Chicago Car Crash Incidents Analysis

Firass Elhouat

Introduction

In this project, the primary aim is to conduct a comprehensive analysis of car crashes within the city of Chicago, Illinois. This involves integrating multiple data sets that contains the details of the crash, vehicles involved, driver/passenger/pedestrian information such as demographics and severity of the injuries.The merging process will be done by finding a common attribute, which in this case will be the

CRASH_RECORD_IDs which are present in all three data sets. The analysis will further allow us to explore the various factors that contribute to these crashes, such as crash type, passenger demographics, vehicle class (high-end, low-end, etc), this of course is opinion's based, as some may categorize a certain car in a different class, based on public knowledge or their own opinion.

Key Findings To Investigate & Methods:

In terms of the questions I plan to answer, it will primary focus on exploring the data, and using various methods by identifying which variable need further investigation, and creating new variables based on the existing ones. This will involve analyzing patterns, relationships, and trends to determine which factors are most relavant for deeper analysis. Since our data includes coordinates and street names, we have the opportunity to visually map the vehicle crash occurrences. This will involve merging our clean dataset with a shapefile for spatial analysis.

Beyond analyzing specific streets within Chicago, our focus will expand to understanding how trends vary across different communities. By visualizing these patterns, we can gain deeper insights into the geographic distributions of these crashes and identify areas may be concerning, depending on the fatality rate. Lastly, the way this would be structured, would require merging and process the data in order to answer these questions while also providing the methods used in order to achieve the results. Thiese questions include;

- Investigating Vehicle Age Distribution

- Most Danagerous Streets By Vehicle Crashes
- Crashes by Age and Speed Limit
- Impact Of Driver's Condition and Vehicle Class
- Crash Frequency By Days and Months
- Visual Mapping

Data Info

In this section, often we want to describe what is our data, and where did we source it from. Firstly,

1. Traffic_Crashes_Crashes: Contains overall details regarding the crash such as data of crash, traffic control condition and sign, weather and lighting conditions, crash type, road condition, number of units involved, injuries and fatalities reported, location information such as coordinates, street name, direction and number. Other columns include whether it was a hit and run, damage in USD value, posted speed limit, time, month and day of week which the crash had occurred.

```
 [1] "CRASH_RECORD_ID"              "CRASH_DATE_EST_I"
 [3] "CRASH_DATE"                   "POSTED_SPEED_LIMIT"
 [5] "TRAFFIC_CONTROL_DEVICE"       "DEVICE_CONDITION"
 [7] "WEATHER_CONDITION"            "LIGHTING_CONDITION"
 [9] "FIRST_CRASH_TYPE"             "TRAFFICWAY_TYPE"
[11] "LANE_CNT"                     "ALIGNMENT"
[13] "ROADWAY_SURFACE_COND"         "ROAD_DEFECT"
[15] "REPORT_TYPE"                  "CRASH_TYPE"
[17] "INTERSECTION_RELATED_I"       "NOT_RIGHT_OF_WAY_I"
[19] "HIT_AND_RUN_I"                "DAMAGE"
[21] "DATE_POLICE_NOTIFIED"         "PRIM_CONTRIBUTORY_CAUSE"
[23] "SEC_CONTRIBUTORY_CAUSE"       "STREET_NO"
[25] "STREET_DIRECTION"             "STREET_NAME"
[27] "BEAT_OF_OCCURRENCE"           "PHOTOS_TAKEN_I"
[29] "STATEMENTS_TAKEN_I"           "DOORING_I"
[31] "WORK_ZONE_I"                  "WORK_ZONE_TYPE"
[33] "WORKERS_PRESENT_I"            "NUM_UNITS"
[35] "MOST_SEVERE_INJURY"           "INJURIES_TOTAL"
[37] "INJURIES_FATAL"               "INJURIES_INCAPACITATING"
[39] "INJURIES_NON_INCAPACITATING"  "INJURIES_REPORTED_NOT_EVIDENT"
[41] "INJURIES_NO_INDICATION"       "INJURIES_UNKNOWN"
[43] "CRASH_HOUR"                   "CRASH_DAY_OF_WEEK"
```

```
[45] "CRASH_MONTH"                    "LATITUDE"
[47] "LONGITUDE"                      "LOCATION"
```

The second dataset;

2. Traffic_Crashes_People: Contains details regarding people involve in the crashes, such person type, demographic information and which seat they were sat in during the crash. Other details includes driver's condition before and after the incident such as vision, condition, and BAC level (Blood Alcohol Content). Furthermore, provides details on the type of injury, hospital location, EMS information and whether cell phone use and other driver actions played a role into the

```
 [1] "PERSON_ID"              "PERSON_TYPE"            "CRASH_RECORD_ID"
 [4] "VEHICLE_ID"             "CRASH_DATE"             "SEAT_NO"
 [7] "CITY"                   "STATE"                  "ZIPCODE"
[10] "SEX"                    "AGE"                    "DRIVERS_LICENSE_STATE"
[13] "DRIVERS_LICENSE_CLASS" "SAFETY_EQUIPMENT"       "AIRBAG_DEPLOYED"
[16] "EJECTION"               "INJURY_CLASSIFICATION" "HOSPITAL"
[19] "EMS_AGENCY"             "EMS_RUN_NO"             "DRIVER_ACTION"
[22] "DRIVER_VISION"          "PHYSICAL_CONDITION"     "PEDPEDAL_ACTION"
[25] "PEDPEDAL_VISIBILITY"    "PEDPEDAL_LOCATION"      "BAC_RESULT"
[28] "BAC_RESULT VALUE"       "CELL_PHONE_USE"
```

The third dataset;

3. Traffic_Crashes_Vehicles: This dataset primarily focuses on the vehicle details that was involved in the crash. This includes the make, model and year of the vehicle, type of vehicle such as personal car, bicycle, motorcycle, and others. The dataset also contains information on what was the vehicle used for the direction it was going on, the maneuver taken prior to the crash, as well as the occupant count. Other information provide includes the first contact point, as well as if a fire occurred and an indicator of whether the vehicle was towed.

```
 [1] "CRASH_UNIT_ID"           "CRASH_RECORD_ID"
 [3] "CRASH_DATE"              "UNIT_NO"
 [5] "UNIT_TYPE"               "NUM_PASSENGERS"
 [7] "VEHICLE_ID"              "CMRC_VEH_I"
 [9] "MAKE"                    "MODEL"
[11] "LIC_PLATE_STATE"         "VEHICLE_YEAR"
[13] "VEHICLE_DEFECT"          "VEHICLE_TYPE"
[15] "VEHICLE_USE"             "TRAVEL_DIRECTION"
[17] "MANEUVER"                "TOWED_I"
```

```
[19] "FIRE_I"                    "OCCUPANT_CNT"
[21] "EXCEED_SPEED_LIMIT_I"      "TOWED_BY"
[23] "TOWED_TO"                  "AREA_00_I"
[25] "AREA_01_I"                 "AREA_02_I"
[27] "AREA_03_I"                 "AREA_04_I"
[29] "AREA_05_I"                 "AREA_06_I"
[31] "AREA_07_I"                 "AREA_08_I"
[33] "AREA_09_I"                 "AREA_10_I"
[35] "AREA_11_I"                 "AREA_12_I"
[37] "AREA_99_I"                 "FIRST_CONTACT_POINT"
[39] "CMV_ID"                    "USDOT_NO"
[41] "CCMC_NO"                   "ILCC_NO"
[43] "COMMERCIAL_SRC"            "GVWR"
[45] "CARRIER_NAME"              "CARRIER_STATE"
[47] "CARRIER_CITY"              "HAZMAT_PLACARDS_I"
[49] "HAZMAT_NAME"               "UN_NO"
[51] "HAZMAT_PRESENT_I"          "HAZMAT_REPORT_I"
[53] "HAZMAT_REPORT_NO"          "MCS_REPORT_I"
[55] "MCS_REPORT_NO"             "HAZMAT_VIO_CAUSE_CRASH_I"
[57] "MCS_VIO_CAUSE_CRASH_I"     "IDOT_PERMIT_NO"
[59] "WIDE_LOAD_I"               "TRAILER1_WIDTH"
[61] "TRAILER2_WIDTH"            "TRAILER1_LENGTH"
[63] "TRAILER2_LENGTH"           "TOTAL_VEHICLE_LENGTH"
[65] "AXLE_CNT"                  "VEHICLE_CONFIG"
[67] "CARGO_BODY_TYPE"           "LOAD_TYPE"
[69] "HAZMAT_OUT_OF_SERVICE_I"   "MCS_OUT_OF_SERVICE_I"
[71] "HAZMAT_CLASS"
```

Methods: Data Merging

In this section, the primary portion focuses on merging the data sets by a common column, in this case will be the CRASH_RECORD_IDs which are present in all three data sets. We first removed duplicates from each dataset based on unique identifiers. This is done by ensuring that each dataset represents a distinct record. As when merging the data without using this step, this leads to redundant information. This ensures that during the merging process, we do not create duplicate entries of a single crash, while also maintaining the one-to-many, and many-to-many relationship which in this case is frequently present in the data we are working with right now.

As part of the questions we wanted to explore in this project, I've only picked out a few columns which look to be interesting to explore, and visualize. The last step of this process, merged crash dataset with the people dataset by CRASH_RECORD_ID, using a left join.

This dataset was then merged with the vehicles dataset "CRASH_RECORD_ID" and "VEHICLE_ID", and using a left join as well. In order to verify that the merging process was successful, I went on to filter the data by selecting any CRASH_RECORD_ID, and verifying that it captures the one-to-many relationship by only selecting the column Make, which in this case it has, we can confirm that the merging process was successful.

```
# A tibble: 2 x 1
  MAKE
  <chr>
1 HONDA
2 HARLEY-DAVIDSON
```

Methods: Data Processing

As part of the questions we aim to answer, it seems that this dataset does not categorize the cars by their class, for instance luxury level, economy, etc. In this section, I aim to look at the frequency of the car makes that have crashed. Doing so, allows me an easy way to output all the car makes, and then start using a str.detect, and start classifying the car makes by a new column named VEHICLE_CLASS, which then classifies them either HIGH-END, MEDIUM-END, LOW-END, Motorbike & Bike, and Other & Unknown.

As part of classifying the car makes, this is mostly based on market perceptions of the car makes, and a few are opinions. As a more accurate way of doing this, would be to only consider the car model, year and furthermore cross referencing this by further researching on whether a specific car makes and the model are indeed "High-End" or "Medium-End", etc. This of course will be touched on, when discussing on the section of future work.

In the next part, I decided to filter the data by looking at two vehicle types: passenger and SUV's, as this is a great way of figuring out the types of car makes that we can use for classifying by vehicle class. The reasoning behind this, is that we are only looking into personal-use vehicles that were recorded in a crash.

