HRM Traffic Collisions

Henry Truong

11/01/2022

1. Executive summary

This topic of this project is on traffic collisions in Halifax Regional Municipality (HRM) over the period of 2018-2021. The project will find out the tendency of traffic accidents in terms of time, space as well as characteristics related to collisions, roads and drivers. In addition, the project will perform hypothesis testing to make inference from the data of HRM. Lastly, a predictive model will be built to make a prediction on the number of daily incidents occurring in HRM

2. Introduction

2.1. Background

Point representation of traffic collisions which have occurred within the road right-of-way. Collisions are mapped according to the Motor Vehicle Collision Report. The dataset was created to support the Road Safety Plan initiative, details available on Halifax.ca. The dataset only contains HRP/RCMP closed ROW collision incidents that were completed electronically.

2.2. Data

The dataset can be accessed via this link

The dataset has following variables:

- `OBJECTID`: System generated ID
- `COLLISION_SK`: Unique identifier for the data set
- `CASE_FILE_NUMBER`: Collision case file number
- `ACCIDENT DATE`: Date of collision accident
- `WGS84_LAT_COORD` and `Y`: Latitude coordinate
- `WGS84_LON_COORD` and `X`: Longitude coordinate
- `ROAD_LOCATION_1`: Road location of collision
- `ROAD_LOCATION_2`: Intersecting road
- `ROAD_CONFIGURATION`: Road configuration
- `COLLISION_CONFIGURATION`: Collision configuration
- `NON_FATAL_INJURY`: If collision involved a non-fatal injury

- FATAL_INJURY: If collision involved a fatal injury
- YOUNG_DEMOGRAPHIC: If collision involved people under 25
- PEDESTRIAN_COLLISIONS: If collision involved any person who is not riding in or on a vehicle
- AGRESSIVE_DRIVING: If collision involved behaviors like following too close, speeding, disobeying traffic control, improper passing or more
- DISTRACTED_DRIVING: If collision involved inattention
- IMPAIRED_DRIVING: If collision involved a driver who is impaired or under the influence of drugs and alcohol
- BICYCLE_COLLISIONS: If collision involved someone on a bicycle
- INTERSCTION_RELATED: If collision occurring within an intersection

3. Procedure

3.1. Getting the data

The very first step is to load packages for the analysis

```
library(tidyverse)
library(modelr)
library(lubridate)
library(broom)
library(infer)
library(leaflet)
library(knitr)
library(kableExtra)
```

To make the analysis reproducible, a random seed needs to be set at the beginning of the analysis

```
set.seed(12345)
```

The data can be read in using the link provided in **Section 2.2** and save as `tc`

```
tc <- read_csv("./Traffic_Collisions.csv")
```

3.2. Understanding the data

The content of the data can be seen using `View(tc)`

3.2.1. Dimension First, the dimension of the data will be shown

```
dim(tc)
```

```
[1] 21127    21
```

There are 21127 rows and 21 columns

3.2.2. NAs Following that, we will look into NAs in each variable of the data. The table below will show variables with NAs along with its count

```
tc %>%
  map_dbl(~ sum(is.na(.))) %>%
  enframe(name = "Variable_name", value = "Count_of_NAs") %>%
  filter(Count_of_NAs > 0) %>%
  kable("html") %>%
  kable_styling(full_width = F)
```

Variable_name

Count_of_NAs

ROAD_LOCATION_1

4

ROAD_LOCATION_2

9068

ROAD_CONFIGURATION

626

COLLISION_CONFIGURATION

897

NON_FATAL_INJURY

18287

FATAL_INJURY

21081

3.2.3. Distinct values of unique identifiers It seems that there are 3 variables serving as the unique identifiers for observations in the data. We will check if their number of distinct values equals the row number of the data

```
tc %>%
  select(OBJECTID:CASE_FILE_NUMBER) %>%
  map_dbl(n_distinct) %>%
  enframe(name = "Variable_name",
          value = "Count_of_distinct_values") %>%
  kable("html") %>%
  kable_styling(full_width = F)
```

Variable_name

Count_of_distinct_values

OBJECTID

21127

COLLISION_SK

21127

CASE_FILE_NUMBER

20319

As can be seen, `OBJECTID` and `COLLISION_SK` are 2 unique IDs for the data because their count of distinct values equals to the data row number. There are some duplicated values in `CASE_FILE_NUMBER`, which may represent some same people involved in different traffic incidents

3.2.4. Missing dates in the timeframe In order to check if there is any missing date in the timeframe of the data, `ACCIDENT_DATE` needs to be modified. Therefore, this checking will be left for **Section 3.3**

There are 2 timestamps in `ACCIDENT_DATE` which are `'03:00:00'` and `'03:59:59'`. These correspond to the fact that daylight savings started and ended throughout the timeframe

3.2.5. Distinct values of character-columns We will count distinct values of each character-columns to see which character-columns can be considered as factor-columns

```r
tc %>%
  select(ROAD_LOCATION_1:INTERSECTION_RELATED) %>%
  map_dbl(n_distinct) %>%
  enframe(name = "Variable_name",
          value = "Count_of_distinct_values") %>%
  kable("html") %>%
  kable_styling(full_width = F)
```

Variable_name

Count_of_distinct_values

ROAD_LOCATION_1

7685

ROAD_LOCATION_2

2492

ROAD_CONFIGURATION

12

COLLISION_CONFIGURATION

17

NON_FATAL_INJURY

2

FATAL_INJURY

2

YOUNG_DEMOGRAPHIC

2

PEDESTRIAN_COLLISIONS

2

AGRESSIVE_DRIVING

2

DISTRACTED_DRIVING

2

IMPAIRED_DRIVING

2

BICYCLE_COLLISIONS

2

INTERSECTION_RELATED

2

Except ROAD_LOCATION_1 and ROAD_LOCATION_2, the other columns can be considered as factor-columns

All levels of these factor-columns will be listed

```
tc %>%
  select(ROAD_CONFIGURATION:INTERSECTION_RELATED) %>%
  map(unique)
```

```
$ROAD_CONFIGURATION
 [1] "Non-intersection"
 [2] "Intersection - two or more public roads"
 [3] "Intersection - private road or access"
 [4] "Ramp"
 [5] "Traffic circle or roundabout"
 [6] NA
 [7] "Express lane of a freeway"
 [8] "Bridge, overpass or viaduct"
 [9] "Tunnel or underpass"
[10] "Passing lane"
[11] "Rail level crossing"
[12] "Light rail transit crossing"

$COLLISION_CONFIGURATION
 [1] NA
 [2] "Multiple vehicle - left turn across opposing traffic"
 [3] "Single vehicle - hit a moving or stationary object on road surface"
 [4] "Multiple vehicle - rear end"
 [5] "Multiple vehicle - head on"
 [6] "Multiple vehicle - left turn into traffic"
 [7] "Multiple vehicle - approaching sideswipe"
 [8] "Multiple vehicle - right angle"
 [9] "Single vehicle - off road to the right"
[10] "Multiple vehicle - hit parked vehicle"
[11] "Multiple vehicle - same direction sideswipe"
[12] "Multiple vehicle - right turn, including turning conflicts"
[13] "Multiple vehicle - one crossing path of other to the left"
[14] "Single vehicle - off road to the left"
[15] "Multiple vehicle - one crossing path of other to the right"
[16] "Multiple vehicle - left turn against traffic"
[17] "Single vehicle - rollover"

$NON_FATAL_INJURY
[1] "Yes" NA

$FATAL_INJURY
[1] NA    "Yes"

$YOUNG_DEMOGRAPHIC
[1] "N" "Y"

$PEDESTRIAN_COLLISIONS
```

```
[1] "N" "Y"

$AGRESSIVE_DRIVING
[1] "N" "Y"

$DISTRACTED_DRIVING
[1] "N" "Y"

$IMPAIRED_DRIVING
[1] "N" "Y"

$BICYCLE_COLLISIONS
[1] "N" "Y"

$INTERSECTION_RELATED
[1] "N" "Y"
```

It should be noted that `ROAD_LOCATION_1` contains the road name which may or may not be preceded by the number (e.g. `'123 KING RD'` and `'BEDFORD HWY'`) where incidents occurred. In **Section 3.4** , the road name will be extracted from `ROAD_LOCATION_1` for the exploratory data analysis

3.2.6. Duplicated rows

We will exclude `OBJECTID` and `COLLISION_SK` to check if there are any duplicated observations

```r
tc %>%
  select(-c(OBJECTID, COLLISION_SK)) %>%
  duplicated() %>%
  sum()
```

```
[1] 0
```

As can be seen, there are none duplicated rows

3.2.7. Meaning of NAs

`NON_FATAL_INJURY` and `FATAL_INJURY`

`NA` for these two columns is equivalent to `'No'`, which means an incident had no fatal injuries or no non-fatal injuries. Therefore, observations with `NA` in both `NON_FATAL_INJURY` and `FATAL_INJURY` represent incidents without injuries

Because an incident cannot be both non-fatal and fatal, there should be no observation where `NON_FATAL_INJURY` and `FATAL_INJURY` are `'Yes'`

```r
tc %>%
  filter(FATAL_INJURY == 'Yes', NON_FATAL_INJURY == 'Yes') %>%
  nrow()
```

```
[1] 0
```

The result is an empty dataframe

`COLLISION_CONFIGURATION` and `ROAD_CONFIGURATION`

`NA` for these two columns is equivalent to `'Not specified'`

`ROAD_LOCATION_2`

This column means intersecting road with the road where an incident happened. `NA` for this column may mean an incident was not related to an intersection. We will check this

```
tc %>%
  filter(is.na(ROAD_LOCATION_2), INTERSECTION_RELATED == 'Y') %>%
  nrow()
```

```
[1] 2027
```

The result is a data frame with 2027 rows, which means there are intersection-related incidents with `NA` in `ROAD_LOCATION_2`. In this case, it is reasonable to assume `NA` is equivalent to `'Not specified'`

`ROAD_LOCATION_1` There are only 4 `NA`s in this column. We can change `NA` to a road name using the latitude and longitude of incidents. Otherwise, we can change `NA` to `'Not specified'` as in the case of `ROAD_LOCATION_2`

3.3. Cleaning the data

3.3.1. Column names All the column names are upper-case so they will be converted to the lower-case format. A spelling mistake is spot on `AGRESSIVE_DRIVING` and this will also be fixed

```
tc <- tc %>%
  rename_with(str_to_lower) %>%
  rename(aggressive_driving = agressive_driving)
```

3.3.2. Date-column `accident_date` is currently a character-column. The column has the date format of "yyyy/mm/dd HH:MM:SS+00". We will extract the date part out of `accident_date`. As explained in the previous section, the time part is not important

```
tc <- tc %>%
  rename_with(str_to_lower) %>%
  mutate(date = accident_date %>% str_extract('^.{10}') %>% ymd())
```

Now we will check if there is any missing date in the timeframe

```
(max(tc$date) - min(tc$date) + 1) - n_distinct(tc$date)
```

```
Time difference of 0 days
```

The difference is 0 so there is none missing date in the timeframe

3.3.3. NA replacement In the case of `non_fatal_injury` and `fatal_infury`, `NA` will be converted to `'No'`. In the case of `collision_configuration`, `road_configuration` and `road_location_2`, `NA` will be converted to `'Not specified'`

```
tc <- tc %>%
  mutate(non_fatal_injury = replace_na(
           non_fatal_injury,
           'No'),
         fatal_injury = replace_na(
           fatal_injury,
           'No'),
         collision_configuration = replace_na(
           collision_configuration,
           'Not specified'),
         road_configuration = replace_na(
           road_configuration,
           'Not specified'),
         road_location_2 = replace_na(
           road_location_2,
           'Not specified'))
```

In the case of `road_location_1`, we will use the latitude and longitude to determine the road name where incidents happened with the help from Google Map

```
tc %>%
  filter(road_location_1 %>% is.na()) %>%
  select(objectid, x, y)
```

```
# A tibble: 4 x 3
  objectid     x     y
     <dbl> <dbl> <dbl>
1     2026 -63.6  44.7
2     4170 -63.5  44.7
3     4478 -63.4  44.9
4    11536 -63.5  44.6
```

Using this approach, `NA`s can be replaced by `'Gottingen St'`, `'Circassion Dr'`, `'Old Guysborough Rd'`, `'Osborne Ave'` for `objectid` of `2026`, `4170`, `4478` and `11536`. It should be noted that the abbreviations like `'St'` and `'Dr'` are used to comply with those in the data

```
tc <- tc %>%
  mutate(road_location_1 = case_when(
    objectid == 2026 ~ 'Gottingen St',
    objectid == 4170 ~ 'Circassion Dr',
    objectid == 4478 ~ 'Old Guysborough Rd',
    objectid == 11536 ~ 'Osborne Ave',
    TRUE ~ road_location_1))
```

Finally, we will check if any `NA` exists in the data

```
tc %>% is.na() %>% sum()
```

```
[1] 0
```

The result is 0 so there is none NA in `tc`.

