New York City Motor Vehicle and People Collision Analysis

Abstract

For this project, our group investigated the details of motor vehicle collisions in New York City. Our key findings include: a higher number of crashes occur during the summer months; distracted driving is the most common cause of collisions; car crashes are more frequent than motorcycle crashes, but motorcycle crashes tend to be more fatal; collisions are most likely to occur between 3 PM and 8 PM; and there is a higher density of motor vehicle collisions in tourist-heavy areas of Manhattan compared to the rest of NYC. Understanding the factors that contribute to collisions is essential for implementing effective precautionary measures.

Introduction

This topic was chosen for our project because motor vehicle collisions in highly populated cities can be a significant issue. Almost 400,000 collisions have occurred in the state of New York alone in the year 2023. We wanted to assess the many factors of these collisions to ensure people, pedestrians or drivers, keep them in mind when in New York City. Specifically:

- What time of day do collisions frequently occur?
- What neighborhood are these collisions most likely to happen?
- Are motorcycles or cars more likely to cause fatal injury?
- What is the most likely cause of collisions?
- Do crash rates vary by months?
- Do crashes occur frequently in tourist locations or are they dispersed throughout?

These statistics are important for people to care about so they can take necessary precautions when in New York City to mitigate collisions. This can take form in pedestrians being sure to not jaywalk, or drivers having a heightened awareness and focus on the road.

Explanation of Data

The datasets that were chosen were selected due to their complexity. They consisted of data from 2016 to 2022. The datasets chosen are from the data.gov website. One dataset focused on the details of the crash event in motor vehicle collisions, and the second one focused on details of the people involved in the motor vehicle collisions. The data is collected by the New York Police Department.

While the data is publicly available, individuals involved in the collisions may not have given explicit consent for their information to be used research studies, but no identifiable information is present in the data so it adheres to ethical standards.

Motor Vehicle Collisions - Crashes - Catalog (data.gov) (https://catalog.data.gov/dataset/motor-vehicle-collisions-crashes)

Motor Vehicle Collisions - Person - Catalog (data.gov) (https://catalog.data.gov/dataset/motor-vehicle-collisions-person)

The data is freely available for anyone to use and view. At the bottom of the webpage where the dataset was retrieved, it was stated that this data was “Intended for public access and use.” The population of the dataset are the people in New York City, and the sample is the details of the people involved in collisions.

Description of Data

In the first dataset, there were 2.11 million rows and 29 columns. In the second data set, there were 5.44 million rows and 21 columns. Each row represents a collision. The relevant variables are most of the columns, minus a few that were deemed irrelevant upon review, such as duplicates of already established variables that were mostly empty. Missing values in specific columns that were kept were omitted based on how each of us organized our data to answer a specific research question.

Cleaning Data Before Working With It

Before working with our data, it was necessary to clean it and make necessary changes before working with it to solve our research questions. First, the necessary libraries had to be loaded.

```
library(tigris)
```

```
## To enable caching of data, set `options(tigris_use_cache = TRUE)`
## in your R script or .Rprofile.
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(leaflet)
library(sp)
library(ggmap)
```

```
## Loading required package: ggplot2
```

```
## ℹ Google's Terms of Service: <https://mapsplatform.google.com>
##   Stadia Maps' Terms of Service: <https://stadiamaps.com/terms-of-service/>
##   OpenStreetMap's Tile Usage Policy: <https://operations.osmfoundation.org/policies/tiles/>
## ℹ Please cite ggmap if you use it! Use `citation("ggmap")` for details.
```

```
library(broom)
library(httr)
library(sf)
```

```
## Linking to GEOS 3.12.1, GDAL 3.8.4, PROJ 9.3.1; sf_use_s2() is TRUE
```

```
library(ggplot2)
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ forcats   1.0.0     ✓ stringr   1.5.1
## ✓ lubridate 1.9.3     ✓ tibble    3.2.1
## ✓ purrr     1.0.2     ✓ tidyr     1.3.1
## ✓ readr     2.1.5
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
e errors
```

```
library(magick)
```

```
## Linking to ImageMagick 6.9.12.98
## Enabled features: cairo, freetype, fftw, ghostscript, heic, lcms, pango, raw, rsvg, webp
## Disabled features: fontconfig, x11
```

```
library(data.table)
```

```
##
## Attaching package: 'data.table'
##
## The following objects are masked from 'package:lubridate':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year
##
## The following object is masked from 'package:purrr':
##
##     transpose
##
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

After this, we loaded our data by storing the datasets into a `crashes` and `person` variable, named after the specification of their datasets. Then to join the datasets, we first had to modify some variable names so that they can automatically join together. Both datasets had columns that indicated their crash date and time, but were named differently. The columns that indicated the crash date and time in the `crashes` dataset were modified to match how the date and time columns were labeled in the `person` dataset. Therefore, the final merge keys were `CRASH_DATE` , `CRASH_TIME` , and `COLLISION_ID`

```
crashes <- read_csv("Motor_Vehicle_Collisions_-_Crashes.csv")
```

```
## Rows: 2109518 Columns: 29
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (16): CRASH DATE, BOROUGH, LOCATION, ON STREET NAME, CROSS STREET NAME,...
## dbl  (12): ZIP CODE, LATITUDE, LONGITUDE, NUMBER OF PERSONS INJURED, NUMBER ...
## time  (1): CRASH TIME
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
person <- read_csv("Motor_Vehicle_Collisions_-_Person.csv")
```

```
## Rows: 5438651 Columns: 21
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (16): CRASH_DATE, PERSON_ID, PERSON_TYPE, PERSON_INJURY, EJECTION, EMOT...
## dbl   (4): UNIQUE_ID, COLLISION_ID, VEHICLE_ID, PERSON_AGE
## time  (1): CRASH_TIME
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
crashes <- crashes %>%
  rename(CRASH_DATE = `CRASH DATE`,
         CRASH_TIME = `CRASH TIME`)
```

After this, the datasets were merged using a `full_join()` and stored in a variable titled `crashesPerson`. Unnecessary columns then had to be deleted. The `subset()` function was used to do so. From both datasets, the deleted functions consisted of:

- OFF STREET NAME
- VEHICLE TYPE CODE 3
- VEHICLE TYPE CODE 4
- VEHICLE TYPE CODE 5
- EJECTION
- PED_LOCATION
- PED_ACTION
- NUMBER OF CYCLIST KILLED
- SAFETY_EQUIPMENT

```
crashesPerson <- full_join(crashes, person)
```

```
## Joining with `by = join_by(CRASH_DATE, CRASH_TIME, COLLISION_ID)`
```

```
crashesPerson <- subset(crashesPerson, select = -c(`OFF STREET NAME`,
                                                   `VEHICLE TYPE CODE 3`,
                                                   `VEHICLE TYPE CODE 4`,
                                                   `VEHICLE TYPE CODE 5`,
                                                   EJECTION,
                                                   PED_LOCATION,
                                                   PED_ACTION,
                                                   `NUMBER OF CYCLIST KILLED`,
                                                   SAFETY_EQUIPMENT))
```

Creating a New Categorical Variable

To create a new categorical variable, there was a focus on categorizing injury severity. This was done by categorizing a certain number of people injured or killed under:

- “Fatal Injury”
- “Major Injury”
- “Minor Injury”
- “No Injury”

```
crashesPerson$INJURY_SEVERITY <-  ifelse(crashesPerson$`NUMBER OF PERSONS KILLED` > 0, "Fatal In
jury",
  ifelse(crashesPerson$`NUMBER OF PERSONS INJURED` > 5, "Major Injury",
    ifelse(crashesPerson$`NUMBER OF PERSONS INJURED` > 0, "Minor Injury","No Injury"
    )
  )
)

head(crashesPerson$INJURY_SEVERITY,7)
```

```
## [1] "Minor Injury" "Minor Injury" "Minor Injury" "Minor Injury" "Minor Injury"
## [6] "Minor Injury" "No Injury"
```

Creating a New Numerical Variable

To create a new numerical variable, there was a focus on quantifying the total number of injuries of that collision

```
crashesPerson$TOTAL_INJURIES <- crashesPerson$`NUMBER OF PERSONS INJURED` +
  crashesPerson$`NUMBER OF PEDESTRIANS INJURED` +
  crashesPerson$`NUMBER OF CYCLIST INJURED` +
  crashesPerson$`NUMBER OF MOTORIST INJURED`

head(crashesPerson$TOTAL_INJURIES,7)
```

```
## [1] 4 4 4 4 2 2 0
```

Creating a Summarization Table

For the summarization table, there was a focus on finding the emotional status of each person relative to their position in the car.

