# milestone3

Frank Guo

2024-05-03

## Milestone 3 - Crashes Analysis

**By Xiaodong Guo**

### 1. Goal.

Our project has a significant objective- to construct models identifying the pivotal factors contributing to severe crashes. This crucial task is based on the data provided by the New Zealand Transport Agency(NZTA).

### 2. Data Source.

The original data, meticulously collected, originated from the Waka Kotahi NZ Transport Agency's open data portal(the tutor provided the link in the assignment piece). We specifically downloaded the dataset named "Crash Analysis System (CAS) data" from the "Crash" catalogue, which encompasses all traffic crashes reported to us by the NZ Police. The data format is a "CVS" file. It was created on 3/25/2020 and last updated on 3/14/2024.

The data includes crash datas from 2000 to 2023.

### 3. Data Processing.

We load the data from csv flie. The dataset we got have 72 columns,and 821744 rows.

```
#load data

library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.0     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ---------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(xgboost)
```

```
##
## Attaching package: 'xgboost'
##
## The following object is masked from 'package:dplyr':
##
```

```
##     slice
```

```
#file_path <- file.choose()
set.seed(230)
data <- read.csv("../data/Crash_Analysis_System_(CAS)_data.csv", header = TRUE, sep = ",")

dim(data)
```

```
## [1] 821744     72
```

```
#glimpse(data)
```

The first things we can do is to drop columns not related to our objective.Thus, we select following columns by common sense of mine(not sure if 100% correct,maybe need some hint from Lisa Chen).There are also have some description columns, that is long string values to desript an event or street name. That looks no sense,should drop them too. Like "crashLocation1","crashLocation2".

```
#clean data

columns_to_drop <- c("X","Y","OBJECTID","areaUnitID","crashDirectionDescription","","crashDistance","tla
"crashFinancialYear","fatalCount","debris","meshblockId","northing","easting","objectThrownOrDropped","n
"otherObject","phoneBoxEtc","seriousInjuryCount")
```

Then drop columns that almost all values are Null(more then 99% of the data is null). columns like these useless for the inference. the column names are crashRoadSideRoad" and "intersection".

```
data <- select(data, -one_of(columns_to_drop))

na_percentage <- colMeans(is.na(data))
columns_with_high_na <- names(na_percentage[na_percentage > 0.99])
#print(columns_with_high_na)

data <- data %>% select(-columns_with_high_na)

#table(data$crashSeverity)
```

Define crashSeverity == "Fatal Crash" and crashSeverity == "Serious Crash" as severe crashes given numeric value 1,
Define crashSeverity == "Minor Crash" | crashSeverity == "Non-Injury Crash" as not severe crashes given numeric value 0.
The "crashSeverity" Label will be the target label.

```
data <- data %>%
  mutate(crashSeverity = ifelse(crashSeverity == "Fatal Crash" | crashSeverity == "Serious Crash", 1,
                        ifelse(crashSeverity == "Minor Crash" | crashSeverity == "Non-Injury Crash", 0,
                               2))) %>%  filter(crashSeverity != 2)

#table(data$crashSeverity)
#table(data$weatherA)
#table(data$weatherB)
```

Attributes "weatherA" and "weatherB", are String values could be treated as factors, the Null value or String "None" are replaced as "Others" condition, not too many factors in each attribute, and the combinations also not too many factors but are more sensitive to understand the whole weather situation. In my opinion, these two could be combined as one attribute "weather",much easier to display and dealing with it later.

```
data$weatherA <- ifelse(data$weatherA %in% c("None", "Null"), "Others", data$weatherA)
data$weatherB<- ifelse(data$weatherB %in% c("None", "Null"), "", data$weatherB)
```

```
data <- data %>% unite(weatherA,weatherB,col=weather,sep=" ")
```