3.3.4. Format of values

To make uniform the format of values in the columns ranging from `road_location_1` to `collision_cofiguration`, we use `str_to_title` function

```
tc <- tc %>%
  mutate(across(road_location_1:collision_configuration,
                str_to_title))
```

The data now is pretty clean and it is ready to perform the exploratory data analysis

3.4. Performing the EDA

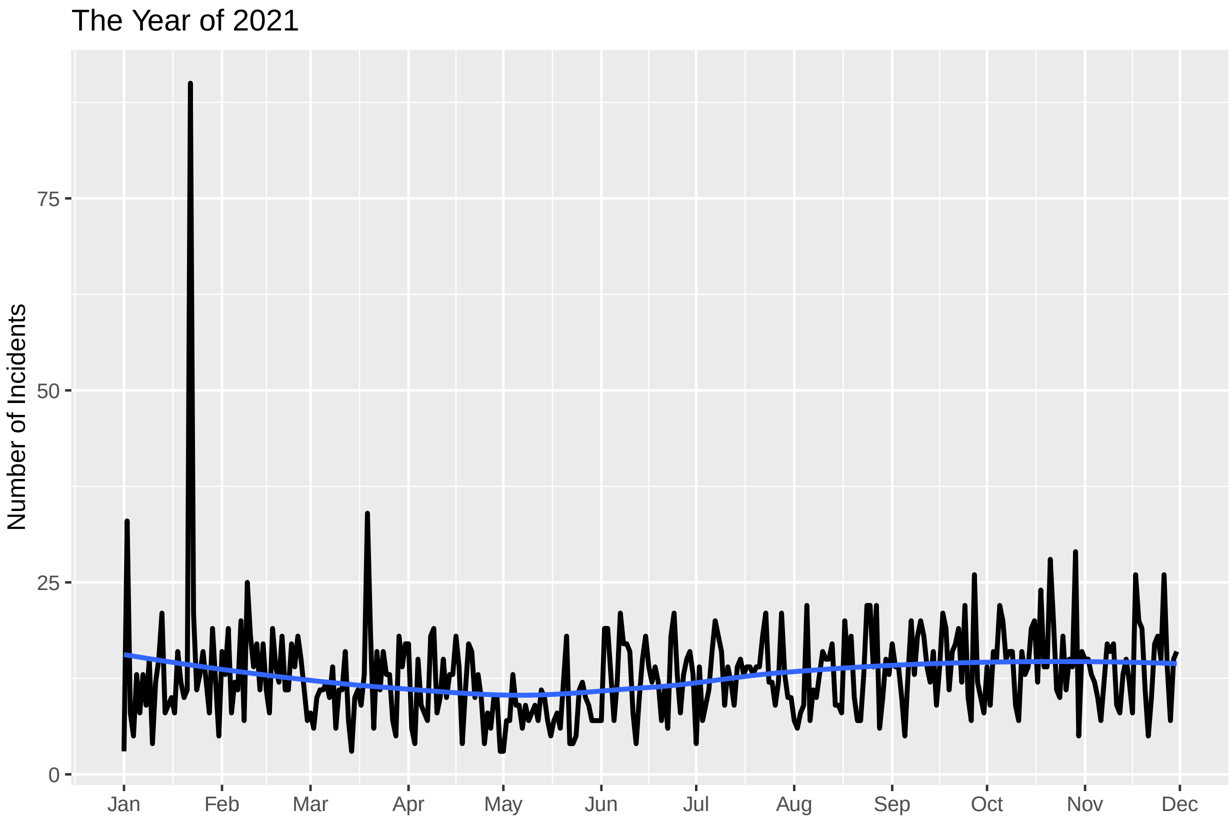
3.4.1. Daily incidents throughout the timeframe

A function named `daily_plot` will be created to plot the total number of incidents on a daily basis in a selected year

```
daily_plot = function(tc, year = 2021) {
  tc %>%
    filter(year(date) == year) %>%
    count(date) %>%
    ggplot(aes(x = date, y = n)) +
    geom_line(size = 1) +
    geom_smooth(se = F) +
    labs(x = NULL, y = 'Number of Incidents',
         title = str_c('The Year of ', year)) +
    scale_x_date(date_breaks = '1 month', date_labels = '%b')
}
```

The plot of daily incidents in 2021 is shown below

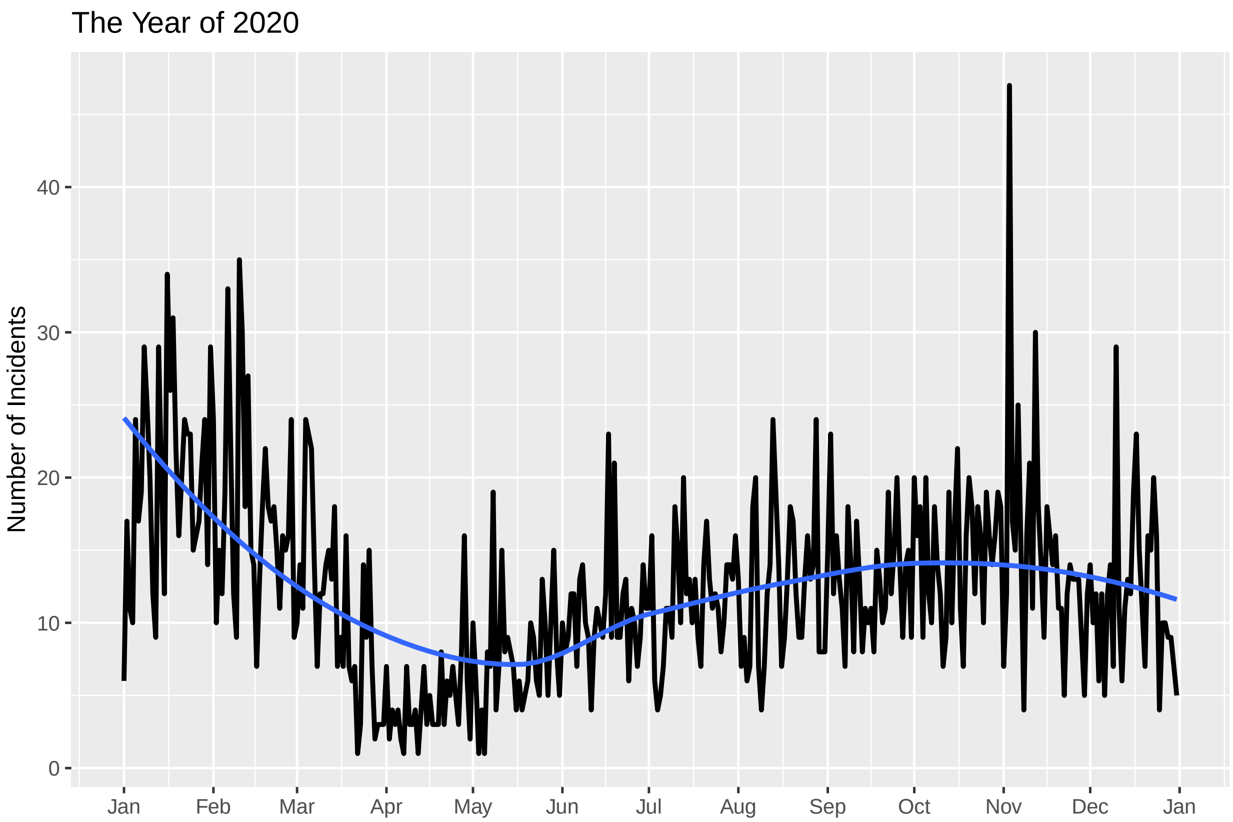
```
daily_plot(tc, 2021)
```



Comment:
- There were remarkable spikes in Jan and Mar
- The data ended on Nov 30, 2021

The plot of daily incidents in 2020 is shown below

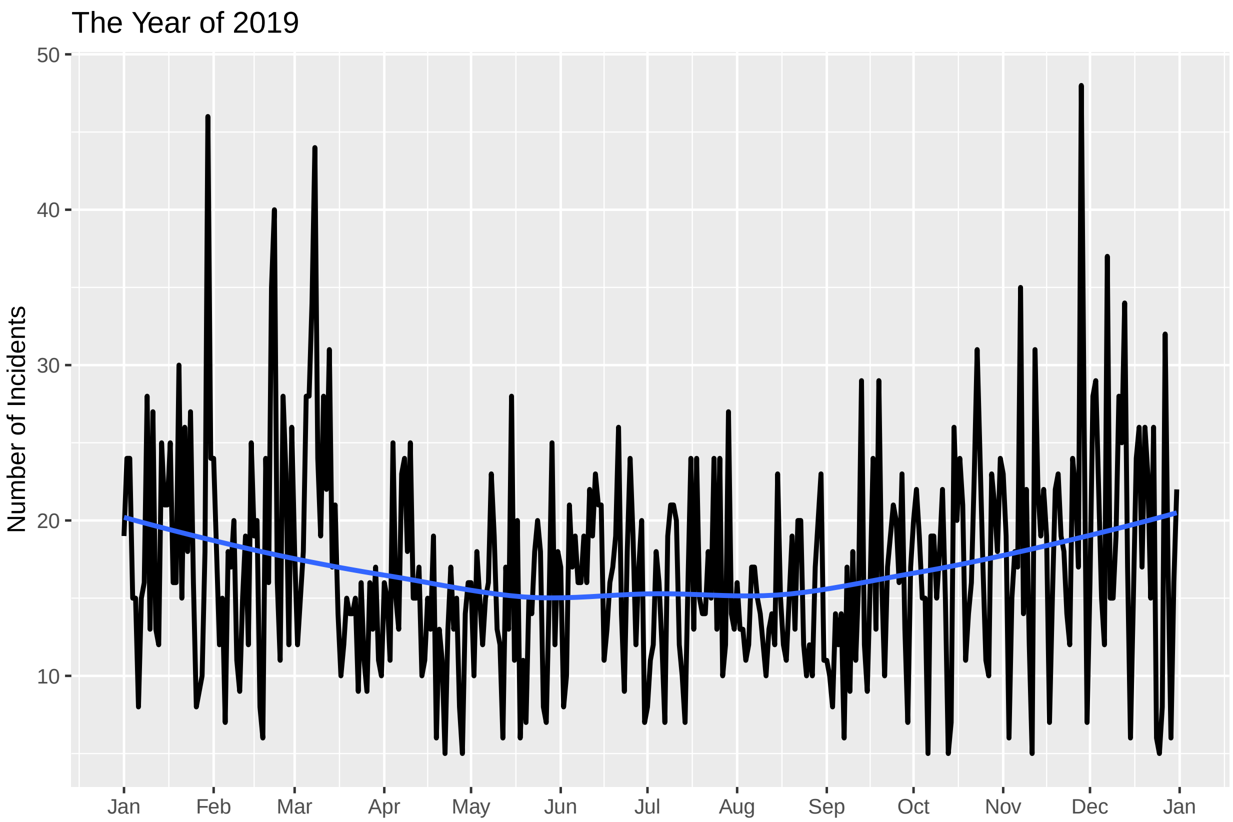
```
daily_plot(tc, 2020)
```



Comment:
- There were remarkable spikes in Jan, Feb and Nov
- There was a noticeable dip in Apr

The plot of daily incidents in 2019 is shown below

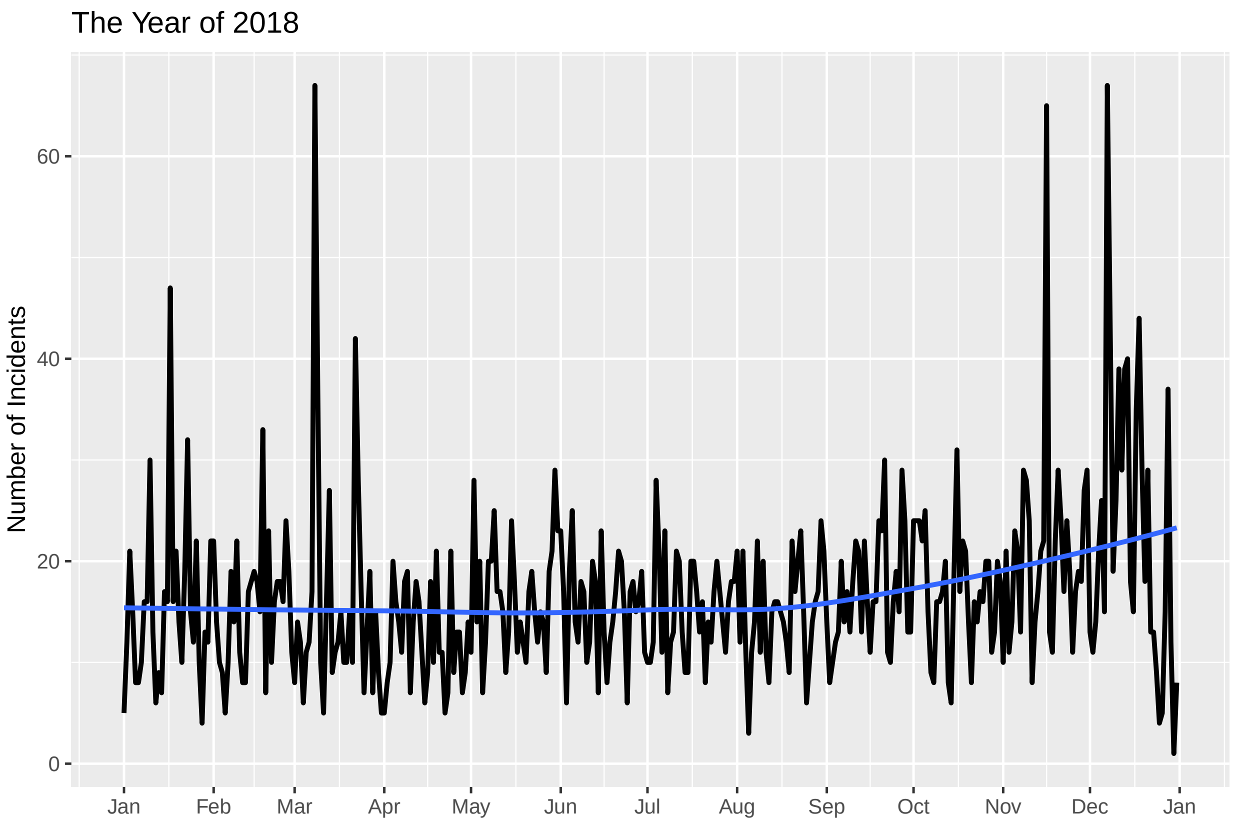
```
daily_plot(tc, 2019)
```



Comment: There were remarkable spikes in Jan-Mar and Nov

The plot of daily incidents in 2018 is shown below

```
daily_plot(tc, 2018)
```



Comment: There were remarkable spikes in Jan, Mar and Nov-Dec

Comment: From the set of plots for 4 years, it can be seen that:
- The spikes in the number of daily incidents generally occurred in the span of time ranging from Nov to Mar, which corresponds to winter seasons
- There was a sharp fall in the number of daily incidents in Apr 2020 which corresponds to the point of time when the policy of social isolation was implemented
- The data ended on Nov 30, 2021

3.4.2. Average daily incidents across selected timescales In this part, we examine the average number of daily incidents in different time scales (e.g. year, month). Note that we cannot use the total number as the metric for a certain time scale because the data ended in Nov 30, 2021 which would create a bias for the year of 2021 or the month of Dec

We will add separate columns representing the year, month and day of week to `tc`