```
emotionPosition <- crashesPerson %>%
  group_by(EMOTIONAL_STATUS, POSITION_IN_VEHICLE) %>%
  summarize(totalPeople = n(),
            totalInjuries = sum(TOTAL_INJURIES, na.rm = TRUE),
            totalFatalities = sum(`NUMBER OF PERSONS KILLED`, na.rm = TRUE),
            avgInjuriesPerPerson = mean(TOTAL_INJURIES, na.rm = TRUE),
            avgFatalitiesPerPerson = mean(`NUMBER OF PERSONS KILLED`, na.rm = TRUE),
            .groups = 'drop') %>%
  arrange(EMOTIONAL_STATUS,POSITION_IN_VEHICLE)

head(emotionPosition,3)
```

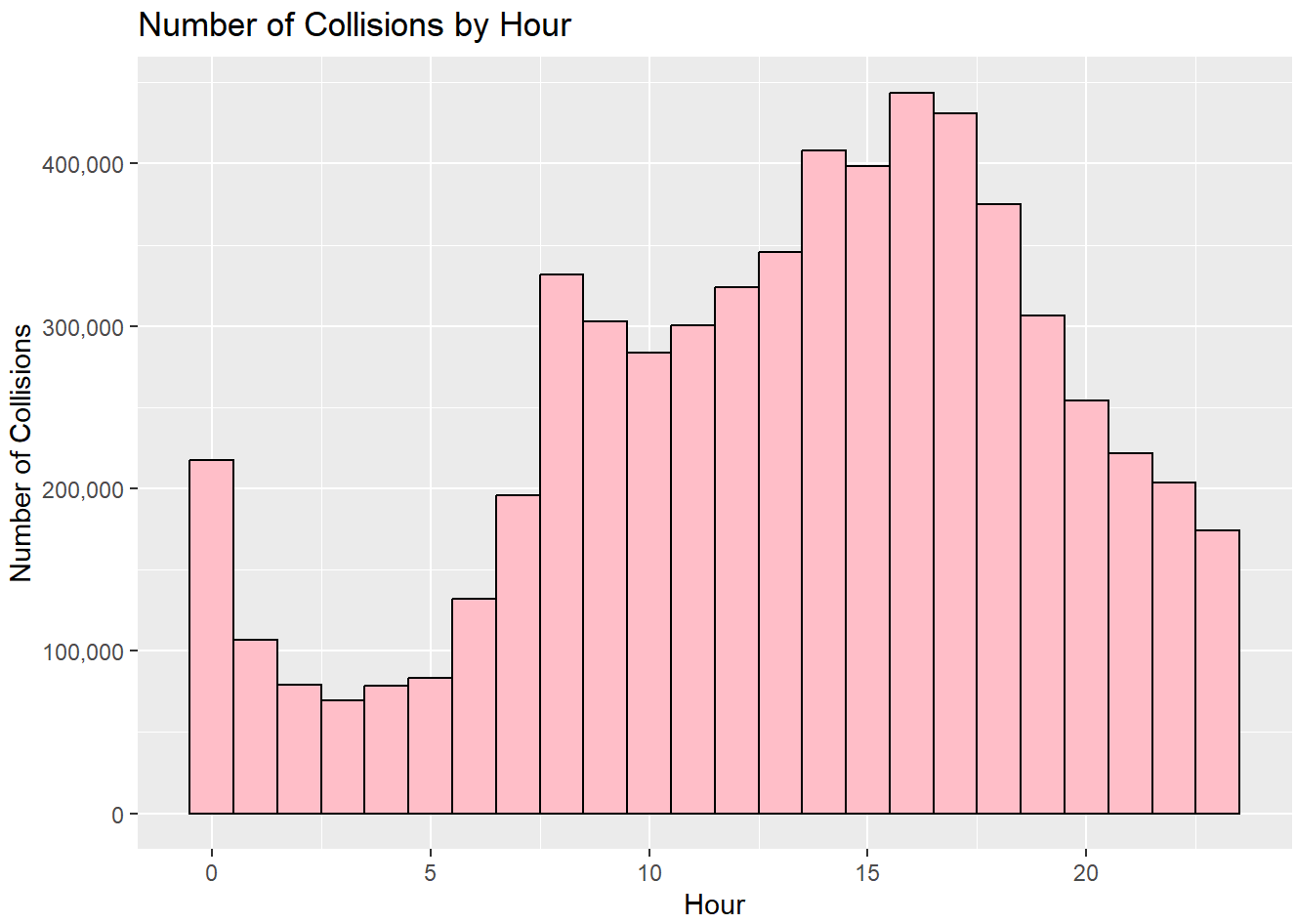
```
## # A tibble: 3 × 7
##   EMOTIONAL_STATUS POSITION_IN_VEHICLE totalPeople totalInjuries totalFatalities
##   <chr>            <chr>                      <int>         <dbl>           <dbl>
## 1 Apparent Death   Any person in the …            6            62               9
## 2 Apparent Death   Does Not Apply                10             2               5
## 3 Apparent Death   Driver                       848           745             694
## # ℹ 2 more variables: avgInjuriesPerPerson <dbl>, avgFatalitiesPerPerson <dbl>
```

Research Questions

What time of day do collisions frequently occur?

The main variable that was used was CRASH_TIME. This was the only variable needed for this particular research question because it was looking into the frequency of what time of day that car crashes occurred. To process the data, I had to extract each value out of CRASH_TIME and change them into a numeric value. Basically, I grabbed hour and minute and changed them into numbers that could be plotted on a bar graph. A bar graph was chosen because I was working exclusively with numerical values. For the sample, I used all possible observations in order to gather the most accurate conclusion about which time of day car crashes occur the most often.

```
crashesPerson$CRASH_TIME <- as.character(crashesPerson$CRASH_TIME)
crashesPerson$HOUR <- as.numeric(sub(":.*", "", crashesPerson$CRASH_TIME))
crashesPerson$MINUTE <- as.numeric(sub(".*:", "", crashesPerson$CRASH_TIME))

ggplot(crashesPerson, aes(x = HOUR)) +
  geom_histogram(bins = 24, fill = "pink", color = "black") +
  labs(title = "Number of Collisions by Hour",
       x = "Hour",
       y = "Number of Collisions") +
  scale_y_continuous(labels = function(x) format(x, big.mark = ",", scientific = FALSE))
```



Trade-Offs:

While this graph does give the time of day that collisions are mostly to occur, it could be more specific to include other variables. Another thing to consider is the accuracy of the data. The dataset fails to mention fully how it went about collecting the crash time, meaning the accuracy of it can be questioned. To add onto this, another question surfaces: were there unrecorded crashes? However, with this question, these unrecorded crashes would likely not be enough to significantly affect the data.

Result:

Based on the graph, the hour range that collisions occur most frequently in is from 15-20. This is around 3pm to 8pm. I was surprised by this observation because I assumed that the number of collisions would be larger in the later evening hours. Another thing to note is that the graph reaches its peak around the 17-18 interval, indicating that there are likely more cars on the road around this time. The amount of accidents around the 0-5 interval are the lowest, but there is a visible outlier at 0:00 or midnight.

Findings:

It is likely that many people commute from work around 3-8pm, making the likelihood of accidents increase with the amount of vehicles on the road. Pushing into the later evening hours, drivers may have a harder time navigating through the roads. For the outlier at midnight, this may be a result of reckless driving as people are often going out drinking or partying at these hours. Another factor that could contribute to this figure is decreased visibility at night, making it all the more likely for drivers to get into accidents. The limitation from these findings is

that my graph fails to mention the specific day that collisions occurred, which would provide more conclusive results. For the future, a graph that includes the time of day could tell a story of whether these crashes are occurring on weekends or working days.

In short, collisions occurred most often at the **3pm to 8pm** mark with the peak being at 4pm. Additionally, collisions have a small, but significant peak at midnight. For future road safety in NYC, the government can look at this data to note which time of day that the frequency of collisions take place the most and go from there to improve road conditions. Drivers can also look at this data to figure out which time of day to be most vigilant as well.

Do crashes occur frequently in tourist locations or are they dispersed throughout the state?

The cleaned data was plotted on a coordinate plane using the `LATITUDE` and `LONGITUDE` variables; cleaned vectors and ggplot package were used. However, because of the vast amount of collisions from the past several years, the density of collisions were unclear. To calculate and compare the actual densities of collisions in the top 3 most visited tourist locations, Times Square, Grand Central Terminal, and Central Park, the surrounding areas for each location became parameters to find how many collisions occurred within these boundaries. The area (km^2) was then calculated for each location, including all of NYC, and this found the crashes/km^2 in all of the 3 locations. This finally was compared to the average crashes/km^2 in the whole city.

```
na.omit(crashesPerson)
```

```
## # A tibble: 0 × 42
## # ℹ 42 variables: CRASH_DATE <chr>, CRASH_TIME <chr>, BOROUGH <chr>,
## #   ZIP CODE <dbl>, LATITUDE <dbl>, LONGITUDE <dbl>, LOCATION <chr>,
## #   ON STREET NAME <chr>, CROSS STREET NAME <chr>,
## #   NUMBER OF PERSONS INJURED <dbl>, NUMBER OF PERSONS KILLED <dbl>,
## #   NUMBER OF PEDESTRIANS INJURED <dbl>, NUMBER OF PEDESTRIANS KILLED <dbl>,
## #   NUMBER OF CYCLIST INJURED <dbl>, NUMBER OF MOTORIST INJURED <dbl>,
## #   NUMBER OF MOTORIST KILLED <dbl>, CONTRIBUTING FACTOR VEHICLE 1 <chr>, …
```

```
leaflet() %>%
  addTiles() %>%
  setView(-74.00, 40.71, zoom = 12) %>%
  addProviderTiles("CartoDB.Positron")
```


```
set.seed(42)
lats <- 40.7544882 + rnorm(10)/100
lngs <- -73.9879923 + rnorm(10)/200
points <- data.frame(lat=lats, lng=lngs)
points
```

```
##         lat       lng
## 1  40.76820 -73.98147
## 2  40.74884 -73.97656
## 3  40.75812 -73.99494
## 4  40.76082 -73.98939
## 5  40.75853 -73.98866
## 6  40.75343 -73.98481
## 7  40.76960 -73.98941
## 8  40.75354 -74.00127
## 9  40.77467 -74.00019
## 10 40.75386 -73.98139
```

