CAPSTONE Project _TTC DELAY ANALYSIS AND PREDICTION

2024-02-17

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

#INITIAL ANALYSIS OF TTC DELAY DATASET:

```
# Load the TTC Delay data readxl package:
library(readxl)

# Read the Excel file
data2 <- read_excel("ttc-bus-delay-data-2021-23.xlsx")
head(data2)
```

```
## # A tibble: 6 x 11
##   Date                Mode of Transportati~1 Route Time  Day   Location Incident
##   <dttm>              <chr>                  <dbl> <chr> <chr> <chr>    <chr>
## 1 2023-01-01 00:00:00 Bus                       91 02:30 Sund~ WOODBIN~ Diversi~
## 2 2023-01-01 00:00:00 Bus                       69 02:34 Sund~ WARDEN ~ Security
## 3 2023-01-01 00:00:00 Bus                       35 03:06 Sund~ JANE ST~ Cleaning
## 4 2023-01-01 00:00:00 Bus                      900 03:14 Sund~ KIPLING~ Security
## 5 2023-01-01 00:00:00 Bus                       85 03:43 Sund~ MEADOWA~ Security
## 6 2023-01-01 00:00:00 Bus                       40 03:47 Sund~ KIPLING~ Emergen~
## # i abbreviated name: 1: `Mode of Transportation`
## # i 4 more variables: `Min Delay` <dbl>, `Min Gap` <dbl>, Direction <chr>,
## #   Vehicle <dbl>
```

```
str(data2)
```

```
## tibble [151,889 x 11] (S3: tbl_df/tbl/data.frame)
##  $ Date                  : POSIXct[1:151889], format: "2023-01-01" "2023-01-01" ...
##  $ Mode of Transportation: chr [1:151889] "Bus" "Bus" "Bus" "Bus" ...
##  $ Route                 : num [1:151889] 91 69 35 900 85 40 336 52 24 36 ...
##  $ Time                  : chr [1:151889] "02:30" "02:34" "03:06" "03:14" ...
##  $ Day                   : chr [1:151889] "Sunday" "Sunday" "Sunday" "Sunday" ...
##  $ Location              : chr [1:151889] "WOODBINE AND MORTIMER" "WARDEN STATION" "JANE STATION" "K
##  $ Incident              : chr [1:151889] "Diversion" "Security" "Cleaning" "Security" ...
##  $ Min Delay             : num [1:151889] 81 22 30 17 1 0 138 30 20 334 ...
##  $ Min Gap               : num [1:151889] 111 44 60 17 1 0 168 60 40 344 ...
##  $ Direction             : chr [1:151889] "W" "S" "N" "N" ...
##  $ Vehicle               : num [1:151889] 8772 8407 1051 3334 1559 ...
```

```
summary(data2)
```

```
##       Date                          Mode of Transportation     Route
##  Min.   :2021-01-01 00:00:00.00   Length:151889          Min.   :   1.0
##  1st Qu.:2021-12-09 00:00:00.00   Class :character       1st Qu.:  37.0
##  Median :2022-08-03 00:00:00.00   Mode  :character       Median :  72.0
##  Mean   :2022-07-21 21:51:46.49                         Mean   : 192.6
##  3rd Qu.:2023-04-06 00:00:00.00                         3rd Qu.: 122.0
##  Max.   :2023-11-30 00:00:00.00                         Max.   :1000.0
##                                                         NA's   :1152
##      Time               Day              Location           Incident
##  Length:151889      Length:151889      Length:151889      Length:151889
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##    Min Delay        Min Gap       Direction            Vehicle
##  Min.   :  0.0   Min.   :  0.0   Length:151889      Min.   :    0
##  1st Qu.:  9.0   1st Qu.: 17.0   Class :character   1st Qu.: 3110
##  Median : 11.0   Median : 22.0   Mode  :character   Median : 7261
##  Mean   : 19.9   Mean   : 32.5                      Mean   : 5467
##  3rd Qu.: 20.0   3rd Qu.: 38.0                      3rd Qu.: 8549
##  Max.   :999.0   Max.   :999.0                      Max.   :99035
##
```

#ASSIGNING CORRECT DATA TYPE:

```
data2$Day <- as.factor(data2$Day)
data2$Incident <- as.factor(data2$Incident)
data2$Direction <- as.factor(data2$Direction)
str(data2)
```

```
## tibble [151,889 x 11] (S3: tbl_df/tbl/data.frame)
##  $ Date                  : POSIXct[1:151889], format: "2023-01-01" "2023-01-01" ...
##  $ Mode of Transportation: chr [1:151889] "Bus" "Bus" "Bus" "Bus" ...
##  $ Route                 : num [1:151889] 91 69 35 900 85 40 336 52 24 36 ...
##  $ Time                  : chr [1:151889] "02:30" "02:34" "03:06" "03:14" ...
##  $ Day                   : Factor w/ 7 levels "Friday","Monday",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ Location              : chr [1:151889] "WOODBINE AND MORTIMER" "WARDEN STATION" "JANE STATION" "K
##  $ Incident              : Factor w/ 12 levels "Cleaning","Collision",..: 3 11 1 11 11 4 3 4 1 3 ...
##  $ Min Delay             : num [1:151889] 81 22 30 17 1 0 138 30 20 334 ...
##  $ Min Gap               : num [1:151889] 111 44 60 17 1 0 168 60 40 344 ...
##  $ Direction             : Factor w/ 5 levels "B","E","N","S",..: 5 4 3 3 3 5 3 2 5 5 ...
##  $ Vehicle               : num [1:151889] 8772 8407 1051 3334 1559 ...
```

#IDENTIFYING PRESENCE OF 'MISSING VALUES' IN THE TTC DATASET:

```
# Identifying Presence of Missing Data
missing_values <- colSums(is.na(data2))
```

```r
# Print the count of missing values for each column
print(missing_values)
```

```
##                    Date Mode of Transportation                   Route
##                       0                       0                    1152
##                    Time                     Day                Location
##                       0                       0                       0
##                Incident              Min Delay                 Min Gap
##                       0                       0                       0
##               Direction                 Vehicle
##                    3735                       0
```

#HANDLING MISSING VALUES:

```r
#Subsetting rows where route is not provided
data2 <- subset(data2, !is.na(Route))
library(DescTools)
```

```
## Warning: package 'DescTools' was built under R version 4.3.1
```

```r
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
```

```r
# Replacing Direction attibutes with NA values based on route and its corresponding mode of direction
data2_filled <- data2 %>%
  group_by(Route) %>%
  mutate(Direction = ifelse(is.na(Direction), Mode(as.integer(Direction), na.rm = TRUE), Direction)) %>%
  ungroup()
missing_values <- colSums(is.na(data2_filled))

# Print the count of missing values for each column
print(missing_values)
```

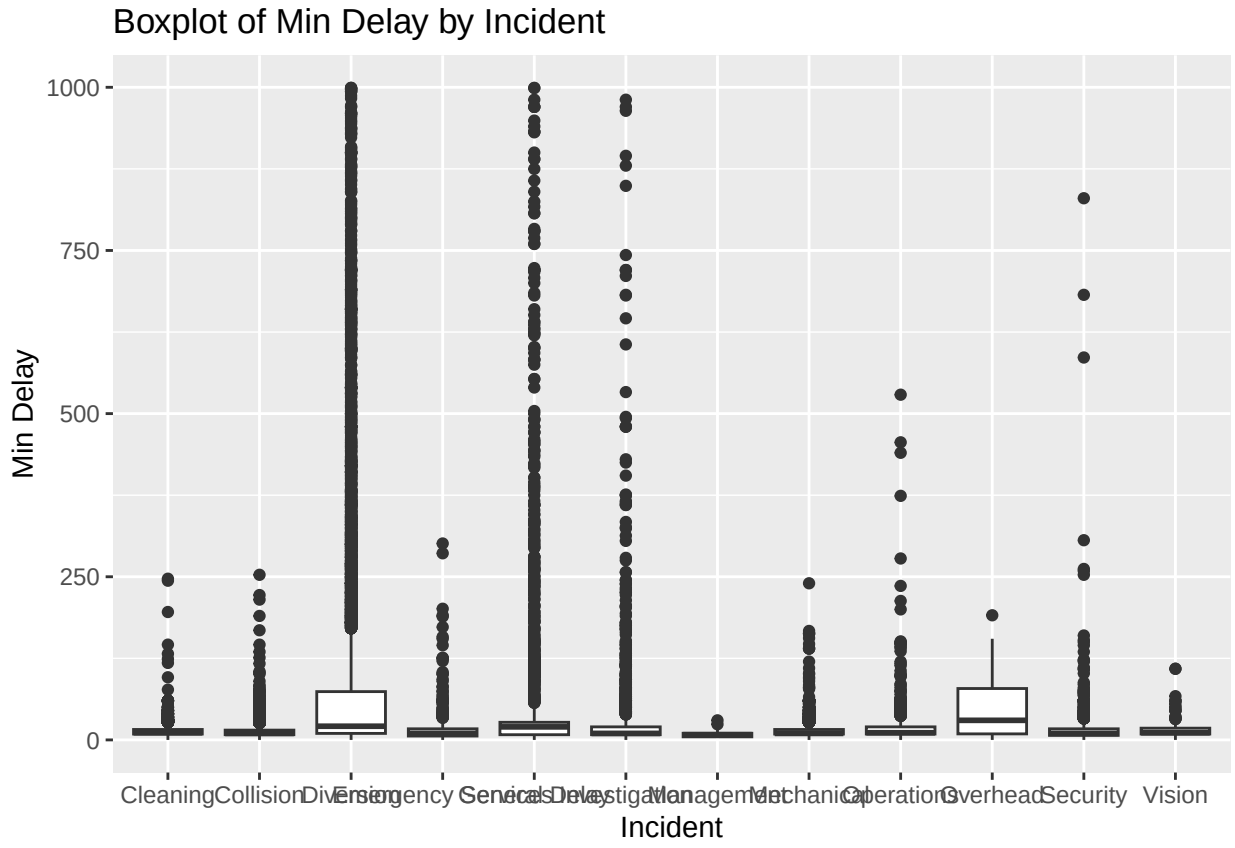
```
##                    Date Mode of Transportation                   Route
##                       0                       0                       0
##                    Time                     Day                Location
##                       0                       0                       0
##                Incident              Min Delay                 Min Gap
##                       0                       0                       0
##               Direction                 Vehicle
##                       0                       0
```

