Analysis of Vehicular Accidents in New York City Before and During COVID-19

Michael Barton
Cory DioGuardi
Alex Taffe

Data Cleaning

Discuss how you modified the data. Did you manually clean the data, or write a program to clean the data? What did you do to the data?

We first started by manually looking at the data to see if the data conformed to any patterns, or if there was just random input from users. This was a large task at first, until we loaded the data into a SQL database (See Data Processing Platform). Using SQL, we were able to look at all the different unique values for each attribute. Some of the attributes already conformed into patterns. For example, the borough attribute provided data clearly organized into a handful of values. For attributes similar to this, we did not need to perform any data cleaning.

Other attributes however, were not as straightforward. The best example of this was the five vehicle type attributes. The first attempt at cleaning the data was to find some common phrases that each potential vehicle type had in common. After spending some time getting more familiar with the data, we discovered that this wasn't feasible. There was some overlap, and the common phrases between types were often too short to be sufficiently unique.

To account for the issues with vehicle types, we ended up using collections of value mappings. This involved some manual work for every different vehicle type. We collected sets of strings from the data in the vehicle type fields and grouped them by type. We noticed that there were many sets of attributes where the only difference was casing. So we decided to ignore the case in all of the mappings. This significantly reduced the amount of mappings we had to store. We reached a point where we had over 99% of the data grouped into under 40 categories. The remaining rows that were unmapped contained abbreviations or strings that were unclear or did not provide enough distinction between groups. These data points were left alone and unchanged.

For question 7, the primary data required was the latitude, longitude, and date of the crashes. A number of crashes had no coordinates, only one of the two coordinates, or coordinates in locations that were not possible. When generating the graphs for question 7, after the data is read in, any rows without a latitude, longitude, or crash date are all thrown out. Rows with dates that cannot be parsed are also thrown out. Finally, any crashes outside the coordinate bounding box: (lat: [40.5544, 40.9129], long: [-74.0529, -73.6630]) are thrown out. An additional form of "cleaning" occurs when the coordinates are binned together, as the binning process is an effective way to reduce some level of noise.

Data Processing Platform

Did you use Mongo-DB, or SQL, or what? What languages did you use?

For the majority of analytics, we used a SQLite database to store and work with the data. This allowed us to write SQL queries to analyze the data. We chose to use SQL for this because it is designed to query data, so getting exactly the points and attributes to analyze a specific problem was turned into a very simple process. SQL is also optimized for large sets of data. Given the large dataset we were working with, SQL allowed us to store the data in a database file that was already properly indexed. This meant that the data would not have to be re-analyzed by a program every time it was run. In our experience, using Python and just loading the dataset into memory was a slow process for the computer, so we wanted to avoid this anywhere we could.

However, for question 7, it was necessary to process the data within Python, rather than pull queries from an SQL database. For this purpose, Pandas was used to read in the whole CSV file in one shot, only pulling in the required columns to save memory. Attempting to read the file with Python's built in csv parser and a numpy array was taking hundreds of gigabytes of RAM and minutes, while Pandas accomplished the same task in a few seconds.

To generate our graphs and visuals, we used Python, with the MatPlotLib library to generate plots. This allowed us to save time creating plots to show what we found, and to spend more time actually analyzing the data. This meant that we were able to do more than just the given set of analytics, one additional comparison which is included in this report.

To load the dataset into the aforementioned SQL database, we used Python to load the data in. We found that the pandas library had convenient functions for inserting dataframes into a database. We wanted to make the database loading script rerunnable, so we created a JSON configuration file. This is where we defined everything from the table columns that needed to be created, to the mappings used for data cleaning. This made tweaking the data cleaning process or names in the database convenient and simple while our code was being developed.

Tools

What software tools or packages did you use?

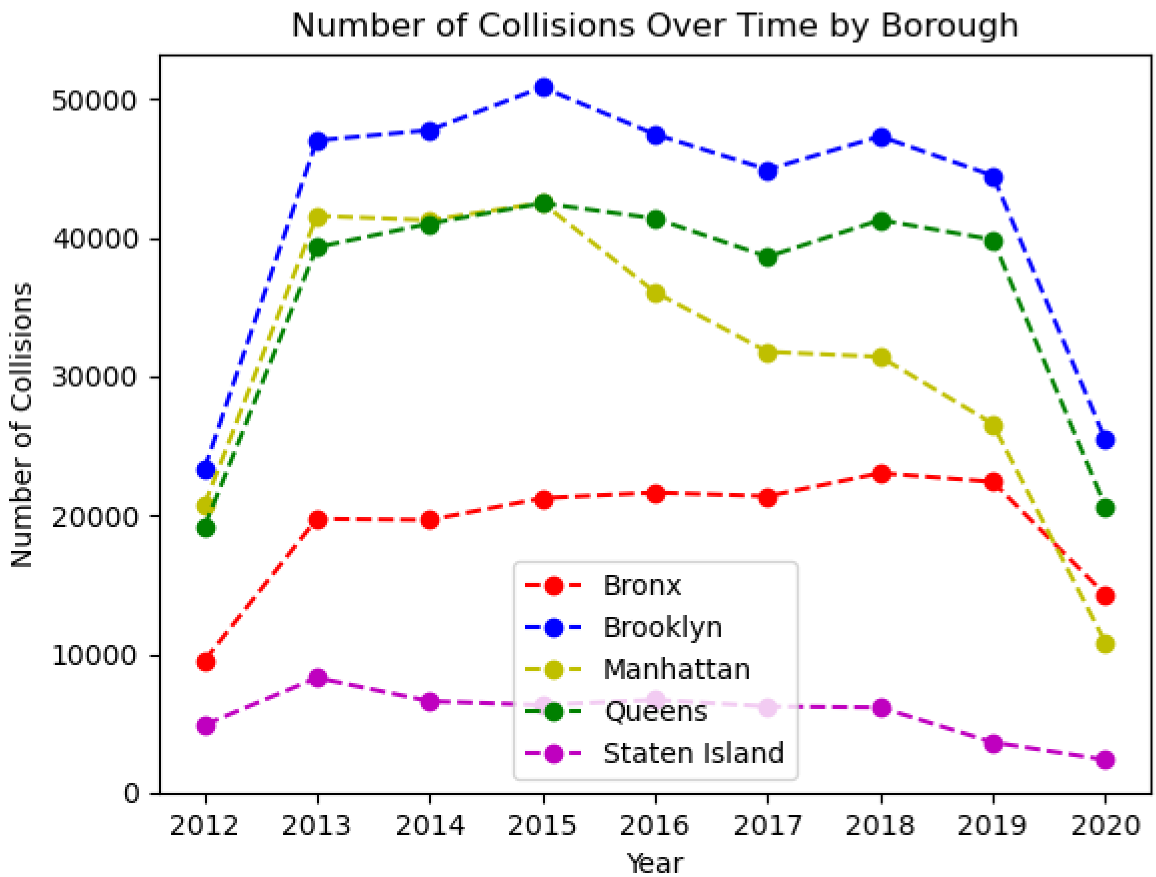
The biggest tool we used for our data analysis was Python. We chose to use Python because it is quick to write, easy to read, and has many packages that make data analysis faster and easier. Some of these packages include pandas, which has many built in functions for and objects to make sorting through and working with data easier. Another package is MatPlotLib, which was extremely useful for graphing the results of our analytics, as it allowed us to make many different types of plots quickly and effectively. We also used a package for interacting with our SQLite database sqlite3. This allowed us to run queries against the database and get the results back. We used this package for data processing, cleaning, and analytics.

In terms of development tools, data exploration, and analysis, we used VSCode for the actual code editing as well as JetBrains DataGrip for database visualization and sample querying. We also used git so that all members could collectively contribute to the work, and check others' work for errors.

For the parzen density estimation, scipy, pandas, numpy, and matplotlib were all used in order to help clean and display the data. Additionally, the map imagery was provided by OpenStreetMap. pbzip2 was also used in order to extract the data from the OpenStreetMap global archive. Finally, a torrent client was used to download the OpenStreetMap data due to its size.