```
v1 <- crashesPerson[["LATITUDE"]]
head(v1,5)
```

```
## [1] NA NA NA NA NA
```

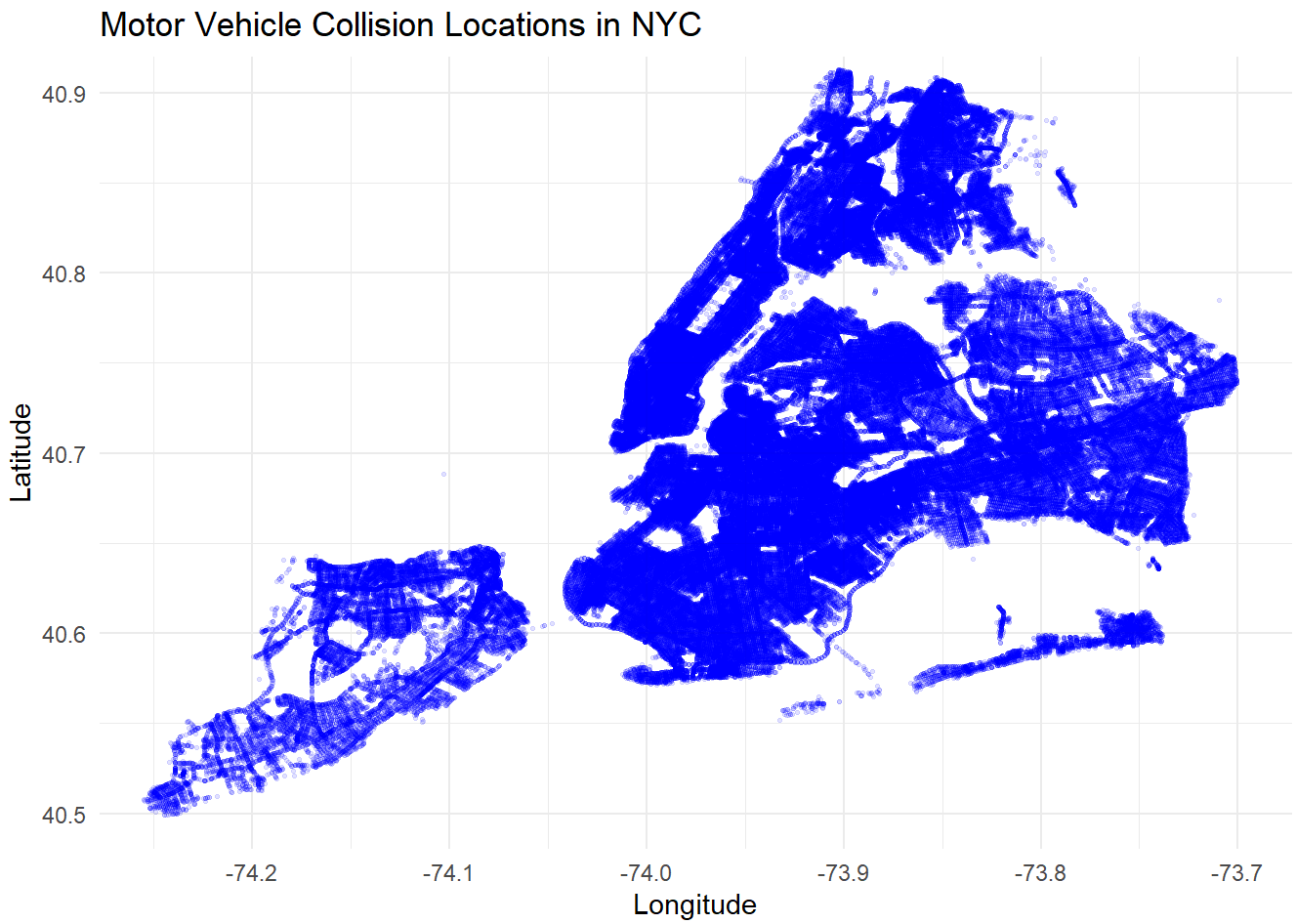
```
v2 <- crashesPerson[["LONGITUDE"]]
head(v2,5)
```

```
## [1] NA NA NA NA NA
```

```
omitV1 <- na.omit(v1)
omitV2 <- na.omit(v2)

cleaned_data <- crashesPerson[!is.na(crashesPerson$LATITUDE) & !is.na(crashesPerson$LONGITUDE),
]
cleaned_data$LATITUDE <- as.numeric(cleaned_data$LATITUDE)
cleaned_data$LONGITUDE <- as.numeric(cleaned_data$LONGITUDE)

cleaned_data <- cleaned_data %>%
  group_by(LATITUDE, LONGITUDE) %>%
  summarize(total_collisions = n())
```

```
## `summarise()` has grouped output by 'LATITUDE'. You can override using the
## `.groups` argument.
```

```
#40.480441, -74.291246 (SW)
#40.914272, -74.276370 (NW)
#40.920498, -73.709201 (NE)
#40.496812, -73.705081 (SE)

ggplot(cleaned_data, aes(x = LONGITUDE, y = LATITUDE)) +
  geom_point(color = "blue", alpha = 0.1, size = 0.5) +
  labs(title = "Motor Vehicle Collision Locations in NYC",
       x = "Longitude",
       y = "Latitude") +
  theme_minimal() +
  coord_cartesian(xlim = c(-74.25, -73.7), ylim = c(40.5, 40.9))
```




```
#get number of observations
# 363,170 clean observations
# 2022: 20,309 observations in 365 days = 55.64 crashes/day
# 2021: 34,571 observations in 365 days = 94.72 crashes/day
# 2020: 40,875 observations in 365 days = 111.99 crashes/day
# 2019: 81,558 observations in 365 days = 223.45 crashes/day
# 2018: 92,884 observations in 365 days = 254.48 crashes/day
# 2017: 92,858 observations in 365 days = 254.41 crashes/day
# 2016: 40,451 observations in 365 days = 110.82 crashes/day

# average crashes per day in years of 2016, 2017, 2018, 2019, 2020, 2021, and 2022
# = 121.58
# (given that NA observations were omitted, leading to likely a lower average of number of crash
es per day in NYC)

avg_per_day <- 121.58
```

```
#times square
min_latitude <- 40.750455
max_latitude <-  40.763504
min_longitude <- -73.995443
max_longitude <- -73.978149

points_in_region <- cleaned_data %>%
  filter(LATITUDE >= min_latitude & LATITUDE <= max_latitude) %>%
  filter(LONGITUDE >= min_longitude & LONGITUDE <= max_longitude)

ts_number_of_points <- nrow(points_in_region)
ts_number_of_points
```

```
## [1] 3126
```

```
#times square area
min_latitude <- 40.750455
max_latitude <- 40.763504
min_longitude <- -73.995443
max_longitude <- -73.978149

latitude_difference <- max_latitude - min_latitude
longitude_difference <- max_longitude - min_longitude


latitude_length_km <- 111
longitude_length_km <- 111 * cos((min_latitude + max_latitude) / 2 * pi / 180)


height_km <- latitude_difference * latitude_length_km
width_km <- longitude_difference * longitude_length_km


ts_area_km2 <- height_km * width_km
ts_area_km2
```

```
## [1] 2.106168
```

```
ts_crash_density <- ts_number_of_points/ts_area_km2
ts_crash_density
```

```
## [1] 1484.212
```

```
#grand central terminal
min_latitude <- 40.748265
max_latitude <- 40.757985
min_longitude <- -73.982760
max_longitude <- -73.971194


points_in_region <- cleaned_data %>%
  filter(LATITUDE >= min_latitude & LATITUDE <= max_latitude) %>%
  filter(LONGITUDE >= min_longitude & LONGITUDE <= max_longitude)


gct_number_of_points <- nrow(points_in_region)
gct_number_of_points
```

```
## [1] 1172
```

```
#grand central terminal area
min_latitude <- 40.748265
max_latitude <- 40.757985
min_longitude <- -73.982760
max_longitude <- -73.971194

latitude_difference <- max_latitude - min_latitude
longitude_difference <- max_longitude - min_longitude

latitude_length_km <- 111
longitude_length_km <- 111 * cos((min_latitude + max_latitude) / 2 * pi / 180)

height_km <- latitude_difference * latitude_length_km
width_km <- longitude_difference * longitude_length_km

gct_area_km2 <- height_km * width_km
gct_area_km2
```

```
## [1] 1.049288
```

```
gct_crash_density <- gct_number_of_points/gct_area_km2
gct_crash_density
```

```
## [1] 1116.947
```

```r
#central park
min_latitude <- 40.761716
max_latitude <- 40.802422
min_longitude <- -73.985928
max_longitude <- -73.945336

points_in_region <- cleaned_data %>%
  filter(LATITUDE >= min_latitude & LATITUDE <= max_latitude) %>%
  filter(LONGITUDE >= min_longitude & LONGITUDE <= max_longitude)

cp_number_of_points <- nrow(points_in_region)
cp_number_of_points
```

```
## [1] 7216
```

```r
#central park area
min_latitude <- 40.761716
max_latitude <- 40.802422
min_longitude <- -73.985928
max_longitude <- -73.945336

latitude_difference <- max_latitude - min_latitude
longitude_difference <- max_longitude - min_longitude

latitude_length_km <- 111
longitude_length_km <- 111 * cos((min_latitude + max_latitude) / 2 * pi / 180)

height_km <- latitude_difference * latitude_length_km
width_km <- longitude_difference * longitude_length_km

cp_area_km2 <- height_km * width_km
cp_area_km2
```

```
## [1] 15.41541
```

```r
cp_crash_density <- cp_number_of_points/cp_area_km2
cp_crash_density
```

```
## [1] 468.1029
```

```r
#nyc: manhattan, brooklyn, queens
min_latitude <- 40.543215
max_latitude <- 40.917797
min_longitude <- -74.042335
max_longitude <- -73.701056

points_in_region <- cleaned_data %>%
  filter(LATITUDE >= min_latitude & LATITUDE <= max_latitude) %>%
  filter(LONGITUDE >= min_longitude & LONGITUDE <= max_longitude)

nyc_number_of_points <- nrow(points_in_region)
nyc_number_of_points
```

```
## [1] 265747
```

```r
#nyc: manhattan, brooklyn, queens area
min_latitude <- 40.543215
max_latitude <- 40.917797
min_longitude <- -74.042335
max_longitude <- -73.701056

latitude_difference <- max_latitude - min_latitude
longitude_difference <- max_longitude - min_longitude

latitude_length_km <- 111
longitude_length_km <- 111 * cos((min_latitude + max_latitude) / 2 * pi / 180)

height_km <- latitude_difference * latitude_length_km
width_km <- longitude_difference * longitude_length_km

nyc_area_km2 <- height_km * width_km
head(nyc_area_km2,5)
```

```
## [1] 1193.575
```

```r
nyc_crash_density <- nyc_number_of_points/nyc_area_km2
nyc_crash_density
```

