Add the basic map base  
A screenshot of a computer

Description automatically generated  
we started by maping traffic counts by merging two excel files given to us,the first file contained the traffic count for specific roads,segments but no coordinates, the second file has the coordinates for each road segment so we created a merged csv:-

Here’s a summary of the steps we took to generate the file with traffic count data and its corresponding X, Y coordinates:

### Steps to Generate the Merged Traffic Count with Coordinates File

1. \*\*Extract and Inspect Traffic Count Data\*\*:

- We started by loading the traffic count data from the CSV file (`H20211092TabMef.csv`). This file contained columns such as road number (`kvish`), segment (`keta`), and traffic counts (`kamut\_kle\_rehev`).

2. \*\*Extract Road Segment Data with Coordinates\*\*:

- We then loaded the road segment data from the Excel file (`H20211092RoadsNonUrban.xlsx`). This file included columns like the road number (`כביש`), segment number (`קטע`), and the corresponding geographic coordinates (`X`, `Y`).

3. \*\*Renaming Columns for Consistency\*\*:

- To facilitate the merge, we renamed the relevant columns in the road segment data to match those in the traffic count data (`kvish` for road number, `keta` for segment number).

4. \*\*Merging the Data\*\*:

- We merged the traffic count data with the road segment data based on the road number (`kvish`) and segment number (`keta`). This allowed us to append the X, Y coordinates to each traffic count entry.

5. \*\*Exporting the Merged Data\*\*:

- Finally, we exported the merged data to a new CSV file (`merged\_traffic\_count\_with\_coordinates.csv`), which now contains traffic counts along with their specific geographic coordinates.

### Resulting File

- The final file (`merged\_traffic\_count\_with\_coordinates.csv`) contains traffic count data with each entry linked to specific X, Y coordinates, making it suitable for spatial analysis in GIS tools like QGIS.

This file can now be used to map and analyze the relationship between traffic counts and road segments, including any further spatial analysis related to traffic accidents or other factors.

CODE

# First, let's rename the relevant columns in the roads\_non\_urban\_data for consistency

roads\_non\_urban\_data.rename(columns={'כביש': 'kvish', 'קטע': 'keta', 'x': 'X', 'y': 'Y'}, inplace=True)

# Now, let's merge the traffic count data with the road segment data based on 'kvish' and 'keta'

merged\_data = pd.merge(traffic\_count\_data, roads\_non\_urban\_data[['kvish', 'keta', 'X', 'Y']], on=['kvish', 'keta'], how='left')

# Display the first few rows of the merged data to verify the merge

merged\_data.head()  
  
  
now that we have x,y data we can visualise the data on the map

A map with many red dots

Description automatically generated  
each point we see on the map is a “radar” that count cars in intervals of 1 hour the whole year  
  
since the accident data has x,y we can plot it as is  
A map of a body of water

Description automatically generated  
we can see that a lot of accidents are on roads we don’t have ‘radars’ for so we cant investigate the relationship between accident and traffic count, we are thinking about what we can do at this point:  
up until this point we dealt only with 2021 data but we have data from 2015 till 2021  
we can take all the accidents and all the traffic count from all years into 2 files and then drop accident on roads we don’t have ‘radars’ for and then see what we can see.  
  
  
Today, we took the next steps in analyzing the relationship between traffic volumes and road accidents by merging traffic count data with accident data. Here's a summary of what we did:

**Steps to Merge Traffic Count Data with Accident Data**

1. **Loading the Datasets**:
   * We began by loading both the traffic count data and the accident data into separate DataFrames. The traffic count data included columns such as road number (kvish), segment (keta), day of the week (yom), and traffic volume (kamut\_kle\_rehev). The accident data included columns like road number (KVISH1 and KVISH2 for the two possible roads involved in an accident), accident severity (HUMRAT\_TEUNA), and geographic coordinates (X, Y).
2. **Renaming Columns for Consistency**:
   * To ensure a smooth merge, we standardized the column names across both datasets, such as renaming HODESH\_TEUNA to hodesh and YOM\_BASHAVUA to yom in the accident data.
3. **Aggregating Traffic Data**:
   * We aggregated the traffic count data by road segment and day of the week to calculate the average traffic volume for each segment (kvish, keta, yom). This gave us the mean traffic volume (kamut\_kle\_rehev) for each road segment on each day of the week.
4. **Merging with Accident Data**:
   * We then merged this aggregated traffic data with the accident data twice: once using KVISH1 and once using KVISH2. This ensured that accidents were matched with the corresponding road segment’s traffic data, regardless of which road column (KVISH1 or KVISH2) contained the matching road number.
   * After merging, we combined both datasets into a single DataFrame that includes accidents along with the average traffic volume for the relevant road segment and day of the week.
5. **Exporting the Merged Data**:
   * We exported the final merged dataset to a CSV file (merged\_traffic\_accidents\_for\_qgis.csv). This file now includes all relevant accidents with their corresponding average traffic volume, ready for visualization in QGIS.