By narrowing the focus to passenger and SUV vehicles, we can refine our analysis to better understand the distribution of the vehicles makes within these categories, which are often seen as the most common vehicle types on the road and yield the highest crash records. This is is seen above, where passenger vehicles had a crash count of 572,790, and SUV's had a crash record of 142,510, and in the next part we narrow this part by looking at the car brands that were involved in these crashes.

```
# A tibble: 23 x 2
   VEHICLE_TYPE                crash_count
   <chr>                             <int>
 1 PASSENGER                        572790
```

```
 2 SPORT UTILITY VEHICLE (SUV)      142510
 3 UNKNOWN/NA                        76755
 4 VAN/MINI-VAN                      40021
 5 PICKUP                            30978
 6 <NA>                              20732
 7 TRUCK - SINGLE UNIT               16229
 8 BUS OVER 15 PASS.                 14679
 9 OTHER                             12902
10 TRACTOR W/ SEMI-TRAILER            7743
# i 13 more rows
```

From this section, I filtered the data by VEHICLE_TYPE, for both PASSENGER and SUV, which I then count the frequency of each car brand that was involved in a car crash. This is a way to few the car brands which we would then use to classify in the later section.

In terms of crash occurrences by car brands for passenger vehicles types, the top five include Toyota, Chevrolet, Nissan, Honda, and Ford, which is not surprising as these are often more frequent on the road. While in terms of the car brands for SUV's, the top five include Jeep, Chevrolet, Ford, Toyota, and Nissan.

This part of the analysis not only provides some valuable insights into the car brands most commonly involved in crashes but also streamlines the process of identifying and categorizing these brands by their vehicle class for further investigation. This method would require using str_detect which would help, and categorize by high-end, medium-end, and low-end, while some other types of vehicles would be categorized as either Motorcycles or as other.

```
# A tibble: 5 x 2
  MAKE      frequency
  <chr>         <int>
1 TOYOTA        80385
2 CHEVROLET     69543
3 NISSAN        57083
4 HONDA         51356
5 FORD          46506
```

```
# A tibble: 5 x 2
  MAKE      frequency
  <chr>         <int>
1 JEEP          18511
2 CHEVROLET     17168
3 FORD          16842
4 TOYOTA        11845
5 NISSAN        11118
```

In this section, I decided to create a new variable called VEHICLE_CLASS, which distinguishes car brands by a class level. In general, we have a wide variety of class based on car brands, the year and model, however much of it depends on the MSRP of the specific car and brand placement. For instance, MERCEDES-BENZ & ROLLS ROYCE, are often segmented as a luxury car brand due to the quality and brand image that they have set themselves as.

However, we do know that MERCEDES-BENZ does sell a few cars that fall in a medium end class, while some of their models such as the S-Class model, are often marketed as a luxury lineup, and is often in the same class as a Rolls Royce. In this case, I decided to try and use both market perceptions of these brands, and additionally involve my opinion on how I will classify each brand by the respective classes.

The classes range from "HIGH-END","MEDIUM-END", "LOW-END", "Motorbike & Bike", and "Other & Unknown". In terms of the Other & Unknown class, this is primarily due to time constraints, as certain brands in the list above, fall into a class of semi-trucks, Vans, Trailers, Wagons, Buses, and other industrial used vehicles.

For implementing this, I used the str_detect function to match vehicle brands to the specific classes, along with str_trim to clean brand names and address any inconsistencies. This approach help streamline the process and ensure accurate classifications. After assigning the classes, I focused primarily on a few specific columns from the dataset, ensuring only few relevant columns were included in this section. If additional insights arise during the stage, I can always revisit and re-select additional columns.

```
# A tibble: 947,464 x 24
   vehicle_class    posted_speed_limit physical_condition make              age
   <chr>                         <dbl> <chr>              <chr>           <dbl>
 1 MEDIUM-END                       15 NORMAL             HONDA              45
 2 Motorbike & Bike                 15 NORMAL             HARLEY-DAVIDSON    69
 3 MEDIUM-END                       30 UNKNOWN            FORD               NA
 4 Other & Unknown                  30 NORMAL             <NA>               14
 5 MEDIUM-END                       30 NORMAL             GENERAL MOTORS ~   36
 6 MEDIUM-END                       20 NORMAL             FORD               29
 7 LOW-END                          30 <NA>               SUBARU             NA
 8 LOW-END                          30 UNKNOWN            SUBARU             70
 9 MEDIUM-END                       30 NORMAL             TOYOTA             30
10 MEDIUM-END                       30 NORMAL             MAZDA              NA
# i 947,454 more rows
# i 19 more variables: injuries_fatal <dbl>, injuries_total <dbl>, sex <chr>,
#   crash_type <chr>, road_defect <chr>, weather_condition <chr>,
#   crash_hour <dbl>, crash_day_of_week <dbl>, crash_month <dbl>,
#   crash_date <chr>, roadway_surface_cond <chr>, latitude <dbl>,
#   longitude <dbl>, street_name <chr>, prim_contributory_cause <chr>,
#   vehicle_year <dbl>, driver_action <chr>, vehicle_use <chr>, model <chr>
```

Key Findings To Investigate: Vehicle Age Distribution

In this section, I am mainly configuring and creating new variables that will streamline the process in analyzing trends and patterns in the number of vehicles crashes in Chicago from 2020 to 2024, as well as preparing much of the data in order to provide a more detailed analysis.

In this part, I wanted to look at possibility of creating a Vehicle_Age column, which would take the year of when the car crash occurred and the year of the car. Doing so, will provide us more insight on the distribution of the life span of a vehicle. While this does not take into consideration into the extent of the damage the vehicle had sustained, which would require additional configuration by looking at whether the vehicle had to be towed, and the exact damage sustained.

Before creating this new column, I did notice that that VEHICLE_YEAR summary, contains a max of 9999 year, which does not make any logical sense. In this case, I wanted to view the max year in that column, to which we could then filter out in order to prevent our new variable from being extremely skewed which may impact our analysis. Furthermore, we filtered out further unrealistic years, for instance if a car crashed in 2023, then surely the year of the car should be 2024.

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
   1900    2009    2014    2015    2018    9999  161171
```

```
# A tibble: 144 x 2
   vehicle_year crash_date
          <dbl> <chr>
 1         9999 05/28/2022 05:20:00 PM
 2         9999 01/28/2022 01:55:00 PM
 3         9999 09/18/2022 08:00:00 PM
 4         9999 08/17/2024 12:00:00 PM
 5         9999 04/06/2023 11:45:00 AM
 6         9999 08/02/2023 03:00:00 AM
 7         9999 12/07/2023 03:20:00 PM
 8         9999 06/14/2021 06:50:00 PM
 9         9999 11/21/2021 09:45:00 PM
10         9999 02/01/2021 04:00:00 AM
# i 134 more rows
```

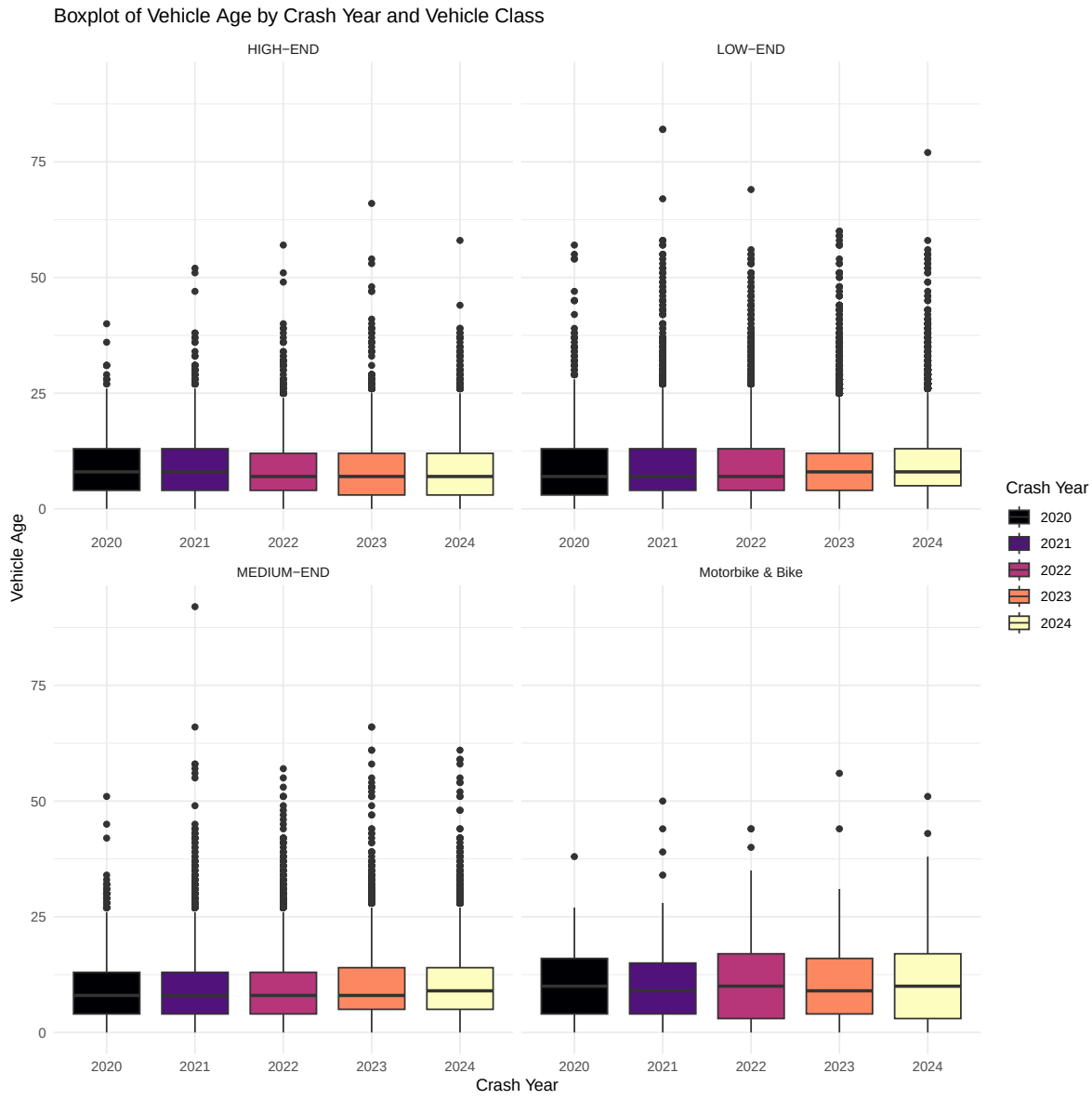
Results: Vehicle Age Distribution

From the results, we can observe that High-end vehicles have an average age of 8.38 years, with the median age 7 years based on the 58670 vehicles we have in this dataset. With outliers ranging all the way to 60+ years.

Low-end have an average age of 8.37 years, with the median age of 8 years based on on 302,308 vehicles, while Medium-end average age of 9.24 years and similar median age of 8 years with 389,287 vehicles. In terms of Motorbike & Bike, have the highest median and average ages, however with only 1,188 observations.

Overall, in terms of the accuracy of these observations, this required a few methods in achieving this, which included verifying the entries, to which I was able to remove vehicle years that exceeded logical sense, as these would have most likley have been input error. For instance, while exploring I found vehicle ages that were below 1929, and ones above 2024, to which this was filtered out.