```
## [1] 222.648
```

```r
#nyc: staten island
min_latitude <- 40.496278
max_latitude <- 40.648945
min_longitude <- -74.255920
max_longitude <- -74.053551

points_in_region <- cleaned_data %>%
  filter(LATITUDE >= min_latitude & LATITUDE <= max_latitude) %>%
  filter(LONGITUDE >= min_longitude & LONGITUDE <= max_longitude)

si_number_of_points <- nrow(points_in_region)
si_number_of_points
```

```
## [1] 22076
```

```r
#nyc: staten island area
min_latitude <- 40.496278
max_latitude <- 40.648945
min_longitude <- -74.255920
max_longitude <- -74.053551

latitude_difference <- max_latitude - min_latitude
longitude_difference <- max_longitude - min_longitude

latitude_length_km <- 111
longitude_length_km <- 111 * cos((min_latitude + max_latitude) / 2 * pi / 180)

height_km <- latitude_difference * latitude_length_km
width_km <- longitude_difference * longitude_length_km

si_area_km2 <- height_km * width_km
si_area_km2
```

```
## [1] 289.1412
```

```r
si_crash_density <- si_number_of_points/si_area_km2
si_crash_density
```

```
## [1] 76.35024
```

```r
avg_crash_density_nyc <- nyc_crash_density + si_crash_density / 2
avg_crash_density_nyc
```

```
## [1] 260.8231
```

```
#nyc observations
90739 + 1774944
```

```
## [1] 1865683
```

```
1193.575 + 289.1412
```

```
## [1] 1482.716
```

```
###conclusion
###avg crash density for nyc = 1643.994 crashes/km^2
###times square crash density = 38,022.61 crashes/km^2
###central park crash density = 5194.931 crashes/km^2
###grand central terminal crash density = 13,924.67 crashes/km^2
```

Trade-Offs:

When trying to plot coordinates on a map using an API key, which likely was the largest challenge, it returned no usable information because the amount of collisions were too numerous to show clear trends. Finding simple numbers such as number of collisions and areas within certain coordinates would have been sufficient, but the coordinate plane was used in conjunction to find these numbers because it had already been coded.

Results:

Conclusion:

- Average crash density for NYC = 1643.994 crashes/km^2
- Times Square crash density = 38,022.61 crashes/km^2
- Central Park crash density = 5194.931 crashes/km^2
- Grand Central Terminal crash density = 13,924.67 crashes/km^2

Listed above are the crash densities for each given location (all NYC, Times Square, Central Park, and Grand Central Terminal). We see that about 1644 crashes happen in all of NYC per km^2 (averaged from the years 2016-2022). Compared to Times Square, with 38,023 crashes/km^2, we could say that there is a creator chance of collisions within and around Times Square. Similar to Central Park with 5194 crashes/km^2 and Grand Central Terminal at 13,925 crashes/km^2, these numbers are relatively higher and worth noting. This is to show that busier places with more foot traffic and popular tourist destinations may have a **higher likelihood** of there being a motor vehicle collision.

Limitations:

It is important to keep in mind that all data used for plotting was already cleaned, so many points were excluded because no coordinates were provided. This reduces the number of total observations, making the numbers not completely accurate. Though lower numbers than actual, the proportions of crash ratios should be relatively similar because of dispersed omitted data over time. Another important factor to consider is that all 3 of these tourist locations are in Manhattan, the central neighborhood in New York with the highest foot traffic. These results could

also be influenced by the actual neighborhood in the city rather than the reason that they are a tourist location. Some instances were always considered, such as Central Park being a non-drivable tourist location, so the surrounding perimeter of about 3 blocks were considered for the area used to make the calculations.

Future Work:

With more time, the average crashes in Manhattan could be used in lieu of all of New York city, given that all 3 of the tourist locations are in Manhattan. This would give a better estimate and comparison if these areas have a higher crash density than a more appropriate baseline.

Another change could be plotting data points only using one year of data. Because several years of data were plotted, it over-crowded the graph and essentially made it unusable. By narrowing down the time factor, it could have given a clearer image.

Findings:

Overall, the findings show that these 3 locations, all considered as tourist locations, do have a higher crash rate than the rest of New York City. There are some limitations with the methodology such as using only cleaned data and comparing Manhattan with the rest of the city, but overall we see that the busiest areas with many tourist destinations are more likely for motor vehicle collisions to occur.

Are motorcycles or cars more likely to cause fatal injury?

New York State has the highest number of motorcycle vehicle registrations (“Number of Registered U.S. Motorcycles by State 2021 | Statista”) (https://www.statista.com/statistics/191002/number-of-registered-motorcycles-in-the-us-by-state/). I wanted to explore this question because motorcycles often weave through pedestrian and street traffic in NYC, and it would be useful to quantify how dangerous these vehicles are.

```
#Analyze motorcycles
deadly_motorcycle_crashes <- crashesPerson %>%
  filter(`NUMBER OF PERSONS KILLED` > 0) %>% #filter crashes that caused death
  filter(`VEHICLE TYPE CODE 1`== "Motorcycle" | `VEHICLE TYPE CODE 2`== "Motorcycle") %>%  #filt
er deadly crashes involving motorcycles
  nrow() %>%  # see how many deadly motorcycle crashes there are
  print()
```

```
## [1] 1445
```

```
#Analyze cars
deadly_car_crashes <- crashesPerson %>%
  filter(`NUMBER OF PERSONS KILLED` > 0) %>%  # Filter crashes that caused death
  filter(`VEHICLE TYPE CODE 1` != "Motorcycle" | `VEHICLE TYPE CODE 2` != "Motorcycle") %>%  # E
xclude crashes involving motorcycles
  nrow() %>%  # Get the number of rows
  print()
```

```
## [1] 11944
```

```
car_to_motorcycle_crashes_ratio <- deadly_car_crashes/deadly_motorcycle_crashes
print(car_to_motorcycle_crashes_ratio) #Ratio of fatal car crashes to fatal motorcycle crashes i
s 8.3
```

```
## [1] 8.265744
```

```
#Upon first glance, it seems that fatal crashes involve cars 8x as much as they involve motorcyc
les.
#But if we look at the vehicle type codes, we see there are motorized bikes under other names th
an Motorcycle.
crashesPerson %>%
  distinct(`VEHICLE TYPE CODE 1`) %>%
  print(n = 10)
```

```
## # A tibble: 1,688 × 1
##    `VEHICLE TYPE CODE 1`
##    <chr>
##  1 Sedan
##  2 <NA>
##  3 Dump
##  4 Station Wagon/Sport Utility Vehicle
##  5 Tanker
##  6 Bus
##  7 Taxi
##  8 Van
##  9 Motorscooter
## 10 Bike
## # ℹ 1,678 more rows
```

```
#there are over a thousand vehicle codes
#to find motorcycle equivalents, pull names that contain "bike" or "motor"
motorcycle_equivalents <- unique(crashesPerson %>%
                           filter(grepl(".*(bike|motor).*", `VEHICLE TYPE CODE 1`, ignore.case =
TRUE)) %>%
                           pull(`VEHICLE TYPE CODE 1`))


# Print the matching names
print(motorcycle_equivalents)
```

```
##  [1] "Motorscooter"   "Bike"           "E-Bike"         "Motorcycle"
##  [5] "Motorbike"      "e bike uni"     "E-bike"         "MOTOR SCOO"
##  [9] "Dirt Bike"      "Minibike"       "MOTOR HOME"     "Motorized Home"
## [13] "Motorized"      "NonMotordS"     "Motorscoot"     "MOTORHOME"
## [17] "DIRT BIKE"      "MOTORSCOOT"     "E Bike w p"     "ebike"
## [21] "MOTOR SKAT"     "E-BIKE"         "E bike"         "motor"
## [25] "Motor"          "MOTOR"          "Motor Home"     "EBIKE"
## [29] "Ebike"          "MOTORCYCLE"     "dirt bike"      "Motor Scoo"
## [33] "E BIKE"         "MOTORIZEDS"     "E MOTORCYC"     "ELE MOTORC"
## [37] "PEDAL BIKE"
```

```
#remove irrelavent categories
Motorcycle_Equivalents <- motorcycle_equivalents[!grepl(
  "Bike|Motorized|Motorized Home|E-Bike|NonMotordS|Motor Home|Ebike|e bike uni|E Bike w p|E BIKE
|E-bike|ebike|E-BIKE|PEDAL BIKE|E bike|MOTOR HOME|MOTORHOME|EBIKE",
  motorcycle_equivalents,
  ignore.case = FALSE
)]
print(Motorcycle_Equivalents)
```