Replace the ““,”Null”,“None” value in region with “Others”; Replace the “ ” in other character attributes with “Others”.

```
knitr::kable(table(data$region))
```

| Var1 | Freq |
|---|---|
| | 3188 |
| Auckland Region | 285346 |
| Bay of Plenty Region | 47177 |
| Canterbury Region | 82146 |
| Gisborne Region | 9784 |
| Hawke’s Bay Region | 32388 |
| Manawatū-Whanganui Region | 46329 |
| Marlborough Region | 8266 |
| Nelson Region | 8076 |
| Northland Region | 33299 |
| Otago Region | 44574 |
| Southland Region | 20234 |
| Taranaki Region | 18604 |
| Tasman Region | 7541 |
| Waikato Region | 87849 |
| Wellington Region | 79725 |
| West Coast Region | 7218 |

```
data$region <- ifelse(data$region %in% c("None", "Null"), "Others", data$region)


data[data == ""] <- "Others"
```

List all the attributes with Na value,got attributes below: “advisorySpeed” “bicycle” “bridge” “bus” “carStationWagon”
“cliffBank” “ditch” “fence” “guardRail” “houseOrBuilding”
“kerb” “moped” “motorcycle” “NumberOfLanes” “otherVehicleType”
“overBank” “parkedVehicle” “pedestrian” “postOrPole” “roadworks”
“schoolBus” “slipOrFlood” “speedLimit” “strayAnimal” “suv”
“taxi” “temporarySpeedLimit” “trafficIsland” “trafficSign” “train”
“tree” “truck” “unknownVehicleType” “vanOrUtility” “vehicle”
“waterRiver”
According the descriptions of these attributes, we can use 0 to fill na value.Here is 2 examples about why 0 be used: For “advisorySpeed” or “temporarySpeedLimit” attribute, the value is mean special speed limitation applied or advised in the road which is involed in the crash. use 0 here means no special speed limit applied(according the rode code, that is open road.follows open road speed limit). For other attributes in the list upon, the value indicates the number of item involved in the crash. the Na value means no item(named by attribute name) is involved,that equals to 0.

At the end, all the Na or missing data is imputated and remedied.

```
na_columns <- sapply(data, function(x) any(is.na(x)))
columns_with_na <- names(data)[na_columns]

print(columns_with_na)
```

