Uncovering the Key Factors Behind Toronto's TTC Bus Delays in 2022 and Proposing Improvements: A Data Analysis

SHAOHAN CHANG

01/26/2023

Abstract

This study analyzed data on bus usage and delays to determine the busiest days and directions for buses. The results show that Friday is the busiest day, with most buses traveling north, and the average delay time is approximately 10 minutes. The majority of delays are caused by operational and mechanical issues, and the highest number of delays occur on Wednesdays and Thursdays. The findings suggest that improving the efficiency of the bus service, particularly by addressing operational and mechanical problems and focusing on service quality on Wednesdays and northbound buses, could significantly improve the overall quality of the bus service for riders.

Introduction

The Toronto TTC Bus Delay Data Analysis is a project aimed at understanding the causes and patterns of bus delays in the City of Toronto in 2022. With the increasing population and traffic congestion in the city, the public transportation system, specifically the bus service, plays a crucial role in ensuring the mobility of citizens. However, delays in bus services can lead to frustration for riders and can ultimately affect the overall efficiency of the city's transportation system.

The project will use data from the Toronto Transit Commission (TTC) Gelfand (2022), which is responsible for the city's public transportation, to analyze the causes and patterns of bus delays. RStudio will be used to analyze the data and find practical solutions to improve the bus service. The goal is to identify the leading causes of bus delays and provide recommendations on how to address them.

The analysis of bus delay data in Toronto is important because it will help the TTC to improve the bus service, making it more efficient and reliable for riders. Additionally, this study fills a gap in the current research by providing a detailed analysis of bus delays in Toronto in 2022, which can be used as a reference for other cities facing similar challenges. The structure of the paper is divided into several sections, including the introduction, methodology, results, and conclusion.

Data Resource

In this data analysis project, I used data from opendatatoronto Gelfand (2022) . In the process of analyzing the data I used some software tools from the RStudio. Here is the RStudio tools that I used in this analysis project, I used tidyverse Wickham et al. (2019) , stringr Wickham (2022), skimr Waring et al. (2022), visdat Tierney (2017), janitor Firke (2021), lubridate Grolemund and Wickham (2011), ggrepel Slowikowski (2022), dplyr Wickham et al. (2022).,bibtex Francois and Hernangómez (2023), and knitr Xie (2014) . This data set is a subset of the bus delay data collected by the Toronto Transit Commission (TTC) in 2022 Gelfand (2022). It has been specifically selected for the purpose of this study, which aims to understand the causes and patterns of bus delays in the city. The data set includes information such as the date and time of the delay, the location of the bus, and the cause of the delay. However, the variable "Vehicle" has been removed from the data set, as it is not relevant to the research question and objectives of this study. The decision to exclude this variable was made to simplify the data analysis and to focus on the key variables that contribute to bus delays. Additionally, the study does not intend to consider the variable 'Vehicle' for the research process.

```
library(tinytex)
library(opendatatoronto)
library(tidyverse)
library(stringr)
library(skimr)
library(visdat)
library(janitor)
library(lubridate)
library(ggrepel)
library(dplyr)
library(bibtex)
library(knitr)

package <- list_package_resources("e271cdae-8788-4980-96ce-6a5c95bc6618")
package <- package %>% mutate(year = str_extract(name, "202.?"))
delay_2022_ids <- package %>% filter(year==2022) %>% select(id) %>% pull()
```

```
delay_2022 <- get_resource(delay_2022_ids)
delay_2022 <- delay_2022 |> select(-Vehicle)
delay_2022
```

```
# A tibble: 58,707 x 9
   Date                Route Time  Day   Locat~1 Incid~2 Min D~3 Min G~4 Direc~5
   <dttm>              <dbl> <chr> <chr> <chr>   <chr>     <dbl>   <dbl> <chr>
 1 2022-01-01 00:00:00   320 02:00 Satu~ YONGE ~ Genera~       0       0 <NA>
 2 2022-01-01 00:00:00   325 02:00 Satu~ OVERLE~ Divers~     131     161 W
 3 2022-01-01 00:00:00   320 02:00 Satu~ YONGE ~ Operat~      17      20 S
 4 2022-01-01 00:00:00   320 02:07 Satu~ YONGE ~ Operat~       4      11 S
 5 2022-01-01 00:00:00   320 02:13 Satu~ YONGE ~ Operat~       4       8 S
 6 2022-01-01 00:00:00   363 02:16 Satu~ KING A~ Operat~      30      60 <NA>
 7 2022-01-01 00:00:00    96 02:18 Satu~ HUMBER~ Securi~       0       0 N
 8 2022-01-01 00:00:00   320 02:38 Satu~ STEELE~ Operat~       4       8 <NA>
 9 2022-01-01 00:00:00   320 02:55 Satu~ YONGE ~ Operat~       4       8 <NA>
10 2022-01-01 00:00:00   300 03:18 Satu~ KENNED~ Emerge~       0       0 E
# ... with 58,697 more rows, and abbreviated variable names 1: Location,
#   2: Incident, 3: `Min Delay`, 4: `Min Gap`, 5: Direction
```

Explanation of Variables

Shows in the table 1(First 5 rows), here is the data on TTC bus delay that downloads from opendatatoronto Gelfand (2022); the dataset are included some of the essential information about the TTC bus information, for example, the information of Date, Route, bus location, and the time minutes delay and minutes.The dataset that is used in this analysis is a compilation of information on TTC bus delays in Toronto, and it is obtained from the opendatatoronto website. This dataset contains a wealth of information that is essential to understanding the causes and impacts of bus delays in the city.

- Date : The date recorded in the dataset is between January 1, 2022 and November 30, 2022. This time frame was chosen to provide a comprehensive analysis of the bus delays that occurred during the year 2022.
- Route : The dataset includes information on various routes that buses take throughout the city of Toronto. Each route is identified by a unique route number, which allows for the analysis of bus delays by route.This information is important in understanding which routes are most affected by delays and what are the possible causes for it. For example, some routes may be more prone to delays due to heavy traffic, construction, or other external factors.