```
##  [1] "Motorscooter" "Motorcycle"   "Motorbike"    "MOTOR SCOO"   "Minibike"
##  [6] "Motorscoot"   "DIRT BIKE"    "MOTORSCOOT"   "MOTOR SKAT"   "motor"
## [11] "Motor"        "MOTOR"        "MOTORCYCLE"   "dirt bike"    "Motor Scoo"
## [16] "MOTORIZEDS"   "E MOTORCYC"   "ELE MOTORC"
```

```
#re-analyze motorcycle equivalents
All_deadly_motorcycle_crashes <- crashesPerson %>%
  filter(`NUMBER OF PERSONS KILLED` > 0) %>%  # filter deadly crashes
  filter(`VEHICLE TYPE CODE 1` %in% Motorcycle_Equivalents |
           `VEHICLE TYPE CODE 2` %in% Motorcycle_Equivalents)  # filter deadly crashes involving
motorcycle equivalents
nrow(All_deadly_motorcycle_crashes) %>% print()
```

```
## [1] 1799
```

```
1779-1425
```

```
## [1] 354
```

```
#there are 354 more observations included in this calculation now that motorcycle equivalents ha
ve been accounted for


#re-analyze cars minus motorcycle equivalents
All_deadly_car_crashes <- crashesPerson %>%
  filter(`NUMBER OF PERSONS KILLED` > 0) %>%  # Filter deadly crashes
  filter(!(`VEHICLE TYPE CODE 1` %in% Motorcycle_Equivalents) &
           !(`VEHICLE TYPE CODE 2` %in% Motorcycle_Equivalents))  # Exclude crashes involving mo
torcycle equivalents


nrow(All_deadly_car_crashes) %>% print()
```

```
## [1] 10490
```

```
cars_only_deadly_crashes <- 11867 - 10433
print(cars_only_deadly_crashes) #filtered out 1,434 observations
```

```
## [1] 1434
```

```
#COMPARE RATIO OF FATAL CAR CRASHES TO MOTORCYCLE CRASHES
print(10433/1779)
```

```
## [1] 5.864531
```

```
#According to this ratio, fatality is 5.7x more common in car-only crashes than in crashes invol
ving motorcycles in NYC
#This insight was interesting to me because I always assumed motorcycles were far more dangerous
than cars
#but not according to this dataset.


#to explore more, I looked at motorcycle vs car crashes and the percentage containing death coun
ts
#motorcycles:
All_motorcycle_crashes <- crashesPerson %>%
  filter(`VEHICLE TYPE CODE 1` %in% Motorcycle_Equivalents |
           `VEHICLE TYPE CODE 2` %in% Motorcycle_Equivalents) %>%  # filter crashes involving mo
torcycle equivalents
  nrow() %>%  print()
```

```
## [1] 75207
```

```
All_motorcycle_crashes_deadly <- crashesPerson %>%
  filter(`VEHICLE TYPE CODE 1` %in% Motorcycle_Equivalents |
           `VEHICLE TYPE CODE 2` %in% Motorcycle_Equivalents) %>%  # filter crashes involving mo
torcycle equivalents
  filter(`NUMBER OF PERSONS KILLED`>0) %>%
  nrow() %>%  print()
```

```
## [1] 1799
```

```
percentage_of_motor_crashes_deadly <- 1779 / 74723 * 100  # Calculate the percentage of motorcyc
le crashes that were deadly
percentage_of_motor_crashes_deadly <- round(percentage_of_motor_crashes_deadly, digits = 1)  # R
ound to 1 decimal place
print(percentage_of_motor_crashes_deadly)
```

```
## [1] 2.4
```

```
#cars
All_car_crashes <- crashesPerson %>%
  filter(!(`VEHICLE TYPE CODE 1` %in% Motorcycle_Equivalents) &
           !(`VEHICLE TYPE CODE 2` %in% Motorcycle_Equivalents)) %>%  # Exclude crashes involvin
g motorcycle equivalents
  nrow() %>% print()
```

```
## [1] 5992941
```

```
All_deadly_car_crashes <- crashesPerson %>%
  filter(`NUMBER OF PERSONS KILLED` > 0) %>%  # Filter deadly crashes
  filter(!(`VEHICLE TYPE CODE 1` %in% Motorcycle_Equivalents) &
           !(`VEHICLE TYPE CODE 2` %in% Motorcycle_Equivalents)) %>%  # Exclude crashes involvin
g motorcycle equivalents
  nrow() %>% print()
```

```
## [1] 10490
```

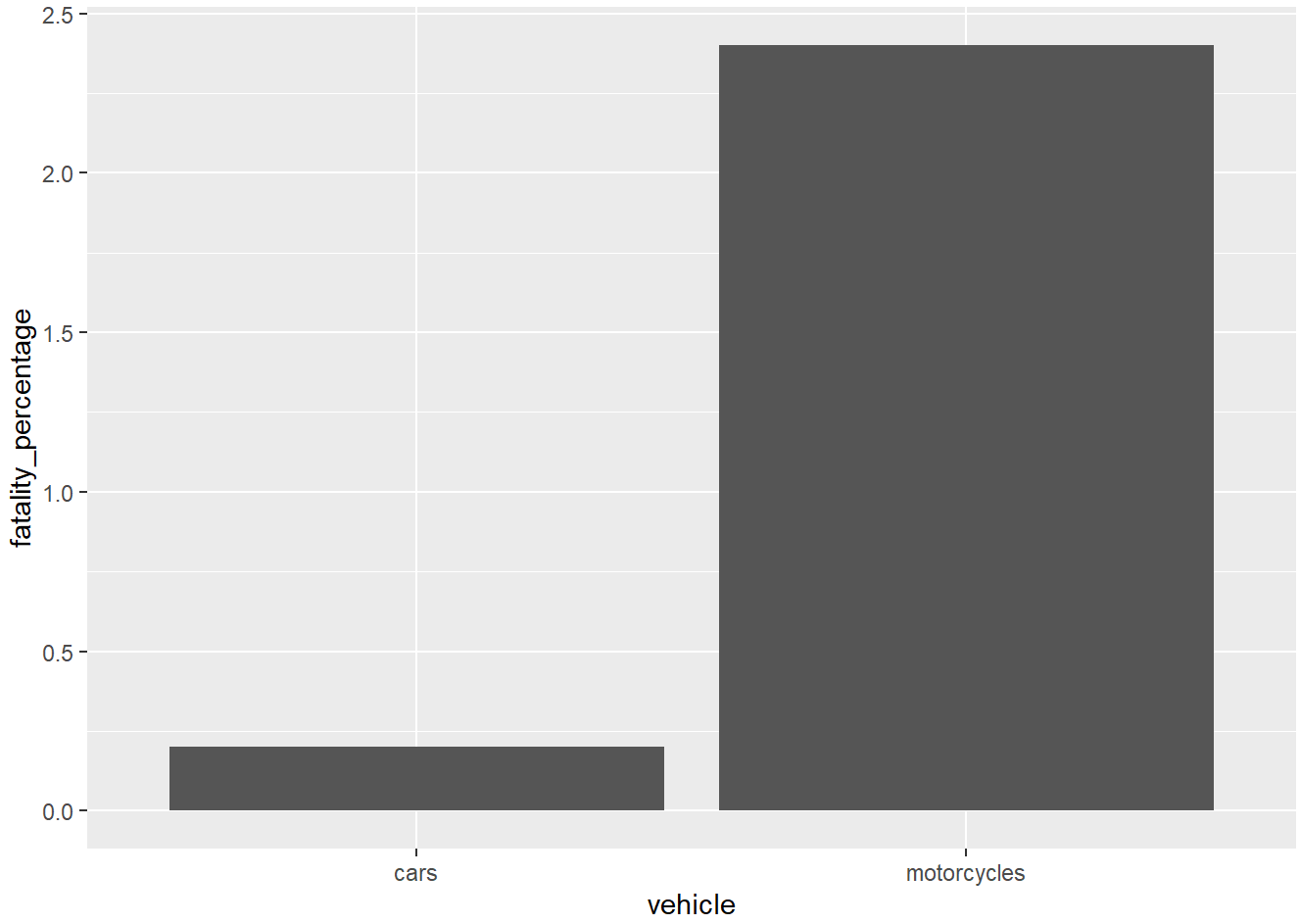
```
percentage_of_car_crashes_deadly <- 10433/5993413 *100 # Calculate the percentage of car crashes
that were deadly
percentage_of_car_crashes_deadly <- round(percentage_of_car_crashes_deadly, digits = 1)
print(percentage_of_car_crashes_deadly)
```