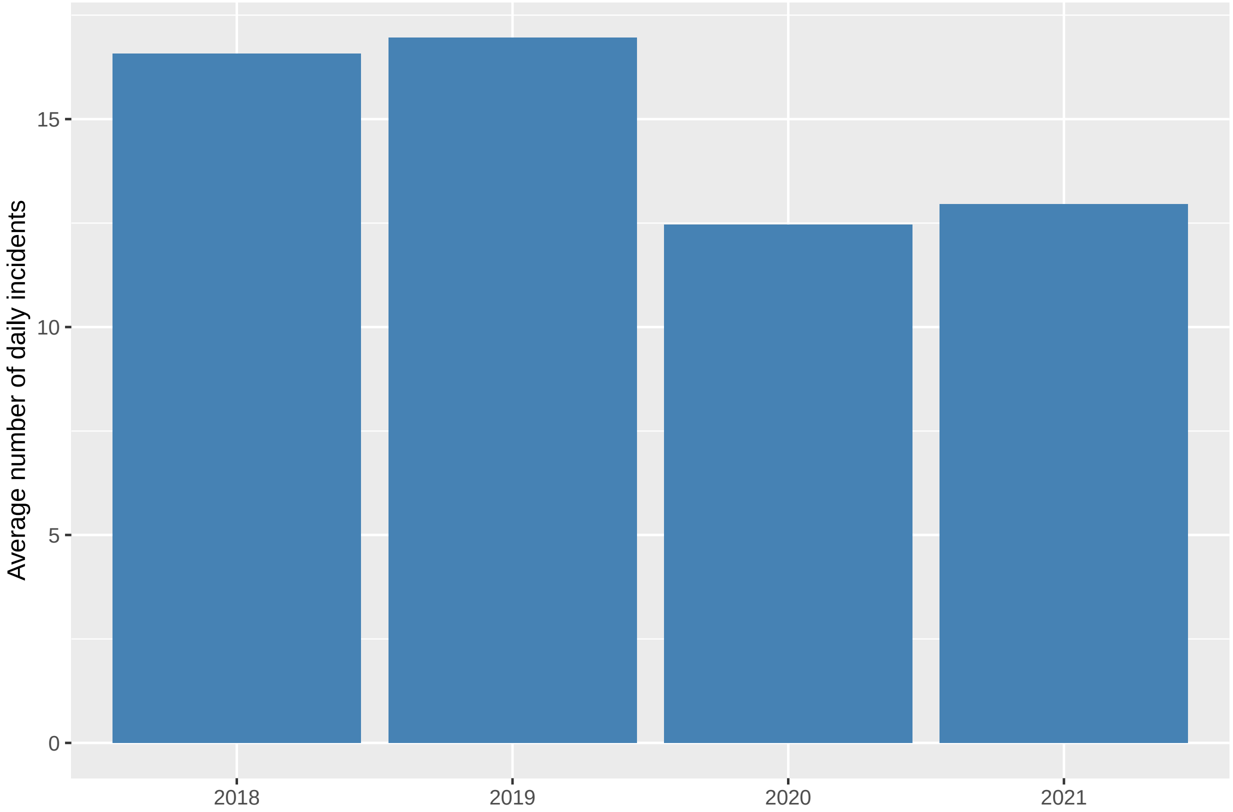
```
tc <- tc %>%
  mutate(year = year(date),
         month = month(date, label = TRUE),
         dow = wday(date, label = TRUE))
```

A couple of functions named `daily_timescale_barplot` and `daily_timescale_boxplot` will be created to create a couple of following graphs:
- A bar chart which compared average daily incidents across the selected timescale
- A box plot which compared the distributions of average daily incidents across the selected timescale

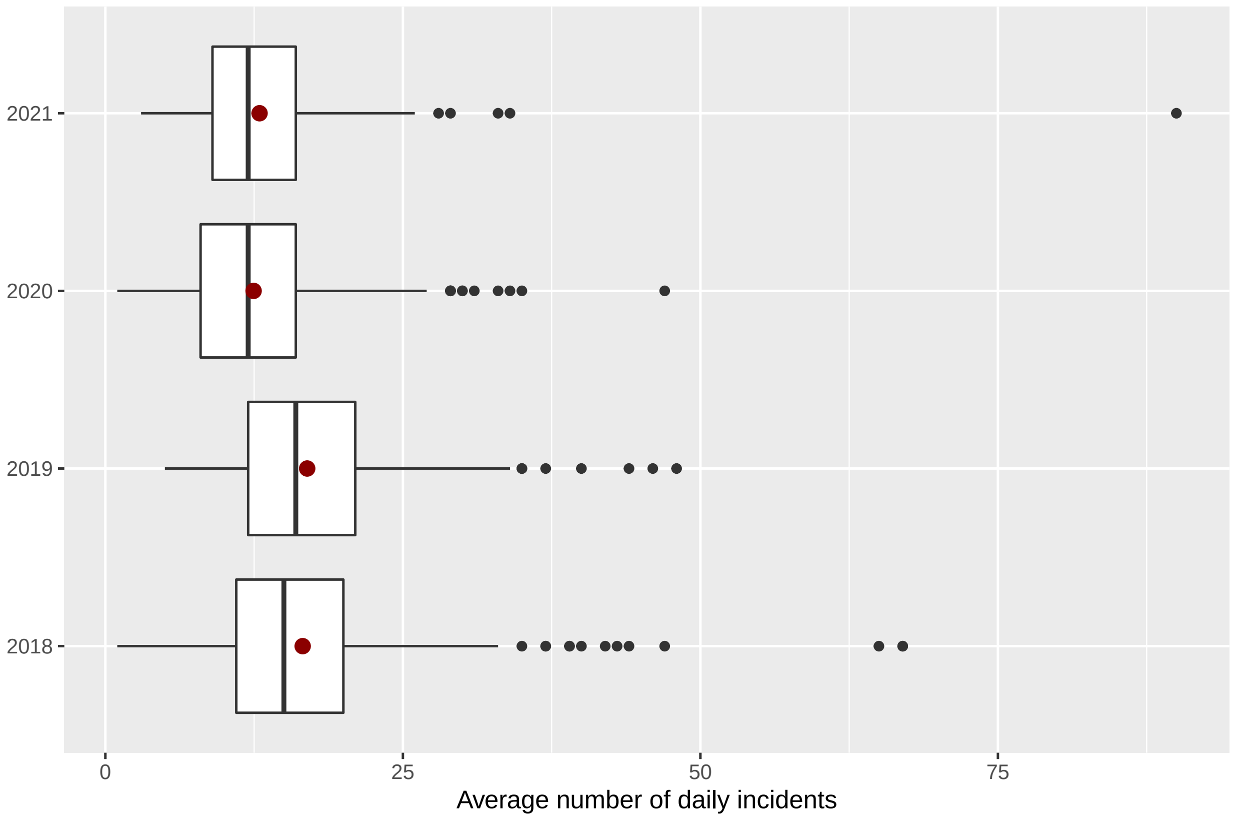
```
daily_timescale_barplot = function(tc, timescale = 'year') {
  tc %>%
    mutate(time_scale = tc[[timescale]]) %>%
    count(date, time_scale) %>%
    group_by(time_scale) %>%
    summarize(n_mean = mean(n)) %>%
    ggplot(aes(x = as_factor(time_scale), y = n_mean)) +
    labs(x = NULL, y = 'Average number of daily incidents') +
    geom_col(fill = 'steelblue')
}
daily_timescale_boxplot = function(tc, timescale = 'year') {
  tc %>%
    mutate(time_scale = tc[[timescale]]) %>%
    count(date, time_scale) %>%
    ggplot(aes(x = as_factor(time_scale),
               y = n)) +
    geom_boxplot() +
    stat_summary(fun = mean, color = 'darkred') +
    labs(x = NULL, y = 'Average number of daily incidents') +
    coord_flip()
}
```

When the selected timescale is `'year'`, we have the following plots

```
daily_timescale_barplot(tc, 'year')
```



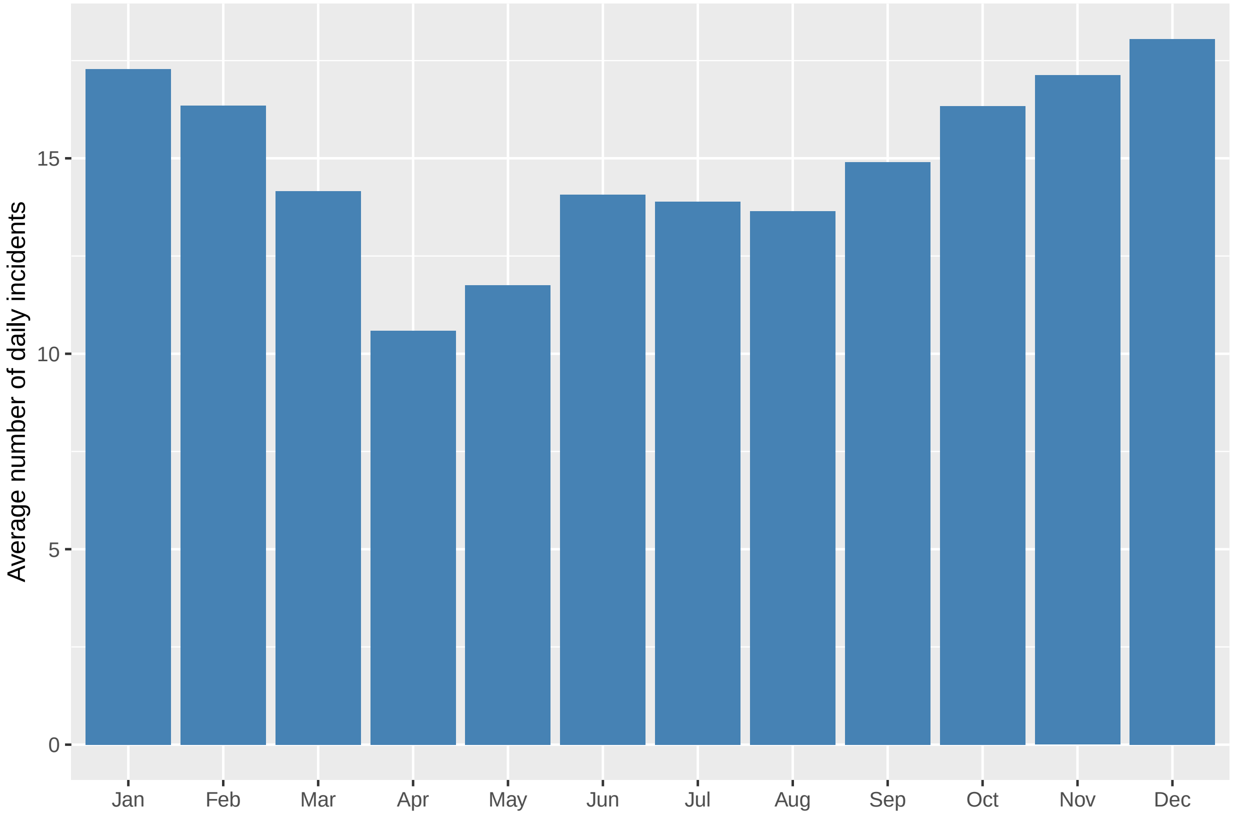
```
daily_timescale_boxplot(tc, 'year')
```



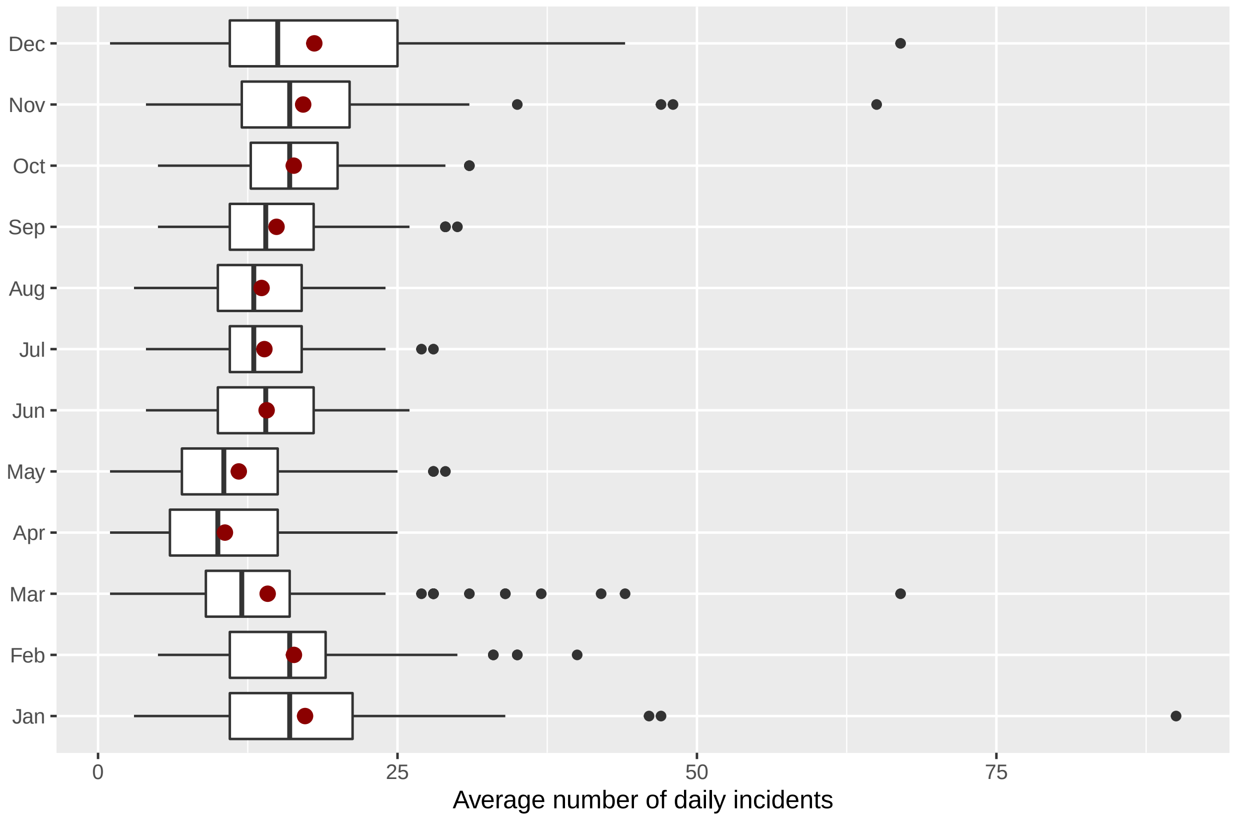
Comment:
- From 2018 to 2019, the average number of daily incidents slightly increased
- From 2019 to 2020, the average number of daily incidents sharply decreased
- From 2020 to 2021, the average number of daily incidents remained low despite an increase
- The sharp decrease in the average number of daily incidents must be attributed to the outbreak of COVID-19
- The slight increase in the average number of daily incidents may be attributed to the increase in the population of the region
- There was one remarkable outlier in 2021

When the selected timescale is `'month'`, we have the following plots

```
daily_timescale_barplot(tc, 'month')
```