- Day : Representing the data by typical weekdays and weekend days can provide a deeper understanding of how bus delays are distributed throughout the week.
- Location : The dataset also includes information on the location of the bus at the time of the delay. This information is important in understanding where delays are occurring within the city of Toronto. By representing the location of the bus, it is possible to identify specific areas of the city that are most affected by bus delays and to develop targeted solutions to address these issues.
- Min Delay : The dataset also includes information on the duration of the delay in minutes. This information is important in understanding how long bus delays are lasting and to identify patterns and trends in the data. The "Min Delay" column in the dataset, represent the minutes of delay for each bus, it helps to understand the severity of the delay, whether it's a minor or major delay.
- Min Gap: The dataset also includes information on the time gap between consecutive buses arriving at a station, measured in minutes. This information is referred to as "Min Gap" in the dataset. It's an important aspect to measure the regularity and punctuality of the bus service. Min Gap can be used to understand if the buses are running on schedule or if they are experiencing delays.
- Direction: The dataset also includes information on the direction in which the bus is traveling, represented as "Direction" in the dataset. This information is important in understanding the movement of buses throughout the city of Toronto and to identify patterns and trends in the data. For example, if a large number of delays are occurring in a specific direction, such as North or East, the Toronto Transit Commission (TTC) could investigate if there are any specific factors that are contributing to the delays in that direction.

```
library(knitr)
kable(head(delay_2022,5), caption = "Table 1: Delay_2022 Data (First 5 rows)")
```

Table 1: Table 1: Delay_2022 Data (First 5 rows)

Date	Route	Time	Day	Location	Incident	Min Delay	Min Gap	Direction
2022-01-01	320	02:00	Saturday	YONGE AND DUNDAS	General Delay	0	0	NA
2022-01-01	325	02:00	Saturday	OVERLEA AND THORCLIFFE	Diversion	131	161	W
2022-01-01	320	02:00	Saturday	YONGE AND STEELES	Operations - Operator	17	20	S

Date	Route	Time	Day	Location	Incident	Min Delay	Min Gap	Direction
2022-01-01	320	02:07	Saturday	YONGE AND STEELES	Operations - Operator	4	11	S
2022-01-01	320	02:13	Saturday	YONGE AND STEELES	Operations - Operator	4	8	S

Variables Test

The "Direction" variable in the dataset sometimes contains unrelated symbols such as "B", "5", "NA" which do not provide any meaningful information about the direction of the bus. These symbols are considered as missing or null values in the dataset and need to be cleaned up during the cleaning phase of data analysis. This is an important step in the data cleaning process as it ensures the quality and accuracy of the data that is used for analysis. If the direction variable contains unrelated symbols, it will lead to inaccurate results and conclusions.

```
unique(delay_2022$Direction)
```

```
 [1] NA   "W"  "S"  "N"  "E"  "/"  "J"  "B"  "D"  "Q"  "I"  "2"  "6"  "3"  "\\"
[16] "T"  "M"  "`"  "8"  "5"
```

Day variable data is fine.

```
unique(delay_2022$Day)
```

```
[1] "Saturday"  "Sunday"    "Monday"    "Tuesday"   "Wednesday" "Thursday"
[7] "Friday"
```

Data Cleaning

There are some NA information in this data. For example, Direction has NA data. In this case, the data will affect the analysis results, such as calculating the wrong number of Directions. So during the process of data cleaning, I deleted the NA data in order to improve the accuracy of the analyze.

```
delay_2022 = na.omit(delay_2022)
```

I found that there is an unsatisfactory direction symbol in the data of Direction of this data, such as $/,2,1$. I deleted the unsatisfactory direction data and used the correct direction information representation, such as (W, E, S, N).

```
delay_2022 = delay_2022 |> filter(Direction %in% c('W','E','S','N') )
```

Variables Distribution

This code represents the time range included in the data, where the recorded of the data were from 2022-01-01 to 2022-11-30.

```
range(delay_2022$Date)
```

```
[1] "2022-01-01 UTC" "2022-12-31 UTC"
```

Specifically, the measurement start to end of each day is obtained, from '00:00' to '23:59'.

```
range(delay_2022$Time)
```

```
[1] "00:00" "23:59"
```

Record the number of unique Routes from TTC buses in this database.

```
length(unique(delay_2022$Route))
```

```
[1] 238
```

In the figure 1 , this plot shows the bus server volume from Monday to Sunday. From the plot, the bus trip volume on Friday is the most among the seven days of the week. On the other hand, the bus trip volume on Sunday is the smallest. Among them, the bus trip volumes on Tuesday, Wednesday, and Thursday are similar.

```
delay_2022%>%
  ggplot(aes(x = Day,fill = Day)) +
  geom_histogram(stat = "count")+
  coord_flip() +
   labs(caption = "Figure 1",
       x = "Day of the Week",
```

```
      y = "Number of Delays") +
  theme(plot.caption = element_text(size = 13))
```

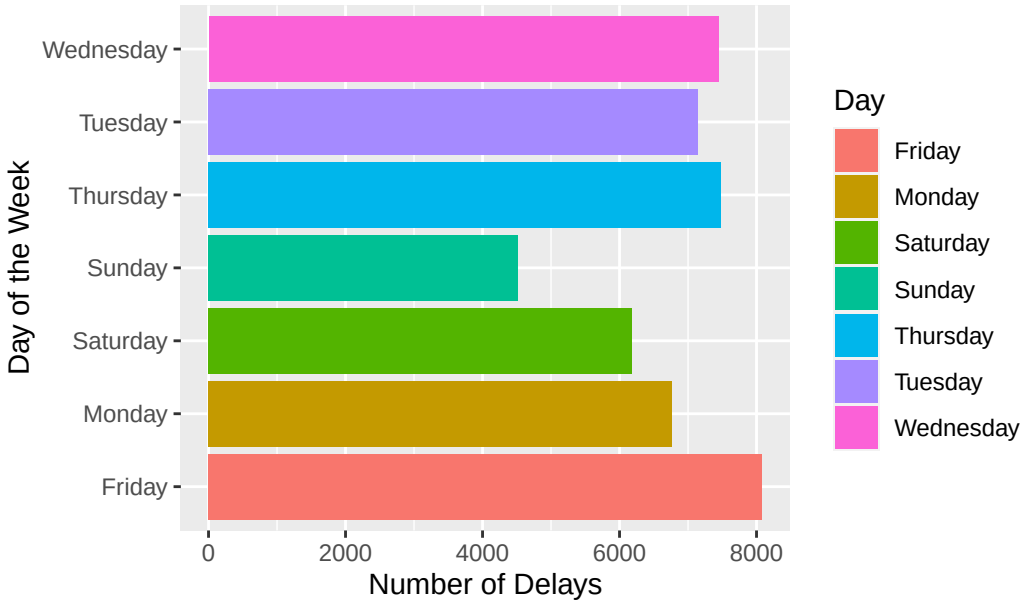

Figure 1

In the figure 2, the plot expresses the number of travel directions for different buses. In the data, the number of buses travelling in the north direction is the highest, and the amount is close to around 12,500.

```
delay_2022%>%
  ggplot(aes(x = Direction,fill = Direction)) +
  geom_histogram(stat = "count")+coord_flip() +
   labs(caption = "Figure 2",
       x = "Drictions",
       y = "Number of Delays") +
  theme(plot.caption = element_text(size = 13))
```