Investigations

Borough with the Most Accidents

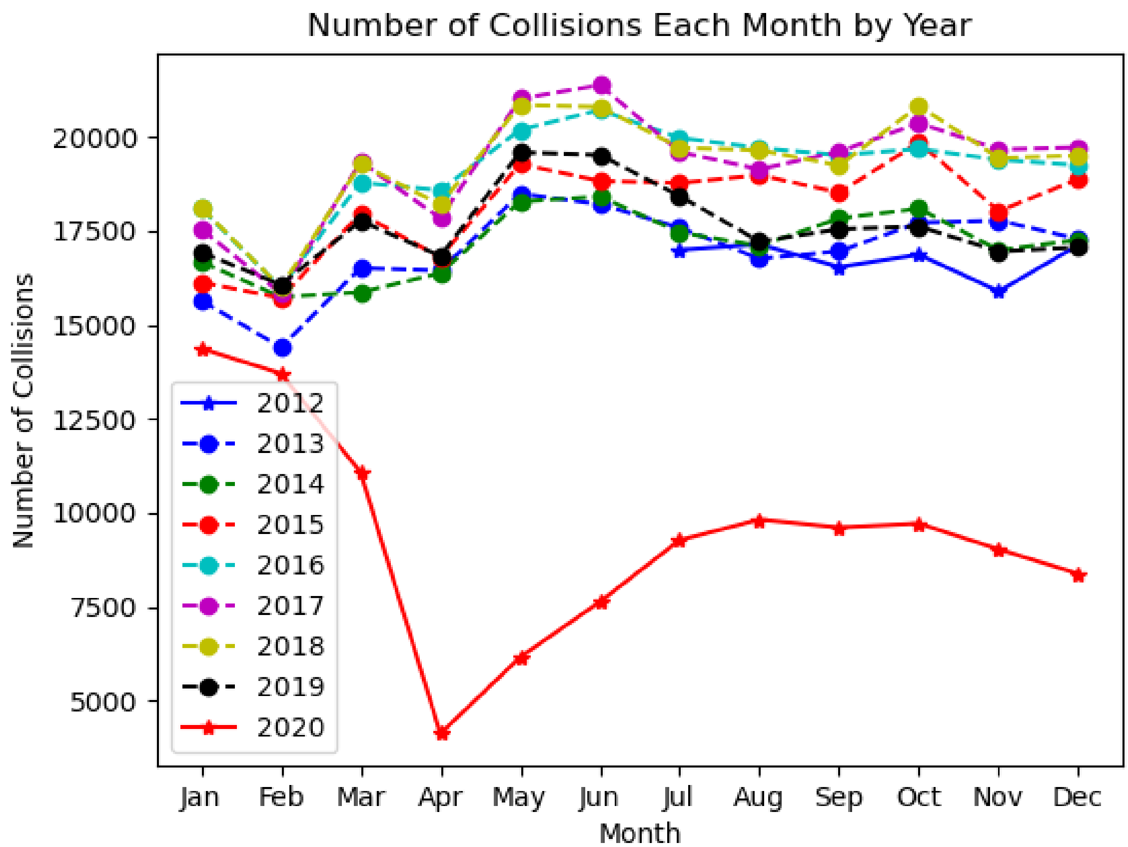


It appears that Brooklyn and Queens tend to have the most accidents out of all five boroughs. The Bronx, Brooklyn and Queens seem to follow the same general trend from 2012 to 2020. They all appear to have a dramatic increase from 2012 to 2013, but I think it is safe to assume that this has something to do with the way the data was collected. Perhaps 2012 was a pilot year for the program before it was fully deployed for city-wide data collecting in 2013.

Manhattan was an interesting trend as it appeared to match the previous three's trends until the 2015-2016 year change. The number of collisions started to steadily decrease until the sharp decline in 2020 which is not really surprising as that is when the COVID-19 pandemic hit. In fact, all five boroughs saw a drop in collisions with some of them seeing sharper ones than others. Upon further research, it was determined that this was not without cause just as most sudden changes in trends are not just by chance. Rather, this steady drop seems to be a result of the borough-wide implementation of Vision Zero. Vision Zero is a citywide initiative that seeks to prevent crashes through engineering, enforcement and education. It was implemented in 2014 and since then the city's officials and government have "used every tool at its disposal to improve the safety of our streets in every neighborhood and in every borough – with expanded enforcement against dangerous moving violations like speeding and failing to yield to pedestrians, new street designs and configurations to improve safety, broad public outreach and communications, and a sweeping legislative agenda to increase penalties for dangerous drivers and give New York City control over the safety of our own streets" (Vision Zero).

Staten Island appears to be the safest borough out of all five boroughs which at a high level makes sense to me. It is the most removed borough from the other four geographically and is usually more associated with family living and a suburban kind of lifestyle as compared to the urban settings of the other boroughs. Given this knowledge, it's easier to see why there are fewer accidents in that borough.

Months with the Most Accidents



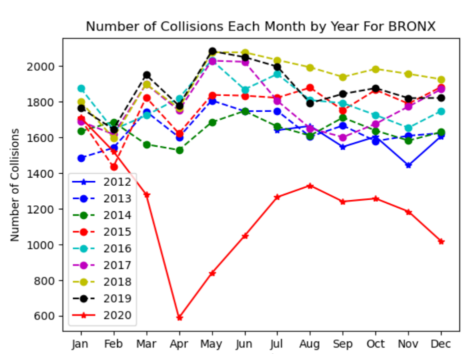
The trends in the monthly accidents seemed to mostly match year to year which is interesting to see. February (likely because it has fewer days) has a drop in the number of accidents every year while June and July have an increase. This is likely due to the summer weather and increased outdoor activities. The sharp decline in accidents in 2020 is quite expected due to the timing of the national "lockdown" due to the pandemic.

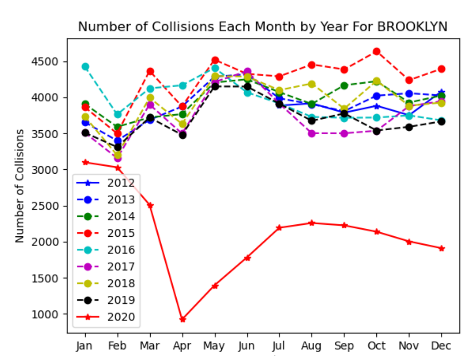
The sharp decline and then steady increase of accidents bears a resemblance to the stock market trends. For reference, the charts below are 5 year charts from Google of the Dow Jones Industrial Average and the S&P 500 (in that order). You can see that they also had sharp declines in the earlier part of 2020 before making a comeback and slowly rising back up. Although 2021 is not included in the above graph of collisions, I would imagine it follows this same general trend.

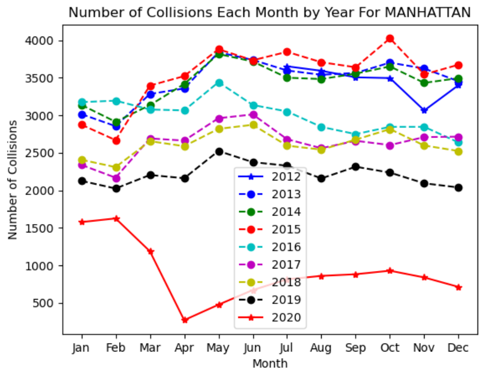
(Above) Dow Jones Industrial Average

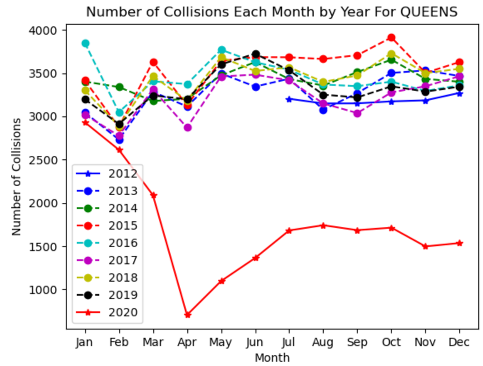
(Above) S&P 500

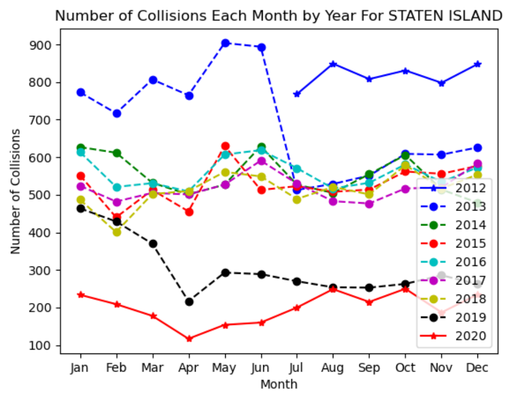
Below are some charts showing the number of collisions each month specifically for each borough. They mostly seem to agree with the monthly trends for all of the boroughs put together. What's more interesting, however, is that the findings in these graphs appear to support the claims we made for the first section above "Boroughs with the Most Accidents". We can again see that Staten Island was much safer as far as car accidents than the other four boroughs. Additionally, we can also see the decline in car accidents for Manhattan when Vision Zero was implemented. The other three boroughs did not seem to waver much year to year.