#IDENTIFYING PRESENCE OF OUTLIERS IN DEPENDENT VARIABLE 'MIN DELAY' ACROSS INCIDENT TYPES:

```
# Load the ggplot2 package
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.3.1
```

```
# OUTLIER ANALYSIS: Boxplot to clearly see the interquartile range
ggplot(data2_filled, aes(x = Incident, y = `Min Delay`)) +
  geom_boxplot(coef = 1.5) +  # Adjust the coef parameter to control the length of the whiskers
  labs(x = "Incident", y = "Min Delay", title = "Boxplot of Min Delay by Incident")
```



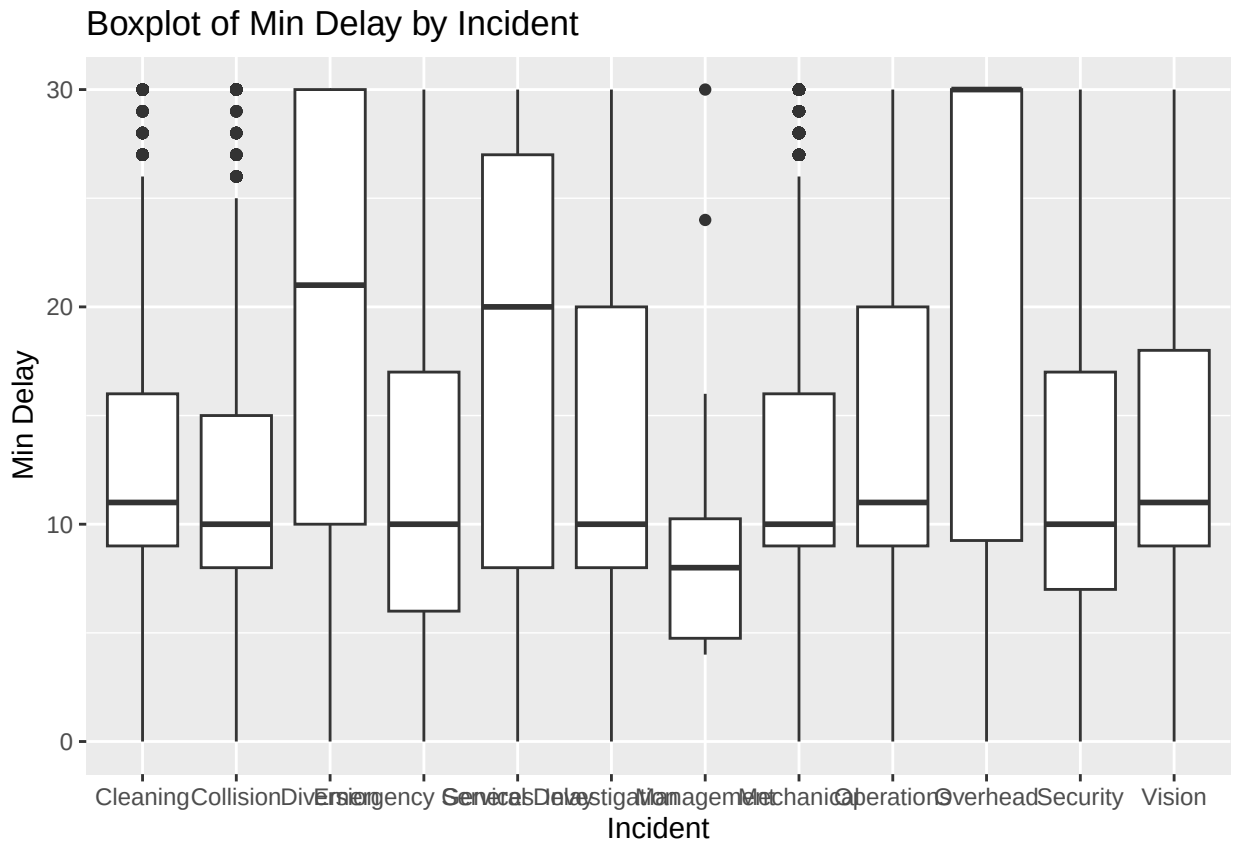
#PERFORMING 'WINSORIZATION' TO FIX OUTLIERS IN DEPENDENT VARIABLE 'MIN DELAY':

```
# Perform 'Winsorization' to Fix Outlier Issues: capping the outliers at a certain percentile. For exam
library(DescTools)

# Winsorize the 'Min Delay' column at the 2ND and 94th percentiles for the entire data
data2_filled$Min_Delay_Winsorized <- Winsorize(data2_filled$`Min Delay`, probs = c(0.02, 0.94))

# Check the results
summary(data2_filled$Min_Delay_Winsorized)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    9.00   11.00   13.91   20.00   30.00
```

BOXPLOT OF 'MIN DELAY' DATA ACROSS INCIDENT TYPES POST WINSORIZATION

```
library(ggplot2)
# OUTLIER ANALYSIS: Boxplot to clearly see the interquartile range
ggplot(data2_filled, aes(x = Incident, y = `Min_Delay_Winsorized`)) +
  geom_boxplot(coef = 1.5) +  # Adjust the coef parameter to control the length of the whiskers
  labs(x = "Incident", y = "Min Delay", title = "Boxplot of Min Delay by Incident")
```



#GROUPING 'MIN DELAY' DEPENDENT VARIABLE TO CHECK DATA IMBALANCE

```
# GROUPING 'MIN DELAY' DATA FOR FUTURE ANALYSIS

data2_filled_updated <- data2_filled %>%
  mutate(Delay_Severity = case_when(
    `Min_Delay_Winsorized` >= 0 & `Min_Delay_Winsorized` < 5 ~ "<5 Min",
    `Min_Delay_Winsorized` >= 5 & `Min_Delay_Winsorized` <= 10 ~ "5-10 Min",
    `Min_Delay_Winsorized` >10 & `Min_Delay_Winsorized` <= 15 ~ "11-15 Min",
    `Min_Delay_Winsorized` >15 & `Min_Delay_Winsorized` <= 20 ~ "16-20 Min",
    `Min_Delay_Winsorized` > 20 ~ ">20 Min",
    TRUE ~ "On Time"  # Handle cases where Min Delay is negative or other values
  ))
data2_filled_updated$Delay_Severity <- as.factor(data2_filled_updated$Delay_Severity)

# CHECK FOR DATA IMBALANCE
```

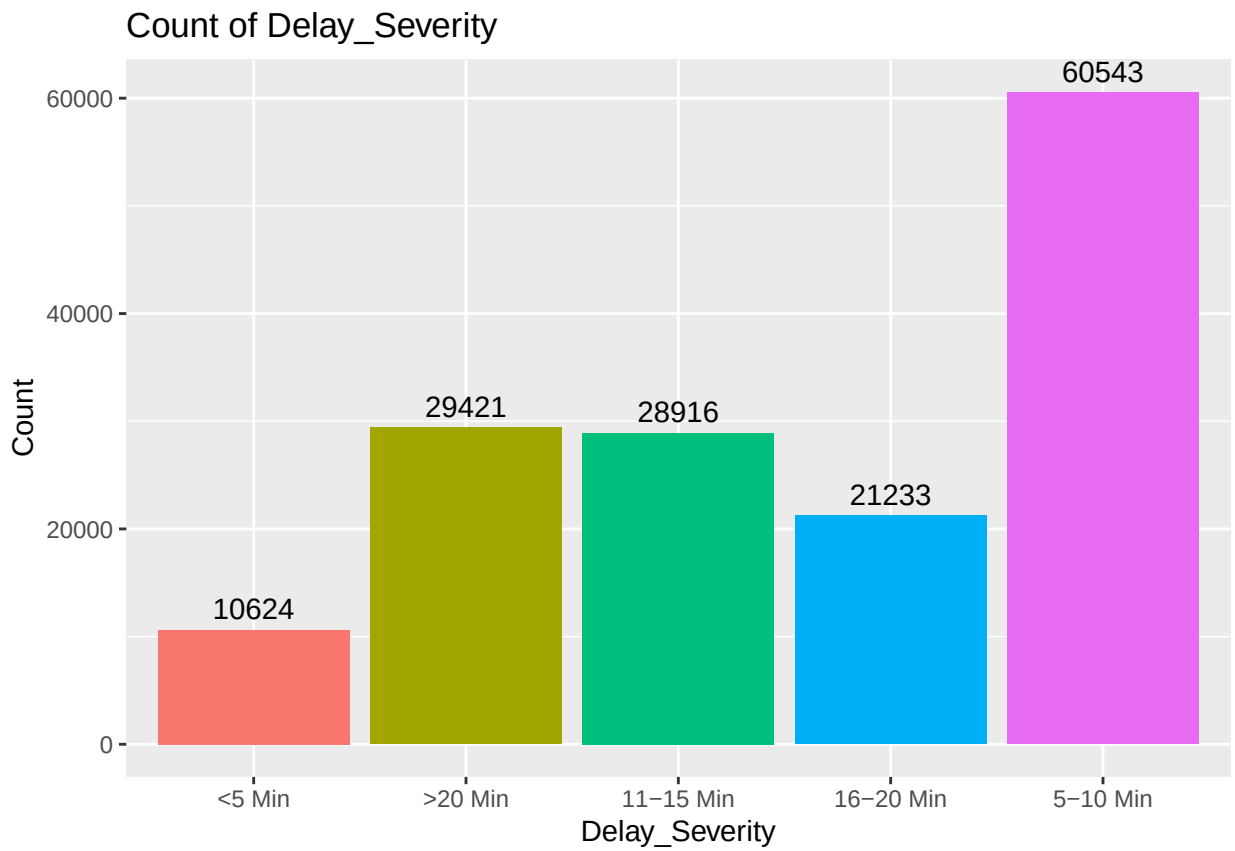
```r
library(dplyr)

# Count the frequency of each level in the 'Incident' column and calculate percentage
Delay_Severity_balance <- data2_filled_updated %>%
  count(Delay_Severity) %>%
  mutate(Percentage = n / sum(n) * 100) %>%
  arrange(desc(n))

# Print the counts and percentages to check for imbalance
print(Delay_Severity_balance)
```

```
## # A tibble: 5 x 3
##   Delay_Severity     n Percentage
##   <fct>          <int>      <dbl>
## 1 5-10 Min       60543      40.2
## 2 >20 Min        29421      19.5
## 3 11-15 Min      28916      19.2
## 4 16-20 Min      21233      14.1
## 5 <5 Min         10624       7.05
```