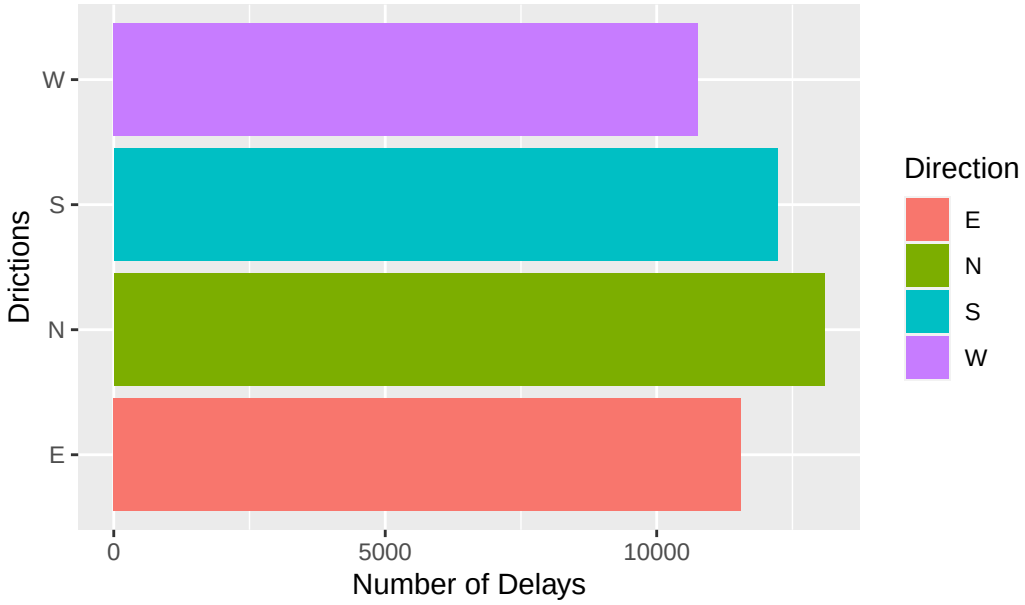

Figure 2

In the figure 3, present each bus delay incident. With more detail, the x-axis represents the number of occurrences in each incident type, and the y-axis represents each incident type. Through the correlation of such two data, it can be more clearly compared which bus delay incident type is the most so as to more intuitively find the main reason for the bus delay. Among them, the incidents of operations and mechanical occurred the most.

```
delay_2022%>%
  ggplot(aes(x = Incident,fill = Incident )) +
  geom_histogram(stat = "count")+
  coord_flip() +
   labs(caption = "Figure 3",
        x = "Type of incident",
        y = "Number of Delays") +
  theme(plot.caption = element_text(size = 14))
```

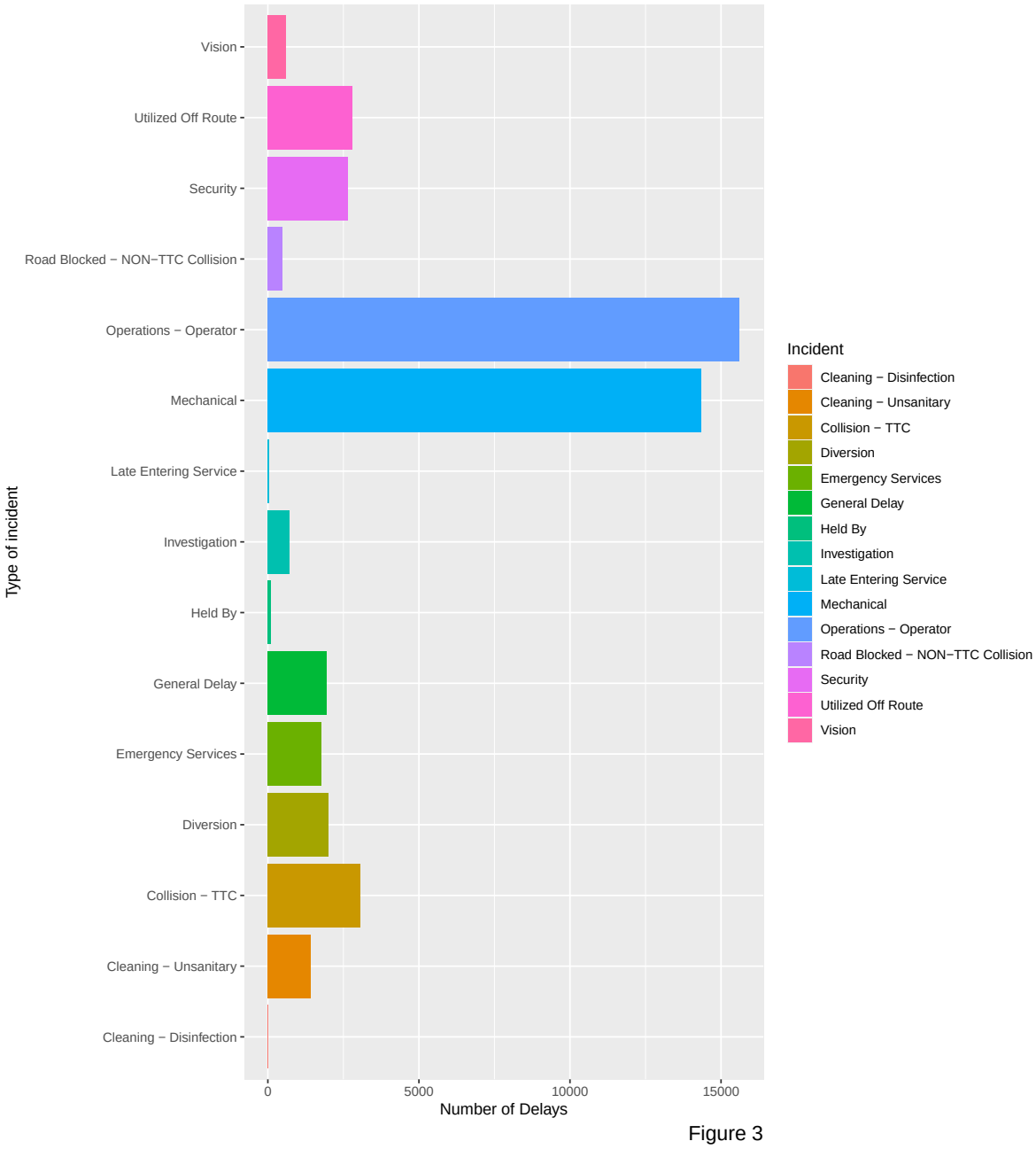

Figure 3

In the figure 4 and figure 5, as shows in the below in the following two plots show the respective densities of Min Delay and Min Gap of the variables. It can be clearly observed from such data expressions that the values of Min Delay and Min Gap density are the most. The **`geom_density`** function is then used to add a density plot to the plot, and the **`stat`** argument is set to "density" to indicate that the plot should show a density estimate Nicholas J. Horton