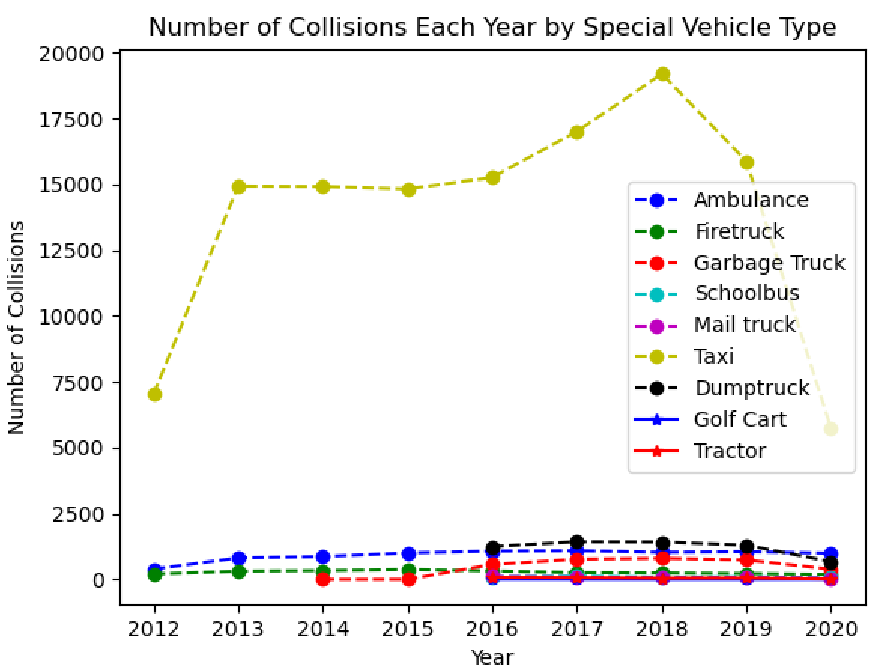


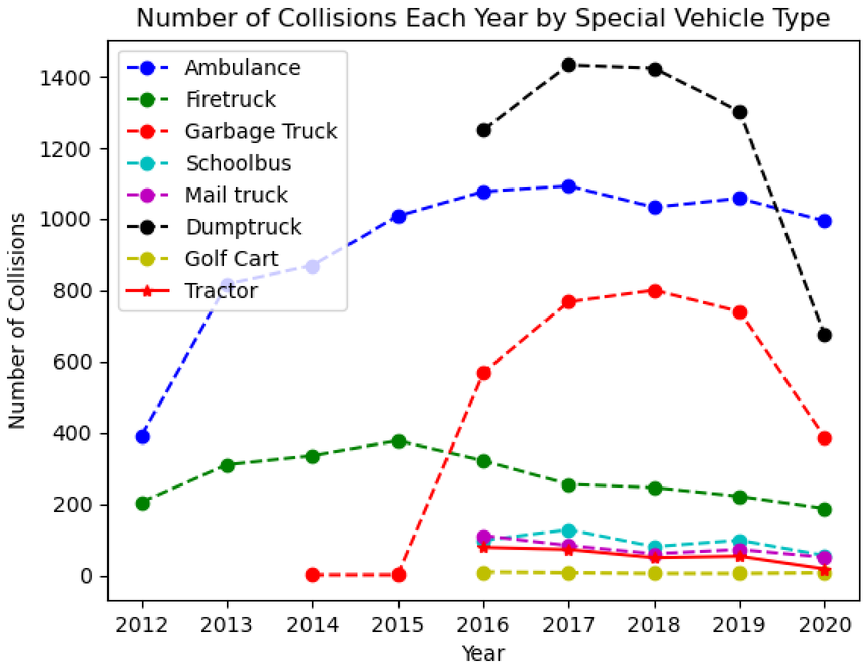






Types of Accidents

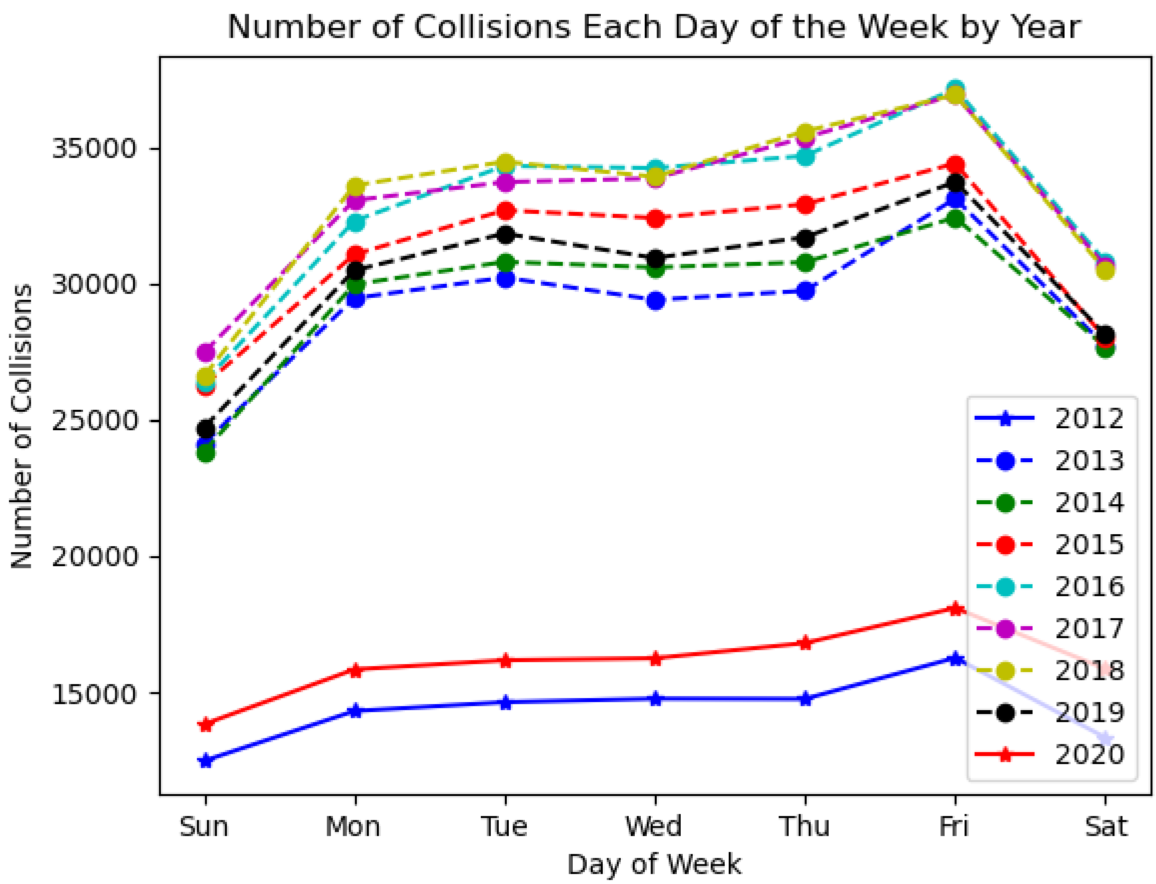




It is almost amusing to note the drastic difference between these two graphs due to the inclusion or exclusion of taxis. One reason for this could be that there are just many, many more taxis on the road as compared to these other types of vehicles. Even compared to regular passenger cars, sometimes it seems that taxis are more prevalent and so may be more likely to be involved in car accidents. However, it may also point to the reckless and aggressive nature that is often tied to the drivers of these taxis. Since they are more desensitized to the traffic of New York City, they may be more prone to getting into an accident.

It's interesting as well to note the trend changes in 2020 for some of these vehicles. Vehicles like dump trucks (used for construction) expectedly were involved in fewer crashes since there were probably fewer on the road. Construction likely halted in many cases due to the pandemic with only essential projects being continued. However, vehicles like fire trucks and ambulances did not have as sharp of a drop. Thinking about it, though, this does make sense. There were probably just as many of these vehicles on the road, especially ambulances, as they are an essential service that can't just stop during a pandemic. For this reason, they're not much less likely to be involved in an accident in 2020 compared to other years.

Days of the Week with the Most Accidents



This chart makes sense to us in general. There are not as many people out on the roads driving on the weekend, so the dip on Saturday and Sunday lines up with that logically. Additionally, an assumption could be made that since more adults will be consuming alcoholic beverages on the weekend that they would be carpooling in taxis or Ubers rather than drunk driving. The dip is more likely due to the smaller amount of people out on the roads, though.

On Friday's, people are probably eager to get home and thus may be a bit more reckless. What strikes me, however, is the continuation of this trend even in 2020. While the numbers drastically dropped from 2019 to 2020 the days of the week in comparison to each other in 2020 still followed the same trends as the other years. Even in 2020, Friday had the highest number of collisions out of all of the days of the week. This would hint that the few people that were commuting to/from work or just out on the road were still in a hurry to get to their destinations.