#BAR GRAPH OF 'DELAY_SEVERITY' DEPENDENT VARIABLE TO CHECK DATA IMBALANCE

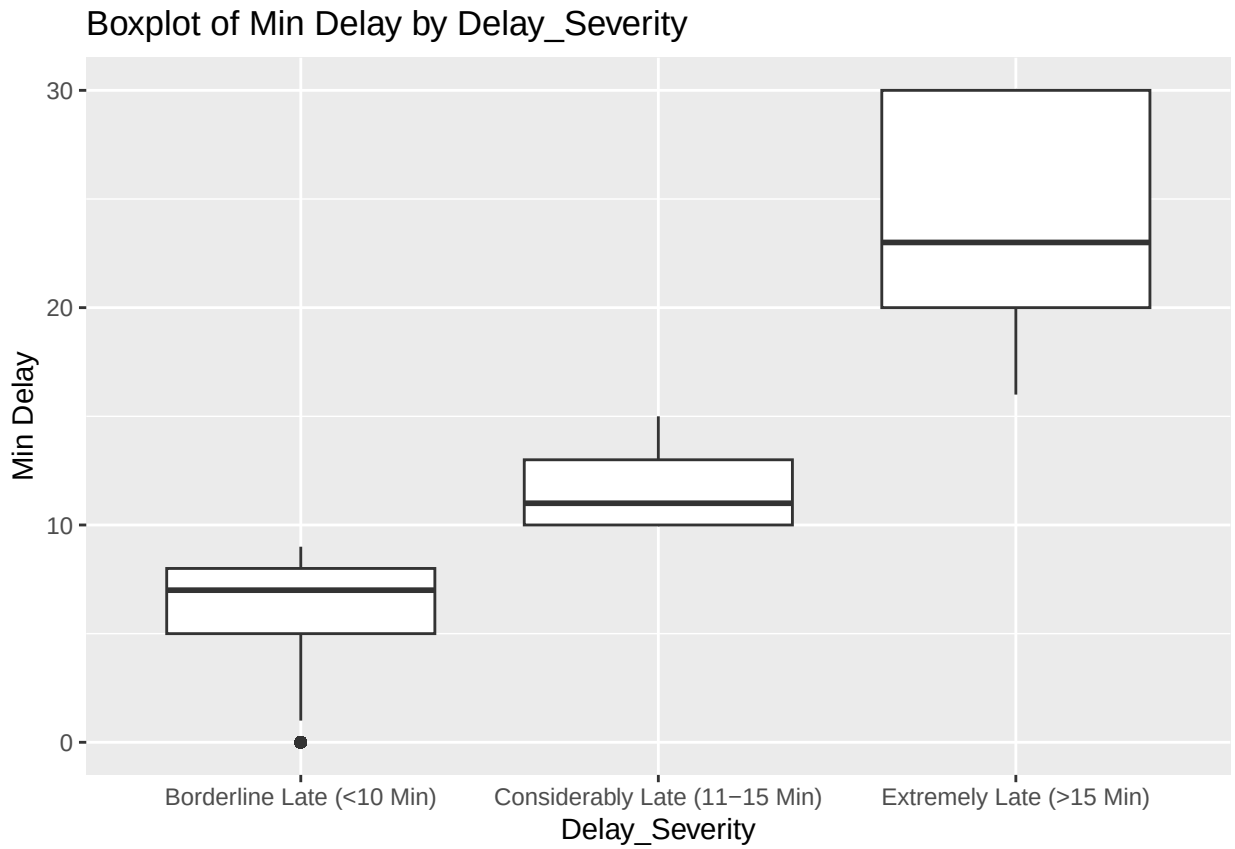
```r
#COUNT OF INCIDENTS
library(ggplot2)

# Create a bar graph with data labels and different bar colors
ggplot(data2_filled_updated, aes(x = Delay_Severity, fill = Delay_Severity)) +
  geom_bar() +
  geom_text(stat = 'count', aes(label = ..count..), vjust = -0.5) +  # Add data labels
  scale_fill_discrete(name = "Delay_Severity") +  # Customize legend title
  labs(x = "Delay_Severity", y = "Count", title = "Count of Delay_Severity") +
  theme(legend.position = "none")  # Hide legend (optional)
```

```
## Warning: The dot-dot notation (`..count..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(count)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



#REDUCING NUMBER OF CLASS/LEVELS WITHIN TARGET VARIABLE FROM 5 TO 3 TO ADDRESS DATA IMBALANCE:

```
#REGROUPING 'DELAY_SEVERITY' DATA FOR EASE OF CLASSIFICATION

data2_filled_updated1 <- data2_filled_updated %>%
  mutate(Delay_Severity = case_when(
    `Min_Delay_Winsorized` >= 0 & `Min_Delay_Winsorized` < 10 ~ "Borderline Late (<10 Min)",
    `Min_Delay_Winsorized` >=10 & `Min_Delay_Winsorized` <= 15 ~ "Considerably Late (11-15 Min)",
    `Min_Delay_Winsorized` > 15 ~ "Extremely Late (>15 Min)",
    TRUE ~ "On Time"  # Handle cases where Min Delay is negative or other values
  ))
data2_filled_updated1$Delay_Severity <- as.factor(data2_filled_updated1$Delay_Severity)

# Create a customized boxplot to clearly see the interquartile range
ggplot(data2_filled_updated1, aes(x = Delay_Severity, y = `Min_Delay_Winsorized`)) +
  geom_boxplot(coef = 1.5) +  # Adjust the coef parameter to control the length of the whiskers
  labs(x = "Delay_Severity", y = "Min Delay", title = "Boxplot of Min Delay by Delay_Severity")
```



#DISTRIBUTION OF DELAY INCIDENCE ACROSS 3 CLASSES OF 'DELAY_SEVERITY' TARGET VARIABLE:

```
# CHECK FOR DATA IMBALANCE
library(dplyr)

# Count the frequency of each level in the 'Incident' column and calculate percentage
Delay_Severity_balance1 <- data2_filled_updated1 %>%
  count(Delay_Severity) %>%
  mutate(Percentage = n / sum(n) * 100) %>%
  arrange(desc(n))

# Print the counts and percentages to check for imbalance
print(Delay_Severity_balance1)
```

```
## # A tibble: 3 x 3
##   Delay_Severity                    n Percentage
##   <fct>                         <int>      <dbl>
## 1 Considerably Late (11-15 Min) 53167       35.3
## 2 Extremely Late (>15 Min)      50654       33.6
## 3 Borderline Late (<10 Min)     46916       31.1
```

#ADDRESSING CATEGORICAL VARIABLES WITH EXCESSIVE LABELS:

#'INCIDENT' VARAIBELE WITH 12 LABELS

```r
# Load necessary library
library(dplyr)

# Count incidents for each type and calculate the percentage of the grand total
incident_type_counts <- data2_filled_updated1 %>%
  group_by(Incident) %>%
  summarise(IncidentCount = n(), .groups = 'drop') %>%
  mutate(TotalIncidents = sum(IncidentCount), # Calculate the total number of incidents
         PercentageOfTotal = IncidentCount / TotalIncidents * 100) %>%
  arrange(desc(IncidentCount)) # Arrange by descending order of incident counts