(December 18, 2020). Among the data about Min Delay, most of the bus delay time is about 10 minutes. Among the Min Gap data, most buses take around 30 minutes.

```
delay_2022%>%
  ggplot(aes(x = `Min Delay`)) +
  geom_density(stat = "density") +
  scale_x_log10()+
   labs(caption = "Figure 4",
        x = "Minutes Delay",
        y = "Density") +
  theme(plot.caption = element_text(size = 13))
```

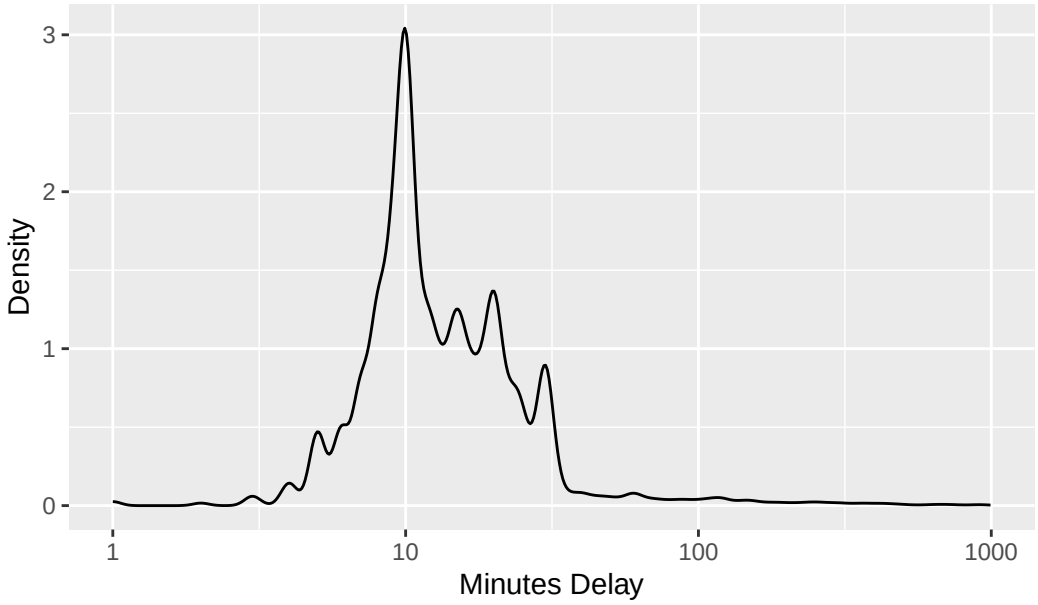

Figure 4

```
delay_2022%>%
  ggplot(aes(x = `Min Gap`)) +
  geom_density(stat = "density") +
  scale_x_log10() +
  labs(caption = "Figure 5",
       x = "Minutes Bus Server Gap",
       y = "Density") +
  theme(plot.caption = element_text(size = 13))
```

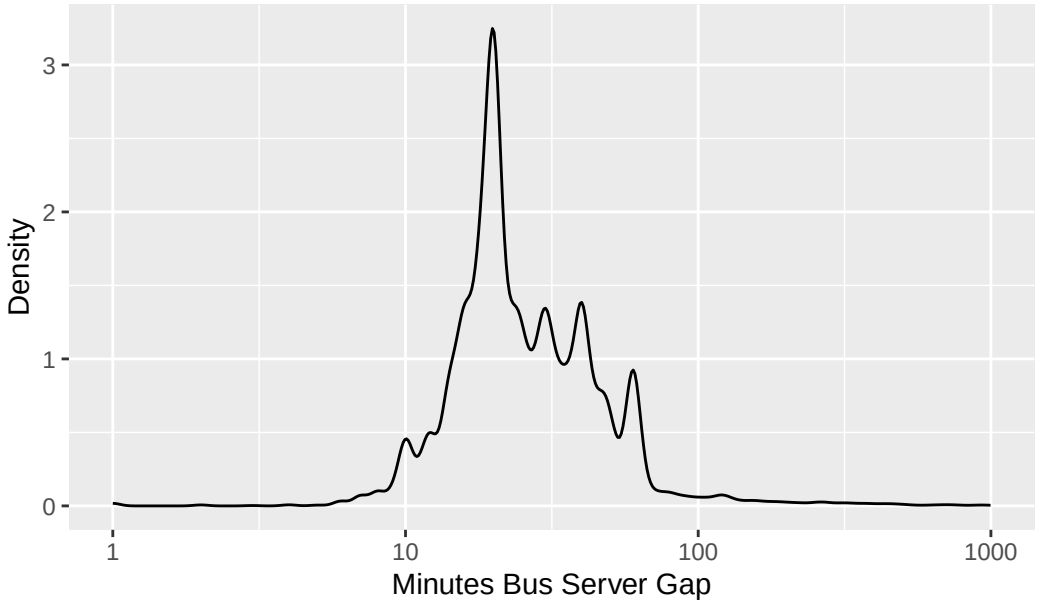

Figure 5

Relationships between the variables

In the figure 6, the plot below depicts the relationship between Min Delay and Date, with each dot representing an individual of independent data. In general, there is little difference between the flow of date and the min delay. As the plot result, most of the Min Delay are gathered within around 250 minutes, and a small number of Min Delays are above around 250 minutes.

```
delay_2022%>%
  ggplot(aes(Date, `Min Delay`)) +
  geom_point() +
  geom_hline(yintercept = 250,color = "red") +
  labs(caption = "Figure 6",
       x = "Minutes Delay",
       y = "Month") +
  theme(plot.caption = element_text(size = 13))
```