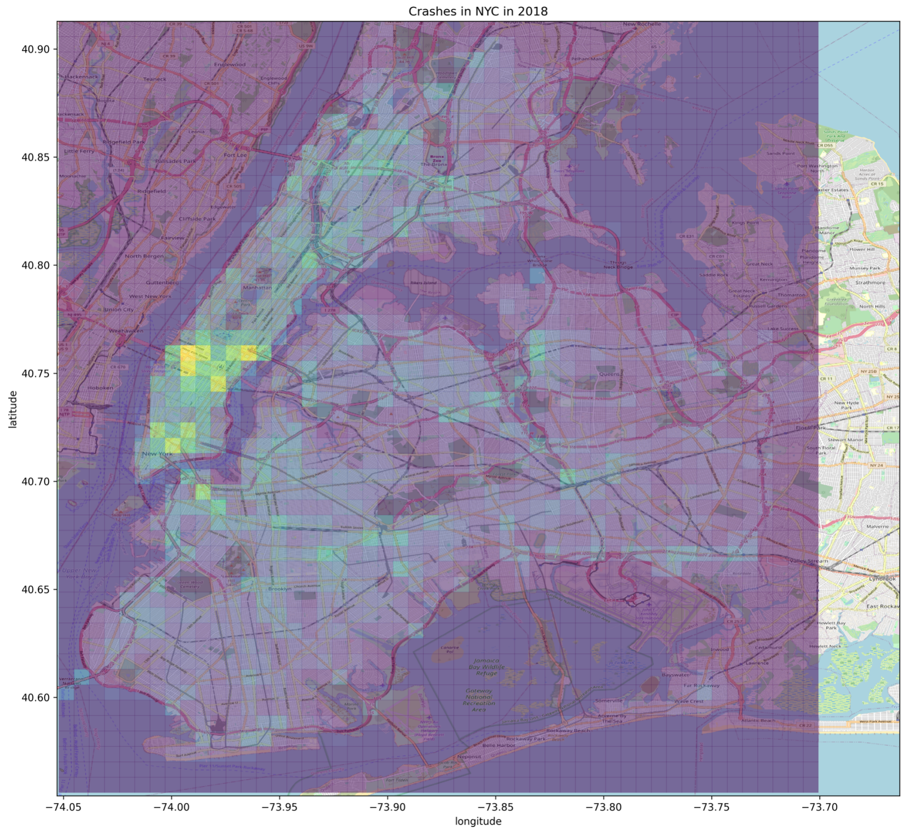
Places with the Most Accidents

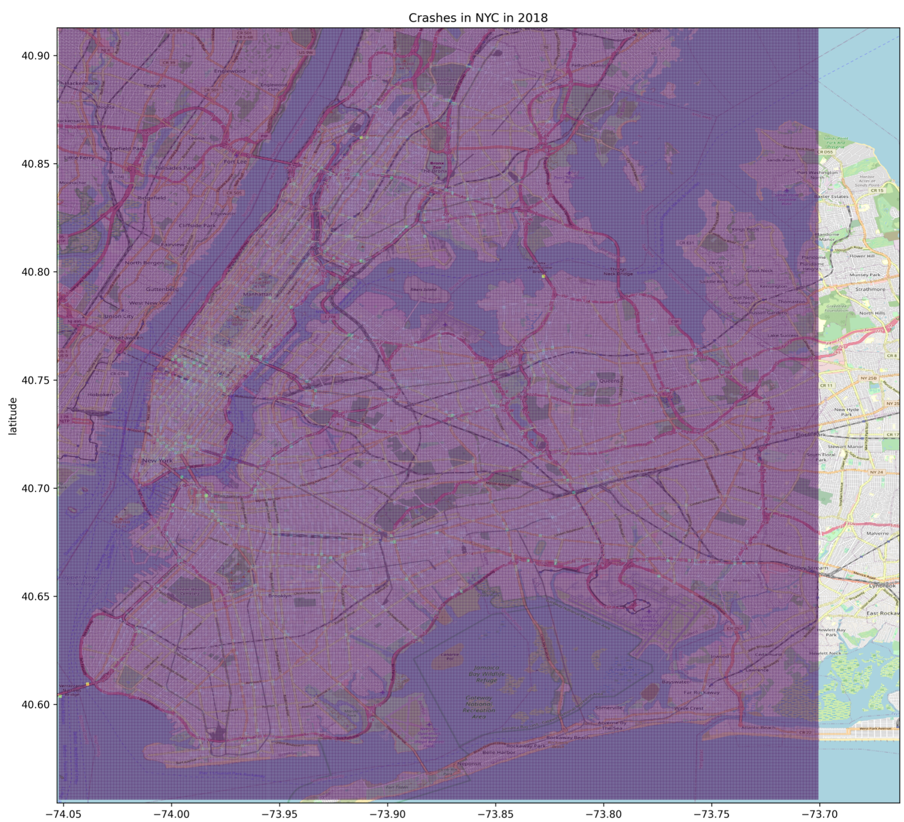
To find the locations with the most accidents, the latitude and longitude was read in, along with the crash dates. The coordinates were then binned and displayed on an image of a map of New York City whose bounds were precisely known. Using a 2d histogram with automatic density mapping and an alpha of 50% produces a histogram produces an effective heat map over the city.

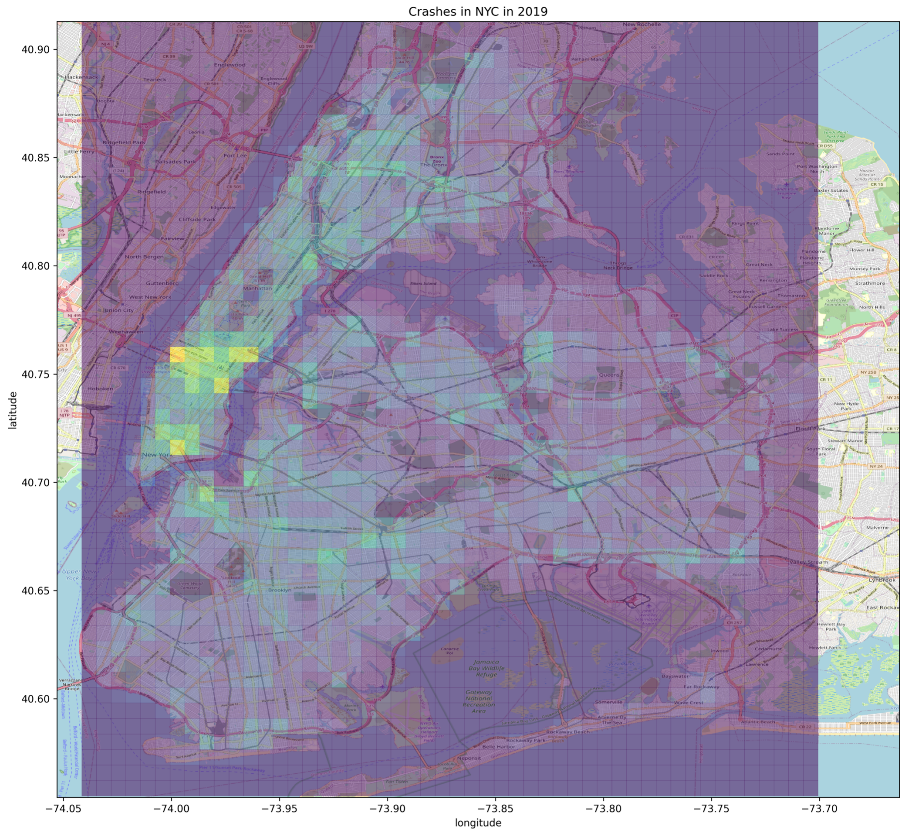
In order to see the general areas of the city where crashes were occuring, a bin size of 50 was selected. This helped reduce the noise present from outliers and anomalies in the data (for example, crashes that occured on what appears on top of a building are very difficult to filter out during preprocessing), while also showing a clear picture. In order to see if there were any micro level changes between the years (occurring more on certain kinds of streets, intersections vs straightaways, etc), a bin size of 250 was chosen. Any smaller and the details became fuzzy, any larger and the data becomes invisible due to the data being too spread out among the bins.