# Print the results
print(incident_type_counts)
```

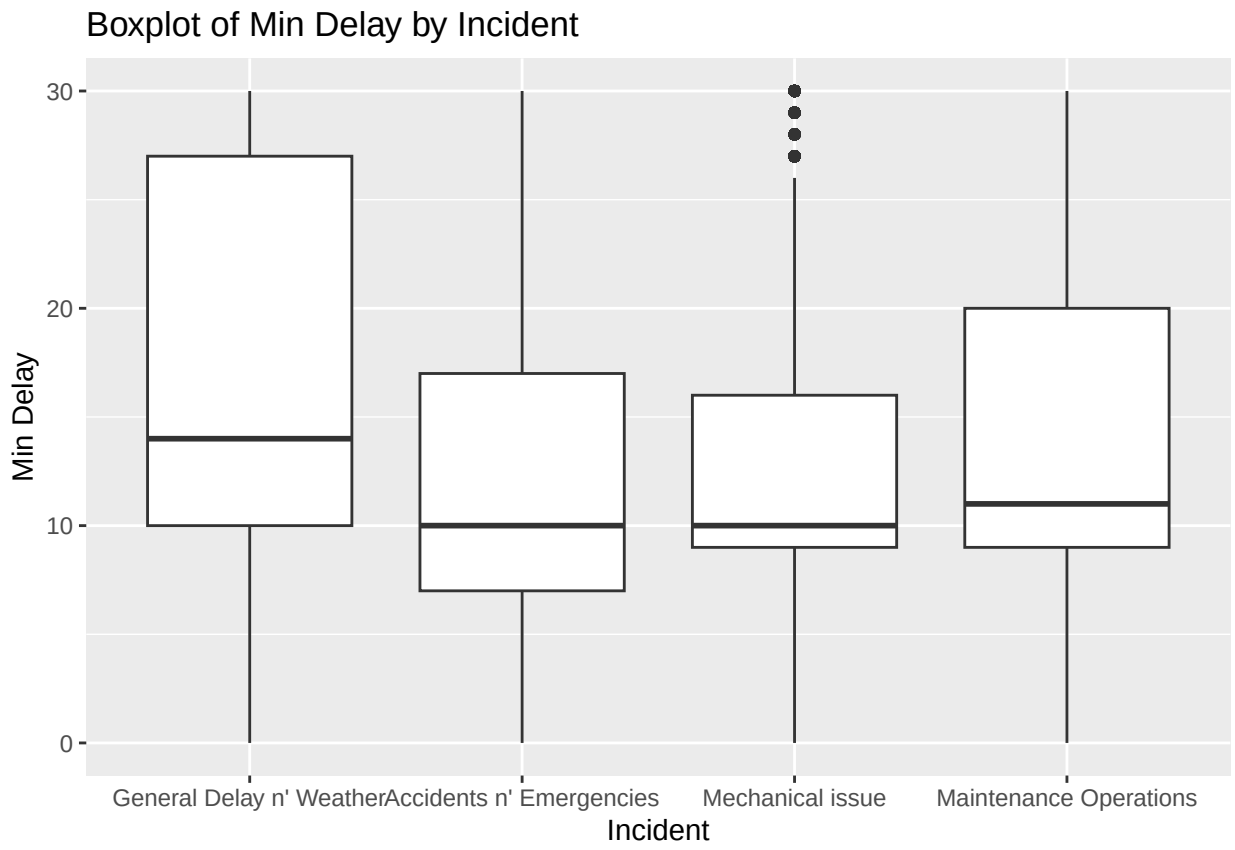
```
## # A tibble: 12 x 4
##    Incident           IncidentCount TotalIncidents PercentageOfTotal
##    <fct>                      <int>          <int>             <dbl>
##  1 Mechanical                 44174         150737           29.3
##  2 Operations                 39366         150737           26.1
##  3 Diversion                  14820         150737            9.83
##  4 Cleaning                   13061         150737            8.66
##  5 Security                   10252         150737            6.80
##  6 Collision                   9097         150737            6.04
##  7 General Delay               7600         150737            5.04
##  8 Emergency Services          6766         150737            4.49
##  9 Investigation               3520         150737            2.34
## 10 Vision                      2035         150737            1.35
## 11 Management                    28         150737            0.0186
## 12 Overhead                      18         150737            0.0119
```

REDUCING 'INCIDENT' LABELS FROM 12 TO 4:

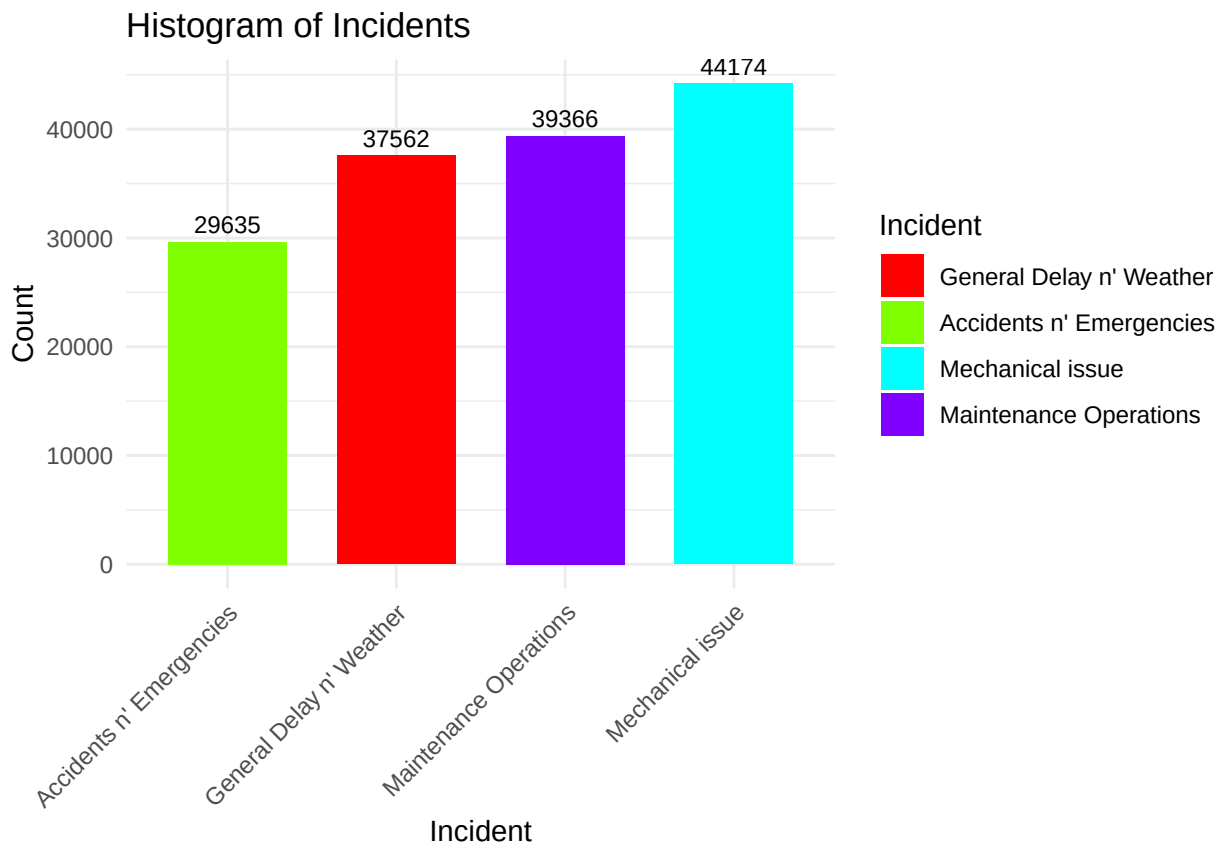
```r
data2_filled_updated2 <- data2_filled_updated1 %>%
  mutate(Incident = recode(Incident,
                           "Cleaning" = "General Delay n' Weather",
                           "Collision" = "Accidents n' Emergencies",
                           "Diversion" = "General Delay n' Weather",
                           "Emergency Services" = "Accidents n' Emergencies",
                           "General Delay" = "General Delay n' Weather",
                           "Investigation" = "Accidents n' Emergencies",
                           "Management" = "General Delay n' Weather",
                           "Mechanical" = "Mechanical issue",
                           "Operations" = "Maintenance Operations",
                           "Overhead" = "General Delay n' Weather",
                           "Security" = "Accidents n' Emergencies",
                           "Vision" = "General Delay n' Weather"
                           ))
```

```r
# Create a customized boxplot to clearly see the interquartile range
ggplot(data2_filled_updated2, aes(x = Incident, y = `Min_Delay_Winsorized`)) +
```

```
  geom_boxplot(coef = 1.5) +  # Adjust the coef parameter to control the length of the whiskers
  labs(x = "Incident", y = "Min Delay", title = "Boxplot of Min Delay by Incident")
```




```
library(dplyr)
library(ggplot2)
library(forcats)  # Load the forcats package for fct_reorder

# Calculate the count of each incident
incident_counts <- data2_filled_updated2 %>%
  count(Incident) %>%
  arrange(desc(n))  # Arrange in descending order of frequency

# Create a ggplot with multiple colors, data labels, and ordered incident types
ggplot(incident_counts, aes(x = fct_reorder(Incident, n), y = n, fill = Incident)) +
  geom_bar(stat = "identity", width = 0.7) +
  geom_text(aes(label = n), vjust = -0.5, size = 3) +  # Add data labels
  labs(title = "Histogram of Incidents", x = "Incident", y = "Count") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +  # Rotate x-axis labels
  scale_fill_manual(values = rainbow(length(incident_counts$Incident)))  # Use multicolor
```

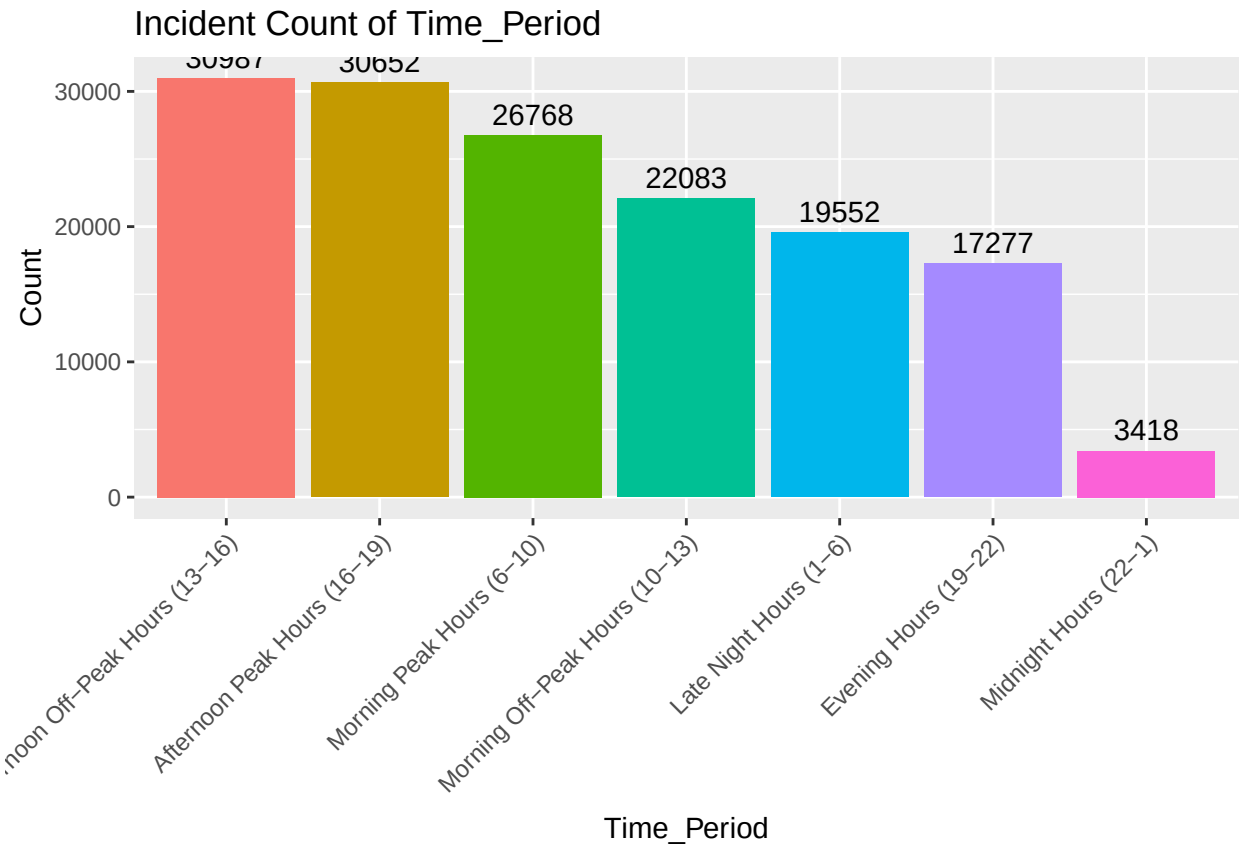


GROUPING 'TIME' VARIABLE ACROSS 4 TIME-PERIODS:

```
library(dplyr)

data2_filled_updated2 <- data2_filled_updated2 %>%
  mutate(Hour = as.integer(substr(Time, 1, 2)),
         Minute = as.integer(substr(Time, 4, 5)),
         Time_Period = case_when(
           (Hour == 6 & Minute >= 0) | (Hour > 6 & Hour < 10) ~ "Morning Peak Hours (6-10)",
           (Hour == 10 & Minute >= 0) | (Hour > 10 & Hour < 13) ~ "Morning Off-Peak Hours (10-13)",
           (Hour == 13 & Minute >= 0) | (Hour > 13 & Hour < 16) ~ "Afternoon Off-Peak Hours (13-16)",
           (Hour == 16 & Minute >= 0) | (Hour > 16 & Hour < 19) ~ "Afternoon Peak Hours (16-19)",
           (Hour == 19 & Minute >= 0) | (Hour > 19 & Hour < 22) ~ "Evening Hours (19-22)",
           (Hour == 23 & Minute >= 0) | (Hour > 23 & Hour < 1) ~ "Midnight Hours (22-1)",
           TRUE ~ "Late Night Hours (1-6)"
         ))
data2_filled_updated2 <- data2_filled_updated2 %>%
  select(-c(Hour, Minute))  # Remove the "Hour" and "Minute" columns

library(ggplot2)