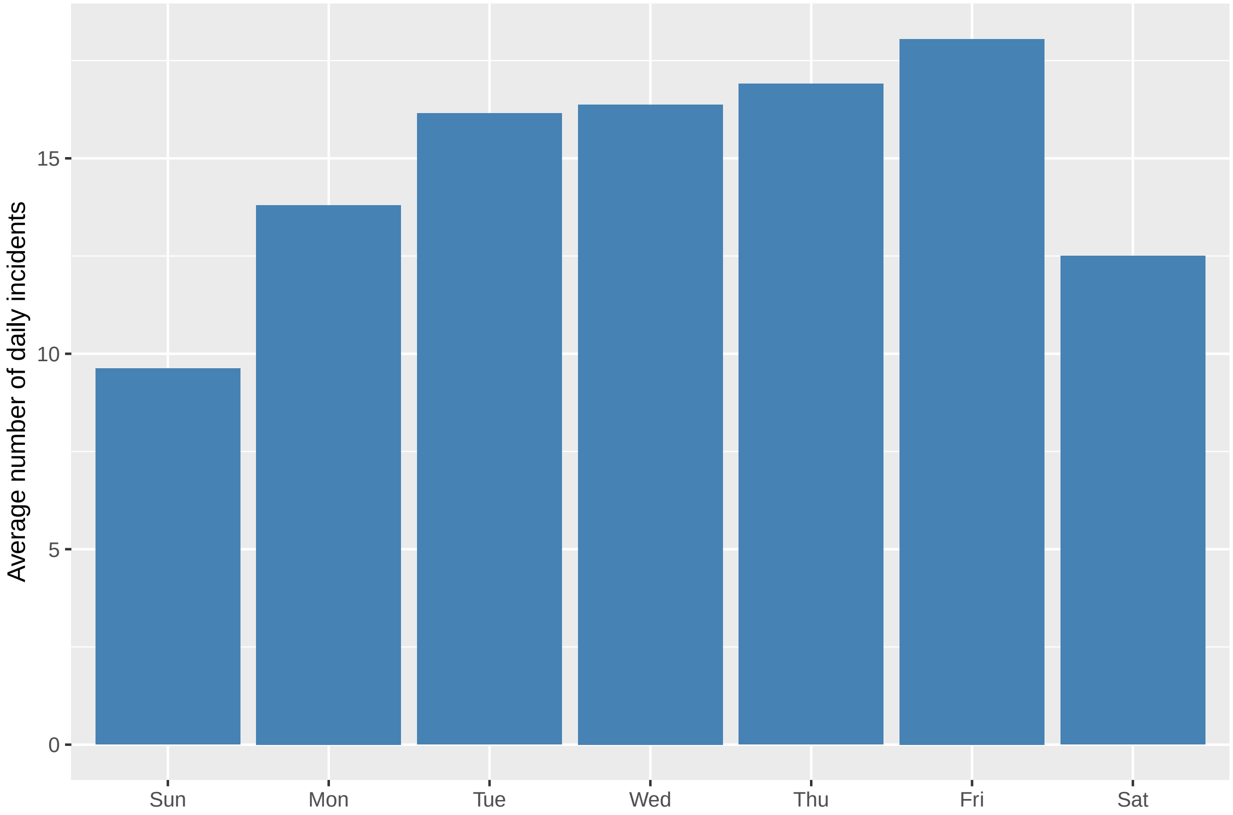
```
daily_timescale_boxplot(tc, 'month')
```



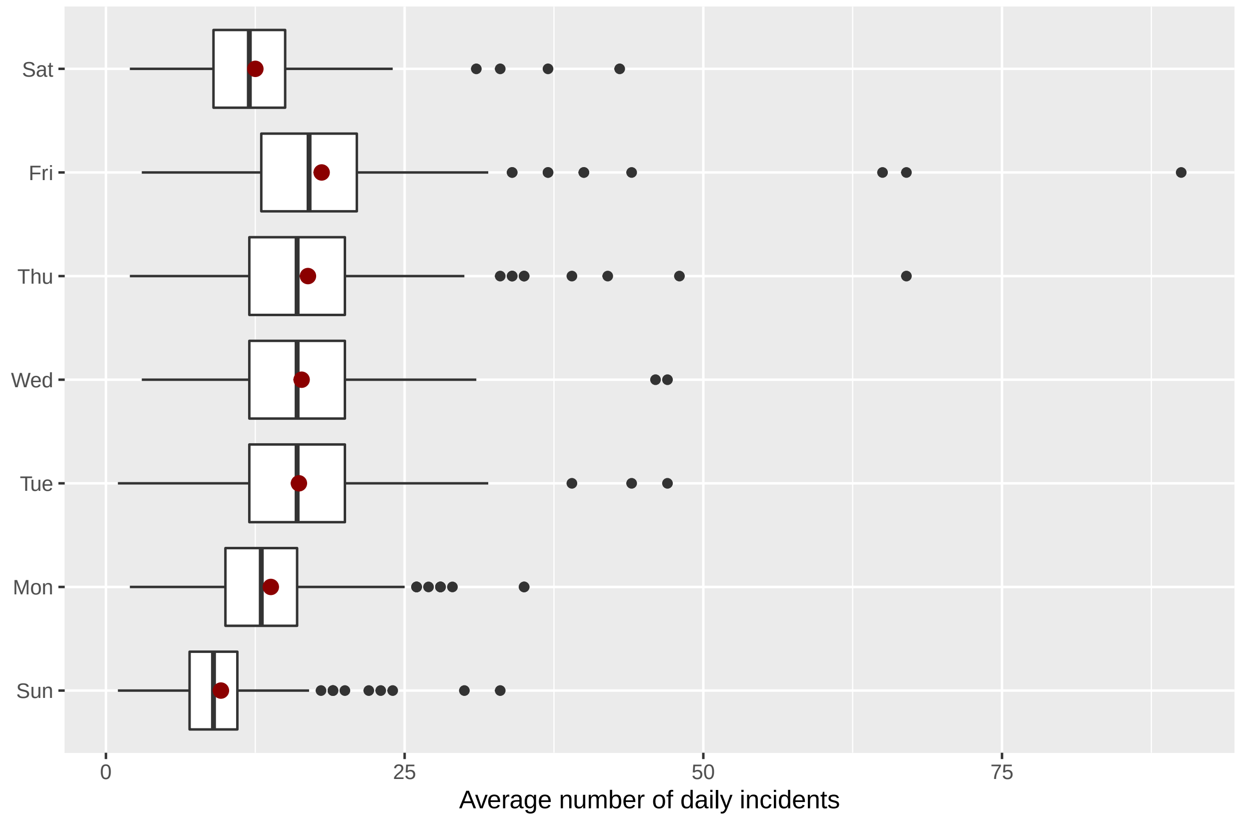
Comment:
- Apr had the lowest average number of daily incidents
- Dec had the highest average number of daily incidents
- Fall-winter term had a higher average number of daily incidents than spring-summer term did
- The number of outliers for Nov-March is higher than that for Apr-Oct
- There was one remarkable outlier in Jan

When the selected timescale is `'dow'` which is day of week, we have the following plots

```
daily_timescale_barplot(tc, 'dow')
```



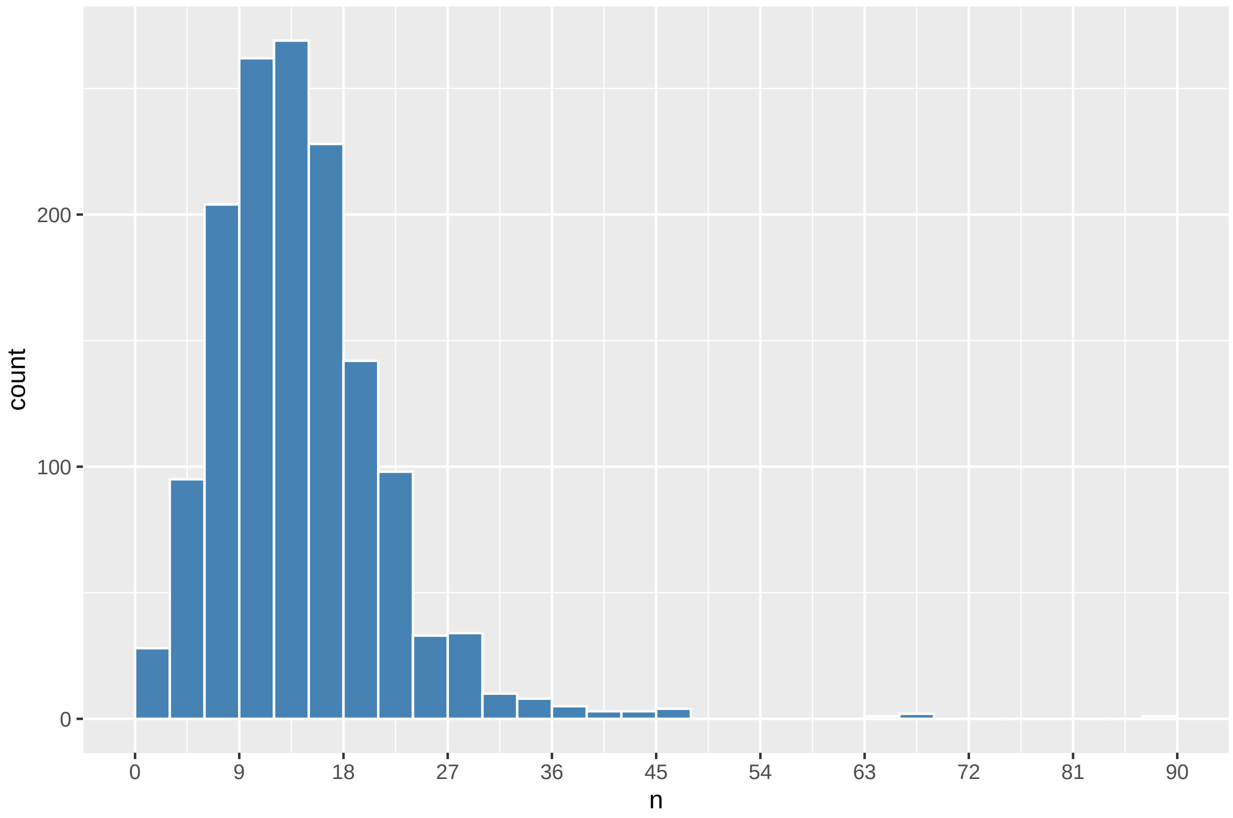
```
daily_timescale_boxplot(tc, 'dow')
```



Comment:
- Weekends had a lower number of daily incidents than weekdays did
- There was fewest incidents on Sunday
- There was most incidents on Friday
- There was one remarkable outlier on Friday

3.4.3. Distribution of daily incidents The distribution of daily incidents can be shown by the following histogram

```
tc %>%
  count(date) %>%
  ggplot(aes(x = n)) +
  geom_histogram(binwidth = 3, boundary = 0,
                 color = 'white', fill = 'steelblue') +
  scale_x_continuous(breaks = seq(0, 90, by = 9))
```



The 6-number summary of daily incidents can be shown by the following table

```
tc %>%
  count(date) %>%
  .$n %>%
  summary() %>%
  enframe() %>%
  kable("html") %>%
  kable_styling(full_width = F)
```

name

value

Min.

1.00

1st Qu.

10.00

Median

14.00

Mean

14.77

3rd Qu.

18.00

Max.

90.00

Comment:
- With the bin width of 3, the most frequent number of daily incidents was 13-15
- The minimum number of daily incidents was 1
- The maximum number of daily incidents was up to 90
- The distribution was positively skewed

Let's find out on what day 90 incidents happened

```
tc %>%
  count(date) %>%
  filter(n == 90) %>%
  mutate(dow = wday(date, label = TRUE)) %>%
  kable("html") %>%
  kable_styling(full_width = F)
```

date

n

dow

2021-01-22

90

Fri

Comment:
- The date when 90 incidents happened was Friday 22nd January 2021, which represents the most noticeable outliers in the 3 previous box plots
- After checking the weather condition, there was a heavy snow storm on that day

Note: It seems that the extreme weather conditions are the reason behind the outliers in the set of previous box plots. This will be checked from the weather news. After that, a new set of data related to weather conditions will be created and then used for model fitting in the upcoming part

3.4.4. Road names and number of incidents

`road_location_1` contains road names with or without their preceding number. We will create a new column which contains only road names

```
road_name <- str_split(tc$road_location_1,
                       '^\\d*\\b',
                       simplify = T)[, 2] %>%
  str_trim()
```

The basic idea behind the above code is to:
- Split

(a) `'123 Abc Rd'` into (1) `'123'` and (2) `'Abc Rd'` or
(b) `'Abc Rd'` into (1) `''` and (2) `' Abc Rd'`

- Take the 2nd element
- Trim the space in the case of (b)

However, there is a pitfall for the above approach. For example, `'123e Abc Rd'` is split into `''` and `'123e Abc Rd'`. Taking the 2nd element still gives the road name with the number in front. The following code will count the fail cases

```
road_name %>%
  str_starts('\\d') %>%
  sum()
```

```
[1] 31
```

So there are 31 cases in which the requirement is not satisfied. Let's check these 31 cases

```
tc$road_location_1[road_name %>% str_starts('\\d')]
```

```
 [1] "637 1"                      "19a Crane Lake Dr"
 [3] "4o Roxham Cl"               "800a Windmill Rd"
 [5] "109a Adelaide Ave"          "5o Gary Martin Dr"
 [7] "5o Nelsons Landing Blvd"  "2a Highway 102"
 [9] "2a Campbell Ave"            "800a Windmill Rd"
[11] "25a Lucien Dr"              "1a Highway 102"
[13] "196b Chain Lake Dr"         "7e Highway 111"
[15] "7th Ave"                    "27a Lahey Rd"
[17] "272c Waverley Rd"           "129a Dorothea Dr"
[19] "1113e Ramp"                 "441692 363"
[21] "294 1"                      "28a Rosedale Ave"
[23] "4a Parkmoor Ave"            "2040b Maynard St"
[25] "361windmill Rd"             "45c Lahey Dr"
[27] "1d Highway 103"             "196b Chain Lake Dr"
[29] "300a Murray Mckay Bridge" "7th Street"
[31] "3b Highway 102"
```