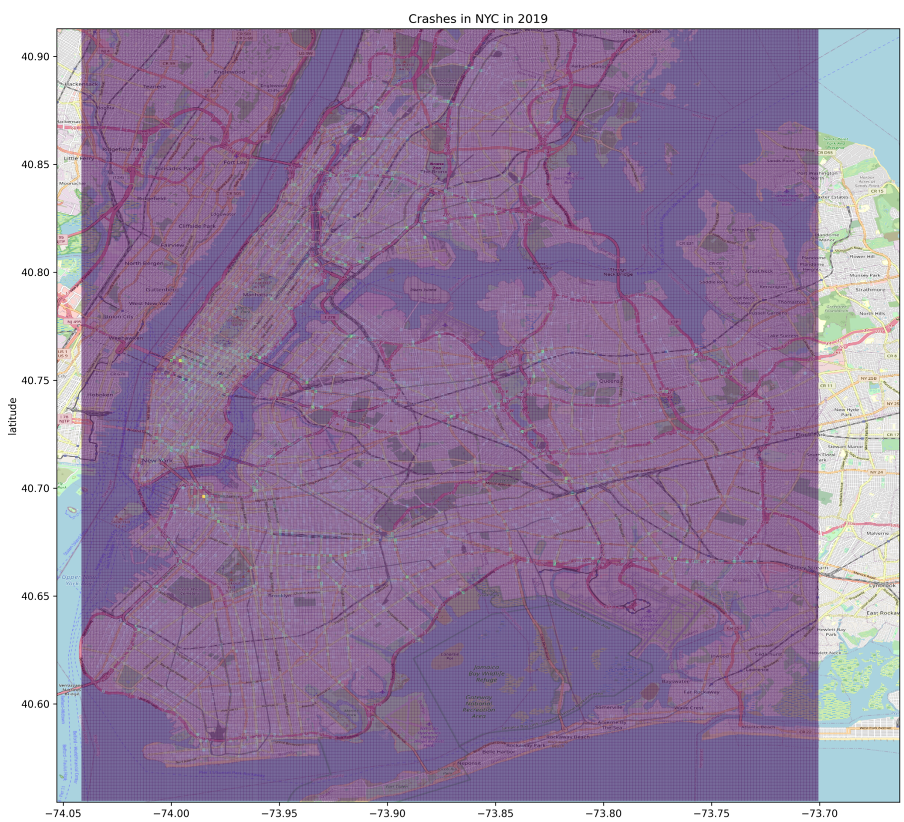
In 2019 and 2018, the majority of crashes occurred in Lower Manhattan, with a lesser but notable amount occuring in Upper Manhattan, Downtown Flushing, and Brooklyn. In 2020, there was a significant shift to the majority of the crashes occurring in Upper Manhattan and Brooklyn. This makes sense given the trends of the pandemic. More people are working from home, rather than visiting office space to go into work. Lower Manhattan (and to a lesser extent, Downtown Flushing) are where the majority of businesses exist in New York City. With fewer deliveries, fewer people taking taxis, fewer people driving to work, and fewer refuse services required for empty buildings, there are less vehicles in the area for accidents to occur. As we saw, the overall crashes in New York City were reduced, but the crashes that did occur were in Upper Manhattan and Brooklyn. These are the areas of the city in which the majority of people live. Vehicular traffic for things like trash collection, food deliveries to grocery stores and restaurants, Amazon deliveries, and other parts of life still need to occur. Additionally, especially in the summer, people were still visiting restaurants, which could mean people driving around. New York City is traditionally navigated by those who live there by taxi, bus, and metro. However, due to health concerns, more people may have been using their own cars, causing an additional uptick in residential accidents.

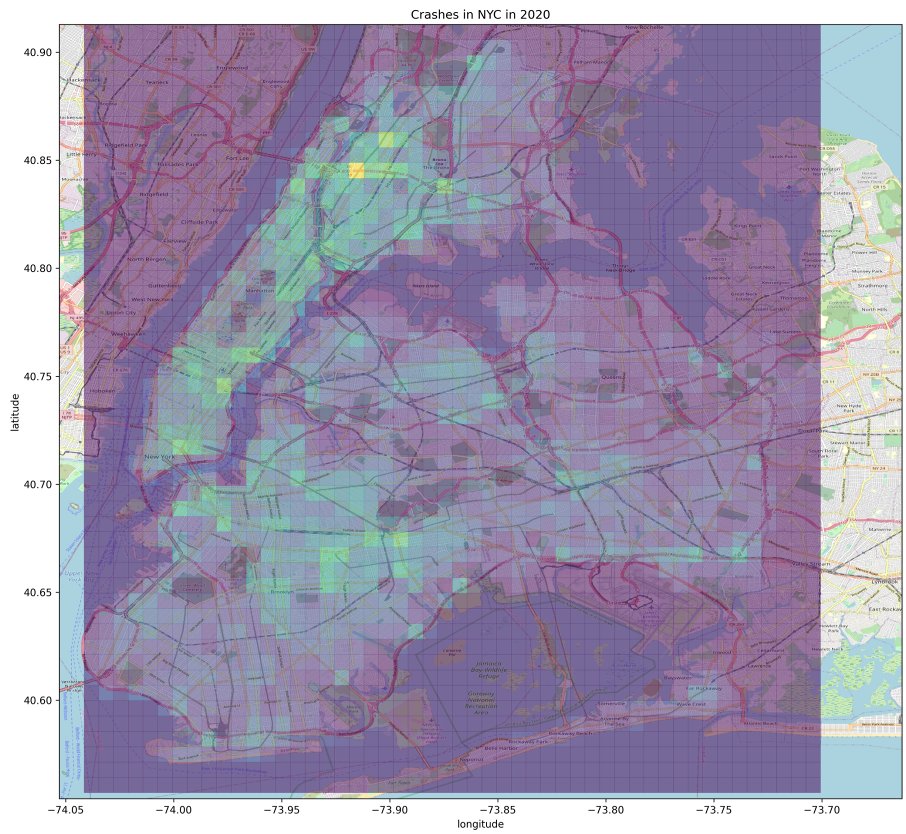
Looking at the maps with smaller bins, in all 3 years, a majority of accidents occurred at cross streets. Vehicles tend to have more accidents when crossing paths (even when controlled by traffic control devices such as stop lights and stop signs) then when traveling in parallel. That being said, it does seem that there was a slight uptick in the percentage of accidents occurring on highways compared to prior years. It can be implied that this was due to reckless or impaired driving early on in the pandemic. During the spring of 2020, the lockdowns were so strict that speed freaks started taking to the streets. News reports began circulating of people drag racing, attempting breaking country speed records, and generally driving poorly on the now empty highways. In addition, an uptick in alcohol consumption due to people being stuck at home may have occurred (Bomey). These combined led to an increase in the percentage of single vehicle accidents, many of which occured on the highways where speeds are higher.

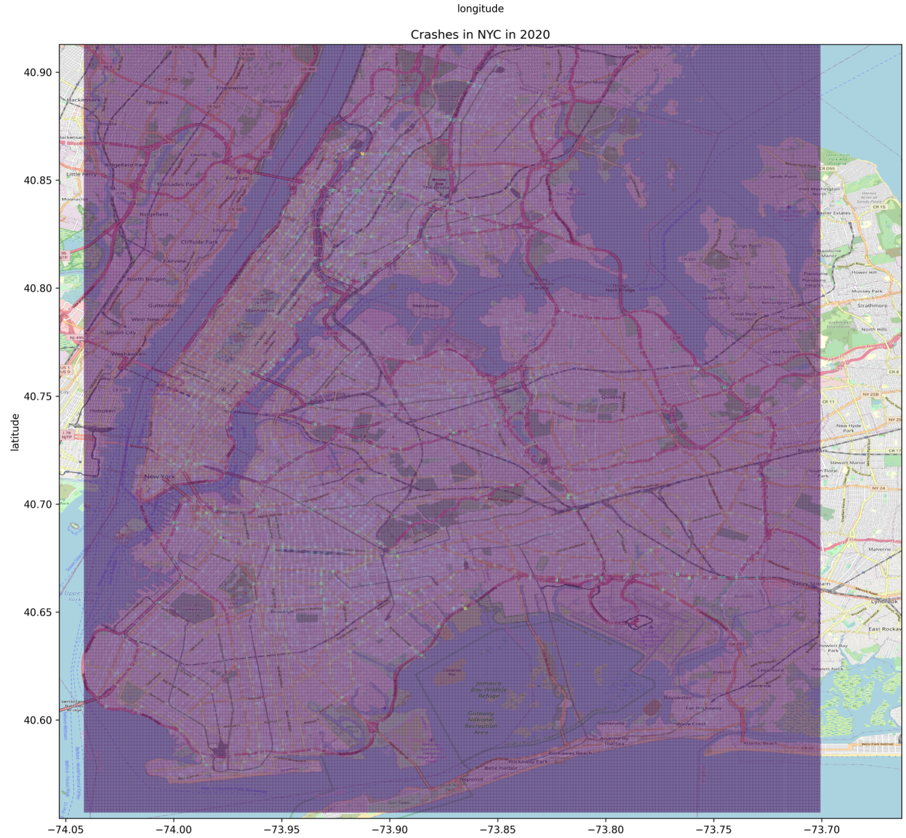

Crashes in NYC in 2018
latitude
longitude
40.90
40.85
40.80
40.75
40.70
40.65
40.60
−74.05
−74.00
−73.95
−73.90
−73.85
−73.80
−73.75
−73.70