**Resulting Insights and Next Steps**

* The exported file now allows us to visualize the relationship between traffic volume and accident severity on a map. Each point represents an accident, and the associated traffic count data provides insight into whether high or low traffic volumes correspond to more severe accidents.
* Moving forward, we plan to visualize this data on a map in QGIS to identify any spatial patterns or correlations between traffic volume and accident severity.
* By overlaying the accident data with traffic counts, we aim to determine whether higher traffic volumes correlate with more severe accidents or if other factors might be at play.

A screenshot of a computer screen

Description automatically generatedeach point is an accident that has the average traffic count for the day of the accident.  
  
next we created a heat map  
A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Heat map of the areas with traffic counts

Now we will create another heatmap of accident severity  
  
  
25/8  
Here’s a detailed step-by-step explanation of the work we accomplished in the notebook, tailored for your colleagues:

### Step 1: \*\*Understanding the Data Structure\*\*

- \*\*Objective:\*\* We started by analyzing the structure of the accident and traffic count data from various years to determine how to standardize and merge them.

- \*\*Analysis:\*\* Each dataset had different columns, some of which needed renaming or recalculating to create a consistent format across all years. We also identified important columns such as `kvish`, `yom`, and `kamut\_kle\_rehev` in the traffic data, and `kvish1`, `kvish2`, and `yom\_bashavua` in the accident data.

### Step 2: \*\*Standardizing the Data\*\*

- \*\*Accident Data:\*\*

- \*\*Renaming Columns:\*\* We renamed the accident columns for consistency. For example, `yom\_bashavua` was mapped to `yom`, ensuring uniformity across datasets.

- \*\*Uniform File Creation:\*\* Created a unified dataset for accidents across all years by aligning the columns and data types, ensuring that all entries were comparable.

- \*\*Traffic Data:\*\*

- \*\*Aggregating Data:\*\* We aggregated the traffic data to calculate the average vehicle count (`kamut\_kle\_rehev`) for each road segment (`kvish`) on every day of the week (`yom`). This aggregation was done across all years, ignoring the month to ensure generalization.

- \*\*Uniform File Creation:\*\* Like the accident data, we created a standardized file for traffic counts, consolidating data across all years.

### Step 3: \*\*Merging Accident and Traffic Data\*\*

- \*\*Objective:\*\* To merge the standardized accident data with the aggregated traffic data so that each accident record has the corresponding average traffic count for the road and day of the week.

- \*\*Method:\*\*

- \*\*Matching Logic:\*\* We attempted to match each accident record’s `kvish1` or `kvish2` with the corresponding `kvish` in the traffic data. The match was based on the road (`kvish`) and the day of the week (`yom`).

- \*\*Result:\*\* The result was a new dataset where each accident record included an additional column for the average traffic count on that road and day across all years.

### Step 4: \*\*Data Cleaning\*\*

- \*\*Objective:\*\* To ensure data quality, we cleaned the merged dataset by removing any accident records that did not have a corresponding traffic count.

- \*\*Outcome:\*\* This step resulted in a clean, final dataset where every accident had an associated average traffic count, ensuring that our analysis would be accurate and reliable.

### Step 5: \*\*Validation using QGIS\*\*

- \*\*Objective:\*\* To validate our data processing and merging, we imported the final dataset into QGIS.

- \*\*Validation:\*\* By visualizing the data on a map, we could check the spatial distribution of accidents and traffic counts, ensuring that the merging process was correctly performed and that the data made sense geographically.

### Files Exported:

1. \*\*cleaned\_accident\_data\_no\_null\_columns.csv:\*\* This file contains the standardized and cleaned accident data across all years.

2. \*\*cleaned\_traffic\_counts\_no\_null\_columns.csv:\*\* This file contains the aggregated traffic counts across all years, showing the average vehicle count per road and day of the week.

3. \*\*accident\_data\_with\_traffic.csv:\*\* This final file merges the accident data with the average traffic counts, providing a comprehensive view of accidents with contextual traffic information.

A map of israel with different colored squares

Description automatically generated  
the map of all the accident where we have traffic count across all years BLACK IS FATAL GREEN IS EASY RED IS SEVERE  
  
next we split the data into fatal,severe,easy accident In three layers the fouth layer had them all comined  
we used the comined layer to make a heat map for traffic infested roads  
and a heatmap for easy accidents locations  
and a heatmap for sever accidents locations

and a heatmap for fatal accidents locations  
  
and we looked at the traffic heatmap overlayed on each of the other heatmaps   
we understood that easy accidents happen everywhere no matter how much traffic there is which makes sense cause fender benders happen everywhere  
we noticed sever accident happen more often when the roads are clearer and fatal accident happen in roads that are far and have few traffic but also happen in roads that have much traffic too..  
the data doesn’t show a lot of correlation but further steps can be taked in order to make the relationship cleared by mapping out intersections and roundabout and traffic lights but it wasn’t possible because of the project scope