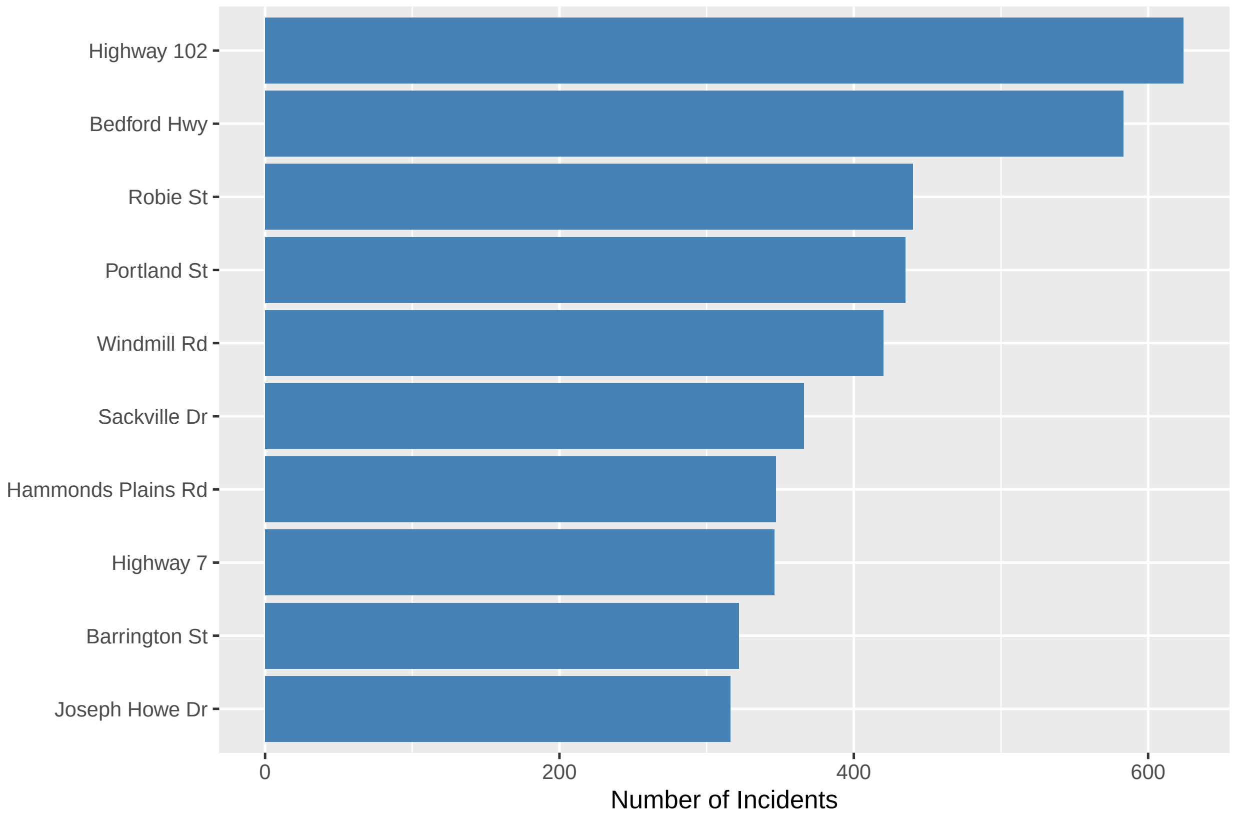
Indeed, all of them have the format of `'123e Abc Rd'` with 3 exceptions under the format of `'123 456'` which are incorrectly entered data

Because 31 is too small when compared to 21127, these 31 rows will be dropped in this part

```
tc_road_name = tc %>%
  mutate(road_name = road_name) %>%
  filter(!str_starts(road_name, '\\d'))
```

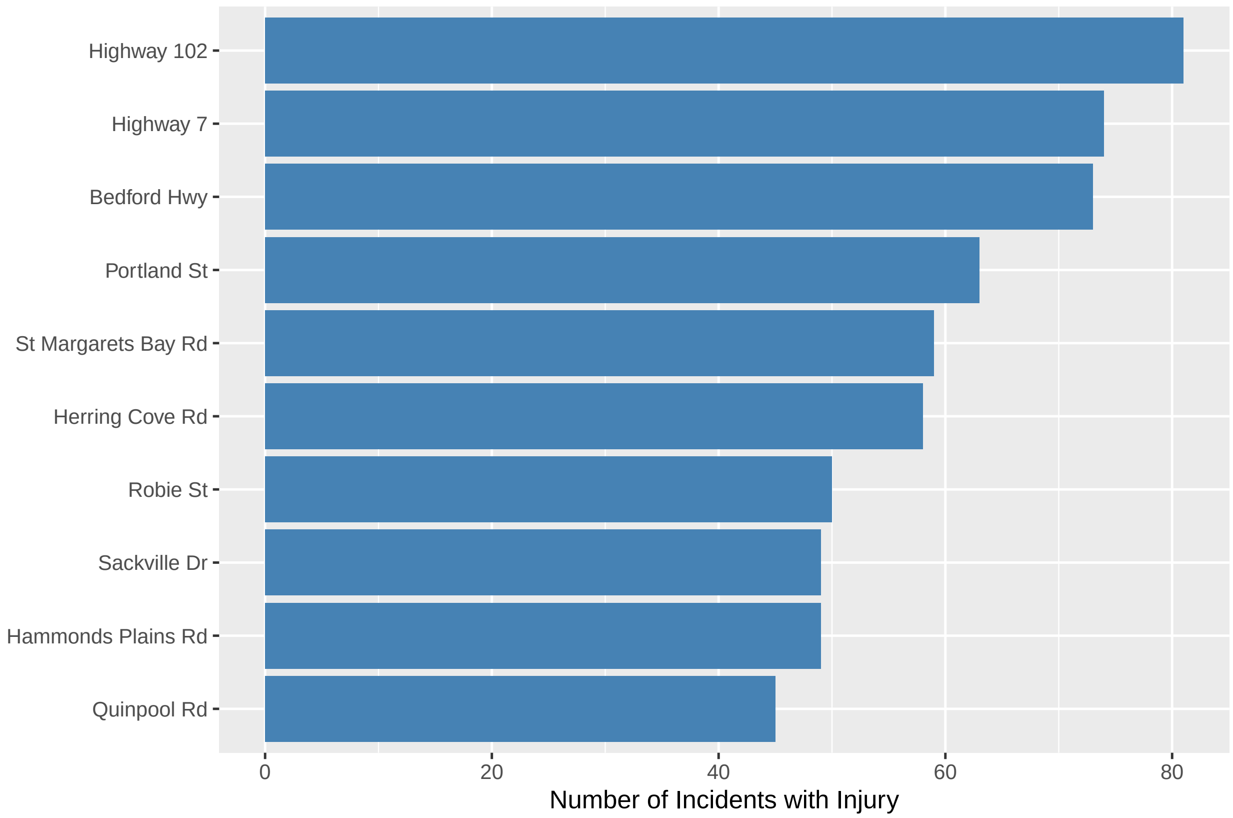
The roads with highest number of incidents are shown in the following bar plot

```
tc_road_name %>%
  count(road_name) %>%
  arrange(desc(n)) %>%
  slice(1:10) %>%
  ggplot(aes(x = fct_reorder(road_name, n), y = n)) +
  geom_col(fill = 'steelblue') +
  labs(x = NULL, y = "Number of Incidents") +
  coord_flip()
```



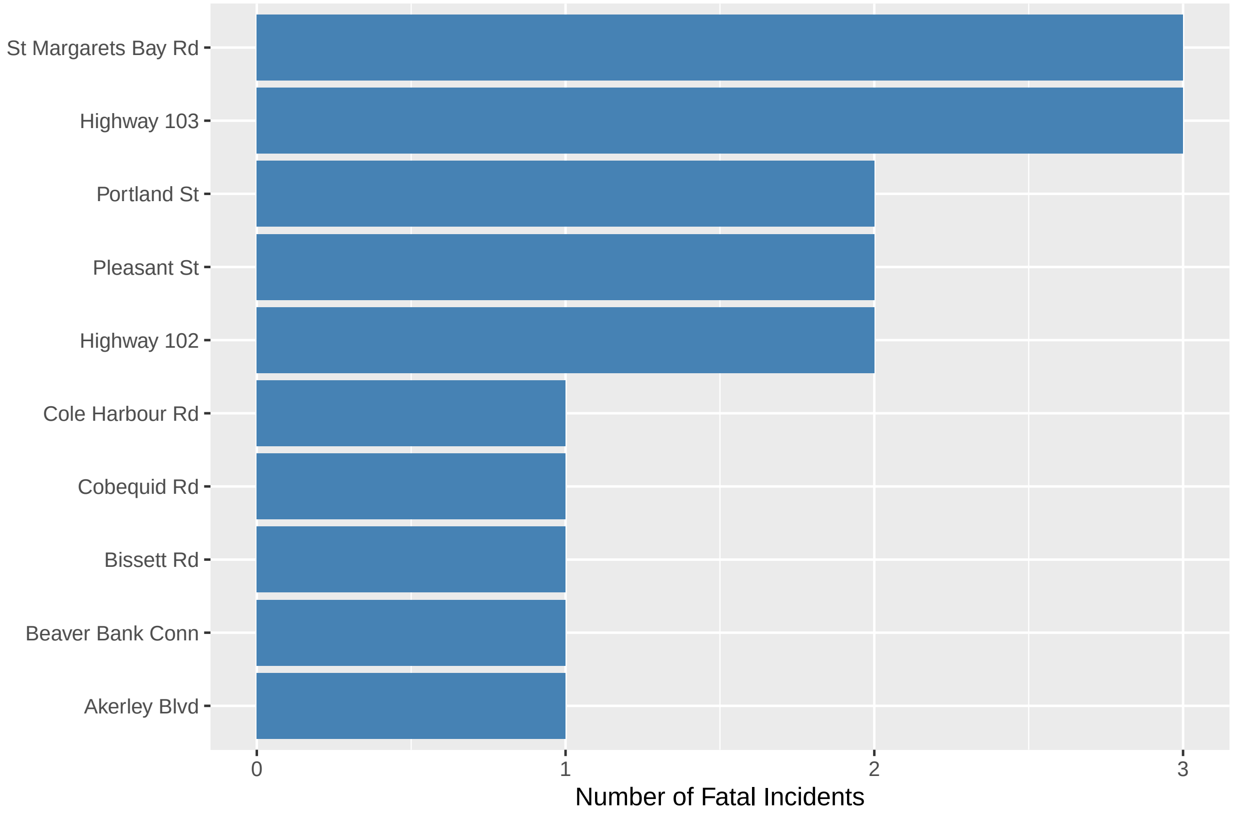
The roads with highest number of incidents with **INJURY** are shown in the following bar plot

```
tc_road_name %>%
  filter(non_fatal_injury == 'Yes' | fatal_injury == 'Yes') %>%
  count(road_name) %>%
  arrange(desc(n)) %>%
  slice(1:10) %>%
  ggplot(aes(x = fct_reorder(road_name, n), y = n)) +
  geom_col(fill = 'steelblue') +
  labs(x = NULL, y = "Number of Incidents with Injury") +
  coord_flip()
```



The roads with highest number of incidents with **FATAL-INJURY** are shown in the following bar plot

```
tc_road_name %>%
  filter(fatal_injury == 'Yes') %>%
  count(road_name) %>%
  arrange(desc(n)) %>%
  slice(1:10) %>%
  ggplot(aes(x = fct_reorder(road_name, n), y = n)) +
  geom_col(fill = 'steelblue') +
  labs(x = NULL, y = "Number of Fatal Incidents") +
  coord_flip()
```



Comment:
- Highway 102, Bedford Hwy and Portland St are present in the list of *"Top 5 Incidents"* and the list of *"Top 5 Incidents with Injury"*
- Highway 103 is the road with the highest number of fatal-injuries but is not in the list of *"Top 5 Incidents"* or the list of *"Top 5 Incidents with Injury"*

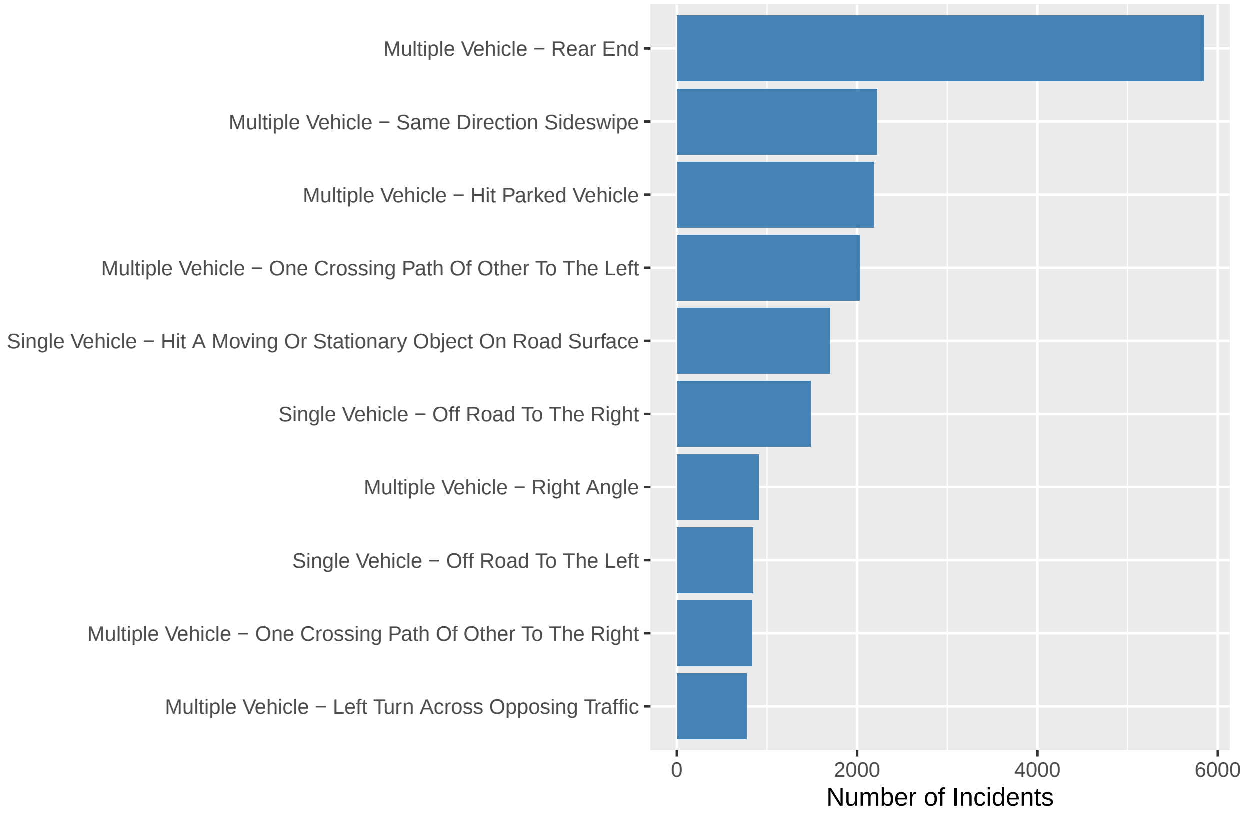
3.4.5. Spatial distribution of incidents A map with clusters of markers will show the spatial distribution of incidents from 2018 to 2021

```
tc %>%
  select(lat = wgs84_lat_coord,
         lng = wgs84_lon_coord) %>%
  leaflet() %>%
  addTiles() %>%
  addMarkers(clusterOptions = markerClusterOptions())
```