Crashes in NYC in 2018
latitude
longitude
40.90
40.85
40.80
40.75
40.70
40.65
40.60
−74.05
−74.00
−73.95
−73.90
−73.85
−73.80
−73.75
−73.70

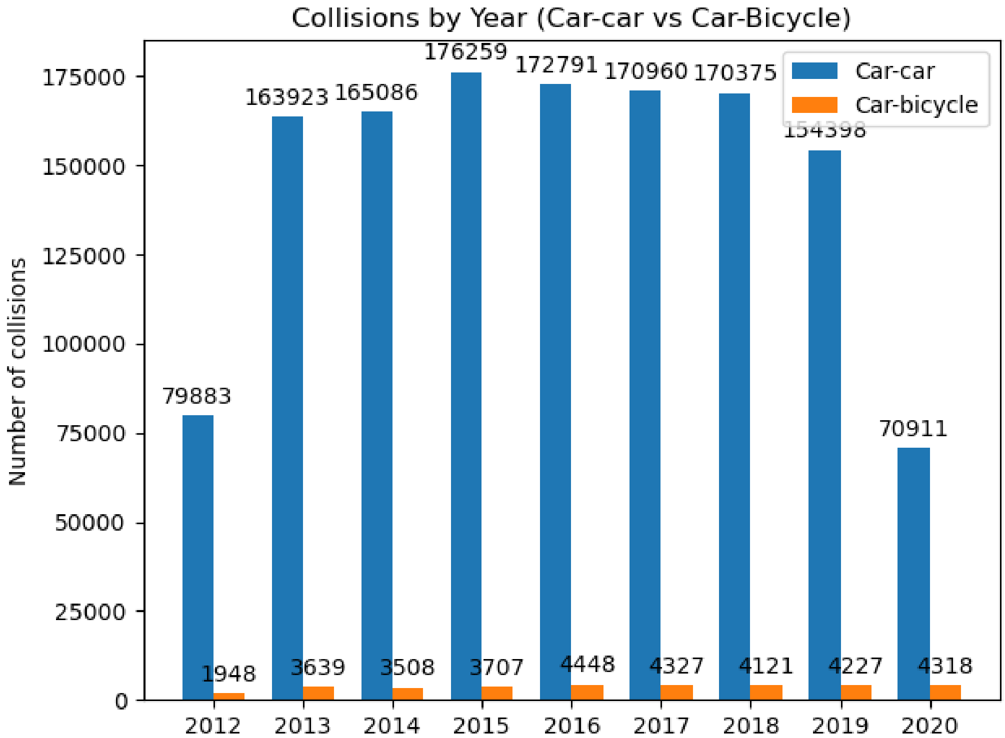

Crashes in NYC in 2019
latitude
longitude
40.90
40.85
40.80
40.75
40.70
40.65
40.60
−74.05
−74.00
−73.95
−73.90
−73.85
−73.80
−73.75
−73.70


Crashes in NYC in 2019
latitude
longitude
40.90
40.85
40.80
40.75
40.70
40.65
40.60
−74.05
−74.00
−73.95
−73.90
−73.85
−73.80
−73.75
−73.70


Crashes in NYC in 2020
latitude
longitude
40.90
40.85
40.80
40.75
40.70
40.65
40.60
−74.05
−74.00
−73.95
−73.90
−73.85
−73.80
−73.75
−73.70


Crashes in NYC in 2020
latitude
longitude
40.90
40.85
40.80
40.75
40.70
40.65
40.60
−74.05
−74.00
−73.95
−73.90
−73.85
−73.80
−73.75
−73.70

Car-only versus Car-pedestrian Accidents



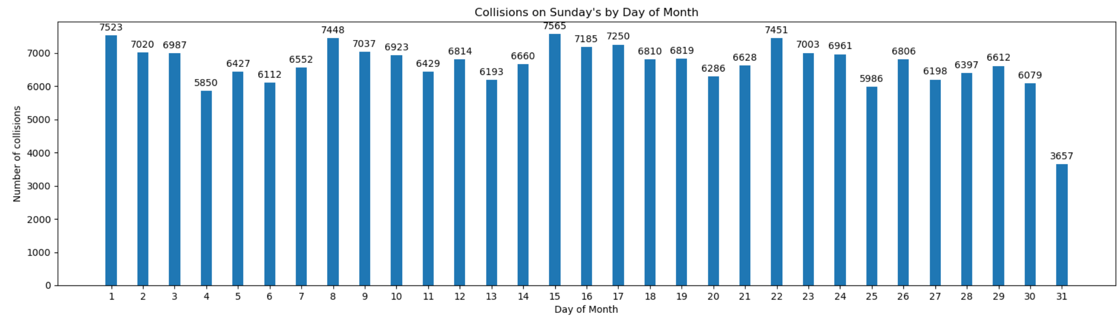
There are some interesting results that we would like to note in this analytic. The first item is the drop from 2018 to 2019. This seems to indicate that the originally slow decline in car-car accidents seemed to greatly speed up (aka decrease much more quickly). The decline from 2019-2020 was clearly caused by the pandemic, so it's interesting to think about what would have happened. Would the numbers have continued trending downwards due to the Vision Zero program? It's also interesting to think about what will happen when the pandemic finally subsides and life returns to normal (or some semblance of normal). Will the number of accidents increase or decrease as compared to 2019?

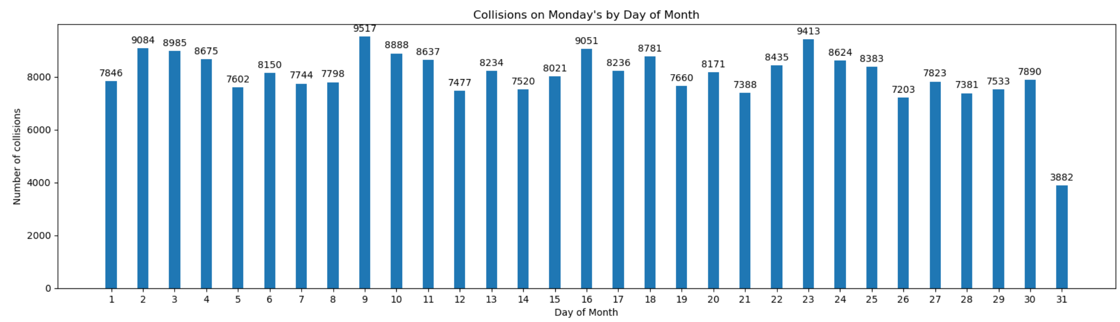
The other result that struck us was the trend in car-bicycle accidents. Not only did the number of accidents of this type not change, but it actually increased in 2020. We could try to speculate different reasons as to why this is such as perhaps there were more bicycles for people just trying to get some fresh air and exercise during the pandemic. Even so, an increase in the number of accidents really would have to mean a very large increase in the number of bicyclists given that the number of cars on the road decreased.