```
# A tibble: 751,453 x 4
   vehicle_year CRASH_YEAR Vehicle_Age vehicle_class
          <dbl>      <dbl>       <dbl> <chr>
 1         2010       2023          13 MEDIUM-END
 2         2005       2023          18 Motorbike & Bike
 3         2023       2023           0 MEDIUM-END
 4         2000       2023          23 MEDIUM-END
 5         2016       2023           7 MEDIUM-END
 6         2016       2023           7 LOW-END
 7         2016       2023           7 LOW-END
 8         2023       2023           0 MEDIUM-END
 9         2023       2023           0 HIGH-END
10         2015       2023           8 LOW-END
# i 751,443 more rows
```



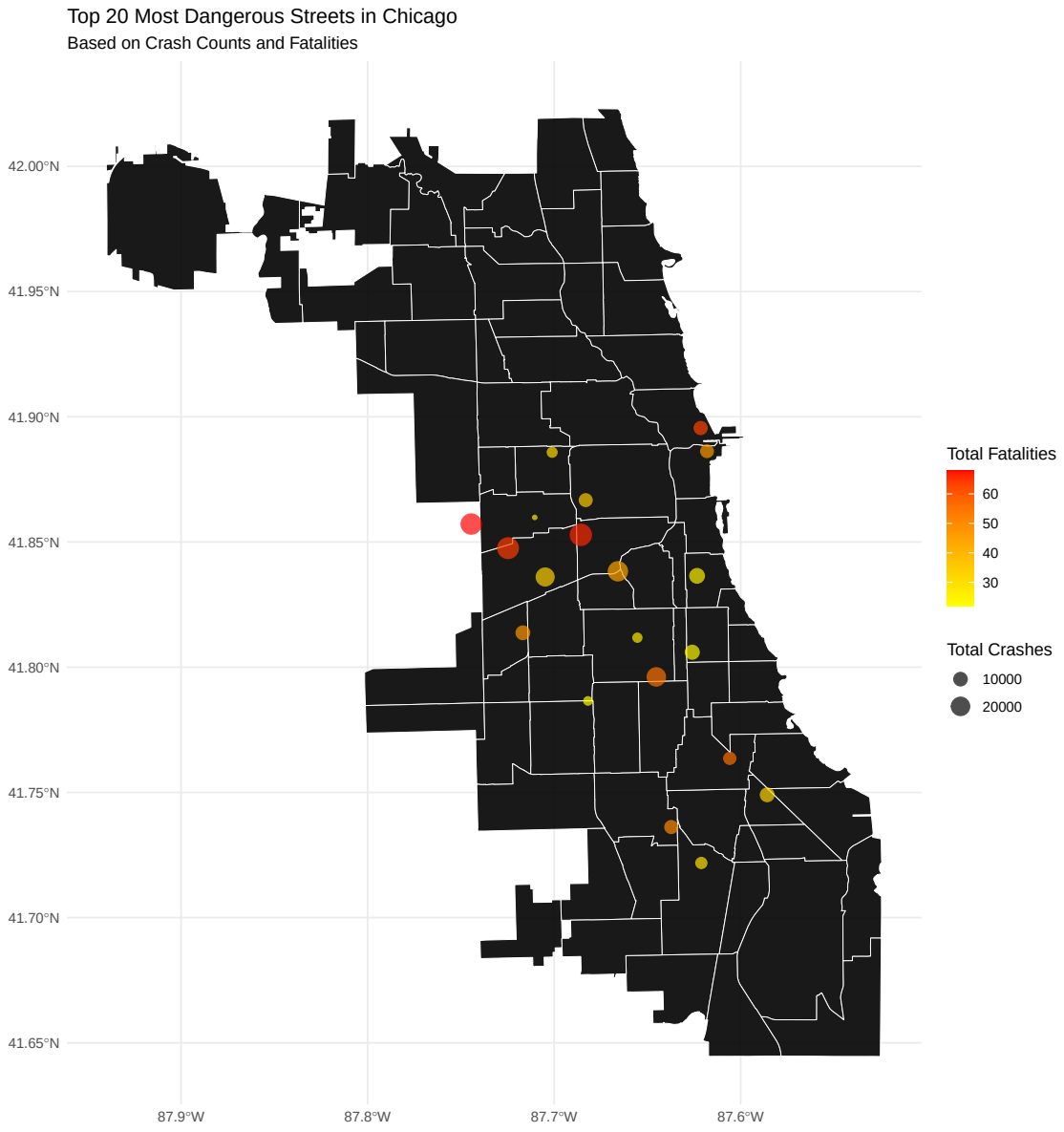

```
# A tibble: 4 x 4
  vehicle_class    mean_vehicle_age median_vehicle_age      n
  <chr>                       <dbl>              <dbl>  <int>
1 HIGH-END                     8.38                  7  58670
2 LOW-END                      8.74                  8 302308
3 MEDIUM-END                   9.24                  8 389287
4 Motorbike & Bike            11.0                  10   1188
```

Key Findings To Investigate: Most Dangerous Streets

- Explore possibility of using a map to better visualize our data below.

  In this section, we explore the range of columns available to us, and we do have a few columns that provides us with the exact coordinates and street names in which these vehicle crashes had occurred. We then counted and summarized the total injuries which were fatal, and order this by the top 20 streets with the highest recorded fatalities between 2020 to 2024. This is the initial step to take before we plot a map, and further add other columns which will aid in the analysis of this dataset.

```
# A tibble: 20 x 5
   street_name       total_crashes total_fatalities latitude longitude
   <chr>                     <int>            <dbl>    <dbl>     <dbl>
 1 CICERO AVE                24932               68     41.9     -87.7
 2 WESTERN AVE               27787               66     41.9     -87.7
 3 PULASKI RD                26297               64     41.8     -87.7
 4 LAKE SHORE DR NB           9318               63     41.9     -87.6
 5 COTTAGE GROVE AVE          7928               56     41.8     -87.6
 6 HALSTED ST                19925               55     41.8     -87.6
 7 87TH ST                    9147               50     41.7     -87.6
 8 ARCHER AVE                10307               47     41.8     -87.7
 9 LAKE SHORE DR SB           8779               47     41.9     -87.6
10 ASHLAND AVE               22094               44     41.8     -87.7
11 ROOSEVELT RD               8966               35     41.9     -87.7
12 KEDZIE AVE                18857               33     41.8     -87.7
13 STONY ISLAND AVE          10340               32     41.7     -87.6
14 LAKE ST                    5305               30     41.9     -87.7
15 95TH ST                    6685               28     41.7     -87.6
16 RACINE AVE                 4631               27     41.8     -87.7
17 HOMAN AVE                  2638               26     41.9     -87.7
18 MICHIGAN AVE              11977               25     41.8     -87.6
19 STATE ST                  10733               24     41.8     -87.6
20 59TH ST                    4096               22     41.8     -87.7
```



Results: Most Dangerous Streets

From the results we found that between 2020 and 2024, CICERO AVE recorded the highest number of total fatalities of 68 and the highest number of recorded vehicle crashes by 24,932. Based on this, we may need to understand why this particular street records such a high number of fatalities, and would other factors need to be considered to fully understand the extent of these occurrences.

Furthermore, the the later section the visual maps will need to be expanded in order to view other factors that may provide further insights, for instance we could explore how these crashes vary by communities in Chicago instead of specific neighborhoods.

Key Findings To Investigate: Crashes by Speed Limit and Age

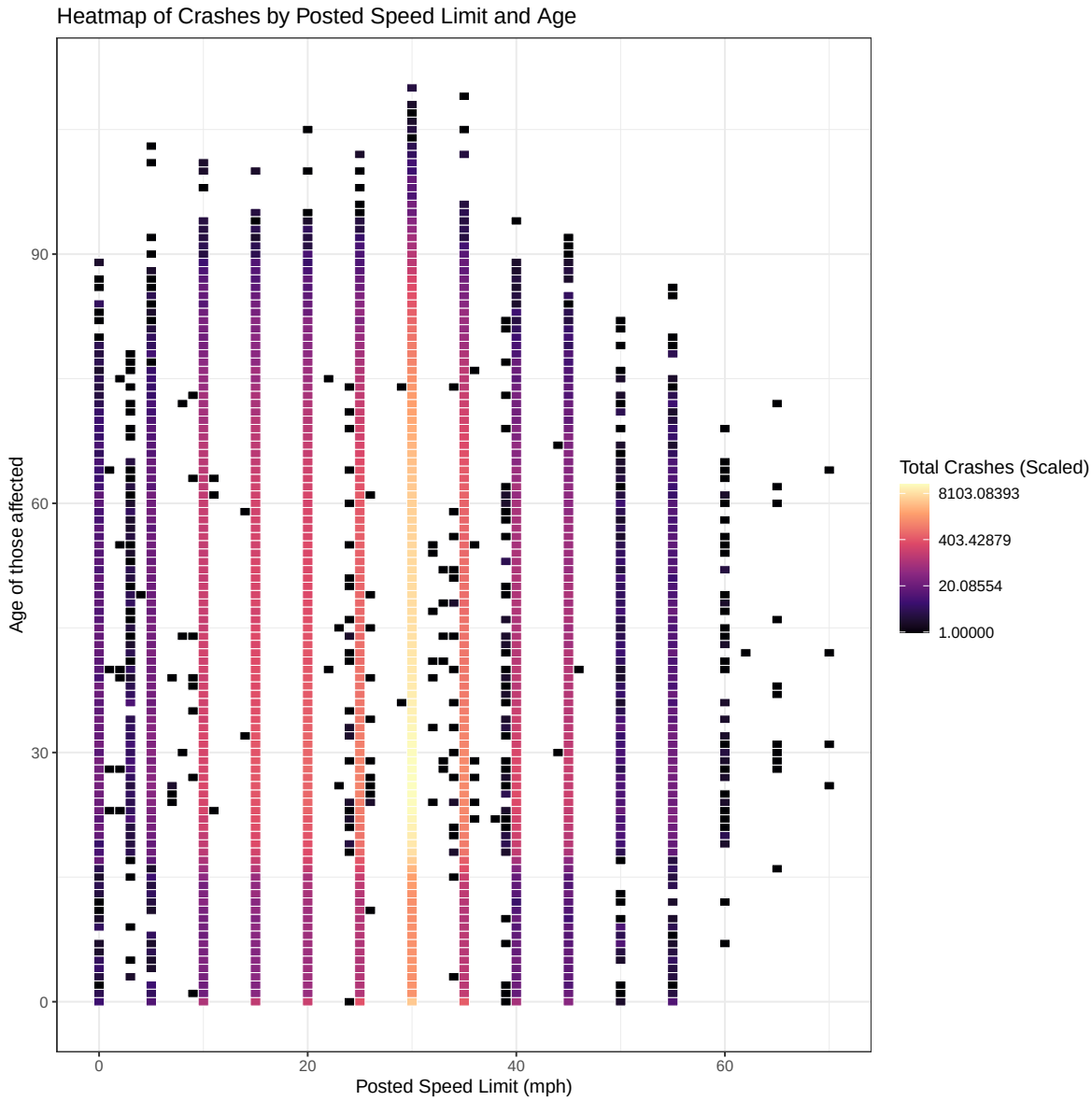
Exploring how the number of crashes differ by the posted speed limit and grouping this by age.

In this case we need first need to look into how we can summarize this data by filtering out any missing values, and exclude any unrealistic ages, as in our previous findings, we had found that some data points were beyond logic. In this case, we would filter missing values in both posted_speed_limit and age, while removing age's below 0.

Results: Crashes by Speed Limit and Age

As a result of this part, we can see that the number of crashes by age and the speed limit is centered right in the middle. It shows us that more vehicles had crashed in a 30 mph speed limit zone, in comparison to the other speed zones. Furthermore, the age of those who were affected by these crashes often ranged in the ages of 30.