```
## [1] 0.2
```

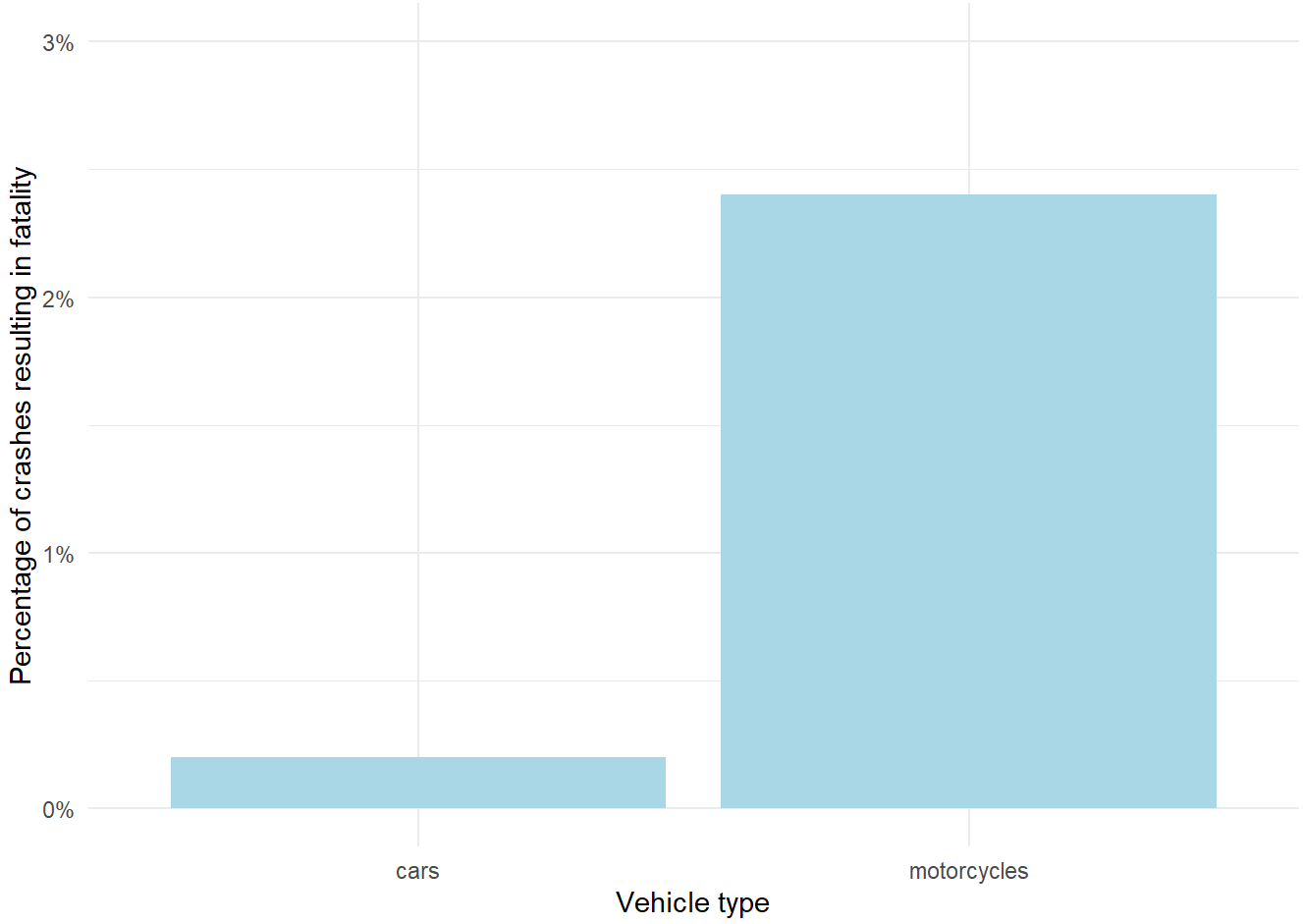
```
#Create a bar plot showing ratio of deadly to non-deadly crashes for motorcycles and cars respec
tively
#step 1: create a new df with crash information columns: vehicle type, number of crashes reporte
d, number of deadly crashes, ratio of deadly crashes to all crashes
vehicle <- c("cars", "motorcycles")
total_crashes <- c(5993413, 74723)
deadly_crashes <- c(10433, 1779)
fatality_percentage <- c(0.2, 2.4)


fatality_df <- data.frame(vehicle, total_crashes, deadly_crashes, fatality_percentage)
fatality_df
```

```
##       vehicle total_crashes deadly_crashes fatality_percentage
## 1        cars       5993413          10433                 0.2
## 2 motorcycles         74723           1779                 2.4
```

```
ggplot(fatality_df, aes(vehicle, fatality_percentage)) +
  geom_col()
```



```
ggplot(fatality_df, aes(vehicle, fatality_percentage)) +
  geom_col(fill = "lightblue") +  # Set bar color to blue
  scale_y_continuous(limits = c(0, 3), labels = scales::percent_format(scale = 1)) +  # Set y-ax
is limits and format as percentage
  labs(x = "Vehicle type", y = "Percentage of crashes resulting in fatality") +  # Set axis labe
ls
  theme_minimal()  # Optional: Use a minimal theme for better aesthetics
```



Method:

Three main variables are involved: `Vehicle Type Code 1` (VTC1), `Vehicle Type Code 2` (VTC2), and `Number of Persons Killed`. The type codes describe what kind of vehicles were involved in the crash. Starting with deadly motorcycle crashes, the conjoined dataset was filtered to omit all observations when zero deaths occured. In other words, looked only at deadly crashes. From there the dataset was filtered to keep only observations where “Motorcycle” appeared as either `VTC1` or `VTC2`. The same was applied for car crashes by instead omitting “Motorcycle” with the ! logical condition.

However, this was not a comprehensive look at motor and car crashes because there exists over 1,000 VTCs in the dataset - many of which were motorcycle equivalents for our purposes. For example, electric bikes and motorized scooters can be classified as Motorcycles for our purposes. To account for motorcycle equivalents, I pulled the column VTC1 and identified names containing the character “motor” and “bike.” Some irrelevant names showed up like “motor home” which is not a motorcycle equivalent. I stored these names in a new variable called Motrocycle_equivalents and removed the irrelevant names.

I re-analyzed my dataset and filtered out these motorcycle equivalents from car crashes and added them to motorcycle crashes. This decreased the ratio of deadly car to deadly motorcycle crashes from 8.3 to 5.7. I also found the ratio of deadly motorcycle crashes to all motorcycle crashes, and applied this analysis to car crashes too.

Results:

My results showed that a crash resulting in fatal injury was 5.7x more likely to be caused by a car crash than motorcycle crash. This reflects the vast amount of cars in NYC in comparison to motorcycles, but a more comprehensive look at the ratio of deadly crashes of all crashes better depicts the dangers of motorcycles. Whereas 0.02% of car crashes involved fatality, 2.3% of motorcycle crashes involved fatality. This means that **motorcycles are 10x more likely to cause death** than car crashes, even though **car crashes are more common**.

My results convey that motorcycles are **more dangerous than cars**. People should be more careful around motorcycles because they are **more likely to cause fatality**.

What is the most likely cause of collisions?

The variables that were most relevant for this question were the variable names:

- `CONTRIBUTING.FACTOR.VEHICLE.1`
- `CONTRIBUTING.FACTOR.VEHICLE.2`
- `CONTRIBUTING.FACTOR.VEHICLE.3`
- `CONTRIBUTING.FACTOR.VEHICLE.4`
- `CONTRIBUTING.FACTOR.VEHICLE.5`
- `CONTRIBUTING_FACTOR_1`
- `CONTRIBUTING_FACTOR_2`

Because there were 5 variables for this data, I needed to combine the data into one variable, to retrieve the frequency of each value under these columns. To do this, the data from these variables were converted into a long format structure from a wide format structure. I named the new column `Factor_Type` and the column name that has the corresponding values was called `Contributing_Factor_in_All_Vehicles` .

Only the values in this column were needed, as in this case, any column that did not pertain to the contributing factors in collisions was unnecessary. To filter out the rest of the data, I used the filter function to filter out columns that were not the contributing factor one, as well as values that were empty in that column. This entire process along with the conversion of the wide format structure to a long format structure was stored into a new variable called `crashesPersonLong`

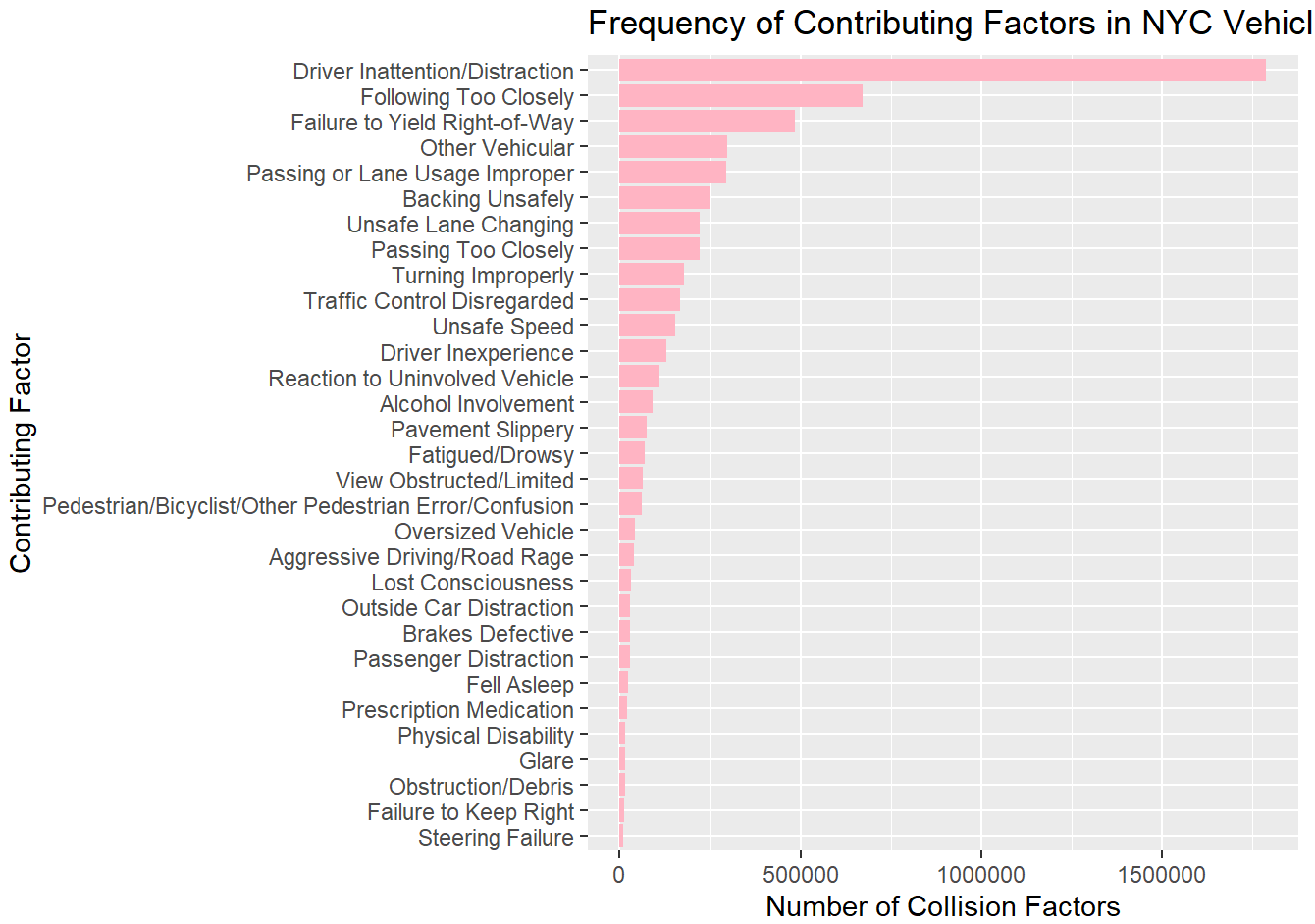
To find the frequency, I made a new variable called counts, that takes in `crashesPersonLong` and counts each occurrence of each value in that column, and those were titled `Frequency` . Then, I filtered `counts` to only display the values that had an occurrence of over 9000, as the values with a frequency less than this are more negligible when considering the scale of the data.

Then, the `counts` variable was updated to exclude the value that the data had as “unspecified’ as this value has no use for the analysis of the data.

To visualize the data and see the results, it was necessary to plot it. The `ggplot()` command was used to take in the `counts` data. The aesthetics that were applied consisted of x equalling the contributing factor, and the y being the frequency. They were ordered using `reorder()` to be displayed from highest to lowest frequency. Columns

were used to visualize the data and they were colored pink. I then flipped the data using `coord_flip()` to display the x values on a vertical axis and y on a horizontal axis for better readability, then titled the plot accordingly. Overall the plot showed the frequency of each contributing variable.