Comment: There is no surprise when Downtown Halifax with a high traffic volume has been a hotspot for traffic incidents over the past 4 years

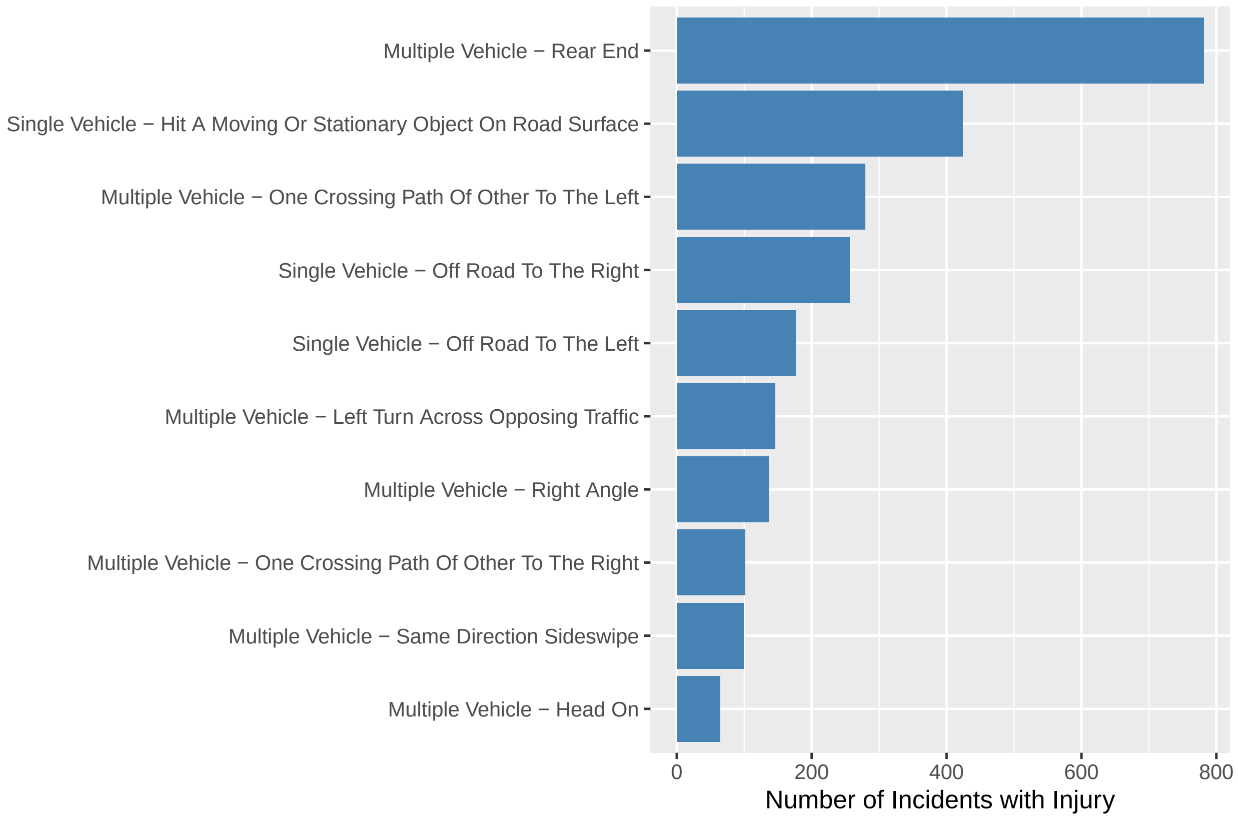
3.4.6. Collision configurations and number of incidents The types of collision configuration with highest number of incidents are shown in the following bar plot

```
tc %>%
  filter(collision_configuration != 'Not Specified') %>%
  count(collision_configuration) %>%
  arrange(desc(n)) %>%
  slice(1:10) %>%
  ggplot(aes(x = fct_reorder(collision_configuration, n), y = n)) +
  geom_col(fill = 'steelblue') +
  labs(x = NULL, y = "Number of Incidents") +
  coord_flip()
```



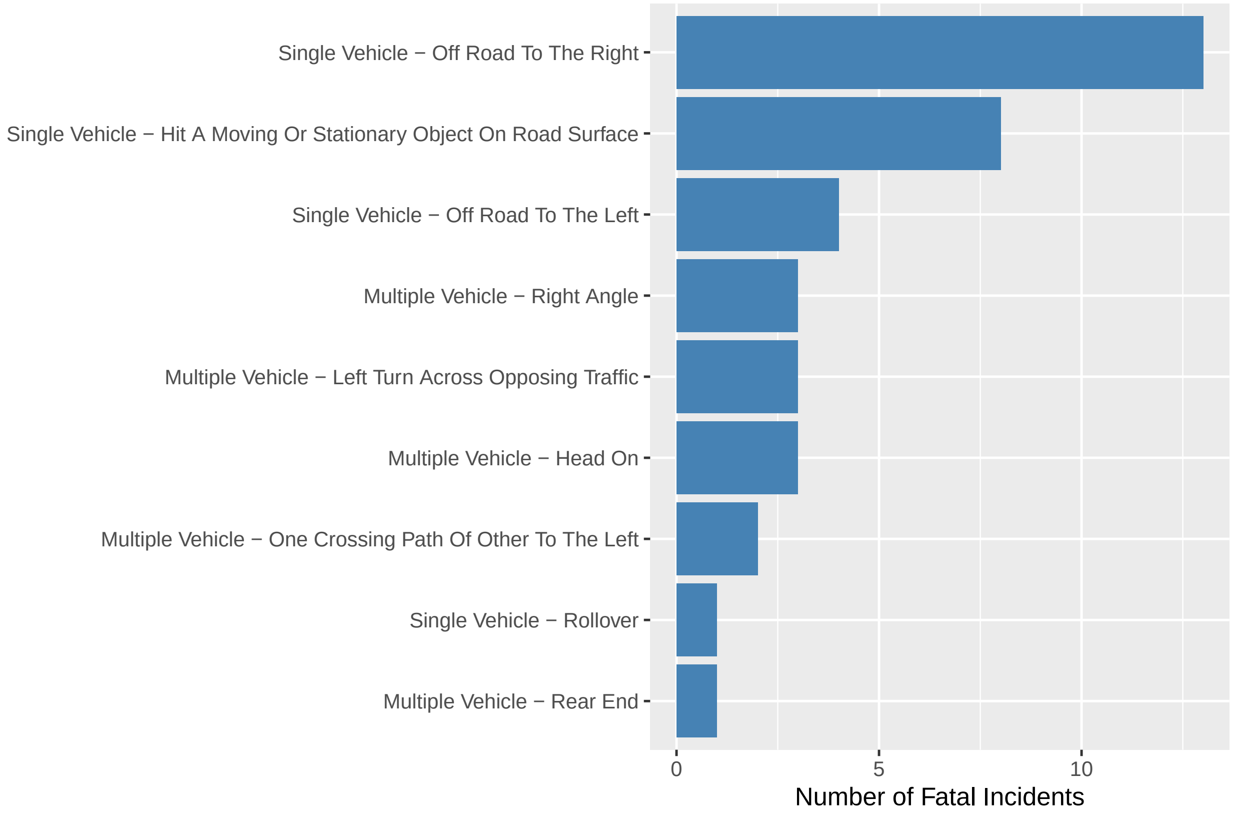
The types of collision configuration with highest number of incidents with **INJURY** are shown in the following bar plot

```
tc %>%
  filter(collision_configuration != 'Not Specified') %>%
  filter(non_fatal_injury == 'Yes' | fatal_injury == 'Yes') %>%
  count(collision_configuration) %>%
  arrange(desc(n)) %>%
  slice(1:10) %>%
  ggplot(aes(x = fct_reorder(collision_configuration, n), y = n)) +
  geom_col(fill = 'steelblue') +
  labs(x = NULL, y = "Number of Incidents with Injury") +
  coord_flip()
```



The types of collision configuration with highest number of incidents with **FATAL-INJURY** are shown in the following bar plot

```
tc %>%
  filter(collision_configuration != 'Not Specified') %>%
  filter(fatal_injury == 'Yes') %>%
  count(collision_configuration) %>%
  arrange(desc(n)) %>%
  slice(1:10) %>%
  ggplot(aes(x = fct_reorder(collision_configuration, n), y = n)) +
  geom_col(fill = 'steelblue') +
  labs(x = NULL, y = "Number of Fatal Incidents") +
  coord_flip()
```



Comment:
- It turns out that incidents with single vehicle were the ones with highest number of fatality
- Meanwhile, incidents related to rear-end collisions were the most frequent types

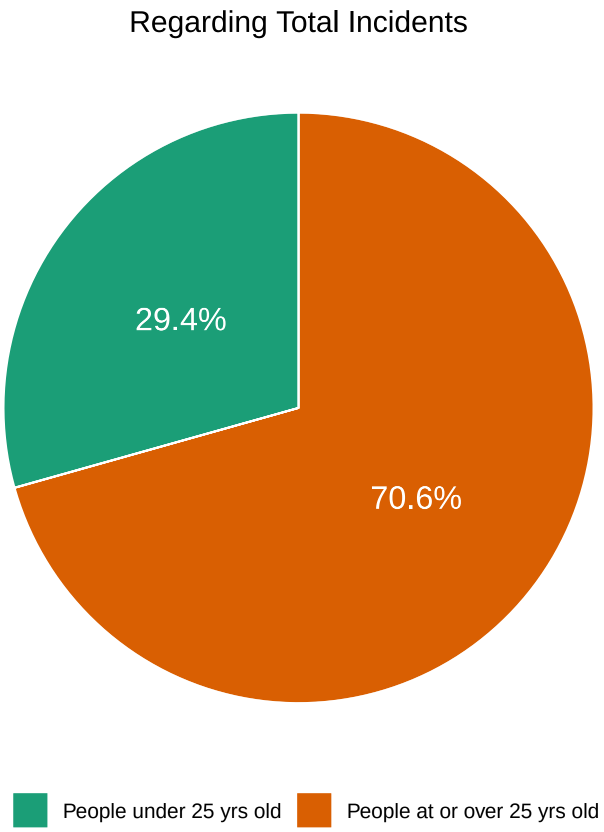
3.4.7. Young demographic and number of incidents In this part, we will look into how much drivers under the age of 25 accounted for traffic incidents

A function named `young_pie` will be created to generate a pie chart showing the percentage of young drivers involved in total incidents, incidents with injury or fatal incidents

```
young_pie <- function(tc, incident = 'total') {
  if (incident == 'fatal') {
    tc_young <- tc %>%
      filter(fatal_injury == 'Yes')
    title <- 'Regarding Fatal Incidents'
  } else if (incident == 'injury') {
    tc_young <- tc %>%
      filter(non_fatal_injury == 'Yes' | fatal_injury == 'Yes')
    title <- 'Regarding Incidents with Injury'
  } else {
    tc_young <- tc
    title <- 'Regarding Total Incidents'
  }
  tc_young %>%
    count(young_demographic) %>%
    mutate(percent = n/sum(n) * 100,
           ypos = cumsum(percent) - 0.5*percent) %>%
    ggplot(aes(x = '', y = percent)) +
    geom_bar(aes(fill = fct_rev(young_demographic)),
             stat = 'identity',
             color = 'white') +
    scale_fill_brewer(palette = 'Dark2',
                      name = NULL,
                      labels = c('People under 25 yrs old',
                                 'People at or over 25 yrs old')) +
    coord_polar(theta = 'y') +
    geom_text(aes(y = ypos,
                  label = str_c(round(percent, 1), '%')),
              color = "white",
              size = 5) +
    theme_void() +
    guides(fill = guide_legend(nrow = 1)) +
    theme(legend.position = 'bottom',
          plot.title = element_text(hjust = 0.5)) +
    ggtitle(title)
}
```

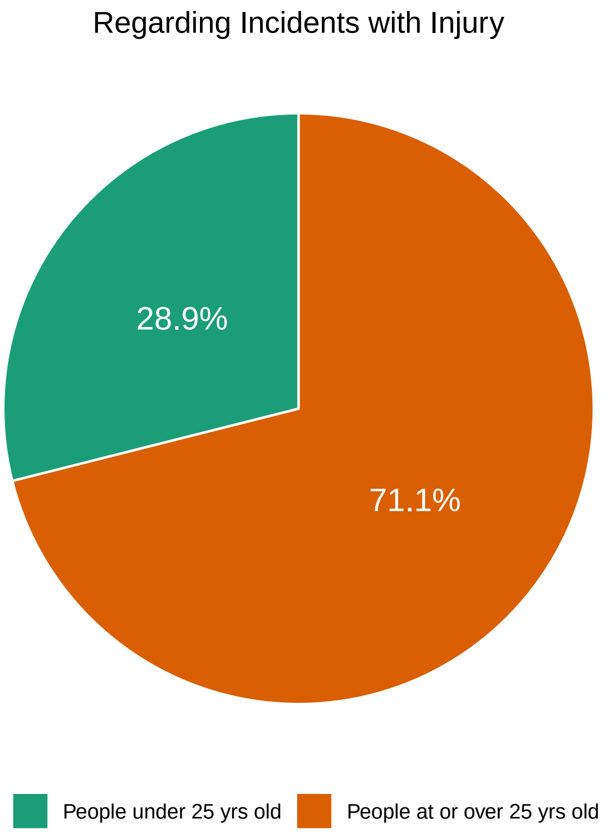
The percentage of young drivers involved in total incidents is shown in the following pie chart

```
young_pie(tc, 'total')
```