# Reorder the levels of Time_Period based on count of incidents
data2_filled_updated2$Time_Period <- factor(data2_filled_updated2$Time_Period,
```

```
                                                levels = names(sort(table(data2_filled_updated2$Time_Period)
```

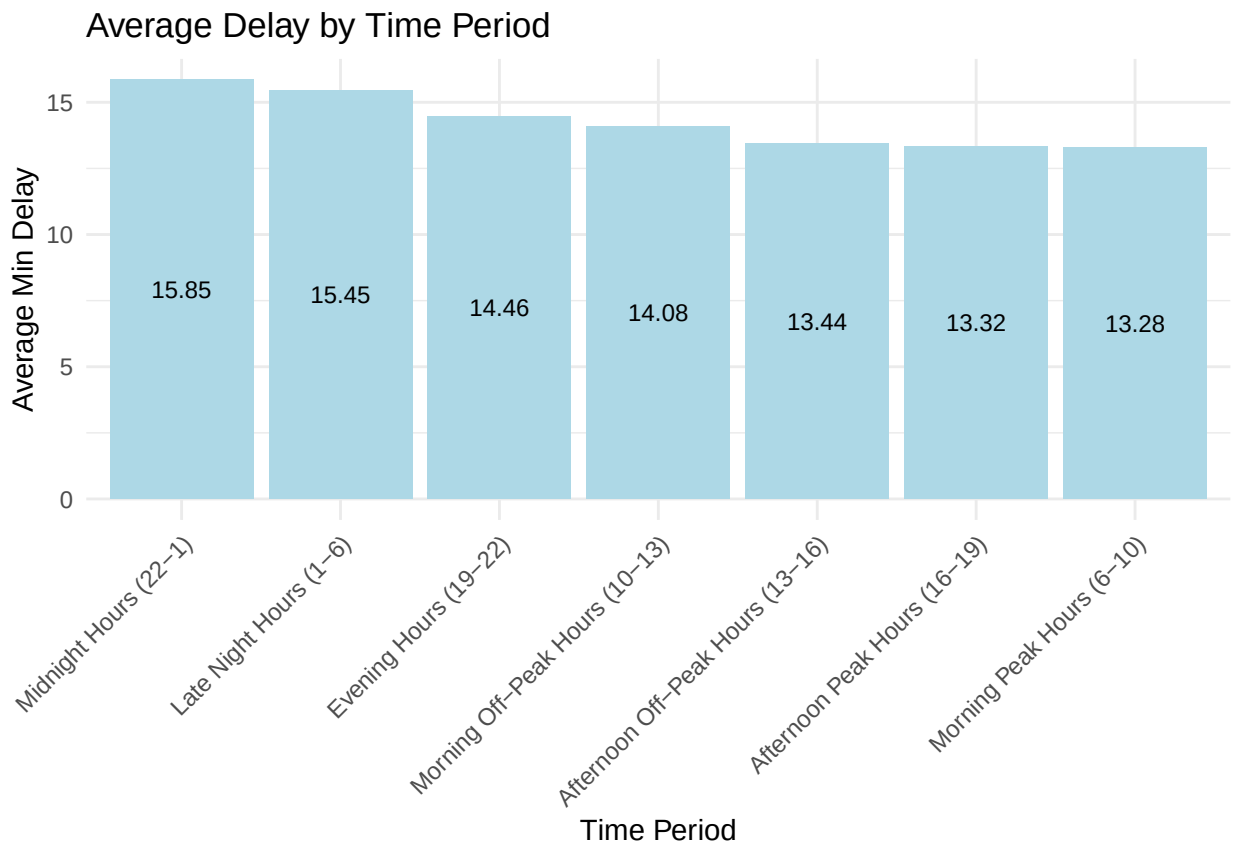
```
# Plotting with ggplot
ggplot(data2_filled_updated2, aes(x = Time_Period, fill = Time_Period)) +
  geom_bar() +
  geom_text(stat = 'count', aes(label = ..count..), vjust = -0.5) +
  scale_fill_discrete(name = "Time_Period") +
  labs(x = "Time_Period", y = "Count", title = "Incident Count of Time_Period") +
  theme(legend.position = "none", axis.text.x = element_text(angle = 45, hjust = 1))
```



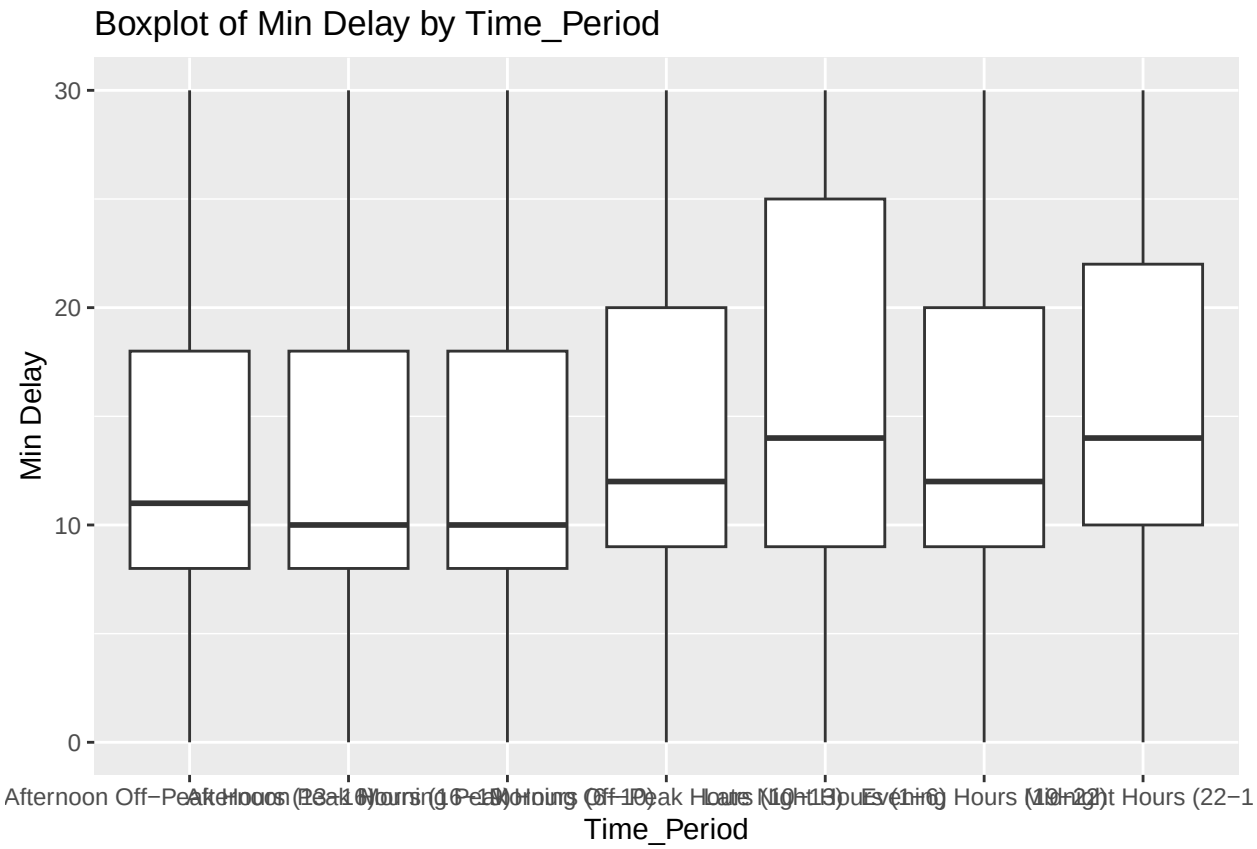

```
data2_filled_updated2$Time_Period <- as.factor(data2_filled_updated2$Time_Period)
```

```
# Aggregate data by time period and calculate the mean Min Delay for each time period
data_by_time_period <- data2_filled_updated2 %>%
  group_by(Time_Period) %>%
  summarise(Avg_Min_Delay = mean(`Min_Delay_Winsorized`))

# Reorder levels of Time_Period according to Avg_Min_Delay in descending order
data_by_time_period <- data_by_time_period %>%
  arrange(desc(Avg_Min_Delay)) %>%
  mutate(Time_Period = factor(Time_Period, levels = Time_Period))

# Create bar plot
ggplot(data_by_time_period, aes(x = Time_Period, y = Avg_Min_Delay)) +
  geom_bar(stat = "identity", fill = "lightblue") +  # Bar plot with different color
```

```
  geom_text(aes(label = round(Avg_Min_Delay, 2)), position = position_stack(vjust = 0.5), color = "blac
  labs(title = "Average Delay by Time Period",
       x = "Time Period",
       y = "Average Min Delay") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))  # Rotate x-axis labels for better readabili
```




```
# Create a customized boxplot to clearly see the interquartile range
ggplot(data2_filled_updated2, aes(x = Time_Period, y = `Min_Delay_Winsorized`)) +
  geom_boxplot(coef = 1.5) +  # Adjust the coef parameter to control the length of the whiskers
  labs(x = "Time_Period", y = "Min Delay", title = "Boxplot of Min Delay by Time_Period")
```



MONTH-WISE DELAY INCIDENTS:

```
library(dplyr)
library(ggplot2)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

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

```
# Extract the month name from the Date column
data2_filled_updated2$Month_Name <- month(data2_filled_updated2$Date, label = TRUE, abbr = FALSE)

# Aggregate count by month name
incidents_by_month_name <- data2_filled_updated2 %>%
  group_by(Month_Name) %>%
  summarise(Total_Incidents = n()) %>%
  mutate(Month_Name = reorder(Month_Name, -Total_Incidents)) # Reorder based on total incidents in desc
```