```
# A tibble: 6 x 3
  posted_speed_limit   age TOTAL_CRASHES
               <dbl> <dbl>         <int>
1                 30    27         14247
2                 30    26         14141
3                 30    28         14091
4                 30    25         13936
5                 30    29         13912
6                 30    30         13573
```



Key Findings To Investigate: Impact Of Driver's Condition and Vehicle Class

Explore driver's physical condition and vehicle class, and how the average injury differs.

On the next section of the analysis, I wanted to examine how crashes differed based on the driver's physical condition (e.g. fatigue, under the influence) and whether these patterns changed by vehicle class.

We then would summarize the data by vehicle class and physical condition to determine the number of average injury recorded to understand the impact of physical condition on the crash outcomes. This analysis helps determine how the physical condition of the driver prior to the crash affect the average injury and whether the vehicle class plays a role in these patterns. For instance, we may find drivers who drive a more expensive vehicle are more likely to have been impaired or fatigued before the crash.

```
 [1] "NORMAL"                      "UNKNOWN"
 [3] NA                            "FATIGUED/ASLEEP"
 [5] "IMPAIRED - ALCOHOL"          "HAD BEEN DRINKING"
 [7] "EMOTIONAL"                   "REMOVED BY EMS"
 [9] "OTHER"                       "IMPAIRED - ALCOHOL AND DRUGS"
[11] "ILLNESS/FAINTED"             "MEDICATED"
[13] "IMPAIRED - DRUGS"
```

Results: Impact Of Driver's Condition and Vehicle Class

Based on the results, we can see that drivers who drove a medium end car had the highest number of recorded crashes when impaired by alcohol and when fatigued/asleep. Furthermore, we that on average drivers who were on a motorbike or bike had the highest average injury of 0.7500000, this would make sense considering that safety is less common in those kind of vehicles.

```
# Print the result
print(crash_summary_condition)
```

```
# A tibble: 13 x 4
   vehicle_class    physical_condition INJURIES_TOTAL_COUNT AVG_INJURY
   <chr>            <chr>                            <int>      <dbl>
 1 HIGH-END         FATIGUED/ASLEEP                     22      0.341
 2 HIGH-END         IMPAIRED - ALCOHOL                  68      0.411
 3 HIGH-END         IMPAIRED - DRUGS                     6      0.667
 4 LOW-END          FATIGUED/ASLEEP                    201      0.308
 5 LOW-END          IMPAIRED - ALCOHOL                 300      0.454
 6 LOW-END          IMPAIRED - DRUGS                    45      0.577
 7 MEDIUM-END       FATIGUED/ASLEEP                    263      0.353
 8 MEDIUM-END       IMPAIRED - ALCOHOL                 338      0.457
 9 MEDIUM-END       IMPAIRED - DRUGS                    36      0.606
10 Motorbike & Bike IMPAIRED - ALCOHOL                   3      0.75
11 Other & Unknown  FATIGUED/ASLEEP                     16      0.392
12 Other & Unknown  IMPAIRED - ALCOHOL                 156      0.834
13 Other & Unknown  IMPAIRED - DRUGS                    34      0.833
```

Key Findings To Investigate: Crash Frequency By Days and Months

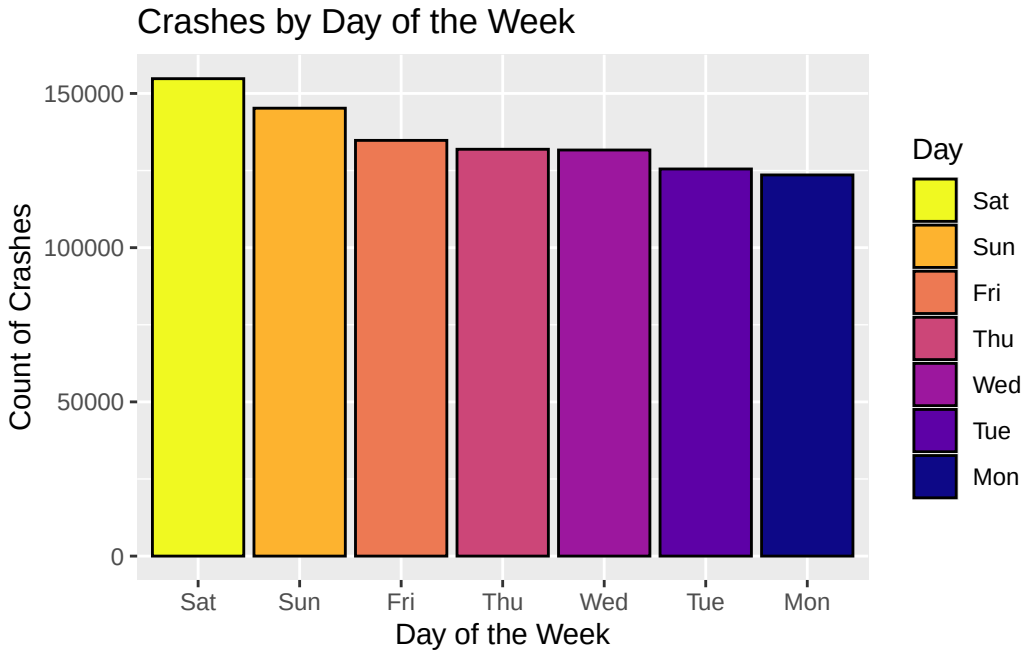
Exploring how crashes vary by days and months of the year.

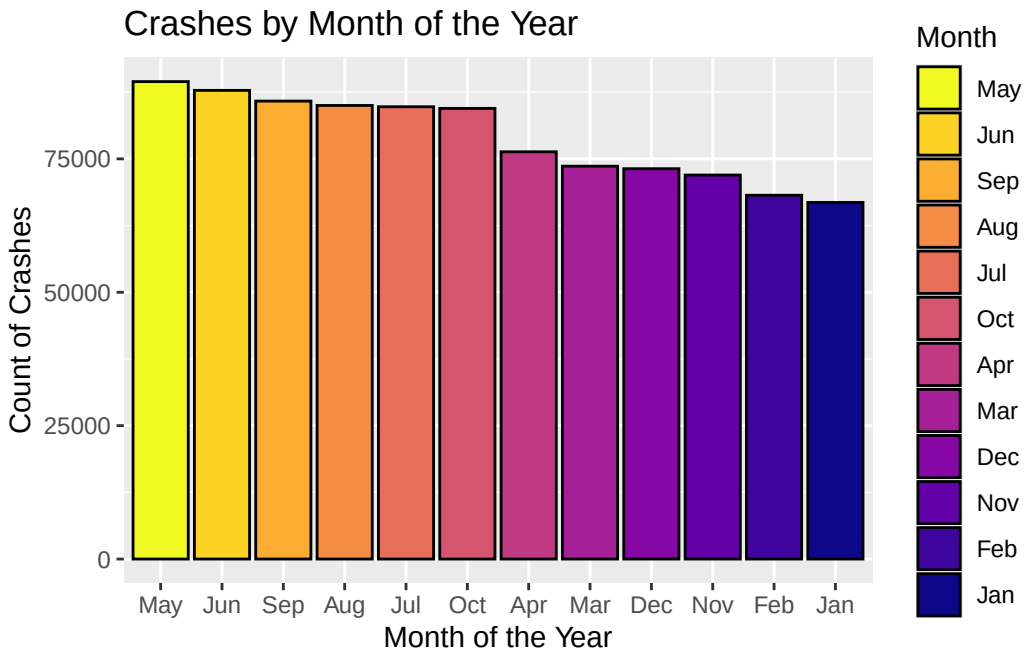
This portion of the study focuses on how vehicle crash occurrences vary across different days of the week and months of the year. This serves as the foundation for further analysis, where we aim to visually map these frequencies by the communities within Chicago. While we may find that crashes are more frequent on Saturdays or during specific months, our goal is to analyze whether these patterns vary across different communities and how do they differ by the years.

Results: Crash Frequency By Days and Months

In the bar plots below, we can observe how vehicle crashes vary by the day of the week and month of the year. For example, between 2020 to 2024, crashes tend to be higher on the weekends, with Saturdays having the highest frequency. This is logical, as more people are out and about during the weekends.

In the second bar plot, an interesting trend emerges, as we can observe more crash incidents occur during the warmer months compared to the colder months. This could be attributed to increased outdoor activities and travel during the warmer seasons, leading to a higher likelihood of accidents.





Key Findings To Investigate: Visual Mapping

In this section, the aim is to creating visual maps of Chicago by the communities to analyze and pinpoint high-risk areas. As these kinds of analysis often contributes to more informed, data-driven decision making for traffic safety and resource allocation, in this case certain areas within Chicago

In our exploration of creating visual maps of these vehicle crash incidents, we want to explore how these occurrences vary depending the severity, primary reasons that lead to these incidents, how these compare by years (2020 - 2024), and how each community varies when apply two layers.

For instance, in our first combined visual map, we are exploring how certain communities differ by crashes and crashes that were fatal. The map reveals patterns in both the total number of crashes and fatal crashes across various communities in Chicago. By comparing these two plots, we can identify areas that experience higher frequencies of crashes, and compare them to when we sum up the frequencies of when they were fatal.

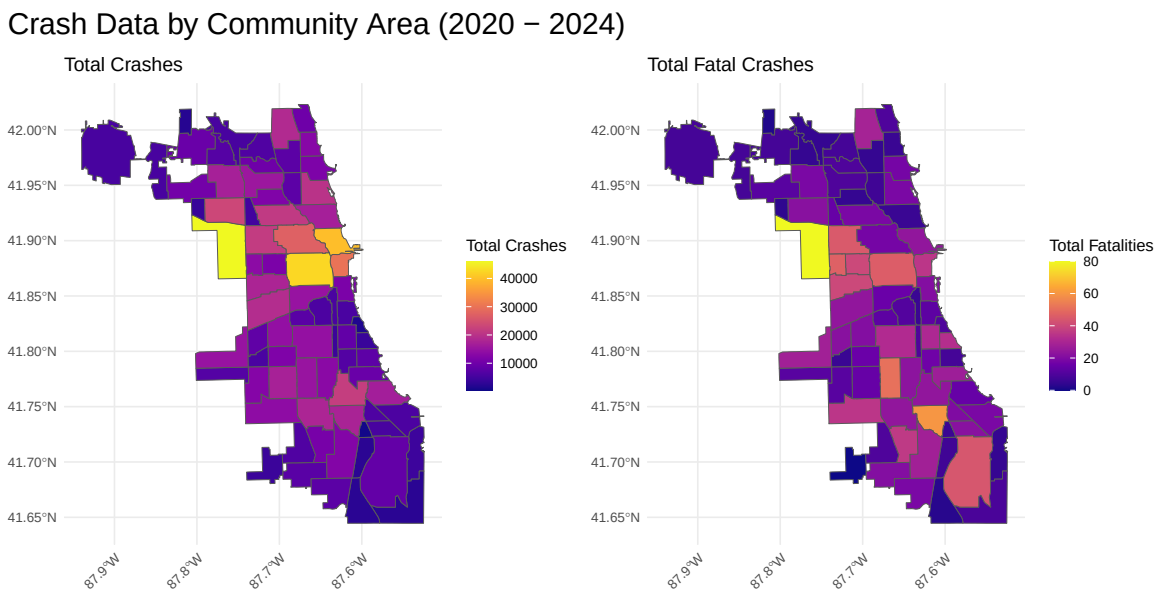
Results: Visual Mapping [Comparison of total and fatal crashes]

In the first map, we can observe that communities like Austin, Near West Side, West Town, Loop, and Near North have the highest total number of crashes between 2020 and 2024. While Austin stands out with the highest total, and three others range in the high 30,000s.

However, when we compare this with the map showing fatal crashes, Austin tops the list with fatalities in the 80s, while the other communities, although still significant, show much lower number. However, Near West Side, is closer to the 40s in fatalities. Interestingly, certain communities that had lower crashes numbers, around the 20,000s, such as Chatham and West Englewood, show higher fatality rates compared to areas with the highest crash counts.

Furthermore, we can see that South Deering, located on the south side of Chicago, has one of the lowest numbers of crashes but a significantly higher fatality rate, with fatalities in the 40s. This emphasizes that areas with fewer crashes might still be prone to more severe outcomes, possibly pointing to specific safety concerns such as driver behaviors, road conditions and traffic enforcement