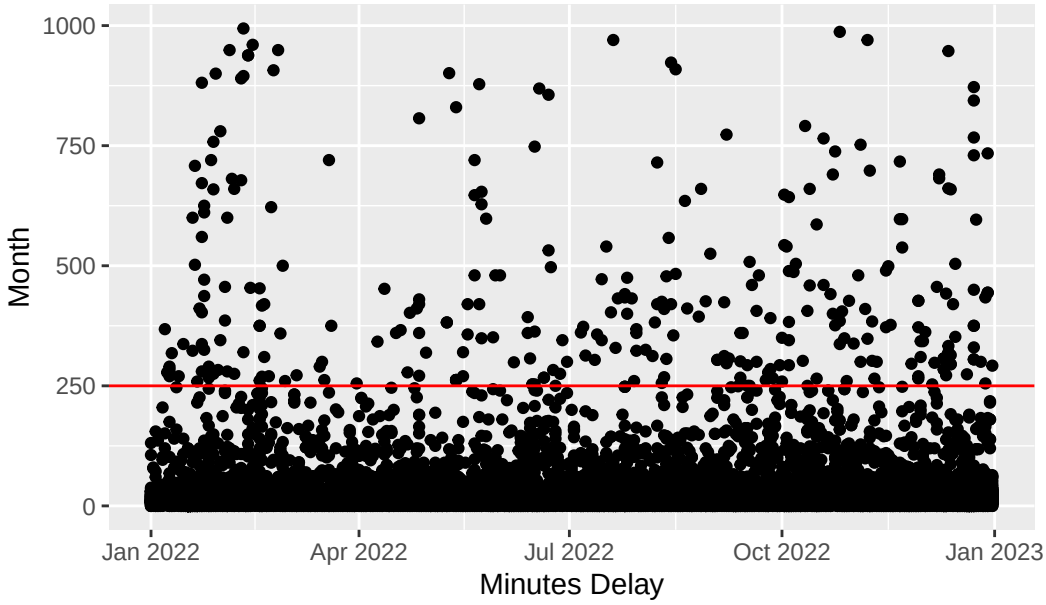

Figure 6

In the figure 7, the plot below depicts the relationship between Min Gap and Date, with each dot representing a separate variable. As a result, there are few significant differences between Date and Min Gap flow. Only a tiny portion of the Min Gap is around 250 minutes, with most of the Min Gap being gathered within this time frame.

```
delay_2022%>%
  ggplot(aes(Date, `Min Gap`)) +
  geom_point()+
  geom_hline(yintercept = 250,color = "red") +
  labs(caption = "Figure 7",
       x = "Minutes Delay",
       y = "Month") +
  theme(plot.caption = element_text(size = 13))
```

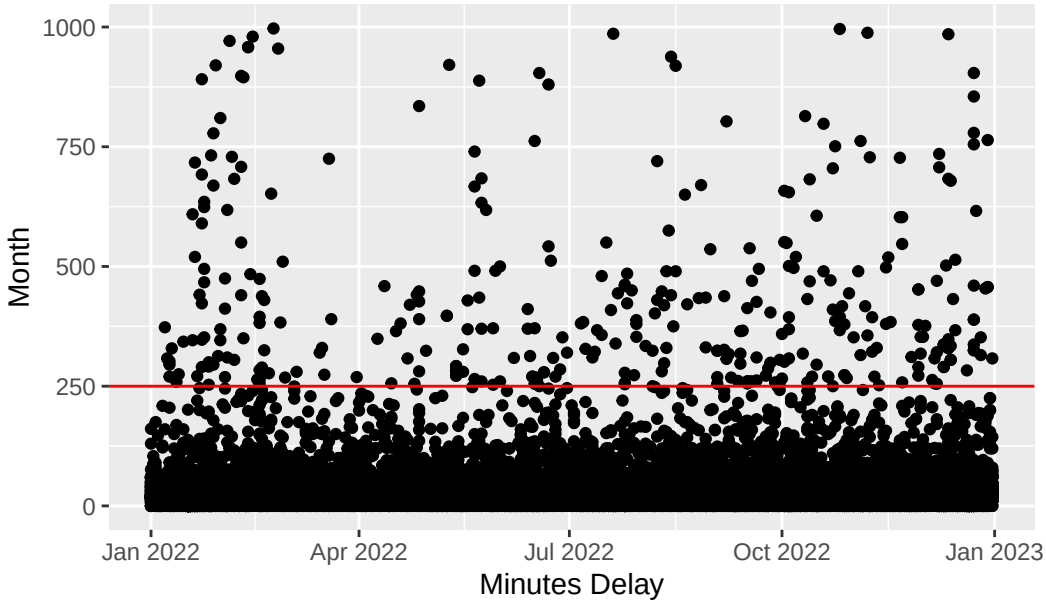

Figure 7

In the figure 8, I described the timeline in detail, and I will classify it into days (Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, and Sunday) to discuss Min Gap separately in order to get a plot of a relationship.But in a broad way, we don't see specific days of the week with the most serious delays. I set the relatively severe delay time at 400 minutes in this case.

In the figure 9, I counted how many delays were longer than 400 minutes from Monday to Sunday. This made the plot easier to see and judge. From a simple look at the numbers, Wednesday has the most delays that last more than 400 minutes.

```
delay_2022%>%
  ggplot(aes(Day,`Min Delay`)) +
  geom_point() +
  geom_hline(yintercept = 400,color = "red") +
  labs(caption = "Figure 8",
       x = "Week of the Day",
       y = "Minute") +
  theme(plot.caption = element_text(size = 13))
```