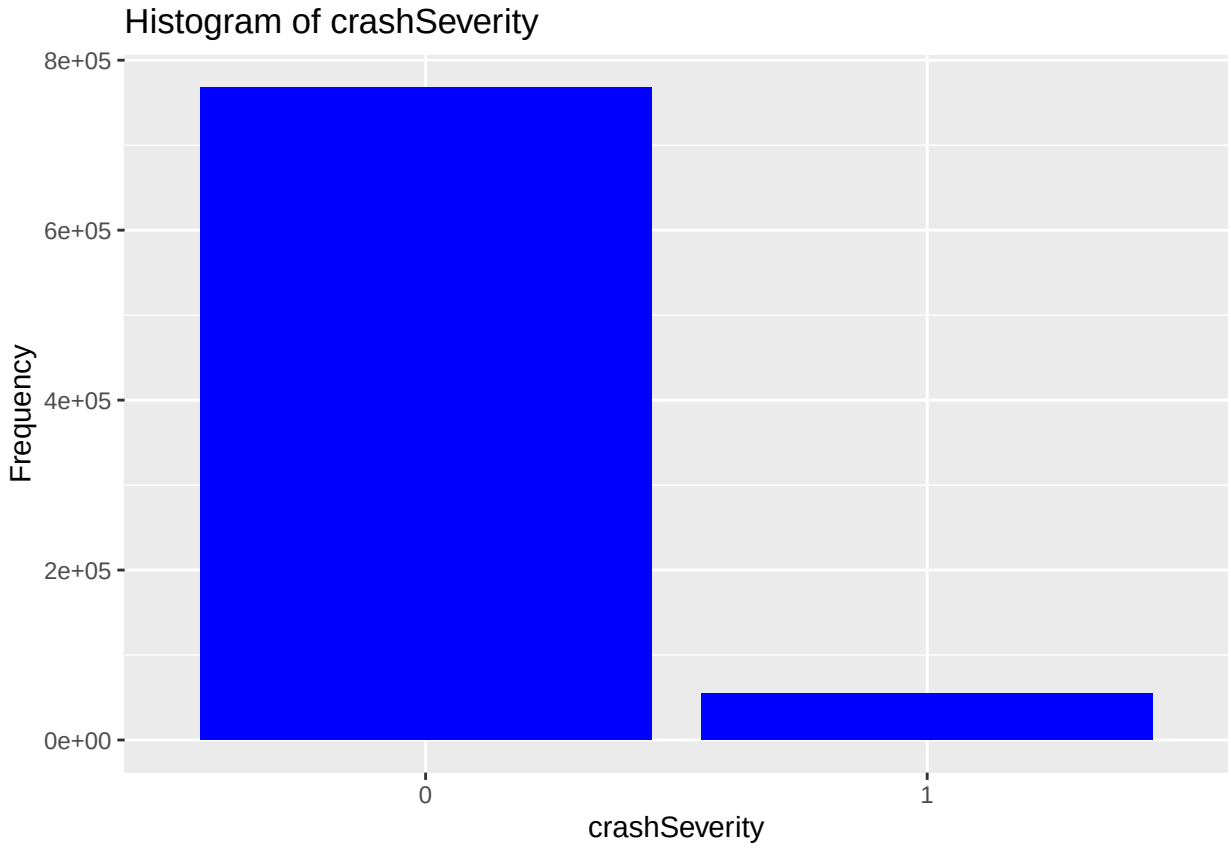
```
##  [1] "advisorySpeed"       "bicycle"             "bridge"
##  [4] "bus"                 "carStationWagon"     "cliffBank"
##  [7] "ditch"               "fence"               "guardRail"
## [10] "houseOrBuilding"     "kerb"                "moped"
## [13] "motorcycle"          "NumberOfLanes"       "otherVehicleType"
## [16] "overBank"            "parkedVehicle"       "pedestrian"
## [19] "postOrPole"          "roadworks"           "schoolBus"
## [22] "slipOrFlood"         "speedLimit"          "strayAnimal"
## [25] "suv"                 "taxi"                "temporarySpeedLimit"
## [28] "trafficIsland"       "trafficSign"         "train"
## [31] "tree"                "truck"               "unknownVehicleType"
## [34] "vanOrUtility"        "vehicle"             "waterRiver"
```

```
data <- data %>%
  mutate_at(vars(one_of(columns_with_na)), ~replace_na(., 0))




#glimpse(data)
```