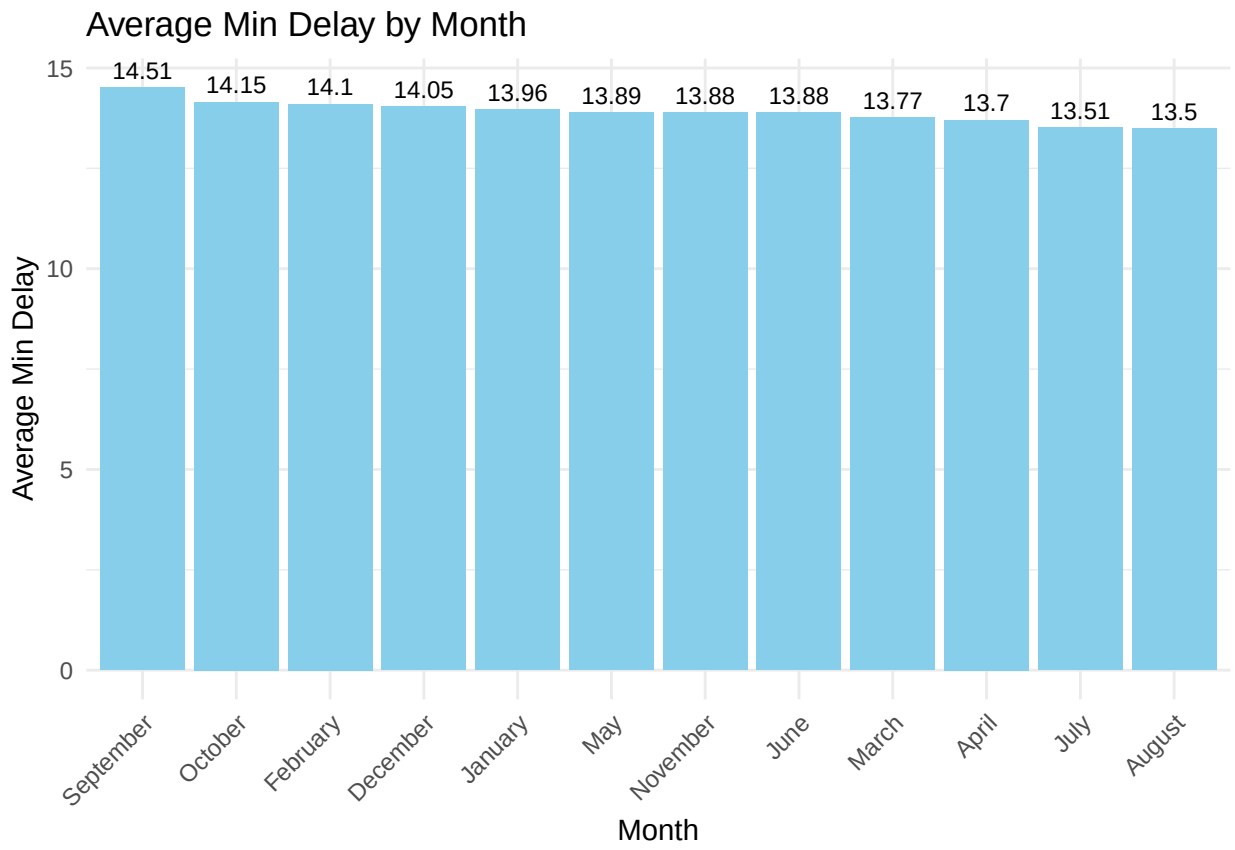
```r
# Extract month from the Date column and count incidents for each month
month_wise_incident_counts <- data2_filled_updated2 %>%
  mutate(Month = month(Date, label = TRUE)) %>%
  count(Month) %>%
  mutate(Percentage = n / sum(n) * 100) %>%
  arrange(Month)

# Print the month-wise counts and percentages
print(month_wise_incident_counts)
```

```
## # A tibble: 12 x 3
##    Month     n Percentage
##    <ord> <int>      <dbl>
##  1 Jan   12508       8.30
##  2 Feb   10912       7.24
##  3 Mar   13003       8.63
##  4 Apr   12367       8.20
##  5 May   13087       8.68
##  6 Jun   13876       9.21
##  7 Jul   12239       8.12
##  8 Aug   12447       8.26
##  9 Sep   13767       9.13
## 10 Oct   13861       9.20
## 11 Nov   12839       8.52
## 12 Dec    9831       6.52
```

AVERAGE 'MIN DELAY' ACROSS MONTHS:

```r
# MONTHWISE AVERAGE DELAY
# Load required libraries
library(dplyr)
library(ggplot2)

# Convert Date column to year-month format
data2_filled_updated2$Month <- format(data2_filled_updated2$Date, "%m")

# Aggregate data by month and calculate the mean Min Delay for each month
data_by_month <- data2_filled_updated2 %>%
  group_by(Month) %>%
  summarise(Avg_Min_Delay = mean(`Min_Delay_Winsorized`))

# Convert month numbers to month names
data_by_month$Month <- factor(month.name[as.numeric(data_by_month$Month)], levels = month.name)

# Reorder levels of Month according to Avg_Min_Delay in descending order
data_by_month <- data_by_month %>%
  arrange(desc(Avg_Min_Delay)) %>%
  mutate(Month = factor(Month, levels = Month))

# Create bar plot
```

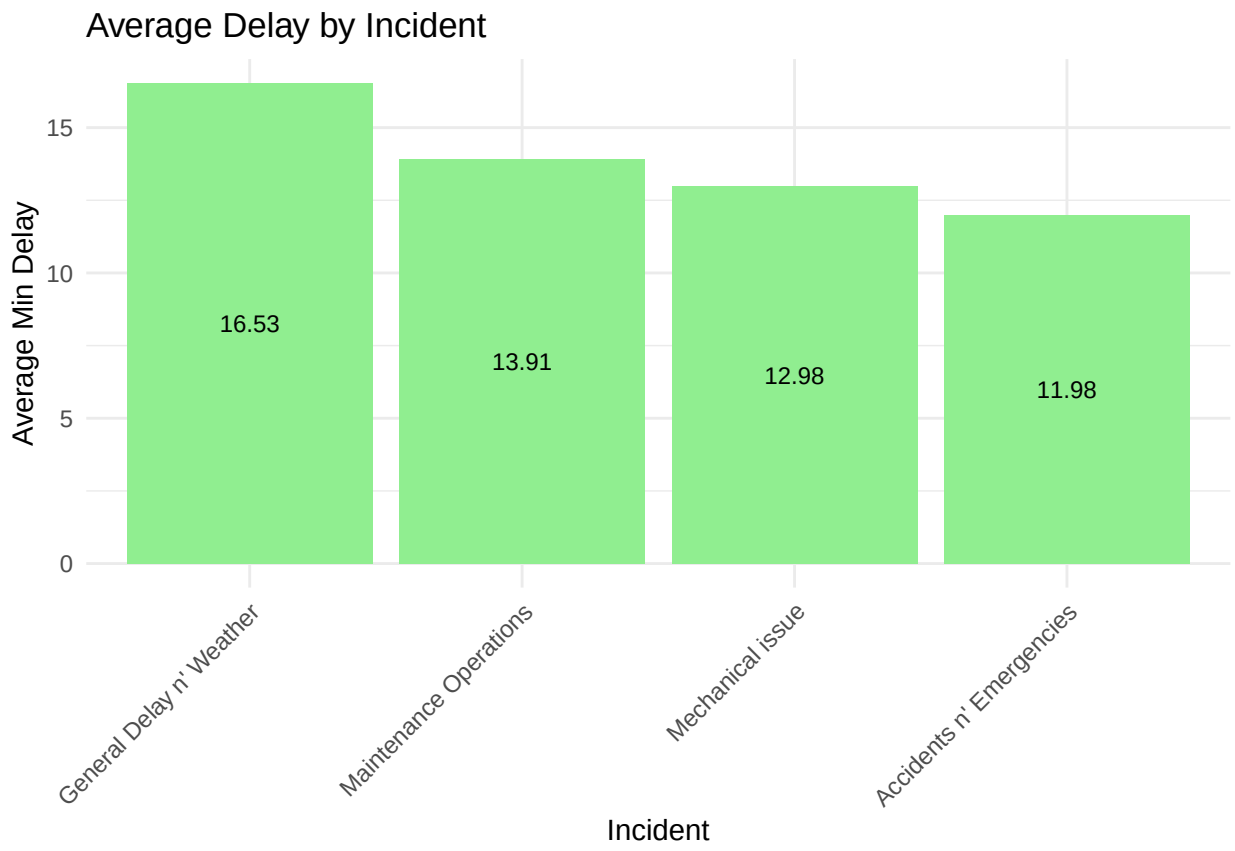
```r
ggplot(data_by_month, aes(x = Month, y = Avg_Min_Delay)) +
  geom_bar(stat = "identity", fill = "skyblue") +  # Bar plot
  geom_text(aes(label = round(Avg_Min_Delay, 2)), vjust = -0.5, color = "black", size = 3) +  # Add dat
  labs(title = "Average Min Delay by Month",
       x = "Month",
       y = "Average Min Delay") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))  # Rotate x-axis labels for better readabili
```



AVERAGE DELAY BY INCIDENT:

```r
library(dplyr)
library(ggplot2)

# Aggregate data by incident and calculate the mean Min Delay for each incident
data_by_incident <- data2_filled_updated2 %>%
  group_by(Incident) %>%
  summarise(Avg_Min_Delay = mean(`Min_Delay_Winsorized`))

# Reorder levels of Incident according to Avg_Min_Delay in descending order
data_by_incident <- data_by_incident %>%
  arrange(desc(Avg_Min_Delay)) %>%
  mutate(Incident = factor(Incident, levels = Incident))

# Create bar plot
ggplot(data_by_incident, aes(x = Incident, y = Avg_Min_Delay)) +
```