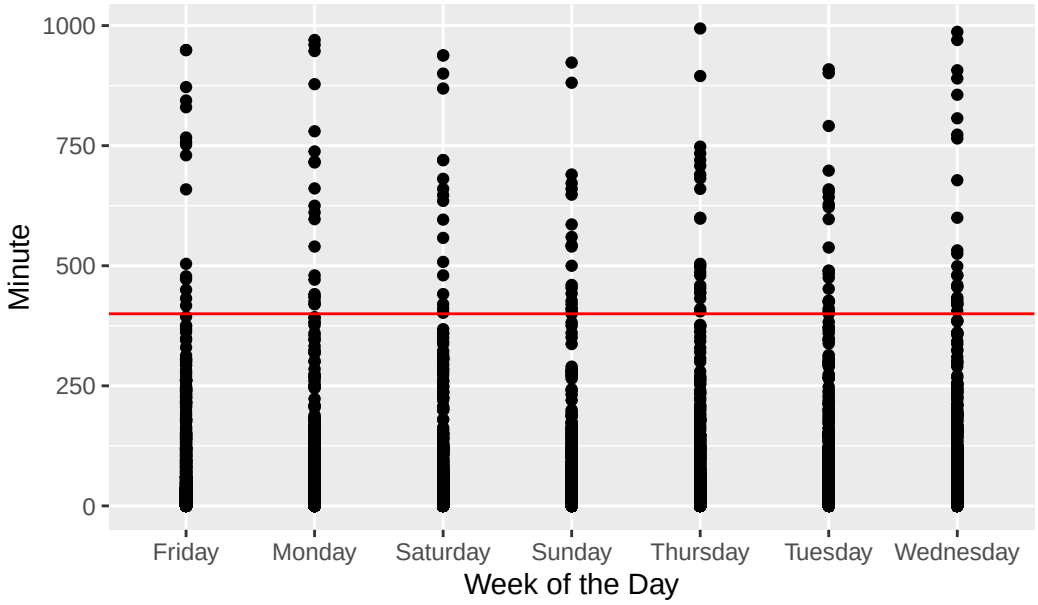

Figure 8

```
over_count_delay_day <- delay_2022 |>
  filter(`Min Delay` > 400)

over_count_delay_day %>%
  ggplot(aes(x = Day,fill = Day)) +
  geom_histogram(stat = "count" )+
  coord_flip()+
  labs(caption = "Figure 9",
       x = "Week of the Day",
       y = "Minute of Delay") +
  theme(plot.caption = element_text(size = 13))
```

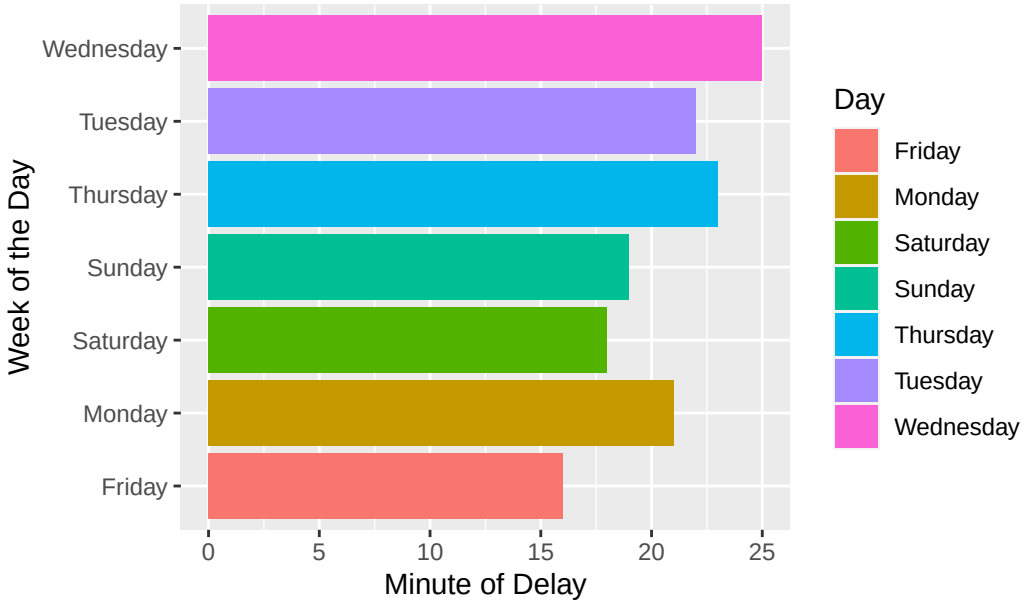

Figure 9

In the figure 10, which is to express the relationship between delay time and incident. Use geom_point() to create a scatter-plot of the variables "Incident" and "Min Delay", then use coord_flip() to flip the coordinates of the plot, and finally draw a horizontal red line at the y-intercept of 300. Using geom_hline(), the 300-minute data represents The dividing line for relatively severe bus delays. By observing the situation where the bus delay is the most in the case of Incident's Diversion.

The figure 11, shows the incidents of the number of delays exceeding 300 minutes in different situations. In this way, it is more intuitive to see what incidents often cause relatively serious bus delays. In this plot, geom_histogram() is used to create a histogram of the variable "Incident", which also uses coord_flip() to flip the coordinates. The fill parameter in the aes() function is used to fill the bars of the histogram with different colours depending on the value of the "Incident" variable. This plot is based on the filtered data frame "over_count_delay_incident", created from "delay_2022", filtered to include only incidents with a "Min Delay" greater than 300. We can see from the icon that the diversion of the incident is the cause of the bus exceeding 300 minutes at the highest.

```
delay_2022%>%
  ggplot(aes(Incident,`Min Delay`)) +
  geom_point()+
  coord_flip()+
  geom_hline(yintercept = 300,color = "red") +
```

```
  labs(caption = "Figure 10",
       x = "Type of incident",
       y = "Minute of Delay") +
  theme(plot.caption = element_text(size = 13))
```