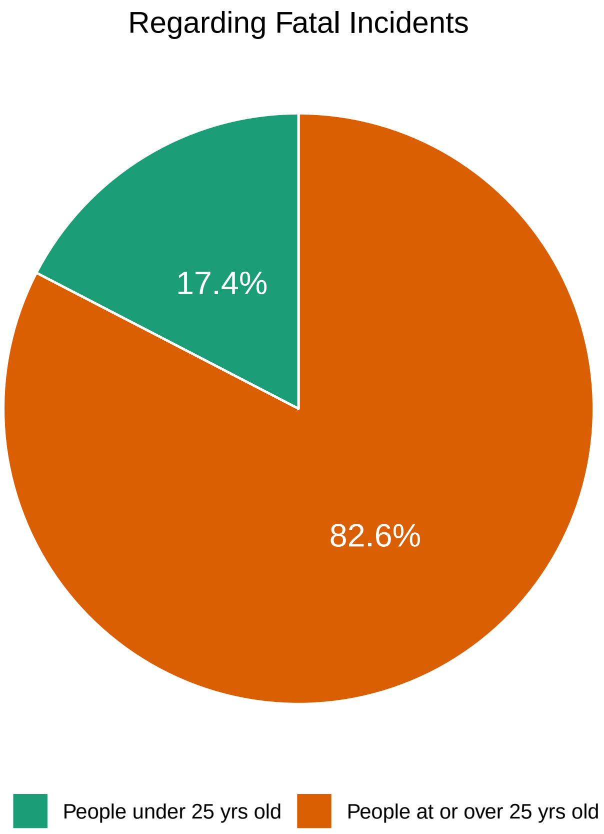
The percentage of young drivers involved in incidents with injury is shown in the following pie chart

```
young_pie(tc, 'injury')
```



The percentage of young drivers involved in fatal incidents is shown in the following pie chart

```r
young_pie(tc, 'fatal')
```



Comment: It is surprising that the percentage of young drivers involved in total incidents, incidents with injury and fatal incidents was much smaller than the figure for older drivers. This seems to be contradictory to the belief that experienced drivers are much safer in driving than new ones

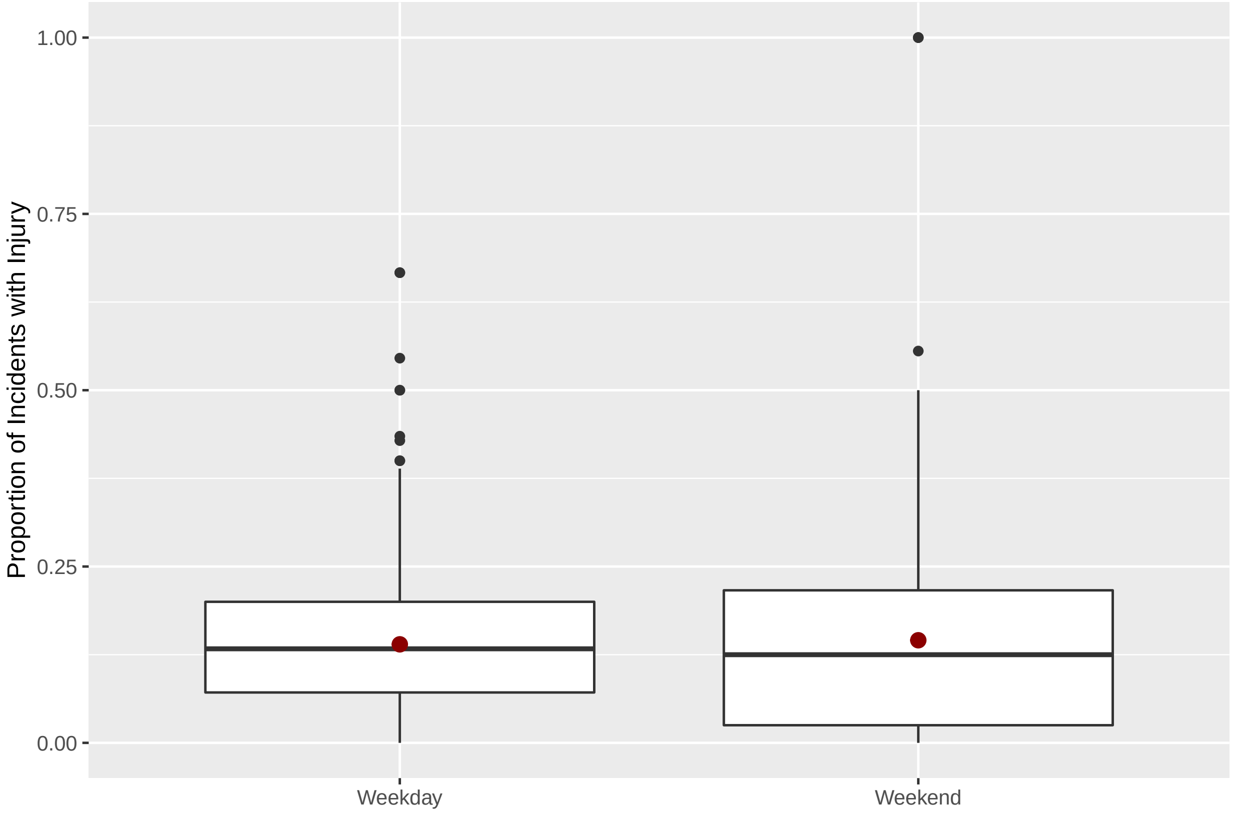
3.4.8. Effect of weekends/weekdays on the risk of injury As suggested above, the average number of daily incidents is lower on weekends than on weekdays which can be explained by the fact that people tend to spend their time with family at home during weekends, resulting to a lower volume of traffic. However, a lower volume of traffic may lead to an increase in speeding and a higher risk of injury which may or may not be fatal when an accident happens. The purpose of this part is to check this argument

First, we have to perform data wrangling to get the data necessary for this part, which is named `wd_we_prop_injury`

```
wd_we_prop_injury <- tc %>%
  group_by(date, dow) %>%
  summarize(
    total = n(),
    injury = sum(non_fatal_injury == 'Yes' | fatal_injury == 'Yes'),
    prop_injury = injury/total) %>%
  mutate(wd_we = case_when(
    dow == 'Sun' | dow == 'Sat' ~ 'Weekend',
    TRUE ~ 'Weekday')) %>%
  select(wd_we, prop_injury)
```

A dual box plot is provided below to see the distribution of the proportion of incidents with injury for weekends and weekdays

```
wd_we_prop_injury %>%
  ggplot(aes(x = wd_we, y = prop_injury)) +
  geom_boxplot() +
  stat_summary(fun = mean, color = 'darkred', size = 0.5) +
  labs(x = NULL, y = "Proportion of Incidents with Injury")
```

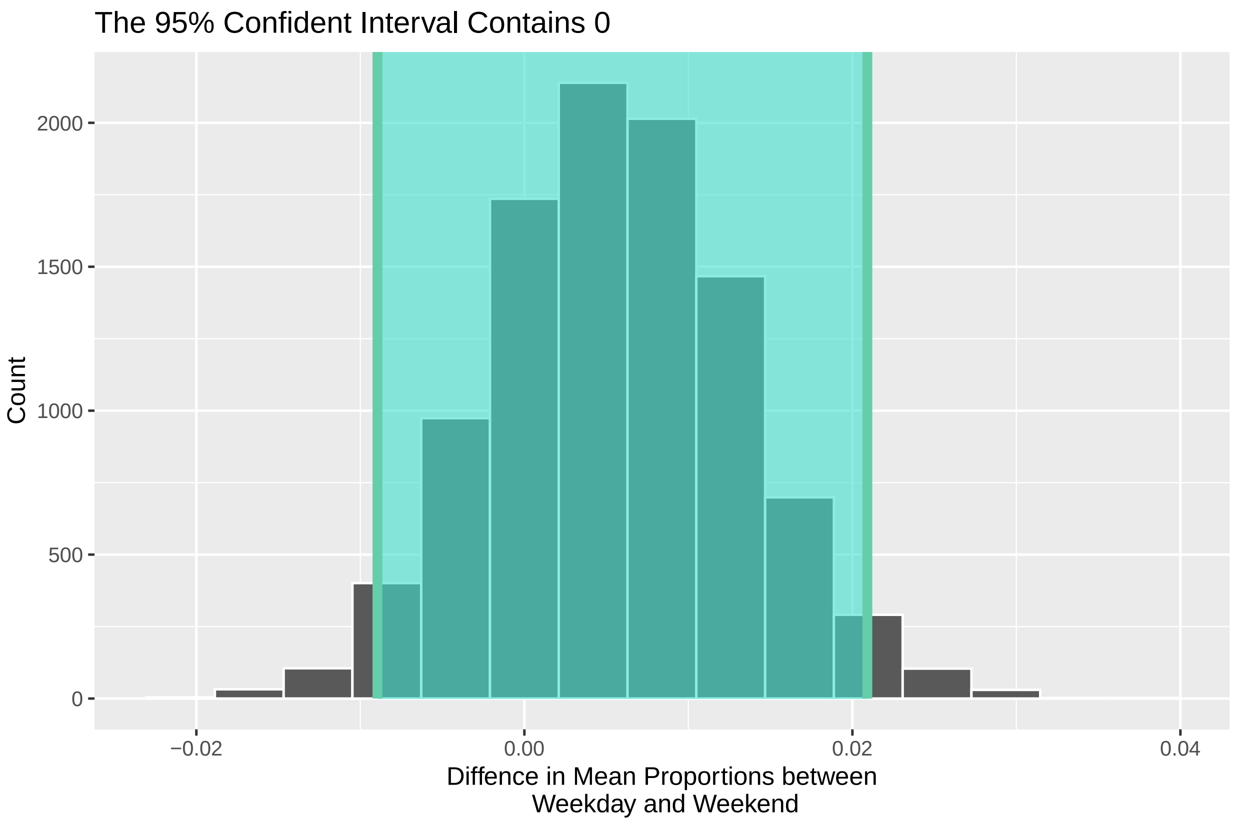


As shown in the box plot, the mean proportion of weekend is marginally higher than the mean proportion of week day. Next, we will construct a confident interval with a confident level of 0.95 for the difference of mean proportions between weekend and weekday using the approach of bootstrap

```
wd_we_boot_distribution <- wd_we_prop_injury %>%
  specify(prop_injury ~ wd_we) %>%
  generate(reps = 10000, type = 'bootstrap') %>%
  calculate(stat = 'diff in means', order = c('Weekend', 'Weekday'))

wd_we_ci_percentile <- wd_we_boot_distribution %>%
  get_ci(type = 'percentile', level = 0.95)

wd_we_boot_distribution %>%
  visualize() +
  shade_ci(wd_we_ci_percentile) +
  labs(x = 'Diffence in Mean Proportions between\n Weekday and Weekend',
       y = 'Count',
       title = 'The 95% Confident Interval Contains 0')
```



Comment: Because the 95% Confident Interval is [-0.009, 0.0209] which contains 0, we cannot claim that the risk of injury on weekend is higher than that on weekday

3.4.9. Effect of aggressive and impaired driving on the risk of injury

In this part, we will compare which driving type is more dangerous: aggressive driving or impaired driving. Personally, I am inclined to impaired driving. I will use the data to test this hypothesis with significance level of 0.05

- Null hypothesis: Risk of injury is the same for both driving conditions
- Alternative: Risk of injury is higher in the case of impaired driving

This code will extract necessary data named `agg_imp` from the clean data `tc`. It should be noted that an incident in `agg_imp` falls into only one of two driving conditions, either aggressive driving or impaired driving

```
tc_agg <- tc %>%
  filter(aggressive_driving == 'Y',
         distracted_driving == 'N',
         impaired_driving == 'N') %>%
  mutate(injury = case_when(
    non_fatal_injury == 'Yes' | fatal_injury == 'Yes' ~ 'Yes',
    TRUE ~ 'No'))

tc_imp <- tc %>%
  filter(aggressive_driving == 'N',
         distracted_driving == 'N',
         impaired_driving == 'Y') %>%
  mutate(injury = case_when(
    non_fatal_injury == 'Yes' | fatal_injury == 'Yes' ~ 'Yes',
    TRUE ~ 'No'))

agg_imp <- tibble(
  condition = c(rep('aggressive', nrow(tc_agg)),
                rep('impaired', nrow(tc_imp))),
  injury = c(tc_agg$injury, tc_imp$injury)) %>%
  sample_n(size = nrow(tc_agg) + nrow(tc_imp))
```

We have the contingency table as follows:

```
agg_imp %>%
  count(condition, injury) %>%
  pivot_wider(names_from = injury, values_from = n) %>%
  kable("html") %>%
  kable_styling(full_width = F)
```

condition

No

Yes

aggressive

4065

804

impaired

194

40

Next, we will calculate the observed difference in risk of injury between impaired and aggressive driving

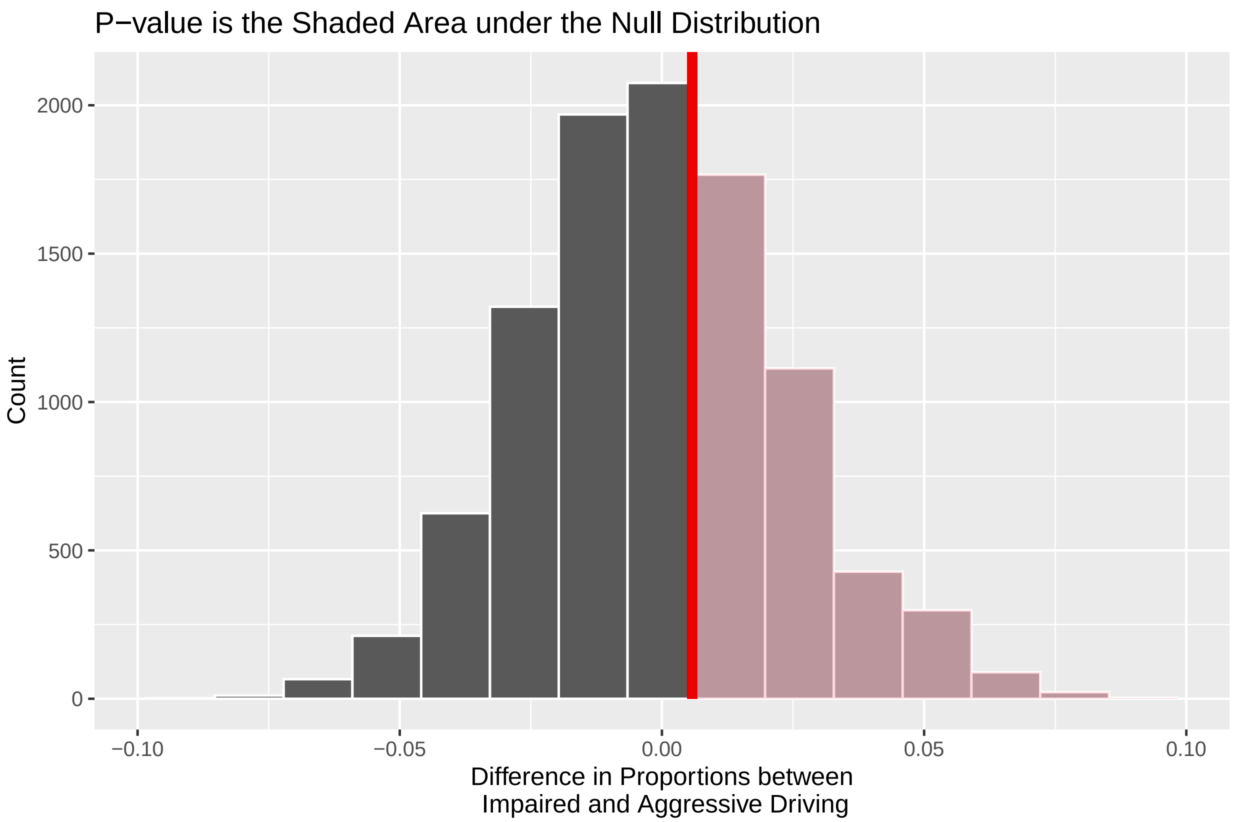
```
obs_diff_prop = agg_imp %>%
  specify(injury ~ condition, success = 'Yes') %>%
  calculate(stat = 'diff in props',
            order = c('impaired', 'aggressive'))
```