```
Joining with `by = join_by(community)`
Joining with `by = join_by(community)`
```



Results: Visual Mapping [Primary Contributory]

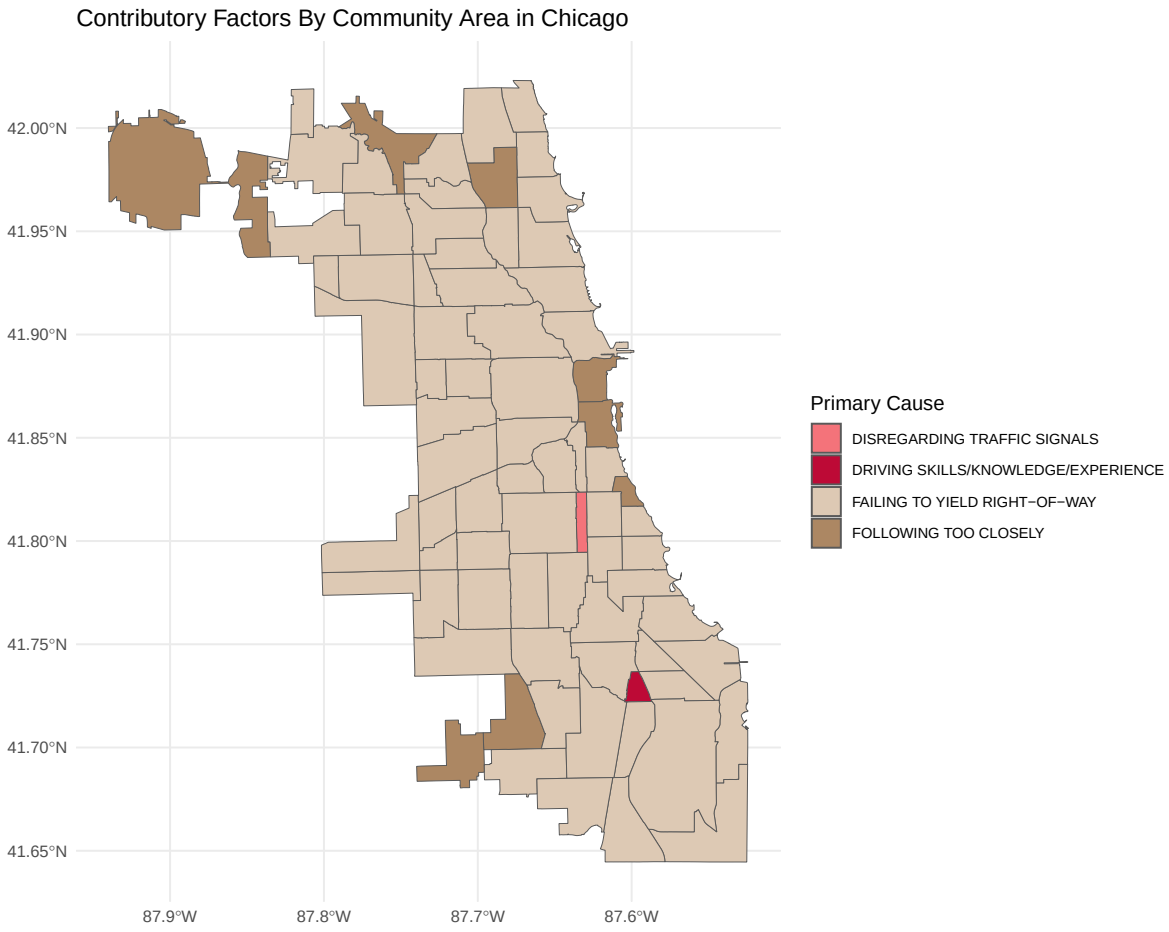
In the next visual map, we are exploring how primary contributory factors to a crash differ by communities, which would essentially lay down a foundation for how we can approach applying multiple layers of data to provide us with key insights.

This comparison not only produces visually ecstatic plots but also provides insights into how each community varies by primary cause. For example, certain communities within Chicago,

such as Fuller Park and Burnside, show a higher frequency of specific primary causes. In Fuller Park, the most frequent contributing factor was disregarding traffic signal, while in Burnside, issues related to driving skills, knowledge, and experience were identified as the primary reason.

By visualizing this data, we can infer that this particular community may have issues with how drivers are responding to traffic signals and how their driving knowledge may need to be further assessed. This highlights the potential need for targeted measures in these areas to address and reduce the frequency of these occurrences.

While visually, we can observe that 'failing to yield the right-of-way' is the most frequent primary contributor to these crashes, excluding 'unable to determine' and 'not applicable' cases. These two levels were removed because they were the most frequent across all of Chicago, which may not provide much meaningful insights. Their high frequency could be due to more obscure reasons, factor beyond my understanding, or even the data entry user may not have known the primary contributory factor of these crashes.



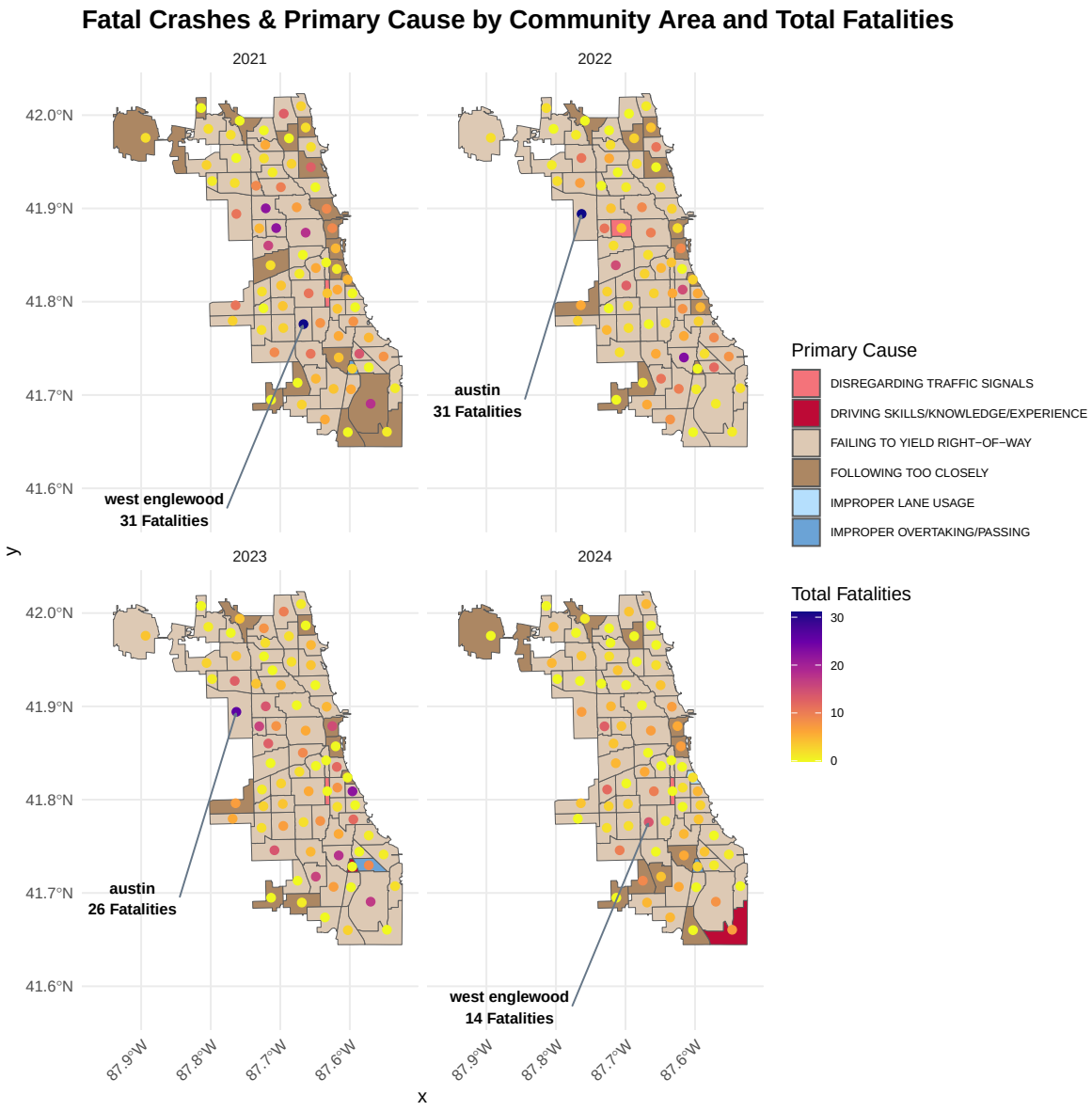
Furthermore, another insight reveals an unsettling consistency in across certain communities, for instance Austin and West Englewood, where the number of fatalities has remained consistently between the tens and thirties over the past four years. These consistencies can be crucial in identifying underlying issues in specific communities within Chicago.

The stability in these number may suggest that certain factors, such as road conditions, traffic, driver behaviors, and even law enforcement practices, have not significantly changed or improved over time. For instance, if the fatalities in Austin and West Englewood, are consistent, despite broader trends, may tell us that specific measured may need to be taken in that area,

such as improved road infrastructure, and stricter enforcement of traffic laws.