```
filteredData <- crashesPerson %>%
  filter(!is.na(`CONTRIBUTING FACTOR VEHICLE 1`) |
         !is.na(`CONTRIBUTING FACTOR VEHICLE 2`) |
         !is.na(`CONTRIBUTING FACTOR VEHICLE 3`) |
         !is.na(`CONTRIBUTING FACTOR VEHICLE 4`) |
         !is.na(`CONTRIBUTING FACTOR VEHICLE 5`) |
         !is.na(CONTRIBUTING_FACTOR_1) |
         !is.na(CONTRIBUTING_FACTOR_2))

filteredData <- filteredData %>%
  select(`CONTRIBUTING FACTOR VEHICLE 1`,
         `CONTRIBUTING FACTOR VEHICLE 2`,
         `CONTRIBUTING FACTOR VEHICLE 3`,
         `CONTRIBUTING FACTOR VEHICLE 4`,
         `CONTRIBUTING FACTOR VEHICLE 5`,
         CONTRIBUTING_FACTOR_1,
         CONTRIBUTING_FACTOR_2)

crashesPersonLong <- filteredData %>%
  pivot_longer(cols = everything(),
               names_to = "Factor_Type",
               values_to = "Contributing_Factor_in_All_Vehicles")

crashesPersonLong <- crashesPersonLong %>%
  filter(!is.na(Contributing_Factor_in_All_Vehicles))

counts <- crashesPersonLong %>%
  count(Contributing_Factor_in_All_Vehicles, name = "Frequency") %>%
  filter(Frequency > 9000)

counts <- counts[counts$Contributing_Factor_in_All_Vehicles != "Unspecified",]

ggplot(counts, aes(x = reorder(Contributing_Factor_in_All_Vehicles, Frequency),
                   y = Frequency)) +
  geom_col(fill = "pink1")+
  coord_flip()+
  labs(title = "Frequency of Contributing Factors in NYC Vehicle Collisions",
       x = "Contributing Factor",
       y = "Number of Collision Factors")
```



Trade-Offs:

One trade-off that accompanied my decisions was the choice to exclude the values that had less than 9000 occurrences. This was because if they were included in the plot, their frequency was so low, it appeared as if they were 0. This showed me that they were negligible and were not a big factor in collisions as they were very specific and not something that is a common cause of a collision that everyday people would have to take into account. Another trade off was focusing solely on the direct contributing factors rather than other potential contextual features of the dataset.

Results:

With the plot visualization, the **most likely cause of collisions is driver inattention/distraction**. Followed by this is “Following Too Closely”, “Failure to Yield Right of Way” and “Other Vehicular” tied with “Improper Passing or Lane Usage”.

Findings:

My results have shown that distracted driving is the most likely cause of collisions. This shows that one should limit all distractions when behind the wheel, be sure to focus on the road ahead, and be aware of what one is doing in regards to other cars and pedestrians. Limitations may include a lack of impact to pedestrians and what they can do to stay safe, as distracted driving falls solely on the driver. Another limitation can include the fact that there were numerous cases with an unspecified cause of collision. This could lead to a potentially incomplete understanding of collision causes. Additionally, the choice to focus on the most common contributing factors may not capture and address nuanced relationships between them. Further steps that can be taken to extend the story can include

understanding different interactions between the contributing factors and other variables. This would potentially need other datasets that are detailed. One can also compare the cause of collisions in New York City to other regions and analyze whether certain contributing factors are more prevalent in specific areas.

Distracted driving is the leading cause of vehicle collisions, highlighting the critical **need for drivers to minimize distractions** and **stay focused** on the road. However, the analysis may overlook the significance of less common factors and lacks consideration of broader contextual elements. Future work should explore interactions between various contributing factors, incorporate additional contextual data, and possibly compare findings across different regions to gain a more comprehensive understanding of collision causes.

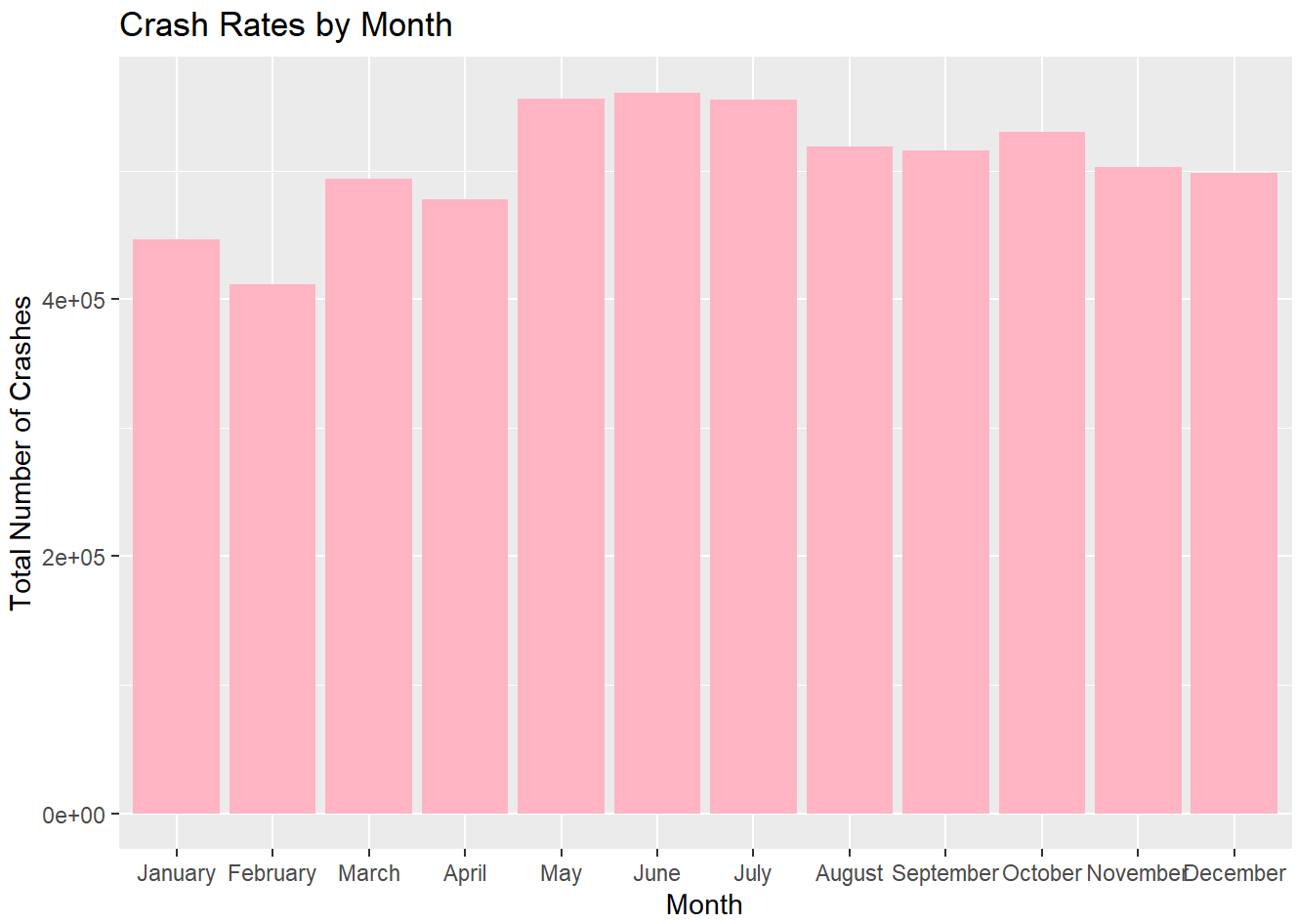
Do crash rates vary by month?

To first find how crash rates vary by month, the month chould be extracted from the `CRASH_DATE` variable. This was done by selecting the numbers in the first and second index of the variable, which is the month in mm/dd/yyyy format. Only the months were needed to answer this question. Labels were then created that assigned each month in numerical form to the corresponding month's full categorical classification. Then, a new column called `month` was created that replaced the numerical form with the categorical form.

After this, a monthly summary data frame was created where a variable titled `totalCrashes` counts the number of crashes that occurs in each month. The months were then put into chronological order by using the `factor()` function and `month.name` function.

To visualize the data, a bar chart was created where the x-axis represented the different months and the y-axis represented the number of crashes.