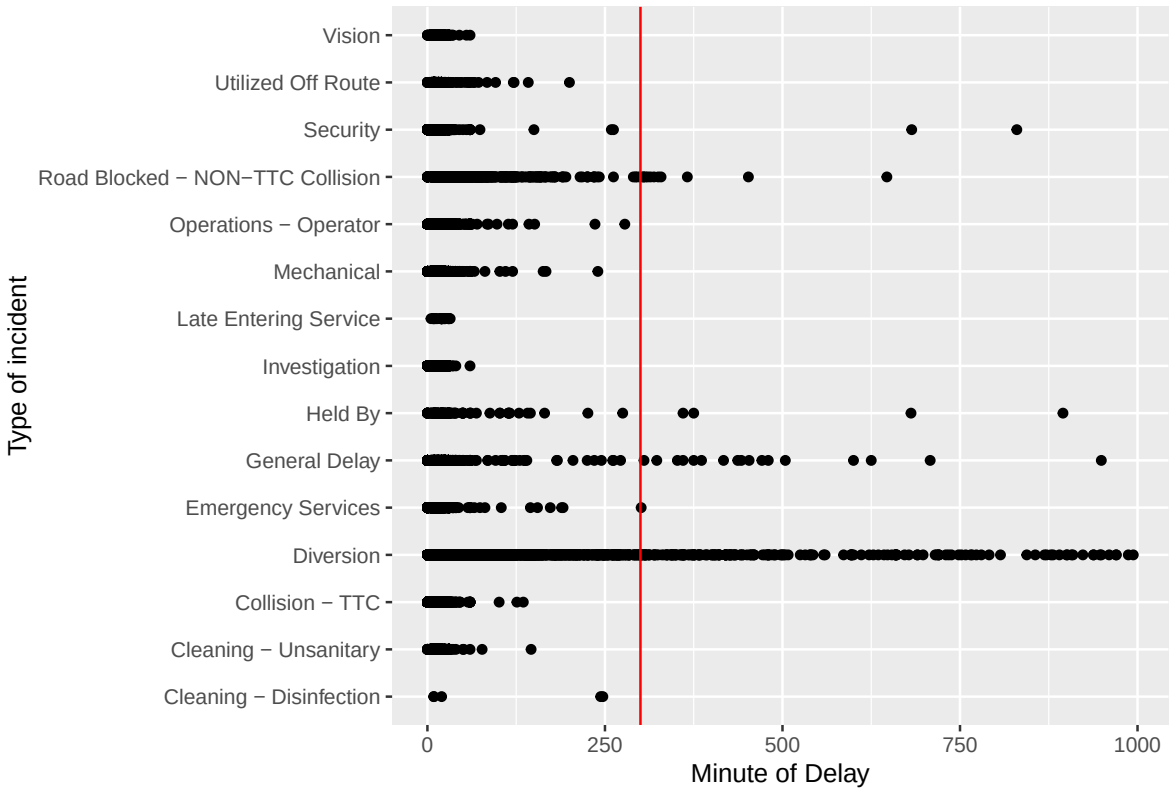

Figure 10

```
over_count_delay_incident <- delay_2022 |>
  filter(`Min Delay` > 300)

over_count_delay_incident%>%
  ggplot(aes(x = Incident,fill = Incident)) +
  geom_histogram(stat = "count" )+
  coord_flip() +
   labs(caption = "Figure 11",
       x = "Type of incident",
       y = "Count of Delay") +
  theme(plot.caption = element_text(size = 13))
```

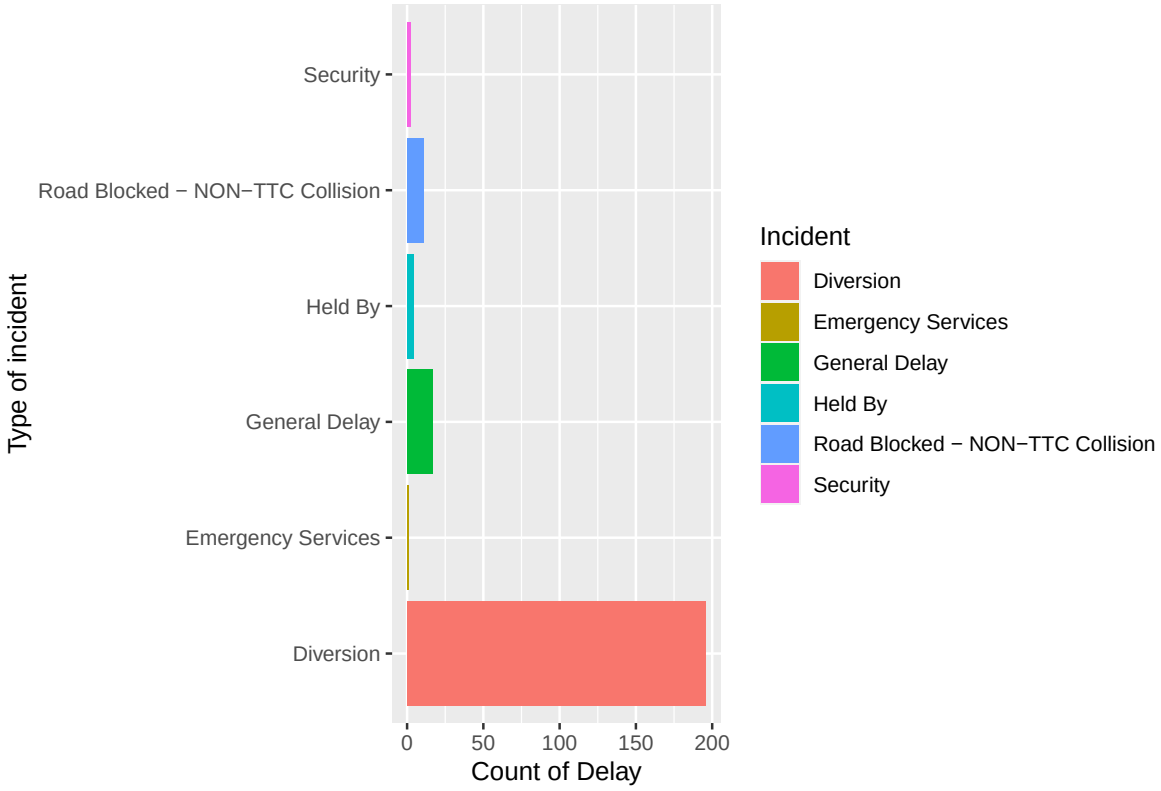

Figure 11

The figure 12 shows the travel time delays for different bus travel directions. Used geom_point() to create a scatter plot of the variables "Direction" and "Min Delay", then used coord_flip() to flip the coordinates of the plot, and finally drew a horizontal red line at the y-intercept of 500 Using geom_hline(), I set the 500 minutes for relatively serious bus time delays.

The figure 13 shows the time delay for different bus travel directions exceeding the 500-minute delay. The histogram of the variable "Direction" is created using geom_histogram(), which also flips the coordinates using coord_flip(). The fill parameter in the aes() function is used to fill the bars of the histogram with different colors depending on the value of the "direction" variable. The plot is based on the filtered data frame "over_count_delay_direction". We can see that the relative time delay of the bus travelling north is the most.

```
delay_2022%>%
  ggplot(aes(Direction,`Min Delay`)) +
  geom_point()+
  coord_flip()+
```

```
  geom_hline(yintercept = 500,color = "red") +
  labs(caption = "Figure 12",
       x = "Type of Direction",
       y = "Minute of Delay") +
  theme(plot.caption = element_text(size = 13))
```