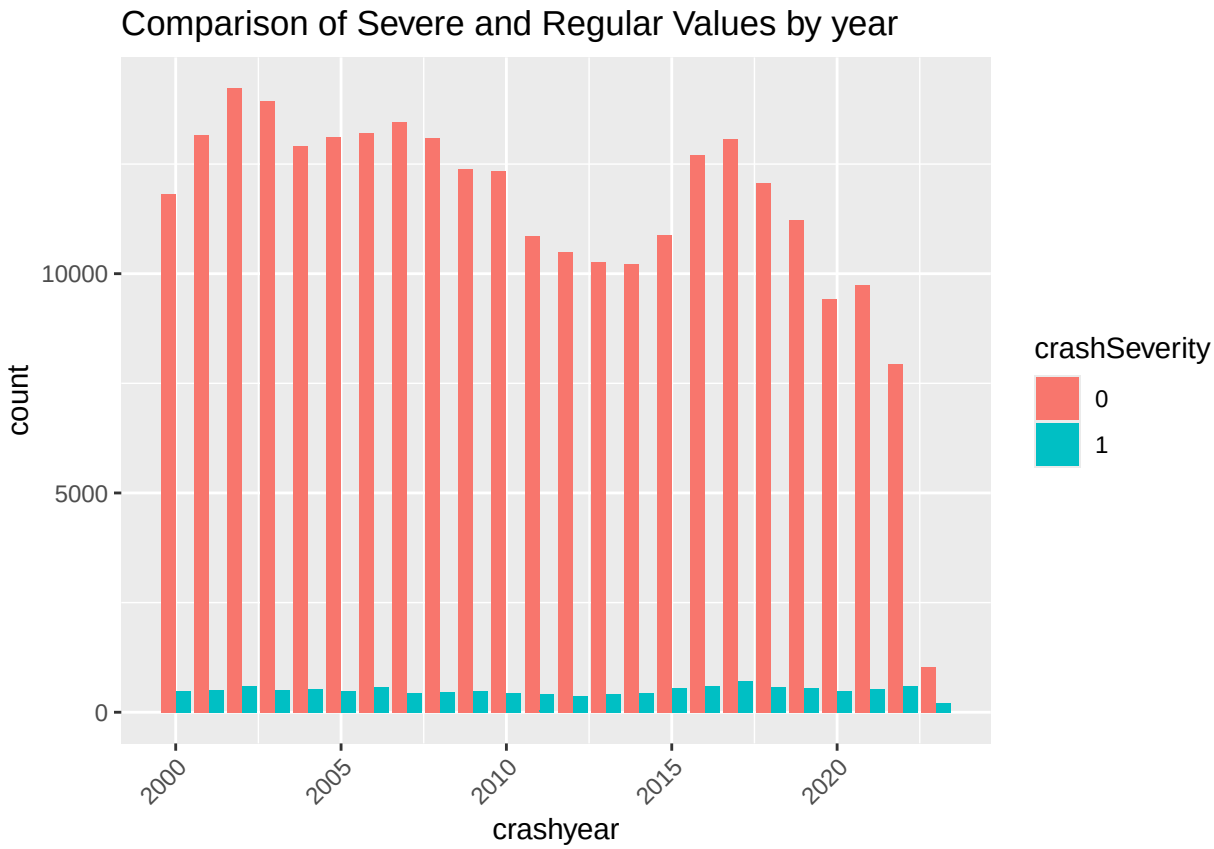
## 4. Data Exploration

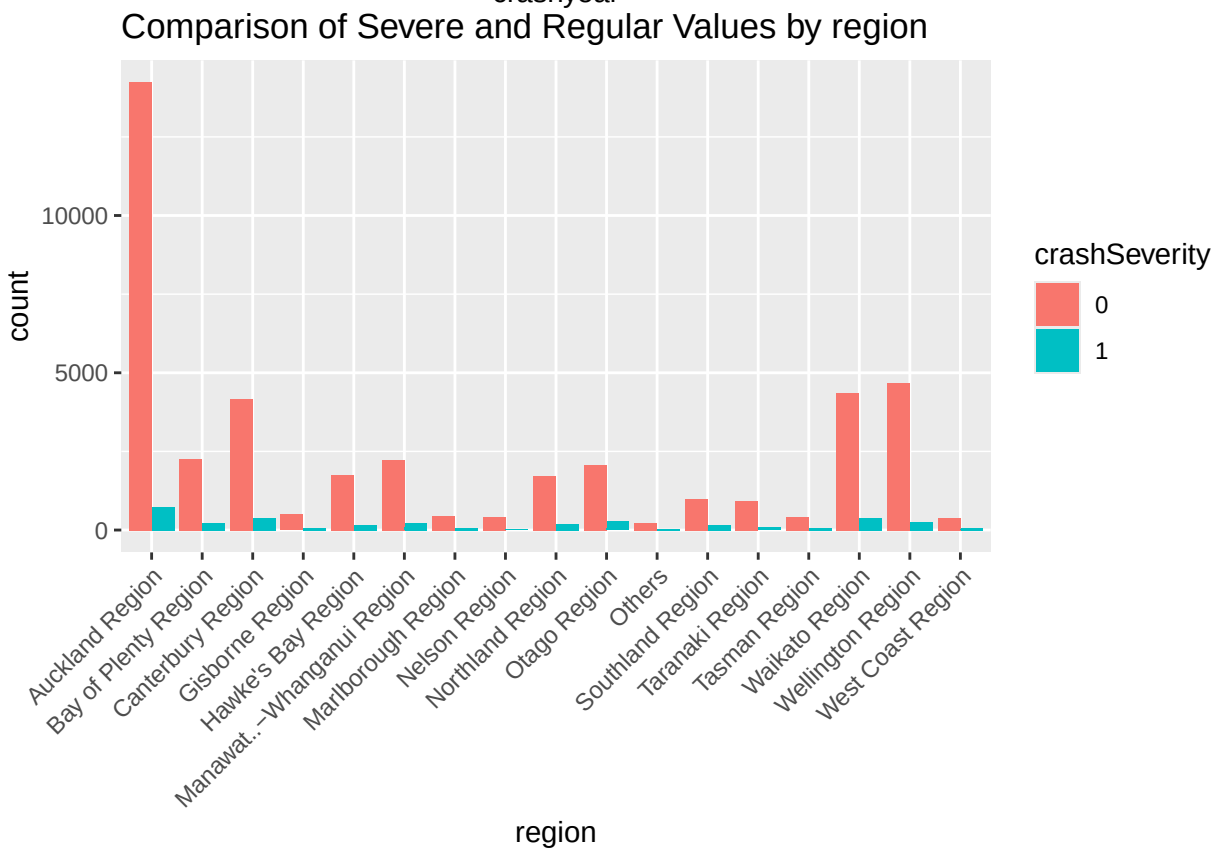
Because all the attributes are categorical attributes, we decide to explore the freqency the most important attributes.
We Plot the target labels and some relations between attributes like crashYear, region, weather and light which are intuitively think that will be key attribute of the cause of incident.
We Can see total crashes decreased by year, but servere crashes not.Also, we found the problem that unbalanced data occurs in predictors.For examples, regarding as weather, we can see most of the crashes(severe and regular) happened in fine day rather than the other weather conditions.
From All of those plots, we found the unbalance of the data.There are 767290 regular crashes,but only 54454 severe crashes. Our target is to find the cause of severe crashes,but the severe observations' size is really small compares to the regular crashes.