The difference in proportions of injury between impaired and aggressive driving is 0.0058. We will perform hypothesis testing using bootstrap

```
agg_imp_null_distribution <- agg_imp %>%
  specify(injury ~ condition, success = 'Yes') %>%
  hypothesize(null = 'independence') %>%
  generate(reps = 10000, type = 'permute') %>%
  calculate(stat = 'diff in props',
            order = c('impaired', 'aggressive'))

agg_imp_p_value <- agg_imp_null_distribution %>%
  get_p_value(obs_diff_prop, direction = 'right')

agg_imp_null_distribution %>%
  visualize() +
  shade_p_value(obs_diff_prop, direction = 'right') +
  labs(title = 'P-value is the Shaded Area under the Null Distribution',
       x = 'Difference in Proportions between\n Impaired and Aggressive Driving',
       y = 'Count')
```



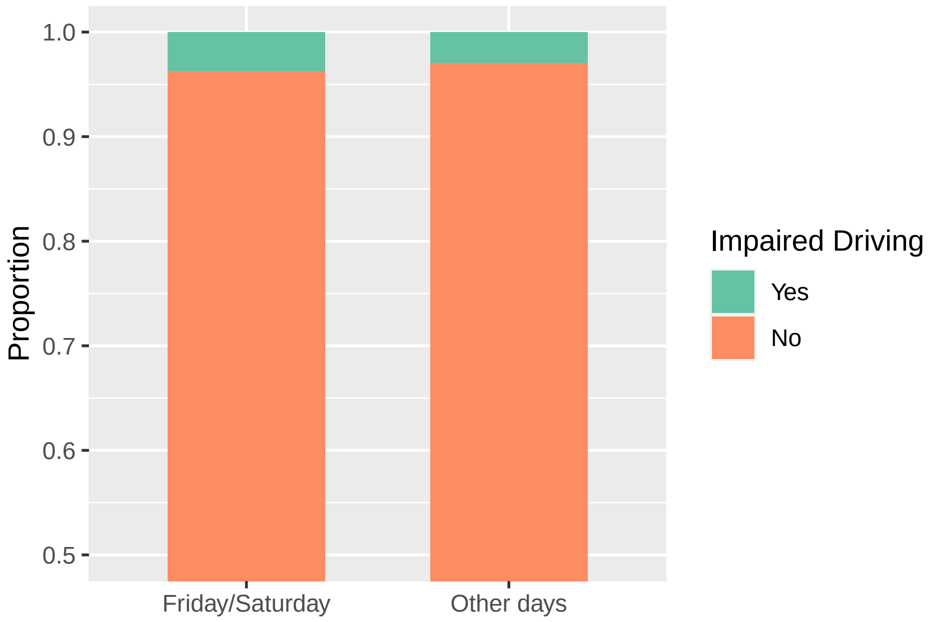
Comment: Because p-value of the test is 0.4393 is larger than the significance level of 0.05, we cannot reject the null hypothesis. Therefore we do not have enough clue to say that the risk of injury due to impaired driving is higher.

3.4.10. Friday and Saturday and incidents related to impaired driving Friday and Saturday evenings are considered as party time! This may be somewhat correlated to the traffic violated cases related to impair driving. This part will look into this argument

As usual, related data need to be extracted

```
fri_sat_imp <- tc %>%
  mutate(fri_sat = case_when(
    dow %in% c('Fri', 'Sat') ~ 'Friday/Saturday',
    TRUE ~ 'Other days')) %>%
  count(fri_sat, impaired_driving)

fri_sat_imp %>%
  ggplot(aes(x = fri_sat, y = n, fill = fct_rev(impaired_driving))) +
  geom_col(position = 'fill', width = 0.6) +
  scale_fill_brewer(name = 'Impaired Driving',
                    palette = 'Set2',
                    labels = c('Yes', 'No')) +
  coord_cartesian(ylim = c(0.5, 1)) +
  labs(x = NULL, y = 'Proportion')
```



It can be seen that on Friday/Saturday, the proportion of incidents related to impaired-driving is marginally higher than on the other days of a week. However, is this difference due to sampling variation?

```
fri_sat_imp
```

```
# A tibble: 4 x 3
  fri_sat         impaired_driving     n
  <chr>           <chr>            <int>
1 Friday/Saturday N                 6003
2 Friday/Saturday Y                  231
3 Other days      N                14457
4 Other days      Y                  436
```

We will construct a confident interval with confident level of 95% to answer this question. Because the number of observations is large, a theory-based approach can be used this time

```
n1 = 6003 + 231
p1 = 231/n1
n2 = 14457 + 436
p2 = 436/n2
se = sqrt(p1*(1 - p1)/n1 + p2*(1 - p2)/n2)
z = qnorm(0.975)
lower_ci = (p1 - p2) - z*se
upper_ci = (p1 - p2) + z*se
ci = c(lower_ci, upper_ci)
ci
```

```
[1] 0.002364781 0.013193943
```

Comment: The 95% confident interval ranges from 0.0024 to 0.0132 which is absolutely greater than 0, which means we are 95% confident that the difference in the proportions of incident related to impaired driving between Friday/Saturday and the other days are withing [0.0024, 0.0132] and there is a higher proportion of incidents related to impaired driving on Friday/Saturday

3.4.11. Model of daily incident number The info on storm events below are retrieved from CBC news. There are two main types of severe weather which are snow storm and tropical hurricane. Dorian and Teddy

Hurricanes stroke Nova Scotia in 2019

```
storm_date <- tribble(
  ~date, ~storm,
  '2021-01-02', 'snow',
  '2021-01-22', 'snow',
  '2021-03-19', 'snow',
  '2020-11-03', 'snow',
  '2020-02-10', 'snow',
  '2020-01-16', 'snow',
  '2019-09-07', 'tropical',
  '2019-09-08', 'tropical',
  '2019-09-10', 'tropical',
  '2019-09-09', 'tropical',
  '2019-09-23', 'tropical',
  '2019-09-22', 'tropical',
  '2019-11-28', 'snow',
  '2018-03-08', 'snow',
  '2018-12-07', 'snow',
  '2018-11-16', 'snow'
)
storm_date <- storm_date %>%
  mutate(date = ymd(date))
```

Let's join `tc` and `storm_date` data frames and change `NA` in `storm` column of the resulting data frame to `'normal'`

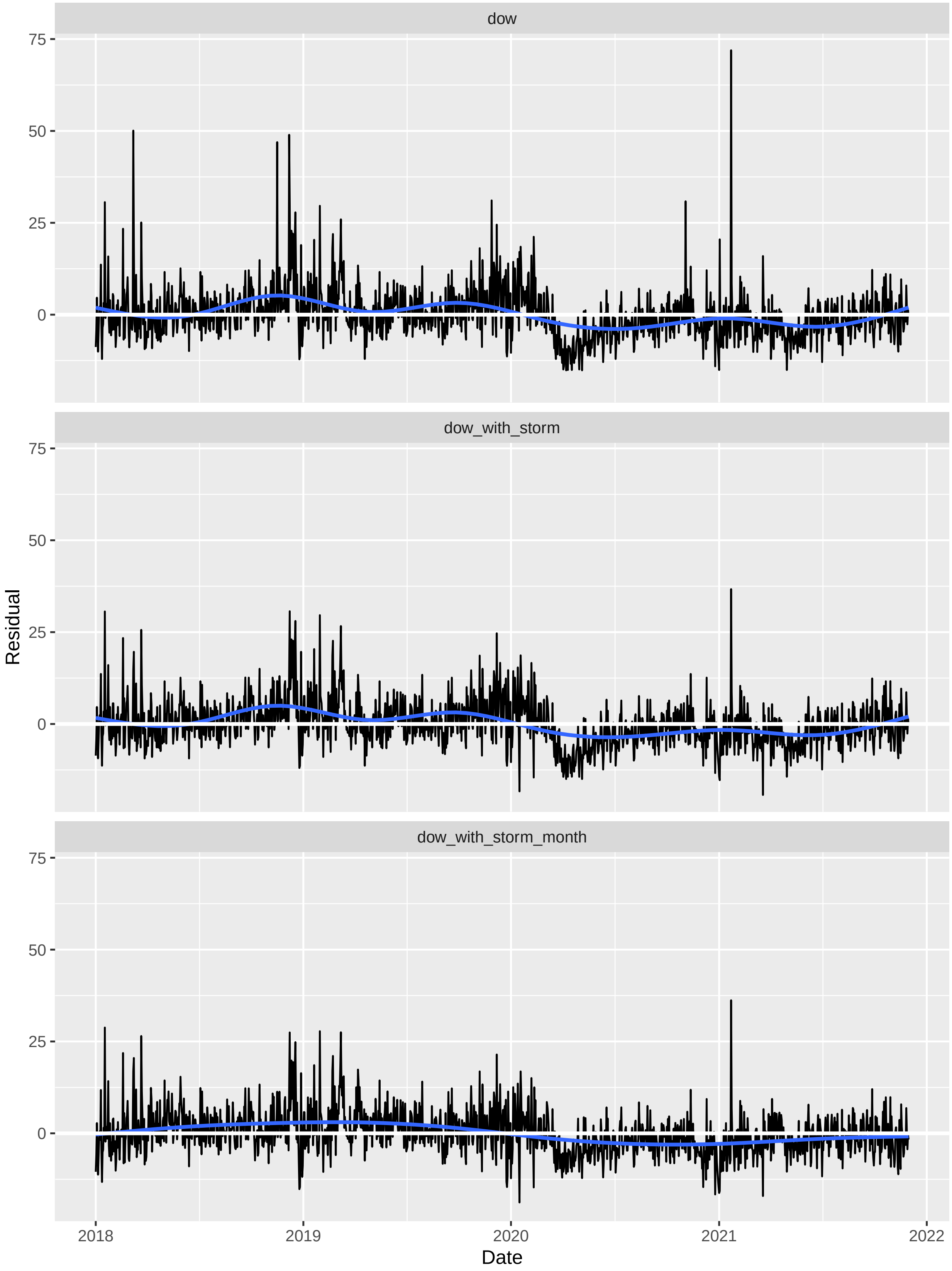
```
tc_model_data <- left_join(tc, storm_date, by = 'date') %>%
  mutate(storm = replace_na(storm, 'normal')) %>%
  group_by(date, dow, month, storm) %>%
  summarize(n = n())
```

We will build models with the number of daily incidents (`n`) as a response variable and day of week (`dow`), month (`month`) and extreme event (`storm`) as predictors

```
model1 <- lm(n ~ dow, data = tc_model_data)
model2 <- lm(n ~ dow + storm, data = tc_model_data)
model3 <- lm(n ~ dow + storm + month, data = tc_model_data)
list_models <- list(dow = model1,
                    dow_with_storm = model2,
                    dow_with_storm_month = model3)
```

Residuals from each model will be examined throughout the whole timeframe using visualizations

```
tc_model_data %>%
  gather_residuals(dow = model1,
                   dow_with_storm = model2,
                   dow_with_storm_month = model3) %>%
  ggplot(aes(date, resid)) +
  geom_line() +
  geom_smooth(se = F) +
  labs(x = "Date", y = "Residual") +
  facet_wrap(~ model, nrow = 3) +
  geom_hline(yintercept = 0, size = 1, colour = "white")
```



The R-squared of 3 models are displayed in the table below

```
list_models %>%
  map_dbl(~ summary(.)[['r.squared']]) %>%
  enframe(name = 'model', value = 'r_squared') %>%
  kable("html") %>%
  kable_styling(full_width = F)
```

model

r_squared

dow

0.1332107

dow_with_storm

0.2922833

dow_with_storm_month

0.3608928

Comment: From the visualization and the table with R squared values, the 3rd model is the best candidate among 3. However, there is much more room to build a better model using other predictors and methods.

4. Conclusion

In terms of temporal scale

- The data ended on Nov 30, 2021
- The spikes in the number of daily incidents generally occurred in the span of time ranging from Nov to Mar, which corresponds to winter seasons
- There was a sharp fall in the number of daily incidents in Apr 2020 which corresponds to the point of time when the policy of social isolation was implemented
- From 2018 to 2019, the average number of daily incidents slightly increased
- From 2019 to 2020, the average number of daily incidents sharply decreased
- From 2020 to 2021, the average number of daily incidents remained low despite an increase
- The sharp decrease in the average number of daily incidents must be attributed to the outbreak of COVID-19
- The slight increase in the average number of daily incidents may be attributed to the increase in the population of the region
- Weekends had a lower number of daily incidents than weekdays did
- There was fewest incidents on Sunday
- There was most incidents on Friday

In terms of spatial scale

- Highway 102, Bedford Hwy and Portland St are present in the list of “Top 5 Incidents” and the list of “Top 5 Incidents with Injury”

- There is no surprise when Downtown Halifax with a high traffic volume has been a hotspot for traffic incidents over the past 4 years

In terms of driver and collision properties

- It turns out that incidents with single vehicle were the ones with highest number of fatality
- Meanwhile, incidents related to rear-end collisions were the most frequent types
- It is surprising that the percentage of young drivers involved in total incidents, incidents with injury and fatal incidents was much smaller than the figure for older drivers

Hypothesis testing and model fitting

- We cannot claim that the risk of injury on weekend is higher than that on weekday
- We do not have enough clue to say that the risk of injury due to impaired driving is higher
- We are 95% confident that there is a higher proportion of incidents related to impaired driving on Friday/Saturday
- The model using day of week, month and date of extreme climatic event has R-squared value of 0.36. However, there is much more room to build a better model using other predictors and methods