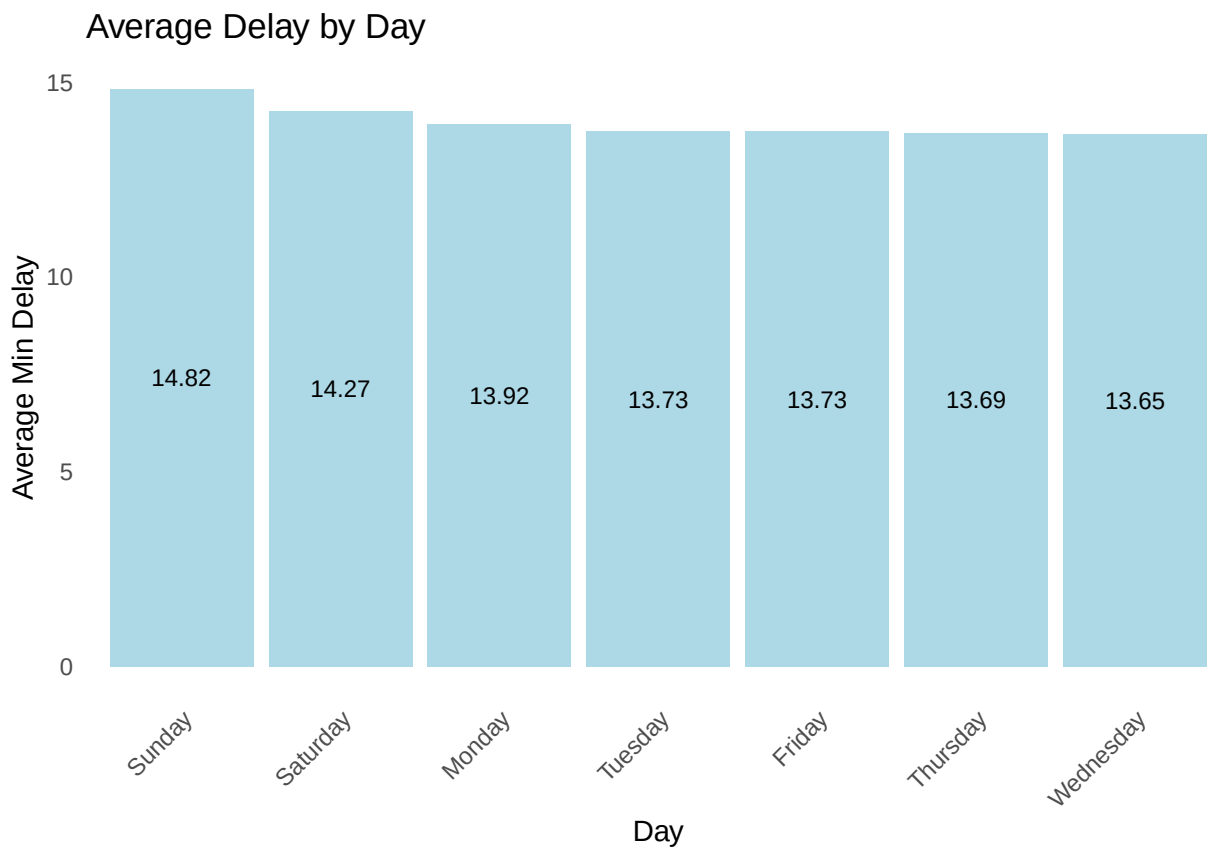
```
  geom_bar(stat = "identity", fill = "lightgreen") +  # Bar plot with different color
  geom_text(aes(label = round(Avg_Min_Delay, 2)), position = position_stack(vjust = 0.5), color = "blac
  labs(title = "Average Delay by Incident",
       x = "Incident",
       y = "Average Min Delay") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))  # Rotate x-axis labels for better readabili
```



AVERAGE DELAY BY DAY

```
# Aggregate data by day and calculate the mean Min Delay for each day
data_by_day <- data2_filled_updated2 %>%
  group_by(Day) %>%
  summarise(Avg_Min_Delay = mean(`Min_Delay_Winsorized`))

# Reorder levels of Day according to Avg_Min_Delay in descending order
data_by_day <- data_by_day %>%
  arrange(desc(Avg_Min_Delay)) %>%
  mutate(Day = factor(Day, levels = Day))

# Create bar plot
ggplot(data_by_day, aes(x = Day, y = Avg_Min_Delay)) +
  geom_bar(stat = "identity", fill = "lightblue") +  # Bar plot with different color
  geom_text(aes(label = round(Avg_Min_Delay, 2)), position = position_stack(vjust = 0.5), color = "blac
  labs(title = "Average Delay by Day",
       x = "Day",
```

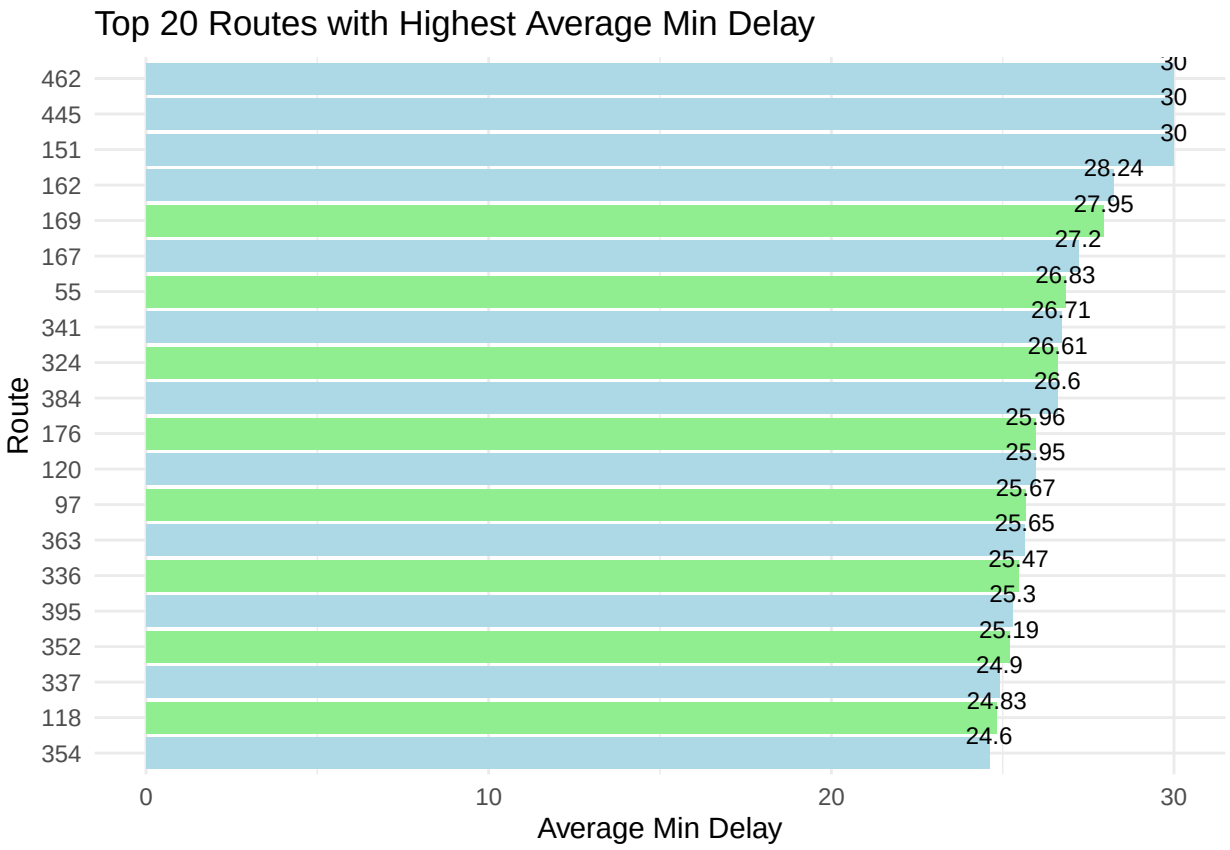
```
       y = "Average Min Delay") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1), panel.grid.major = element_blank(), panel.gri
```



TOP 20 ROUTES WITH HIGHEST DELAY INCIDENT

```
# Aggregate data by route and calculate the mean Min Delay for each route
data_by_route <- data2_filled_updated2 %>%
  group_by(Route) %>%
  summarise(Avg_Min_Delay = mean(`Min_Delay_Winsorized`))

# Reorder levels of Route according to Avg_Min_Delay in descending order
data_by_route <- data_by_route %>%
  arrange(desc(Avg_Min_Delay)) %>%
  mutate(Route = factor(Route, levels = Route))

# Select the top 25 routes with the highest average minimum delay
top_routes <- head(data_by_route, 20)

# Create bar plot for the top 20 routes
ggplot(top_routes, aes(x = reorder(Route, Avg_Min_Delay), y = Avg_Min_Delay)) +
  geom_bar(stat = "identity", fill = ifelse(rank(-top_routes$Avg_Min_Delay) %% 2 == 0, "lightblue", "lig
```

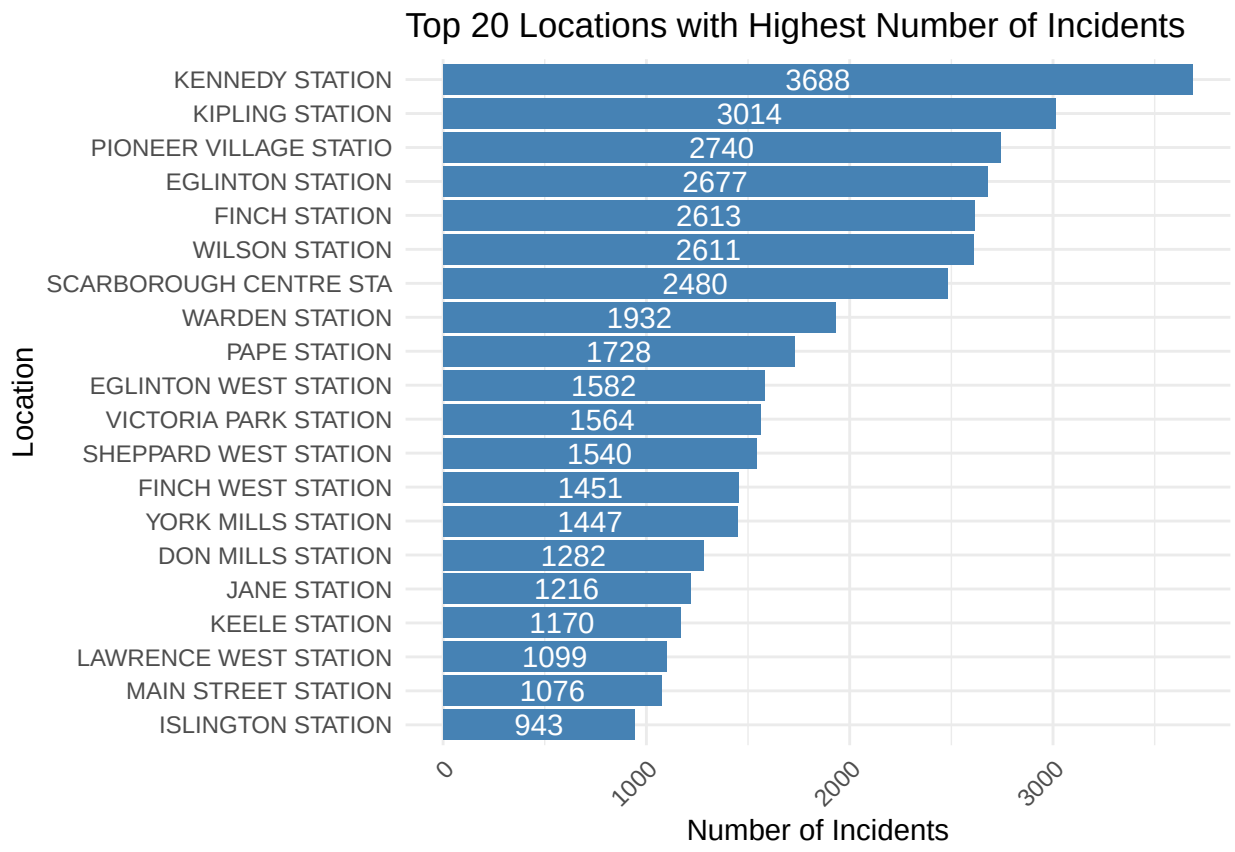
```
  geom_text(aes(label = round(Avg_Min_Delay, 2)), position = position_dodge(width = 0.9), vjust = -0.5,
  coord_flip() +  # Flip coordinates for horizontal bar chart
  labs(title = "Top 20 Routes with Highest Average Min Delay",
       x = "Route",
       y = "Average Min Delay") +
  theme_minimal()
```