| Var1 | Freq |
|---|---|
| 0 | 767290 |
| 1 | 54454 |



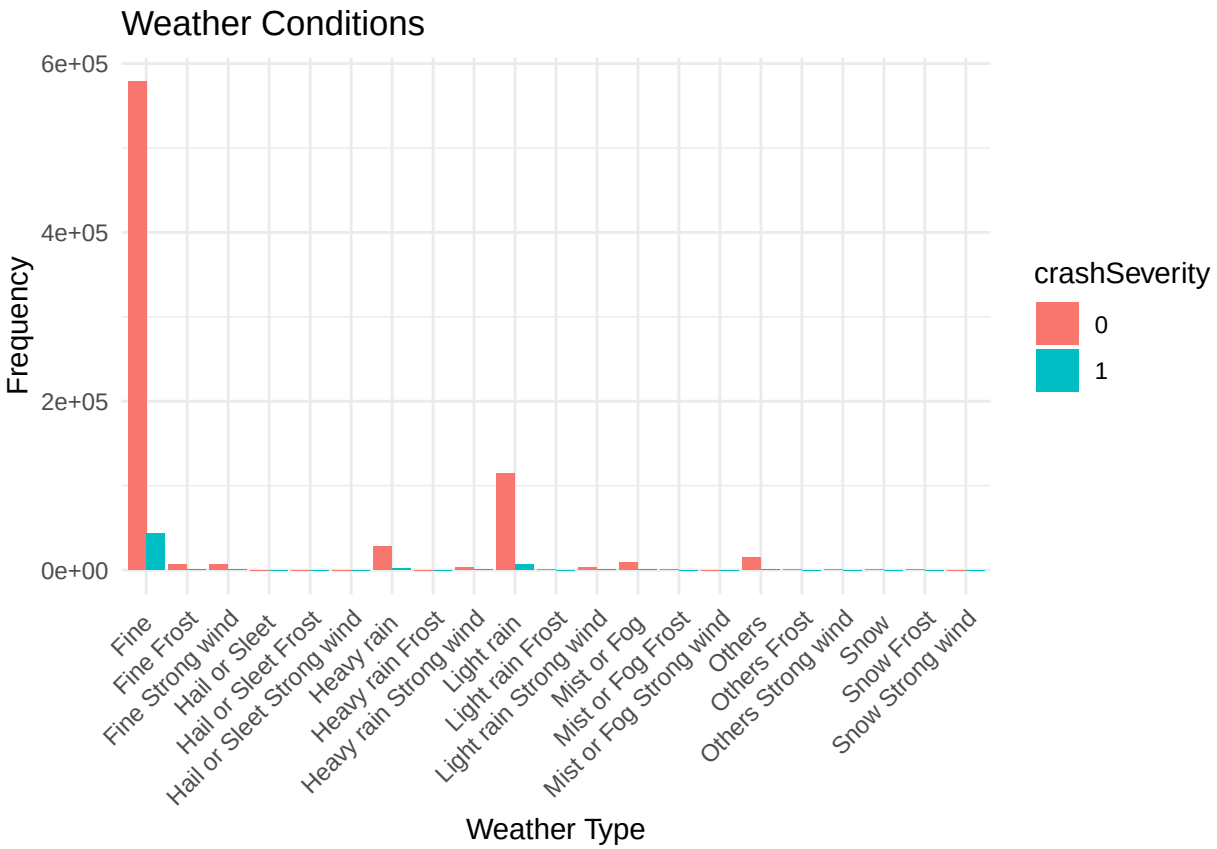

```
## `summarise()` has grouped output by 'crashYear', 'region'. You can override
## using the `.groups` argument.
```