```
Joining with `by = join_by(community)`
```

```
Warning in st_point_on_surface.sfc(sf::st_zm(x)): st_point_on_surface may not
give correct results for longitude/latitude data
Warning in st_point_on_surface.sfc(sf::st_zm(x)): st_point_on_surface may not
give correct results for longitude/latitude data
Warning in st_point_on_surface.sfc(sf::st_zm(x)): st_point_on_surface may not
give correct results for longitude/latitude data
Warning in st_point_on_surface.sfc(sf::st_zm(x)): st_point_on_surface may not
give correct results for longitude/latitude data
```

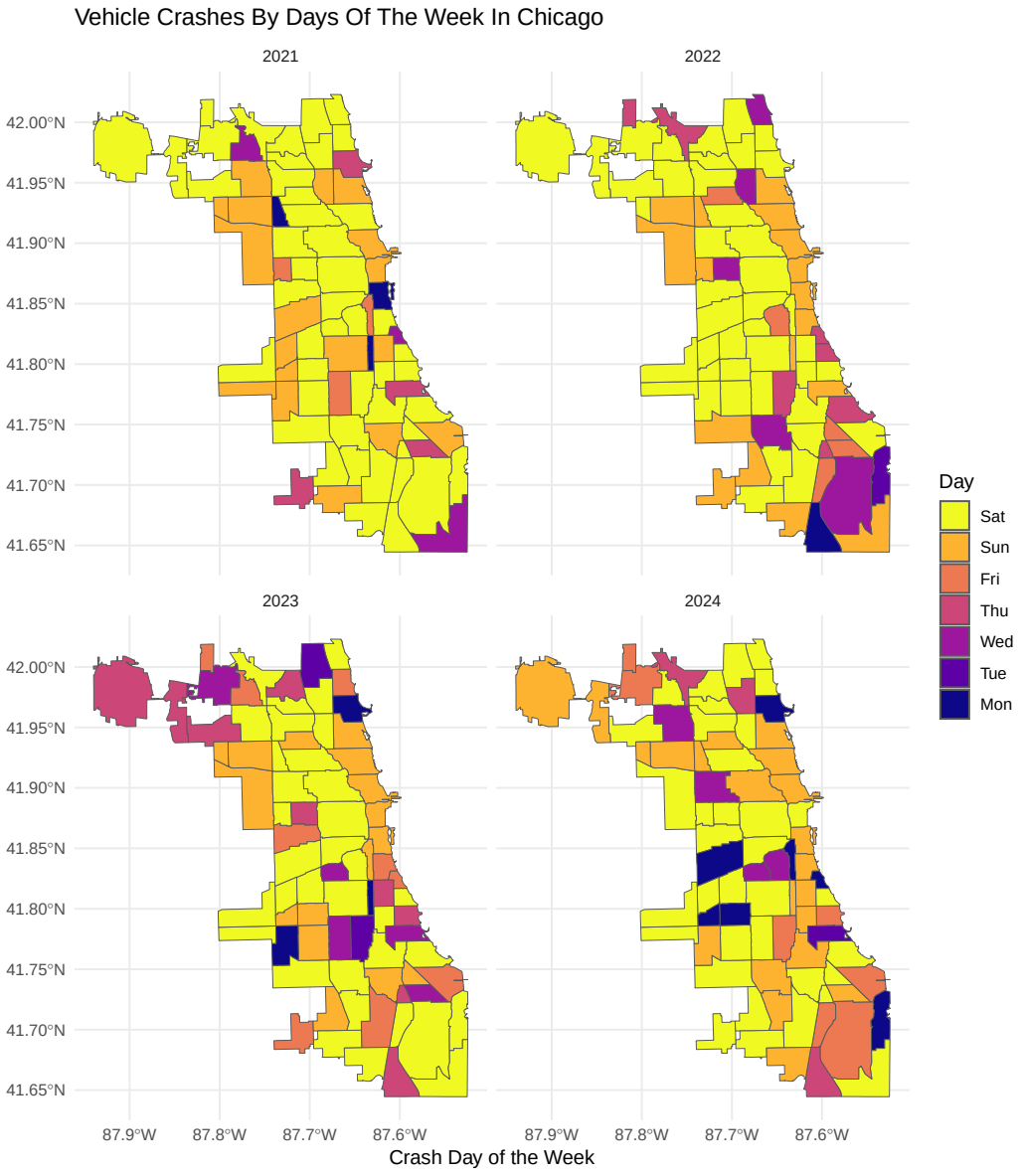


Results: Visual Mapping [Days Of The Week]

In our final visual map, we had previously examined how the frequency of crashes varies by the days of the week. However, I wanted to explore how these trends differ by communities and identify if certain community areas experience higher crash rates on specific days. In this case, i wanted to expand this by visually mapping and facet wrap by year, to further our analysis in how these vary from 2021 to 2024.

In doing so, we can observe that the weekends tend to have a higher frequency of crash across most of the communities. However, upon closer inspection, we find that some communities show fluctuations in crash rates across different days of the week over the year. This variation provides further added insight into how community specific factors and yearly trends may influence the frequency of crashes on particular days of the week.

```
Joining with `by = join_by(community)`
```



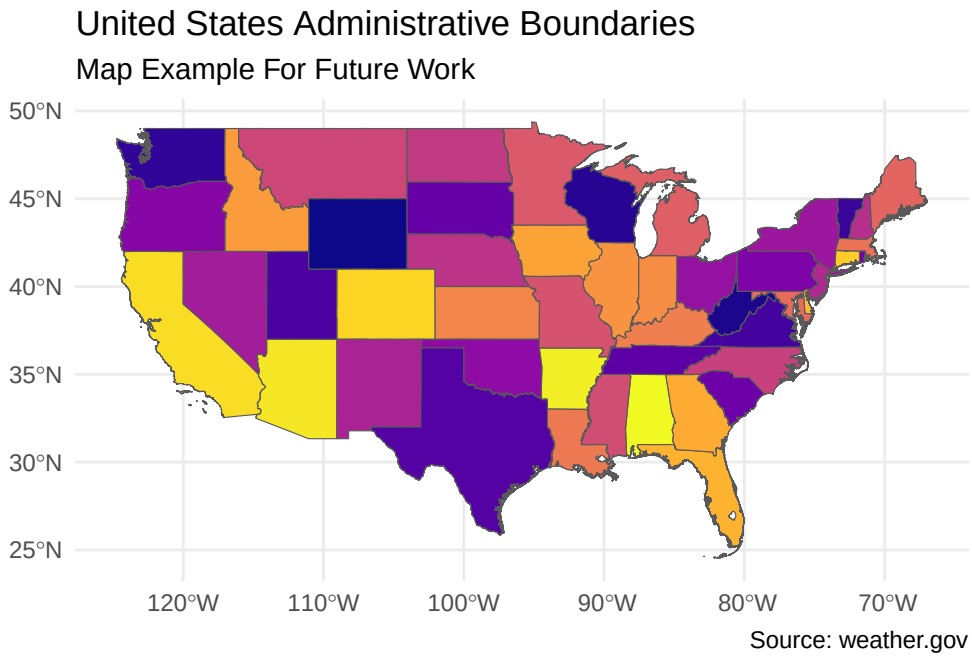
Conclusion

This analysis provided much needed insights into vehicle crashes within Chicago by examining various factors, such as vehicle type, driver conditions, and geographical trends. By exploring fatality rates, crash frequencies, and primary contributory causes, with the study highlighting significant patterns and areas of concern. Through mapping these crash occurrences revealed that specific communities and streets experienced higher crash frequencies. While these findings offer valuable direction, incorporating traffic density data would provide a more comprehensive understanding of how these trends correlate with road usage, which would be vital in future work.

In terms of addressing the predefined questions, the analysis showed that the distribution of vehicle ages by class mean and median did not vary significantly. However, a notable proportion of vehicles over 50 years old, with some exceeding 70 years. This raises the potential for categorizing vehicles into "new","old", and "vintage" groups. Such categorization would require additional knowledge of vehicle value and model to ensure high accuracy. The study also explored how the age of drivers and the speed limit influenced crash occurrences, with most occurring in the 30 mph zones, significantly more than in the higher-speed zones.

Furthermore, by examining crash trends by the days of the week and months, provided valuable context. This analysis revealed that weekends, especially Saturdays, had the highest number of crashes, reflecting increased road usage during leisure and social activities. Overall, this study underscores the complexity of vehicle crash dynamics in Chicago, providing actionable insights to improve the decision making process for resource allocation and road sefety improvements. These findings can inform targeted measures, such as optimzing road designs in high-crash areas, and addressing specific areas to reduce the crash occurrences.

Moreover, the results of this study could be extended to a broader comparative analysis across states in the U.S. to explore variations in primary crash factors. Such a comparison could identify unique trends and patterns, providing a deeper understanding of regional differences and informing nationwide traffic safety measures. For example, the visual map below provides a small sample of how this could shape up, and compare across states.



Appendix

```
library(tidyverse) # for data manipulation
library(paletteer) # for palette colors
library(sf) # for visual maps
library(janitor) # for column name cleaning
library(viridis) # for palette colors
library(paletteer) # for palette colors
library(patchwork) # for combined plots
library(ggrepel) # for plotting labels and arrows
```

```
# Load data ----------------------------
# loading chicago community area shapefile
comm_area <- st_read("FINAL PROJECT DATA /comm_areas.shp") %>%
  clean_names() %>%
  mutate(community = str_to_lower(community))
```