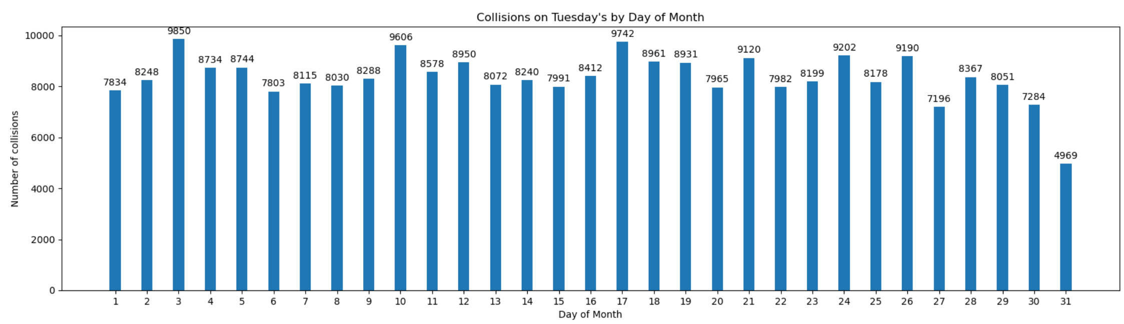
Other Discoveries

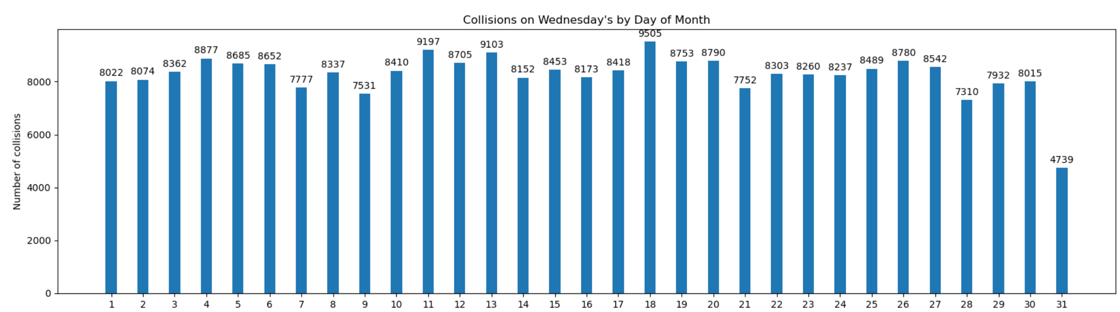
As part of our exploratory data analysis, we thought it would be interesting to see if Friday the 13th had more collisions than any other Friday (Friday the 1st, Friday the 2nd, etc.). For further analysis and possible discovery, we extended this to include all 7 days of the week. The below seven graphs are our results. They all appeared to follow the same general trends where there was a wave up and down of accidents by day of the month for that particular day of the week except for the 31st. See our notes under the Friday graph for some remarks on the differences between Friday and the other graphs.

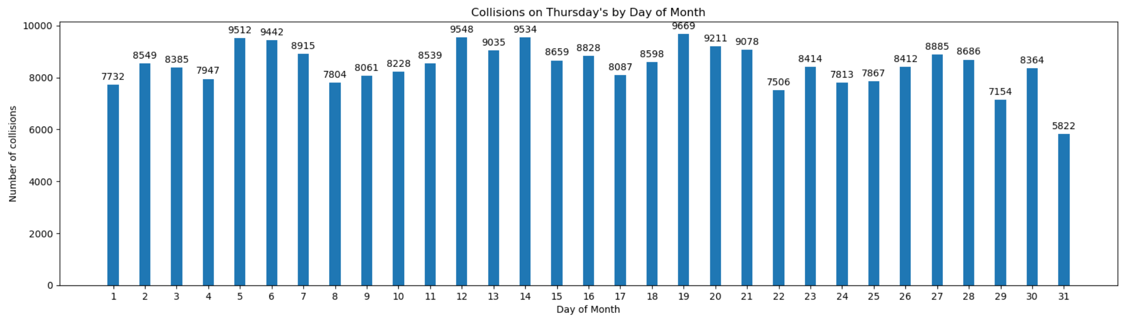
Another graph/analytic that we could potentially explore would be the sum of these graphs together. So, instead of differentiating between days of the week, we could just look at the days of the months by themselves and see what trends there are, if any. You can almost see a matching wave pattern throughout these 7 graphs. This leads to the consideration that what if there is some other association that leads to this wave. Are moon cycles which are a temporally cycling pattern associated with car accidents? If we had more time, we would consider this and run the analytics.

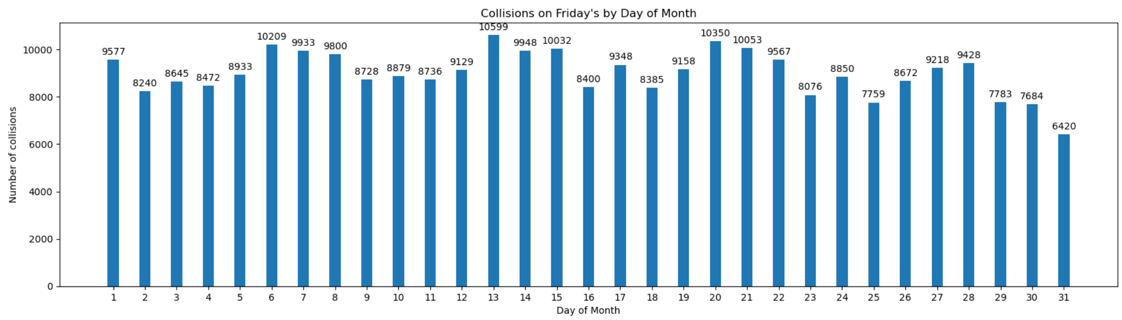






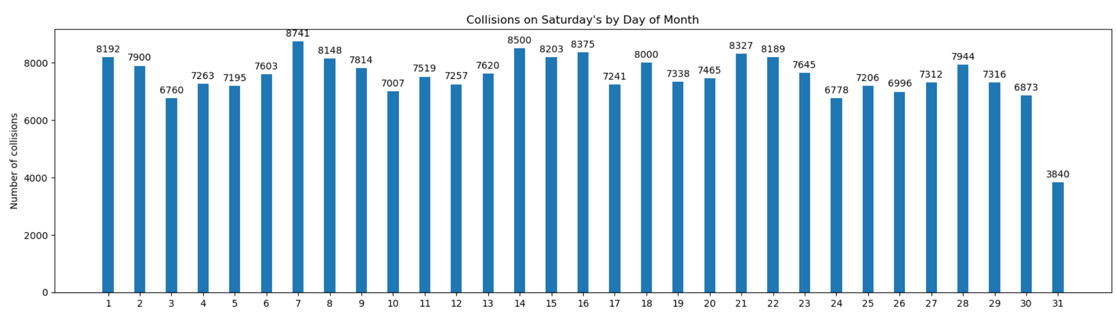






As was our original intention with this data exploration, Friday the 13th did indeed appear to have the highest number of collisions out of all of the Friday's. However, the difference between that particular Friday and all of the other ones was not as drastic as we were hoping for. Thus, while it is an interesting thing to note, it does not appear to be a very important trend.

One thing that we would like to point out for sure, though, is the 31st. If you look at the other days of the week, their 31st is roughly half of the other days of the month. This makes sense as 7 of the 12 months have a 31st day of the month which is roughly half. However, for Friday the 31st, the number of collisions is much more than the half of all of the other days. This kind of ties back into the other analytic for days of the week where Friday seems to just have many more collisions than not.



Looking at the numbers of the graphs, we can see the correlation between what these graphs tell us and what the Days of the Week with the Most Accidents graph shows us. The numbers for Saturday and Sunday tend to lie in the 6000-8000 range, while the weekdays (excluding Friday's) tend to lie in the 7000-9000 range. Finally, as expected, Friday's were the highest of the all the days lying in the 8000-10,000 range.

So, to recap, we did not really find any evidence that any one day of the month tends to be worse than all others, but the graphs did show some interesting general wave patterns that could perhaps be investigated further. We did see that Friday the 13th had the most collisions out of all the Friday's which was fun to observe.

Discussion

Other relevant information to relate.
What went right?
What went wrong?
What was difficult?

For the actual data exploration and analysis (excluding the heat map question), there was not a ton of work that was difficult necessarily. Since we had cleaned the data and inserted it into a database, it was easy to simply write SQL queries for the different analytics we wanted to view. This was a really smart way to go as it allowed for quick querying and modifications to the queries to get the exact results we wanted. We were able to use JetBrains' DataGrip software to connect to our SQLite database and run queries there as well as view the actual database. Using SQL also allowed us to create views on the data to reuse certain results that were used across the different analytics we did.

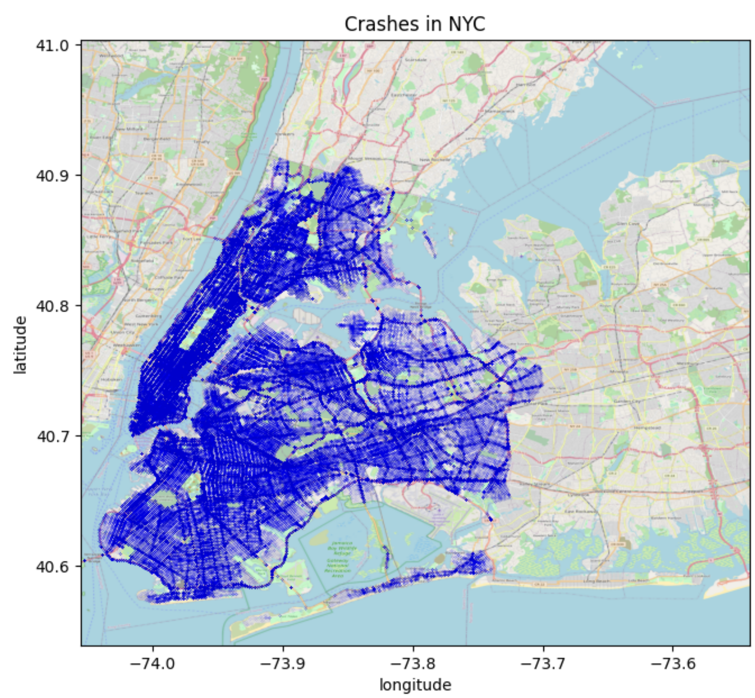
The one difficult part that was a result of some garbage data that was too difficult to clean was the exclusion of some data from our analytics. There were some vehicle types that just could not be recognized by our cleaning mapper. This was not a huge deal but nevertheless does have some impact on our data accuracy and inclusion of all data points. We saw it as a good lesson, though, that in the real world, data collection and cleaning is hard and even harder to perfect. Another example to go along with that is that it does not include pedestrians as a vehicle type. While yes, pedestrians could be considered as not being a vehicle, it still made it difficult and confusing to decide whether to use the columns for pedestrians injured or pedestrians killed as the numbers didn't seem to be 100% consistent. This is the reason why we decided to look at car-car vs car-bicycle accidents rather than car-car vs car-pedestrian accidents.