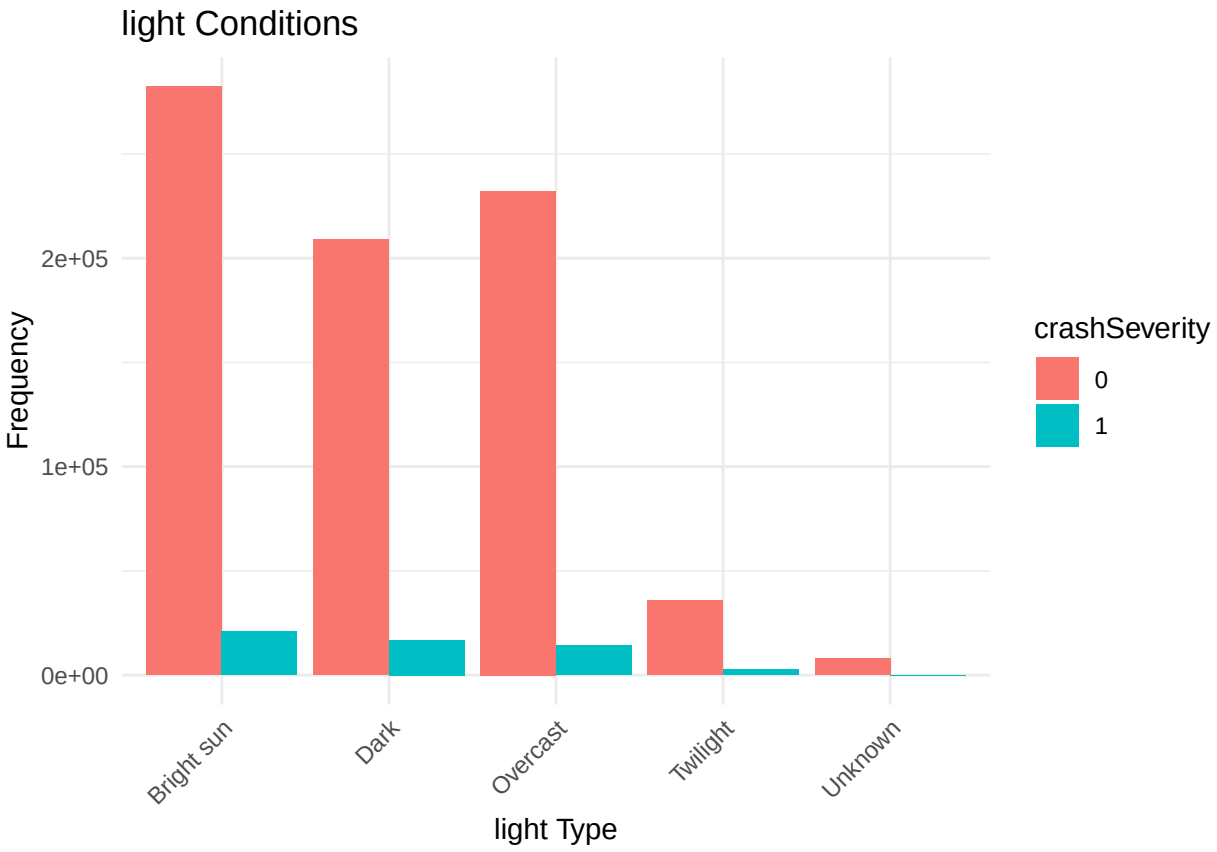




```
## `summarise()` has grouped output by 'weather'. You can override using the
```