```
crashesPerson$month <- substr(crashesPerson$CRASH_DATE, 1, 2)

monthLabel<- c("01" = "January", "02" = "February", "03" = "March", "04" = "April",
                  "05" = "May", "06" = "June", "07" = "July", "08" = "August",
                  "09" = "September", "10" = "October", "11" = "November", "12" = "December")

crashesPerson$month <- monthLabel[crashesPerson$month]

monthlySummary <- crashesPerson %>%
  group_by(month) %>%
  summarize(
    totalCrashes = n())

monthlySummary$month <- factor(monthlySummary$month, levels = month.name)

ggplot(monthlySummary, aes(x = month, y = totalCrashes)) +
  geom_col(fill = "pink1") +
  labs(title = "Crash Rates by Month",
       x = "Month",
       y = "Total Number of Crashes")
```



Trade-Offs:

A trade-off with this research question is that I grouped the data by month and created discrete, monthly groups that may not completely capture other factors such as trends that a more advanced time series could potentially show. The var chart also does not show variation within a month, so a more nuanced view that shows changes over time or peak crashes within a month isn't possible.

Results:

The result of the data is that **June is the month with the highest collisions**. It is then followed by May, then July. These are summer/late spring months and the increase in crash counts could be due to the increased foot and vehicle traffic in warmer weather. Outdoor activities could also lead to roads being closed off, which can result in a more complicated driving experience. There could also be riskier driving behaviors due to a lack of caution about the weather when it is sunny. Additionally, higher temperatures can impact vehicle equipment, such as tire pressure, and cause blowouts. (Buckingham and Vega Law Firm) (https://www.medmal-law.com/blog/2022/june/how-do-high-temperatures-affect-tires-/#:~:text=For%20every%2010%C2%B0F,essential%20component%20of%20driving%20safely.)

Findings:

Knowing that June, May, and July are the months with the highest collision rates, one can infer that the increase in travel and outdoor activities during warmer months could contribute to this heightened rate. Increasing traffic control presence during summer months could help manage the surge in traffic and reduce accidents. Limitations to consider could be that the data does not account for unreported collisions or almost collisions, which could

provide additional context. To further answer this question, one could assess and differentiate between different types of collisions, such as those involving only vehicles versus those involving pedestrians or cyclists, which can provide more insight.

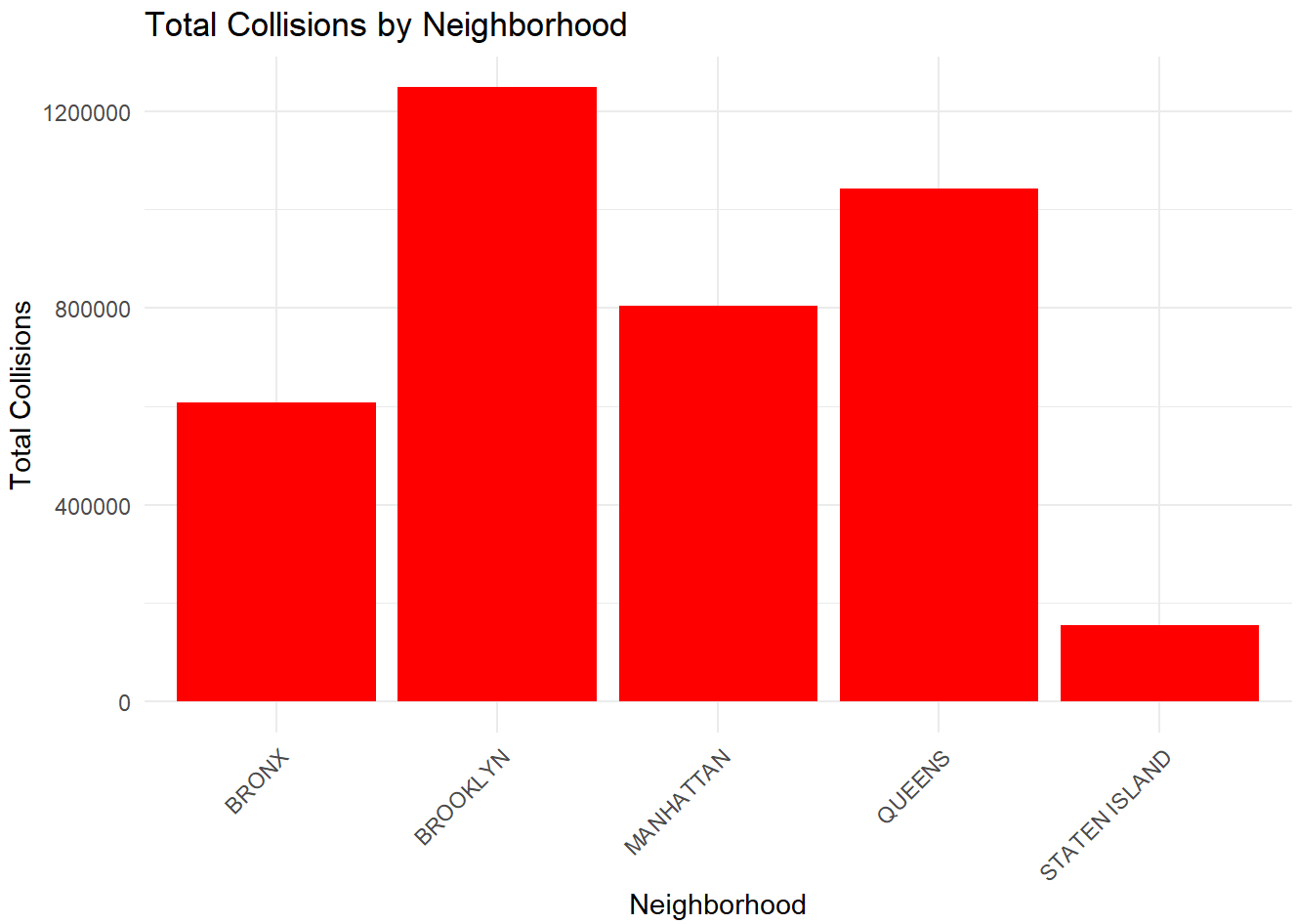
Knowing *when* motor vehicle collisions are most likely to happen is the crucial first step in preventing them. By using this knowledge, pedestrians, cyclists, and drivers in New York City can take proactive measures to protect themselves.

What Neighborhood are these Collisions Most Likely to Occur?

Method:

The variables that were used to project this data were total collisions and the names of the five boroughs in New York City. Before graphing the data, one problem that needed to be fixed was processing the NA values. To fix this, the NA and blank values had to be filtered out. The sample that was used for this research question was all the observations that could be processed by the computer and R. The samples were taken from the `CRASH` and `PERSON` datasets that were later merged.

```
BOROUGH_collisions <- crashesPerson %>%
  filter(!is.na(BOROUGH) & BOROUGH != "") %>%
  group_by(BOROUGH) %>%
  summarise(total_collisions = n())

ggplot(BOROUGH_collisions, aes(x = BOROUGH, y = total_collisions)) +
  geom_bar(stat = "identity", fill = "red") +
  theme_minimal() +
  labs(title = "Total Collisions by Neighborhood",
       x = "Neighborhood",
       y = "Total Collisions") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Trade-Offs:

One of the biggest trade-offs with filtering out the NA values is that some observations in the dataset were unrecorded. Before filtering out the NA values, I noticed that when they were graphed, there were a significant number of values that belonged to this category. Whether they are high enough in number to affect the data is doubtful, but these NA values prevent the dataset from being 100% accurate.

Results:

Based on the results of the bar graph, the **borough with the largest collisions is Brooklyn**. The borough with the least number of collisions is Staten Island, with significantly less compared to the other four boroughs. Another thing to note is that the volume of collisions in collisions is around the same amount of collisions of Manhattan and the Bronx combined.

Findings:

In order to figure out the reason why collisions occur the most frequently in Brooklyn and a low number in Staten Island, I decided to read into the population of each borough to see if this was correlated to the number of collisions. **Brooklyn is the borough with the highest population** and Staten Island is the borough with the least number of people, which could indicate that the number of people in a given area is related to the number of collisions. This could be measured in the future. One limitation is that because there were some NA values that had to be filtered out, the graph doesn't depict all the crash data. In the future, the coordinates of these NA values could be used to determine which borough these collisions occurred in.

All in all, Brooklyn is the borough with the highest number of collisions and the largest population. On the other hand, Staten Island has the least number of collisions and the smallest population.

Stakes: Why Does This Matter?

Our results matter because our topic involves public safety, policy-making, and urban planning. We now understand the fatal causes and injuries, neighborhood impact, and which locations are more prone for accidents (eg. tourist locations).

This also can tie into neighborhood disparities and if certain areas are more affected by collisions.

Knowing the time of day crashes are more likely to occur or the time of year can influence traffic regulation or have implemented safety measures.

This also give us an idea of the economic impact and how many resources go to damaged property and injuries.