In order to determine in what areas most accidents occurred, a number of technical challenges had to be overcome. The first of which was reading in all of the CSV in a way that matplotlib could understand. Normally this is trivial, but due to the CSV being nearly 400MB and millions of lines long, the normal way of using Python's built in CSV parser and a Numpy array was simply not feasible. It took nearly 400GB of RAM (compressed by macOS so the system didn't crash) and nearly 10 minutes to accomplish this. The development speed simply wouldn't be adequate. Luckily, moving to a Pandas dataframe and the `read_csv` method entirely solved the problem. The reads now take only a few seconds. The data frame also made it extremely easy to filter out bad data, including crashes outside the desired date range, NaN rows, and crashes that occured in seemingly impossible locations, such as in the Atlantic Ocean.

The biggest issue with creating the mapping image of where crashes occurred was actually generating the map. Due to the precise nature of plotting millions of accident coordinates on the map, the bounds of the map need to be precisely known. Additionally, because New York City is an extremely large area, to glean any useful information from some of the images with small bin sizes, the image of the city had to be extremely high resolution to allow for zooming. Taking a screenshot of Google Maps was simply not high quality enough, nor would it be possible to know the edges of the map. Google has an API for Maps which allows for image export with precise bounds, but they do not allow the quality to be high enough for our purposes due to the fear of competitors stealing their maps. A solution to this could be exporting tiny, higher quality images of slices of the map, but this would require precisely stitching the images back together, which isn't always trivial on a map, not to mention the API quota limit.

OpenStreetMap is an alternative to the major mapping providers such as Google, Bing, and MapQuest. It is fully open source and any one can download the data powering it. On the cloud hosted version of the site, there is a tool to export both the raw map data and an image of a given bounding box. This would be perfect, but unfortunately New York City as a whole was far too large. The site would either time out or throw an error saying to reduce the size of the bounding box. Downloading the data to the whole planet seemed like the next logical step. It appeared to only be 130GB and took around an hour to download from the torrent mirror. However, upon beginning to expand the bz2 archive, it became clear this was going to be a problem. Bz2 is an extremely slow compression algorithm and macOS by default only expands on one thread at a time. The uncompressed file was also over 1.3TB, far too large for an internal SSD. Eventually, a utility called pbzip2 was located that can expand bz2 archives in a multithreaded fashion. When expanded onto an external 2TB hard drive, this allowed the file to fully expand in a relatively short 3 hours. After spinning up a local instance of the OpenStreetMap web server and playing around with the timeout parameters, it was possible to export an extremely precise map of New York City at 1806 × 2201 pixels at 72ppi. Higher resolution would've been possible, but with this PNG already cresting 5MB and the detail being acceptable, more seemed unnecessary.

From this point, it was a matter of figuring out how to display the data on the map. Plotting the individual accident points with extremely small dots as a scatter plot was interesting in that it effectively traced the roads of NYC over the map, but was not very useful. Areas like Central Park are clearly visible with the lack of crashes.



The next step was figuring out how to bin the data appropriately to display in a heatmap. Luckily, matplotlib's `hist2d` method was able to handle this automatically, and generate a heatmap. It took adjusting the bin size, figure size, and output image ppi to eventually get graphs/maps that matched what was needed appropriately.

One area that would be nice to explore is smoothing out the heatmap. As is, the heat map is chunked into square bins, which is not not necessarily the most visually appealing. One potential solution to this is convolving the data with a Gaussian 2D Kernel from something like the astropy library. Initial attempts at this produced either wildly incorrect graphs, took too long to finish executing, or created data structures that were not correct.

Exploring parzen density estimation would also be interesting, although proved to be nonessential with the binned heatmaps. It could create linear graphs of how accidents occurred on latitude versus longitude. With the axes combined, this would most likely reveal the same data as the heatmaps, although potentially with more precision. The binning process can only be so precise, because otherwise the hotspots begin to disappear. The same would happen with parzen, but probably to a lesser to degree.

Conclusion

A general conclusion about what you did and what you learned.

This project gave us a lot to do and helped us learn a lot. One major thing we learned was the hassle of data cleaning. The data we have worked with in past homeworks has been ready-to-go data, meaning there is almost no work to be done before the analytics could happen. This was certainly not the case with the traffic data. There was a lot of work to be done to get some of the attributes in a usable state. One of the important points from class was how a large part of the time spent on data science is cleaning and understanding the data. This project definitely helped that set in and gave us a full view on what data science is actually like. We had to make important decisions about how far to go with our data cleaning process. For example, "how good is good enough?", in terms of grouping and organizing data. This question really came up when it came to dealing with attributes that were clearly raw user input. There was inconsistent spelling, abbreviations, random casing, excess whitespace, and other issues that all had to be dealt with. It was something that you only pick up by actually sitting down and doing it, that can't be fully appreciated from just being told. It was also surprisingly difficult to clean the data when the data set is so large that it cannot be easily visualized.

Another thing that we learned was how sometimes surprising trends may emerge from the data that you weren't originally expecting. When we did the analytic on the number of accidents by day of the week, we were expecting something more along the lines of higher numbers of collisions on the weekends or on Monday's due to people perhaps not being as careful due to intoxication or lack of sleep. However, it turned out that Friday's were the most likely to have accidents. In hindsight, this did make sense as people may be in more of a rush to get home or perhaps traveling for the weekend. The same thing occurred when analyzing where accidents occured. Intuitively, it makes sense that a higher percentage of accidents occurred on the highway compared to at street corners due to single vehicle accidents, but it is not something that we thought of immediately. Data tends to manifest itself in ways that make sense later on, but are not intuitive at first. The point here is that we went in expecting one thing and got another back which was refreshing in a way. The data does not lie, so it was cool to see that in action and not allow bias to be a part of the issue, at least this time.

While doing more of this exploratory data analysis, something else that we noticed was the insanely huge amount of possibilities of ways we can analyze the data. As we hinted at earlier in this report, we could have considered moon cycles or weather patterns and looked at how those affected (or didn't affect) the number of collisions. We could also have looked when certain events happened like marches or sporting events or performing art events or parties or anything else similar to that and seen if that affected the number. This taught us that in data analysis, there is truly a wide universe of possibilities that we can explore. It is for this reason that we as data scientists must decide what is important enough to look at within the scope of the given project. We must determine what is relevant, actionable information for the client of the project and what is either just interesting but irrelevant or completely irrelevant results. This is an important responsibility and one that should not be taken lightly as data scientists are the ones directly dealing with the data and transferring that information to the end users.

One last thing that was discovered in general data analysis was the power that comes with designing graphs and other visualizations for our data. Sometimes, it seems straightforward as far as which graphs to use or how to lay them out, but sometimes it’s not. There are important decisions to be made as far as which axes to use, how to appropriately scale them and whether to include values in the actual graph or let the human eye make their own analysis. Fortunately for us, most of our decisions were just which type of graph to use which mostly was straightforward and logical to do. However, in general, there are many decisions to make and possibilities to consider that are crucial to the effective display of data trends for those not intimately familiar with the data.

A surprising source of frustration was with combining multiple sources of data together. Something like a map is so commonplace in applications such as Waze or Google Maps that it seems trivial to get one. However, getting high quality maps was surprisingly difficult. It took terabytes of map data, hours of processing, and trial and error to get to the finished product. This again just proves the point that data cleaning and analysis is half the battle, the actual code is not on the forefront.

Works Cited

Bomey, Nathan. "Rate of Traffic Deaths Rises during COVID as Open Roads Tempt Drivers to Speed Up." USA Today, Gannett Satellite Information Network, 28 Jan. 2021, www.usatoday.com/story/money/cars/2021/01/28/iihs-aaa-crash-test-speeding-traffic-safety/4265112001/.

Vision Zero. www1.nyc.gov/content/visionzero/pages/.