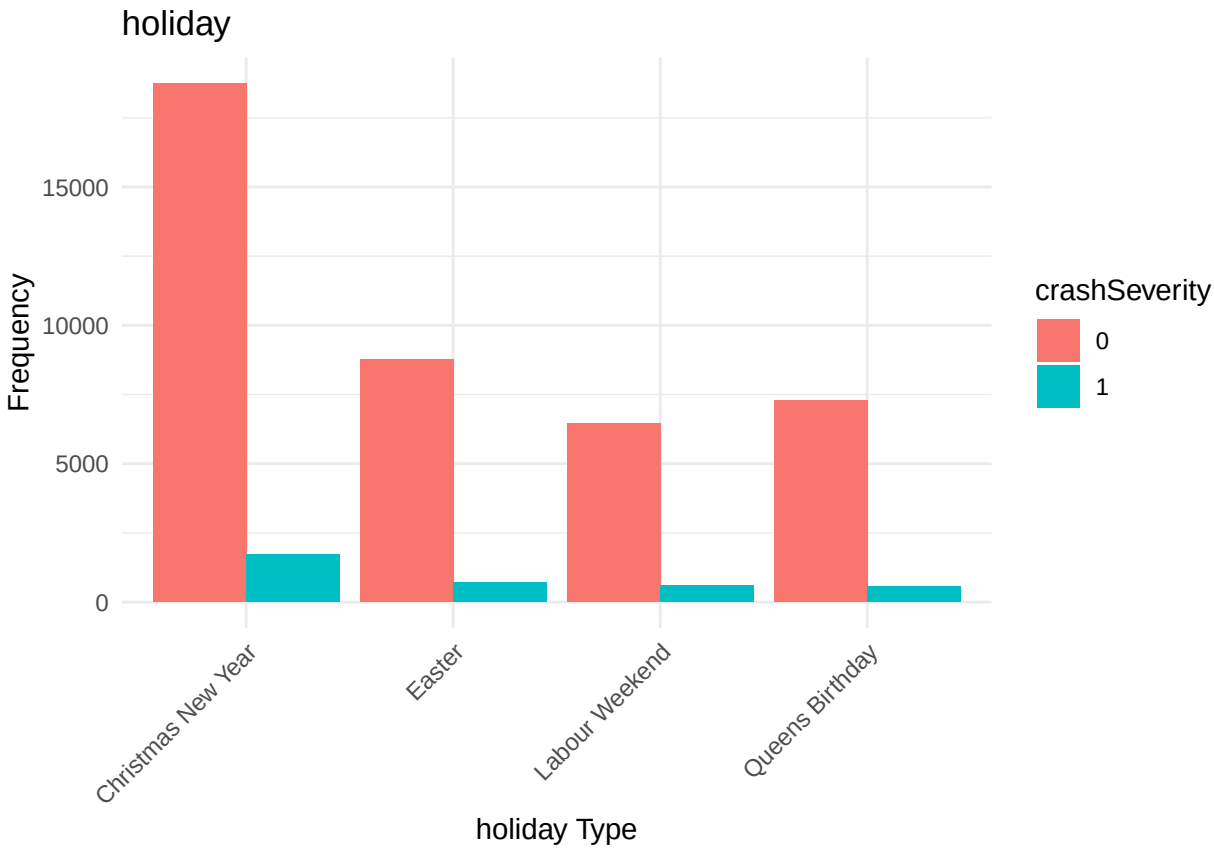
```
## `.groups` argument.
```




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




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


holiday
15000
10000
5000
0
Frequency
Christmas New Year
Easter
Labour Weekend
Queens Birthday
holiday Type
crashSeverity
0
1