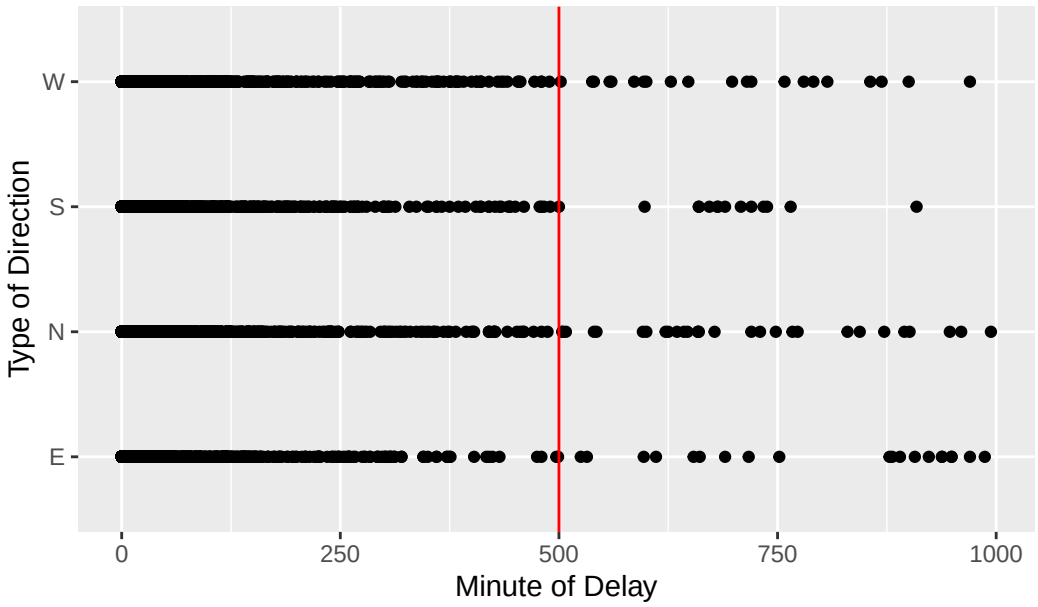

Figure 12

```
over_count_delay_direction <- delay_2022 |>
  filter(`Min Delay` > 500)

over_count_delay_direction %>%
  ggplot(aes(x = Direction ,fill = Direction)) +
  geom_histogram(stat = "count" )+
  coord_flip() +
  labs(caption = "Figure 13",
       x = "Type of Direction",
       y = "Count of Delay") +
  theme(plot.caption = element_text(size = 13))
```

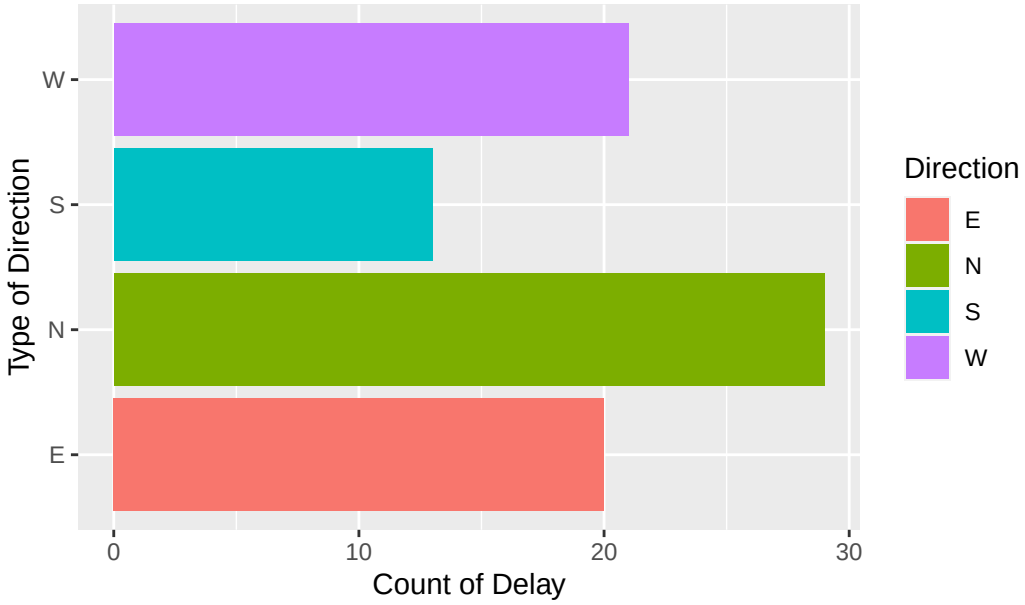

Figure 13

Conclusion

By the above data analysis and Plots, the busiest day of the week for buses is Friday. In addition, most vehicles are in the direction of the bus travelling north.

This statistical analysis can help to improve the efficiency of the TTC bus service. First of all, in terms of overall direction, the general TTC bus delay time is about 10 minutes. Wednesday and Thursday have the most delays among the weekly days. Most of the time, orations and mechanical problems cause TTC buses to be late.

To summarize, TTC BUS should first avoid operations and mechanical problems. Second, more emphasis should be placed on the quality of bus service on Wednesdays because the number of bus trips is the smallest of the week while the number of TTC bus delays is the highest. Third, more attention should be paid to northbound TTC buses; even though the number of northbound buses is the largest, the number of delays is also the highest.

References

Firke, Sam. 2021. *Janitor: Simple Tools for Examining and Cleaning Dirty Data*. https://CRAN.R-project.org/package=janitor.

Francois, Romain, and Diego Hernangómez. 2023. *Bibtex: Bibtex Parser*. https://CRAN.R-project.org/package=bibtex.

Gelfand, Sharla. 2022. *Opendatatoronto: Access the City of Toronto Open Data Portal*. https://CRAN.R-project.org/package=opendatatoronto.

Grolemund, Garrett, and Hadley Wickham. 2011. "Dates and Times Made Easy with lubridate." *Journal of Statistical Software* 40 (3): 1–25. https://www.jstatsoft.org/v40/i03/.

Nicholas J. Horton, Ken Kleinman. December 18, 2020. *Using r and RStudio for Data Management, Statistical Analysis, and Graphics*. Chapman & Hall. chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://englianhu.files.wordpress.com/2016/01/using-r-and-rstudio-for-data-management-statistical-analysis-and-graphics-2nd-edit.pdf.

Slowikowski, Kamil. 2022. *Ggrepel: Automatically Position Non-Overlapping Text Labels with 'Ggplot2'*. https://CRAN.R-project.org/package=ggrepel.

Tierney, Nicholas. 2017. "Visdat: Visualising Whole Data Frames." *JOSS* 2 (16): 355. https://doi.org/10.21105/joss.00355.

Waring, Elin, Michael Quinn, Amelia McNamara, Eduardo Arino de la Rubia, Hao Zhu, and Shannon Ellis. 2022. *Skimr: Compact and Flexible Summaries of Data*. https://CRAN.R-project.org/package=skimr.

Wickham, Hadley. 2022. *Stringr: Simple, Consistent Wrappers for Common String Operations*. https://CRAN.R-project.org/package=stringr.

Wickham, Hadley, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D'Agostino McGowan, Romain François, Garrett Grolemund, et al. 2019. "Welcome to the tidyverse." *Journal of Open Source Software* 4 (43): 1686. https://doi.org/10.21105/joss.01686.

Wickham, Hadley, Romain François, Lionel Henry, and Kirill Müller. 2022. *Dplyr: A Grammar of Data Manipulation*. https://CRAN.R-project.org/package=dplyr.

Xie, Yihui. 2014. "Knitr: A General-Purpose Package for Dynamic Report Generation in r." *Journal of Statistical Software* 40 (1): 1–30.