install MongoDB Community Server (version 4.45) from <https://www.mongodb.com/try/download/community>
2. How do we download the data you used for your project? Please do NOT submit ALL the data with your code (Submitting a very small portion (< 5 MB) so that we can run the demo might be okay)
  - a. The project data is located in the backend folder and in the data folder in the frontend folder and it's called data.csv. The data.csv is shortened to 5 MB on the main branch and the original data is located in the backend folder on the backend branch with the same data.csv and it is 45 MB.
3. How do we load this data into the database system?
  - a. Start by installing all dependencies listed in section 2, especially MongoDB Compass. Then, navigate to the project directory /frontend/src. From the src folder, run '`node server.js`' to initiate the database. Open your MongoDB Compass application and you will see a new database called 'LATrafficCollisions'. Navigate into the database and select the only table called 'collisions' (you may need to click refresh for it to show up). You will see a large green button to 'Add Data' which will allow you to select the 'data.csv' file from your computer. Make sure to specify the correct data types according to the Schema.js file (there are a couple Number columns and one Date column). You may need to run 'node server.js' again in case the data is not being fetched from the backend when calling from the frontend.
4. Do you have some scripts to do that? If yes, how do we execute them?
  - a. N/A
5. Did you use some tools for loading? If yes, what are they? Provide appropriate details and links.
  - a. N/A
6. If you are benchmarking different database systems, did you make changes to the configurations? If yes, what are they?
  - a. N/A
7. If you are generating your own data, how do we generate it?
  - a. We did not generate our own data. We sanitize an existing database using custom python scripts and excel operations (located on the 'backend' branch in GitHub).

### **Application and Code**

1. Which programming language(s) and version(s) are you using (Python, Java 8, C++, etc.)?
  - a. Python (version 3.0 or higher)
  - b. Javascript (java version 1.8 or higher)

- c. HTML / CSS (version 5)
- 2. List the third-party libraries needed to execute your code and how do we install them (For ex. MySQL/neo4j connector for Python)
  - a. Node.js (<https://nodejs.org/en/download/>)
  - b. React.js (npm install react react-dom --save)
  - c. MongoDB (<https://www.mongodb.com/try/download/compass>)
  - d. Mongoose (npm install mongoose)
  - e. Express.js (npm install express)
  - f. Axios (npm install axios)
  - g. Cors (npm install cors --save)
  - h. Bootstrap (npm install bootstrap --save)
  - i. Pymongo (pip install pymongo)
  - j. Pandas (pip install pandas)
- 3. If you have a GUI, how do we run it?
  - a. To run the React app: **npm start**
  - b. Tableau (no installation needed)
  - c. React-vis (npm install react-vis)
  - d. D3 (npm install d3)
- 4. Anything else we need to know about running your application, like creating virtual envs?

The application does not use any virtual environments and should work on both Mac and Windows. Make sure to import the data into the MongoDB Compass database before running the frontend (using `npm start`).

### Code Documentation and References

1. Did you use some code from GitHub or other sources? If yes, provide the link. If you used some online code, what changes did you make to the code?
  - a. Creating a Pie Chart in d3.js:  
<https://ihsavru.medium.com/react-d3-implementing-a-pie-chart-dc7bf13ff418>. We modified the size of the Pie Chart.
  - b. Creating a Line Graph in d3.js:  
[https://www.d3-graph-gallery.com/graph/line\\_basic.html](https://www.d3-graph-gallery.com/graph/line_basic.html) We modified the axes and data for the Line Graph as well as integrate it into React like the Pie Chart.
  - c. Embedding Tableau sheets to a React app:  
[https://www.youtube.com/watch?v=hc4UCBTACTU&t=14s&ab\\_channel=AndredeVries/](https://www.youtube.com/watch?v=hc4UCBTACTU&t=14s&ab_channel=AndredeVries/). We changed the urls of the sheets to our personal Tableau sheets. Most of the work on the Tableau end was done within the Tableau app, we uploaded our sheets to Tableau Public and used the code from this video tutorial to embed it into our React app.
  - d. Setup MERN (MongoDB, Express, React, Node):  
<https://medium.com/how-to-react/setup-mern-mongodb-express-js-react-js-and-node-js-environment-and-create-your-first-mern-7774df0fff19>. We used this tutorial to learn how to connect our DB to our app and query off it. We made different routes and queries based on the needs of the app.

- e. Tableau level-of-detail Expressions:  
<https://community.tableau.com/s/question/0D54T00000F33yNSAR/calculated-field-for-most-frequently-occurring-value-of-a-dimension>. We used this forum to learn about level-of-detail expressions in Tableau in order for us to get the visualizations we need. We made some adjustments to the LOD expression to fit the visualizations we wanted.
  - f. React-vis simple bar chart:  
<https://www.tuomokankaanpaa.com/blog/how-to-create-a-bar-chart-with-react>. We used this tutorial in order to create a simple bar graph with react-vis. From the end of the tutorial, we flushed out more feature details such as adding inputs, filtering the bar graph, and hovering displays that we researched through many StackOverflow forums.
2. Give a list of files in your submission which are written by you.
- a. App.js
  - b. index.js
  - c. server.js
  - d. BarGraph.js
  - e. BarGraphContainer.js
  - f. BarGraphContainer.css
  - g. BarGraphGenderByMonth.js
  - h. BarGraphHoliday.js
  - i. BarGraphRaceAgeGender.js
  - j. Home.js
  - k. MultiBarGraph.js
  - l. MultiBarGraphContainer.js
  - m. NavBar.js
  - n. collisionRoute.js
  - o. Schema.js
  - p. collisionController.js
  - q. database.py
  - r. quadrantCalculator.py
  - s. Tableau Sheets:
    - i. <https://public.tableau.com/profile/nabil.patel#!/vizhome/LACollisionsHeatmap/Sheet1>
    - ii. [https://public.tableau.com/views/LACollisionsBargraph/Sheet3?:language=en&:display\\_count=y&:origin=viz\\_share\\_link](https://public.tableau.com/views/LACollisionsBargraph/Sheet3?:language=en&:display_count=y&:origin=viz_share_link)
    - iii. [https://public.tableau.com/views/LACollisions/Sheet22?:language=en&:display\\_count=y&:origin=viz\\_share\\_link](https://public.tableau.com/views/LACollisions/Sheet22?:language=en&:display_count=y&:origin=viz_share_link)

Enter Race:White

Enter Min Age:35

Enter Max Age:40

Enter gender:M

Submit

Number of Collisions by Race, Age Group, Gender per Area Name