```
# Load data ----------------------------
# dataset of car crashes with overall details
crashes <- read_csv("FINAL PROJECT DATA /Traffic_Crashes_-_Crashes (2024 - 2020).csv")
```

```r
colnames(crashes)
```

```r
# Load data ---------------------------
# dataset of the Vehicles involved in the accidents
vehicles <- read_csv("FINAL PROJECT DATA /Traffic_Crashes_-_Vehicles(2024-2020).csv")
```

```r
# Load data ---------------------------
# dataset of people and their details who where involved in car accidents
people <- read_csv("FINAL PROJECT DATA /Traffic_Crashes_-_People (1)(2024-2020).csv")
```

```r
colnames(people)
```

```r
# Remove duplicates from each dataset based on unique identifiers
crashes <- crashes %>% distinct(CRASH_RECORD_ID, .keep_all = TRUE)
people <- people %>% distinct(CRASH_RECORD_ID, PERSON_ID, VEHICLE_ID, .keep_all = TRUE)
vehicles <- vehicles %>% distinct(CRASH_RECORD_ID, VEHICLE_ID, .keep_all = TRUE)

# Select only the relevant columns from the 'crashes' dataset
crashes_selected <- crashes %>%
  select(CRASH_RECORD_ID, CRASH_DATE, POSTED_SPEED_LIMIT, TRAFFIC_CONTROL_DEVICE,
         DEVICE_CONDITION,WEATHER_CONDITION, LIGHTING_CONDITION, FIRST_CRASH_TYPE,
         TRAFFICWAY_TYPE, LANE_CNT, ALIGNMENT,ROADWAY_SURFACE_COND, ROAD_DEFECT, REPORT_TY
         CRASH_TYPE, INTERSECTION_RELATED_I,NOT_RIGHT_OF_WAY_I, HIT_AND_RUN_I, DAMAGE,
         PRIM_CONTRIBUTORY_CAUSE, SEC_CONTRIBUTORY_CAUSE,STREET_NO, STREET_DIRECTION,
         STREET_NAME, NUM_UNITS, INJURIES_TOTAL, INJURIES_FATAL, INJURIES_INCAPACITATING,
         INJURIES_NON_INCAPACITATING, INJURIES_REPORTED_NOT_EVIDENT,INJURIES_NO_INDICATION
         CRASH_HOUR, CRASH_DAY_OF_WEEK, CRASH_MONTH, LATITUDE, LONGITUDE, LOCATION)

# Select relevant columns from 'people'
people_selected <- people %>%
  select(CRASH_RECORD_ID, PERSON_ID, PERSON_TYPE, VEHICLE_ID, SEAT_NO, SEX, AGE,
    DRIVERS_LICENSE_STATE, DRIVERS_LICENSE_CLASS, SAFETY_EQUIPMENT, AIRBAG_DEPLOYED,EJECTI
    INJURY_CLASSIFICATION, DRIVER_ACTION, DRIVER_VISION,PHYSICAL_CONDITION, PEDPEDAL_ACTIO
    BAC_RESULT, `BAC_RESULT VALUE`, CELL_PHONE_USE)

# Select relevant columns from 'vehicles'
vehicles_selected <- vehicles %>%
  select(CRASH_RECORD_ID, VEHICLE_ID, NUM_PASSENGERS, UNIT_NO, MAKE, MODEL, VEHICLE_YEAR,
         VEHICLE_TYPE, VEHICLE_USE, TRAVEL_DIRECTION, MANEUVER, EXCEED_SPEED_LIMIT_I,
         FIRST_CONTACT_POINT)
```

```r
# Merge 'crashes_selected' and 'people_selected' using 'CRASH_RECORD_ID'
# This keeps all crash records and matches people data to them
merged_crashes_people <- left_join(crashes_selected, people_selected, by = "CRASH_RECORD_I

# Merge 'merged_crashes_people' with 'vehicles_selected' using 'CRASH_RECORD_ID' and 'VEHI
# This ensures that each vehicle remains associated with its specific crash and people dat
Chicago_Crash <- left_join(merged_crashes_people, vehicles_selected, by = c("CRASH_RECORD_

# Viewing one record to verify no records are lost
Chicago_Crash %>%
  filter(CRASH_RECORD_ID == "6c1659069e9c6285a650e70d6f9b574ed5f64c12888479093dfeef179c034
  select(MAKE) %>%
  print()
```

```r
# summarising by crash count, and grouping by VEHICLE_TYPE
Vehicle_type_crash <- Chicago_Crash %>%
  group_by(VEHICLE_TYPE) %>%
  summarise(crash_count = n()) %>%
  arrange(desc(crash_count))
# output results
print(Vehicle_type_crash)
```

```r
# Counting the freqency of car makes used for PERSONAL use
PERSONAL_Data <- Chicago_Crash %>%
  filter(VEHICLE_TYPE == "PASSENGER") %>%count(MAKE, name = "frequency") %>%
  arrange(desc(frequency)) %>% slice(1:5)

# output results
print(PERSONAL_Data)
```

```r
# Counting the freqency of car makes used for PERSONAL use
SUV_Data <- Chicago_Crash %>%
  filter(VEHICLE_TYPE == "SPORT UTILITY VEHICLE (SUV)") %>%
  count(MAKE, name = "frequency") %>%
  arrange(desc(frequency)) %>% slice(1:5)

# output results
print(SUV_Data)
```

```r
  # Applying the classification logic to classify vehicle classes by their values on the m
  Chicago_Crash_DATA <- Chicago_Crash %>%
    mutate(MAKE = str_trim(MAKE),
           VEHICLE_CLASS = case_when(
             str_detect(MAKE, "ALFA ROMEO|MERCEDES-BENZ|ROLLS ROYCE|ASTON MARTIN|BMW|MASER
             str_detect(MAKE, "TOYOTA|SMART|MUSTANG|HONDA|HUMMER|FORD|GENESIS|NISSAN|VOLKS
             str_detect(MAKE, "CHEVROLET|KIA|HYUNDAI|SAAB|RAMBLER|FIAT|ISUZU|SUBARU|FORD|D
             str_detect(MAKE, "HARLEY-DAVIDSON|KTM|YAMAHA|VESPA|DUCATI|ELECTRIC CYCLE|ECO-
             TRUE ~ "Other & Unknown"
           )) %>%
  select(VEHICLE_CLASS, POSTED_SPEED_LIMIT, PHYSICAL_CONDITION, MAKE, AGE, INJURIES_FATAL,
 # main columns will focus on for our analysis
# View the result
print(Chicago_Crash_DATA)
```

```r
# using summary() to see the distribution of the vehicle_year
summary(Chicago_Crash_DATA$vehicle_year)

# output max year
max_vehicle_year <- max(Chicago_Crash_DATA$vehicle_year, na.rm = TRUE)

# Filter rows where vehicle_year is the maximum value
filtered_max_year <- Chicago_Crash_DATA %>%
  filter(vehicle_year == max_vehicle_year) %>%
  select(vehicle_year, crash_date)  # Select only the vehicle_year and crash_date columns

# Print the filtered data
print(filtered_max_year)

# Compute Vehicle_Age and filter out rows where vehicle_year > 2024
Vehicle_Life <- Chicago_Crash_DATA %>%
  mutate(crash_date = mdy_hms(crash_date), # Converting to a time variable
         CRASH_YEAR = year(crash_date), # Extracting just the year to create a new variabl
         Vehicle_Age = CRASH_YEAR - vehicle_year) %>% # Computing the Vehicle_Age
  filter(vehicle_year <= 2024 & vehicle_year >= 1920) %>%
  filter(vehicle_class != "Other & Unknown") %>% # Filter out "Other & Unknown" vehicle cl
  filter(Vehicle_Age >= 0)
```

```r
# print the results
print(Vehicle_Life[, c("vehicle_year", "CRASH_YEAR", "Vehicle_Age","vehicle_class")])
```

```r
# boxplot using facet wrap by vehicle_class
ggplot(Vehicle_Life, aes(x = factor(CRASH_YEAR),
                         y = Vehicle_Age,
                         fill = factor(CRASH_YEAR))) +
  geom_boxplot() +
  labs(title = "Boxplot of Vehicle Age by Crash Year and Vehicle Class",
       x = "Crash Year",
       y = "Vehicle Age") +
  theme_minimal() +
  scale_fill_viridis_d(option = "magma") +  # Apply magma palette
  facet_wrap(~ vehicle_class, scales = "free_x")

# computing the mean and median of the Vehicle_Age
Vehicle_Life2 <- Chicago_Crash_DATA %>%
  mutate(crash_date = mdy_hms(crash_date),  # converting to a time variable
         CRASH_YEAR = year(crash_date),  # extracting just the year to create a new variab
         Vehicle_Age = CRASH_YEAR - vehicle_year) %>%
  filter(vehicle_year <= 2024 & vehicle_year >= 1920) %>%
  filter(vehicle_class != "Other & Unknown") %>% # Filter out "Other & Unknown" vehicle cl
  filter(Vehicle_Age >= 0) %>%
  select(vehicle_class, CRASH_YEAR, Vehicle_Age) %>%
  group_by(vehicle_class) %>%
  summarise(mean_vehicle_age = mean(Vehicle_Age, na.rm = TRUE),  # compute mean vehicle ag
            median_vehicle_age = median(Vehicle_Age, na.rm = TRUE),n = n())

# output results
print(Vehicle_Life2)
```

```r
# Aggregate crash counts and fatalities by STREET_NAME
street_summary <- Chicago_Crash_DATA %>%
  group_by(street_name) %>%
  summarise(
    total_crashes = n(),
    total_fatalities = sum(injuries_fatal, na.rm = TRUE),
    latitude = mean(latitude, na.rm = TRUE),
    longitude = mean(longitude, na.rm = TRUE)) %>%
  arrange(desc(total_fatalities), desc(total_crashes))

# Select top 20 streets that recorded the highest total_fatalities
top_dangerous_streets <- street_summary %>%
  slice_head(n = 20)
```

```r
# output
print(top_dangerous_streets)

top_dangerous_sf <- st_as_sf(
  top_dangerous_streets,
  coords = c("longitude", "latitude"),
  crs = st_crs(comm_area))

# Create the map
ggplot() +
  geom_sf(data = comm_area, fill = "black", color = "white", alpha = 0.9) +
  geom_sf(data = top_dangerous_sf, aes(size = total_crashes, color = total_fatalities), al
  scale_color_gradient(low = "yellow", high = "red") +
  labs(
    title = "Top 20 Most Dangerous Streets in Chicago",
    subtitle = "Based on Crash Counts and Fatalities",
    size = "Total Crashes",
    color = "Total Fatalities"
  ) +
  theme_minimal()
```

```r
# Create a crash summary by posted_speed_limit and age
crash_summary_speed_age <- Chicago_Crash_DATA %>%
  filter(!is.na(posted_speed_limit) & !is.na(age) & age >= 0) %>%  # Remove rows with miss
  group_by(posted_speed_limit, age) %>%
  summarise(TOTAL_CRASHES = n(), .groups = "drop") %>%
  arrange(desc(TOTAL_CRASHES))

# output the summary
head(crash_summary_speed_age)

# plot the plot to see where most crashes occur by age and posted speed limit.
ggplot(crash_summary_speed_age, aes(x = posted_speed_limit, y = age, fill = TOTAL_CRASHES)
  geom_tile(color = "white") +  # White borders between tiles
  scale_fill_viridis_c(option = "magma", direction = 1, trans = "log") +
  labs(
    title = "Heatmap of Crashes by Posted Speed Limit and Age",
    x = "Posted Speed Limit (mph)",
    y = "Age of those affected",
    fill = "Total Crashes (log scale)"
  ) +
```

```
  theme_minimal() +
  theme(legend.position = "right")
```

```
# print the unqiue values of the physical condition of the drivers
unique(Chicago_Crash_DATA$physical_condition)

# Summarizing crash severity by driver's condition and vehicle class
# based on the condition of the driver if they were impaired by alcohol or FATIGUED
crash_summary_condition <- Chicago_Crash_DATA %>%
  filter(physical_condition %in%
           c("IMPAIRED - ALCOHOL", "FATIGUED/ASLEEP","IMPAIRED - DRUGS")) %>%
  group_by(vehicle_class, physical_condition) %>%
  summarise(INJURIES_TOTAL_COUNT = sum(injuries_total > 0, na.rm = TRUE),
    AVG_INJURY = mean(injuries_total, na.rm = TRUE),
    .groups = "drop")

# Print the result
print(crash_summary_condition)
```

```
# visualizing the crash occurances by the days of the week
Chicago_Crash_DATA %>%
  tibble() %>%
  mutate(crash_date = mdy_hms(crash_date),
    year = year(crash_date)) %>%
  mutate(Day = fct_infreq(factor(crash_day_of_week,
                                 levels = 1:7,
                                 labels = c("Mon", "Tue",
                                            "Wed", "Thu",
                                            "Fri", "Sat",
                                            "Sun")))) %>%
  ggplot(aes(x = Day, fill = Day)) +
  scale_fill_viridis_d(option = 'C', direction = -1) +
  geom_bar(color = 'black') +
  labs(x = "Day of the Week",
       y = "Count of Crashes",
       title = "Crashes by Day of the Week")


# visualizing the crash occurances by the weeks of the week
Chicago_Crash_DATA %>%
  tibble() %>%
```

```
  mutate(crash_date = mdy_hms(crash_date),
    year = year(crash_date)) %>%
  mutate(Month =
           fct_infreq(factor(crash_month,
                             levels = 1:12,
                             labels = c("Jan","Feb",
                             "Mar","Apr",
                             "May","Jun",
                             "Jul","Aug",
                             "Sep","Oct",
                             "Nov","Dec")))) %>%
           ggplot(aes(x = Month, fill = Month)) +
           scale_fill_viridis_d(option = 'C', direction = -1) +
           geom_bar(color = 'black') +
           labs(x = "Month of the Year", y = "Count of Crashes",
        title = "Crashes by Month of the Year")
```

```
# merging shapefile with main dataset with coordinates
Chicago_Crash_DATA <- Chicago_Crash_DATA %>%
  clean_names() %>%
  filter(!is.na(longitude) & !is.na(latitude) &
           longitude != 0 & latitude != 0) %>%
  st_as_sf(coords = c("longitude", "latitude"),
           crs = st_crs(comm_area))

# Spatial Join each crash point assigned to their community
joined_sf <- st_join(Chicago_Crash_DATA, comm_area, join = st_within)
```

```
# visual map of communities in chicago with number of crashes.
plot_total_crashes <- joined_sf %>%
  tibble() %>%
  group_by(community) %>%
  summarise(total_crashes = n(),.groups = 'drop') %>%
  left_join(comm_area) %>%
  st_as_sf() %>%
   ggplot() +
  geom_sf(aes(fill = total_crashes)) +
  scale_fill_viridis(option = "C") +
  theme_minimal() +
  labs(title = "Total Crashes",
       fill = "Total Crashes") +
```

```
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

plot_fatal_crashes <- joined_sf %>%
  tibble() %>%
  group_by(community) %>%
  summarise(total_fatalities = sum(injuries_fatal > 0, na.rm = TRUE),
            .groups = 'drop') %>%
  left_join(comm_area) %>%
  st_as_sf() %>%
   ggplot() +
  geom_sf(aes(fill = total_fatalities)) +
  scale_fill_viridis_c(option = "C") +
  theme_minimal() +
  labs(title = "Total Fatal Crashes",
       fill = "Total Fatalities") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

# Combined Plot of total and fatal crashes
Crash_Plot <- (plot_total_crashes / plot_fatal_crashes) +
  plot_annotation(title = "Crash Data by Community Area (2020 - 2024)",
  subtitle = "Comparison of total and fatal crashes across Chicago communities")

plot(Crash_Plot)
```

```
# creating a visual map of vehicle crashes by prim_contributory_cause
  joined_sf %>%
  tibble() %>%
  filter(!prim_contributory_cause %in%
           c("UNABLE TO DETERMINE", "NOT APPLICABLE")) %>%
  group_by(community, prim_contributory_cause) %>%
  summarise(frequency = n(), .groups = 'drop') %>%
  group_by(community) %>%
  slice_max(frequency, n = 1, with_ties = FALSE) %>%
  left_join(comm_area, by = "community") %>%
  st_as_sf() %>%
  ggplot() +
  geom_sf(aes(fill = prim_contributory_cause)) +
  scale_fill_paletteer_d(
    palette = "ggthemes::Classic_Blue_Red_12",
    name = "Primary Cause",
    direction = -1
```

```
  ) +
  theme_minimal() +
  labs(
    title = "Total Crashes by Community Area in Chicago",
    fill = "Primary Cause"
  ) +
  theme(legend.text = element_text(size = 8)) # Adjust size if needed
```

```
# defining our plot data for prim_contributory_cause
suppressWarnings({
  B  <- joined_sf %>%
    tibble() %>%
    filter(!prim_contributory_cause %in%
             c("UNABLE TO DETERMINE","NOT APPLICABLE")) %>%
    mutate(crash_date = mdy_hms(crash_date),
           year = year(crash_date)) %>%
    group_by(community, year, prim_contributory_cause) %>%
    summarise(frequency = n(), .groups = 'drop') %>%
    group_by(community, year) %>%
    slice_max(frequency, n = 1, with_ties = FALSE) %>%
    filter(year != 2020) %>%
    left_join(comm_area, by = "community") %>%
    st_as_sf()

  # defining our plot data for injuries_fatal
  D <- joined_sf %>%
  tibble() %>%
  mutate(crash_date = mdy_hms(crash_date),
         year = year(crash_date)) %>%
  group_by(community, year) %>%
  summarise(total_fatalities = sum(injuries_fatal >= 1, na.rm = TRUE),
            .groups = 'drop') %>%
  filter(year != 2020) %>%
  left_join(comm_area) %>%
  st_as_sf()

# Computing centroids of the community areas in D
  D_centroids <- D %>%
    st_centroid()

# Identifying the communities with the highest frequency of fatalities
```

```
D_centroids_labels <- D_centroids %>%
  group_by(year) %>%
  slice_max(total_fatalities, n = 1, with_ties = FALSE)


# Plotting two layers in a visual map for Primary Cause and Total Fatalities
suppressWarnings({
ggplot() +
  geom_sf(data = B, aes(fill = prim_contributory_cause)) +
  geom_sf(data = D_centroids,
          aes(color = total_fatalities),
          alpha = 1, size = 2) +
  scale_fill_paletteer_d(palette = "ggthemes::Classic_Blue_Red_12",
                         name = "Primary Cause",
                         direction = -1) +
  scale_color_viridis_c(name = "Total Fatalities",
                        option = "plasma",
                        direction = -1) +
  labs(title = "Fatal Crashes & Primary Cause by Community Area and Total Fatalities",
       fill = "Frequency of Primary Cause", size=2) +
  theme_minimal() + theme(legend.text = element_text(size = 7),
                          plot.title = element_text(size = 10, face = "bold")) +
  facet_wrap(~year) +
  geom_text_repel(data = D_centroids_labels,
                  aes(geometry = geometry,
                      label = paste0(community, "\n",
                                     total_fatalities, " Fatalities")),
                  stat = "sf_coordinates",size = 3,
                  arrow = arrow(length = unit(0.02, "inches")),
                  segment.color = "#678",
                  nudge_x = -0.2, nudge_y = -0.2, fontface = "bold") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))})})
```

```
# Using fct_infreq to
joined_sf %>%
  tibble() %>%
  mutate(crash_date = mdy_hms(crash_date),
         year = year(crash_date)) %>% filter(year != 2020) %>%
  mutate(Day = fct_infreq(factor(crash_day_of_week,
                                 levels = 1:7,
                                 labels = c("Mon", "Tue",
```

```
                                          "Wed", "Thu",
                                          "Fri", "Sat",
                                          "Sun")))) %>%
  group_by(community, year, Day) %>%
  summarise(frequency = n(), .groups = 'drop') %>%
  group_by(community,year ) %>%
  slice_max(frequency, n = 1, with_ties = FALSE) %>%
  left_join(comm_area) %>%
  st_as_sf() %>%
  ggplot() +
  geom_sf(aes(fill = Day)) + scale_fill_viridis_d(option = 'C', direction =-1) +
  labs(title = "Vehicle Crashes By Days Of The Week In Chicago",
       x = "Crash Day of the Week") + facet_wrap(~year)  +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) + theme_minimal()
```

```
# Load data ---------------------------
# loading United States Boundaries shapefile
comm_area22 <- st_read("FINAL PROJECT DATA /s_05mr24.shp") %>%
  clean_names() %>%
  mutate(state = str_to_lower(state)) %>%
  filter(!name %in% c("American Samoa",
                      "Guam",
                      "Northern Mariana Islands",
                      "Puerto Rico",
                      "United States Virgin Islands",
                      "Alaska","Hawaii",
                      "Fed States of Micronesia","Palau",
                      "Virgin Islands","Marshall Islands"))
```

```
# visual map of the united states and its boundaries
ggplot(data = comm_area22) +
  geom_sf(aes(fill = state)) +
  scale_fill_viridis_d(option = "C", direction =-1) +
  theme_minimal() + # Minimal theme for cleaner appearance
  theme(legend.position = "none") + # Hides the legend
  labs(title = "United States Administrative Boundaries",
       subtitle = "Map Example For Future Work",
       caption = "Source: weather.gov")
```