TOP 20 LOCATION WITH HIGHEST DELAY INCIDENCE

```
# Load necessary libraries
library(ggplot2)
library(dplyr)

# Aggregate data to count the number of incidents per location
incident_counts <- data2_filled_updated2 %>%
  group_by(Location) %>%
  summarise(Incidents = n()) %>%
  arrange(desc(Incidents)) %>%
  top_n(20, Incidents)

# Plot the top 20 locations with the highest number of incidents, including data labels
ggplot(incident_counts, aes(x = reorder(Location, Incidents), y = Incidents)) +
```

```
  geom_bar(stat = "identity", fill = "steelblue") +
  geom_text(aes(label = Incidents), position = position_stack(vjust = 0.5), color = "white") +
  coord_flip() + # Flip the coordinates to make it a horizontal bar plot
  labs(title = "Top 20 Locations with Highest Number of Incidents",
       x = "Location",
       y = "Number of Incidents") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) # Improve label readability
```




```
# Display the plot
ggsave("Top20_Locations_Incidents_with_Labels.png", width = 10, height = 8, dpi = 300)
```

WINSORIZATION OF 'MIN GAP' VARIABLE

```
# Perform 'Winsorization' to Fix Outlier Issues: capping the outliers at a certain percentile. For exam
library(DescTools)

# Winsorize the 'Min Delay' column at the 5th and 95th percentiles for the entire data
data2_filled_updated2$Min_Gap_Winsorized <- Winsorize(data2_filled_updated2$`Min Gap`, probs = c(0.02, 

# Check the results
summary(data2_filled_updated2$Min_Gap_Winsorized)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   17.00   22.00   27.13   38.00   60.00
```

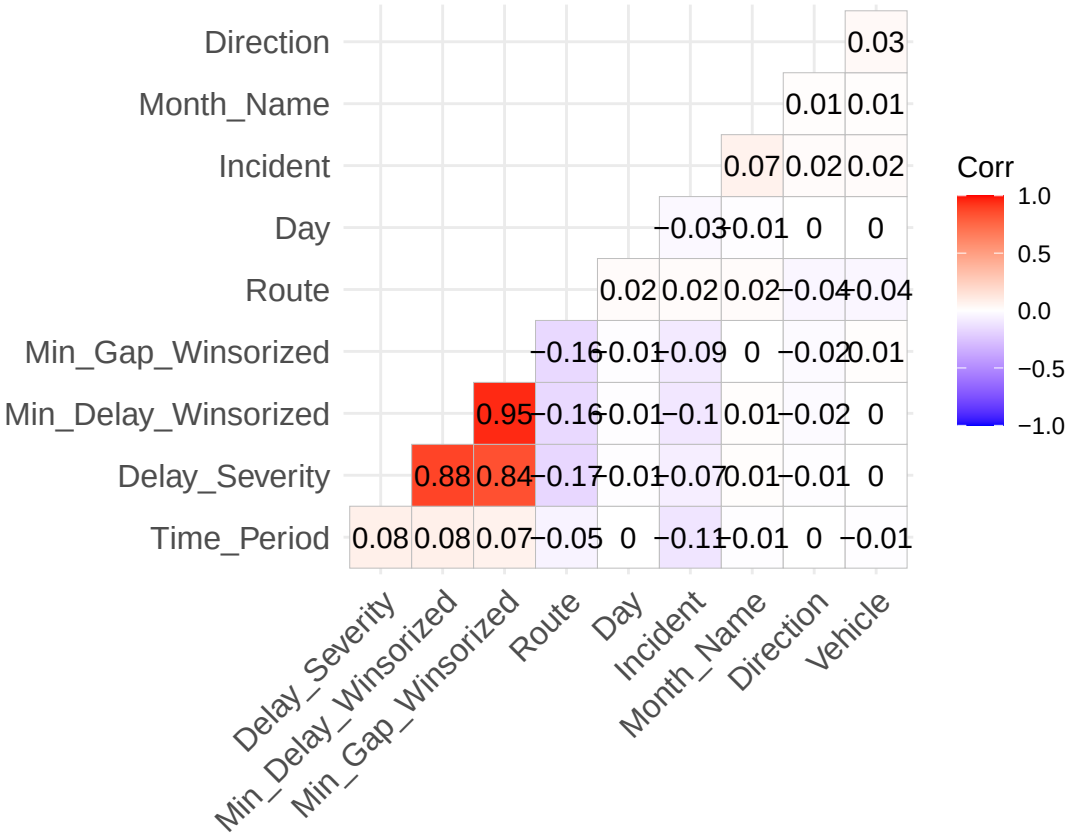
```
numeric_data1 <- data2_filled_updated2
str(numeric_data1)
```

```
## tibble [150,737 x 17] (S3: tbl_df/tbl/data.frame)
##  $ Date                  : POSIXct[1:150737], format: "2023-01-01" "2023-01-01" ...
##  $ Mode of Transportation: chr [1:150737] "Bus" "Bus" "Bus" "Bus" ...
##  $ Route                 : num [1:150737] 91 69 35 900 85 40 336 52 24 36 ...
##  $ Time                  : chr [1:150737] "02:30" "02:34" "03:06" "03:14" ...
##  $ Day                   : Factor w/ 7 levels "Friday","Monday",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ Location              : chr [1:150737] "WOODBINE AND MORTIMER" "WARDEN STATION" "JANE STATION" "K
##  $ Incident              : Factor w/ 4 levels "General Delay n' Weather",..: 1 2 1 2 2 2 1 2 1 1 ...
##  $ Min Delay             : num [1:150737] 81 22 30 17 1 0 138 30 20 334 ...
##  $ Min Gap               : num [1:150737] 111 44 60 17 1 0 168 60 40 344 ...
##  $ Direction             : int [1:150737] 5 4 3 3 3 5 3 2 5 5 ...
##  $ Vehicle               : num [1:150737] 8772 8407 1051 3334 1559 ...
##  $ Min_Delay_Winsorized  : num [1:150737] 30 22 30 17 1 0 30 30 20 30 ...
##  $ Delay_Severity        : Factor w/ 3 levels "Borderline Late (<10 Min)",..: 3 3 3 3 1 1 3 3 3 3 ..
##  $ Time_Period           : Factor w/ 7 levels "Afternoon Off-Peak Hours (13-16)",..: 5 5 5 5 5 5 5 5
##  $ Month_Name            : Ord.factor w/ 12 levels "January"<"February"<..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Month                 : chr [1:150737] "01" "01" "01" "01" ...
##  $ Min_Gap_Winsorized    : num [1:150737] 60 44 60 17 1 0 60 60 40 60 ...
```

CORRELATION ANALYSIS OF VARIABLES

```
library(ggcorrplot)
library(dplyr)
numeric_data1$Time_Period<-as.numeric(numeric_data1$Time_Period)
numeric_data1$Delay_Severity<-as.numeric(numeric_data1$Delay_Severity)
numeric_data1$Incident<-as.numeric(numeric_data1$Incident)
numeric_data1$Day<-as.numeric(numeric_data1$Day)
numeric_data1$Month_Name<-as.numeric(numeric_data1$Month_Name)
numeric_data1$Direction<-as.numeric(numeric_data1$Direction)

numeric_data1 <- numeric_data1 %>%
  select(-`Min Delay`, -`Min Gap`)
```

```
# Select numeric variables
numeric_data <- numeric_data1 %>%
  select_if(is.numeric)

# Calculate the correlation matrix
correlation_matrix <- cor(numeric_data)

# Plot the correlation matrix using ggcorrplot
ggcorrplot(correlation_matrix, hc.order = TRUE, type = "lower", lab = TRUE)
```



#APPLYING MULTIPLE LINEAR REGRESSION TO UNDERSTAND PREDICTIVE POWER OF KEY VARIABLES:

```
summary(lm (Min_Delay_Winsorized ~ Incident + Time_Period + Route + Day + Direction , data = data2_fill
```

```
##
## Call:
## lm(formula = Min_Delay_Winsorized ~ Incident + Time_Period +
##     Route + Day + Direction, data = data2_filled_updated2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -19.774  -5.336  -1.669   5.184  22.260
##
## Coefficients:
##                                            Estimate Std. Error t value
## (Intercept)                               1.707e+01  9.720e-02 175.660
## IncidentAccidents n' Emergencies         -4.585e+00  6.158e-02 -74.453
## IncidentMechanical issue                 -3.300e+00  5.560e-02 -59.343
## IncidentMaintenance Operations           -2.442e+00  5.732e-02 -42.600
## Time_PeriodAfternoon Peak Hours (16-19)  -8.563e-02  6.355e-02  -1.347
## Time_PeriodMorning Peak Hours (6-10)      2.467e-02  6.609e-02   0.373
## Time_PeriodMorning Off-Peak Hours (10-13) 4.114e-01  6.968e-02   5.904
## Time_PeriodLate Night Hours (1-6)         2.121e+00  7.250e-02  29.250
## Time_PeriodEvening Hours (19-22)          7.490e-01  7.518e-02   9.963
## Time_PeriodMidnight Hours (22-1)          2.032e+00  1.427e-01  14.241
```

```
## Route                                       -3.953e-03  6.883e-05 -57.429
## DayMonday                                    1.709e-01  7.405e-02   2.308
## DaySaturday                                  2.825e-01  7.601e-02   3.717
## DaySunday                                    8.565e-01  8.236e-02  10.399
## DayThursday                                 -7.899e-02  7.129e-02  -1.108
## DayTuesday                                  -5.955e-02  7.239e-02  -0.823
## DayWednesday                                -1.533e-01  7.117e-02  -2.155
## Direction                                   -1.301e-01  1.839e-02  -7.079
##                                             Pr(>|t|)
## (Intercept)                                  < 2e-16 ***
## IncidentAccidents n' Emergencies             < 2e-16 ***
## IncidentMechanical issue                     < 2e-16 ***
## IncidentMaintenance Operations               < 2e-16 ***
## Time_PeriodAfternoon Peak Hours (16-19)   0.177866
## Time_PeriodMorning Peak Hours (6-10)      0.708883
## Time_PeriodMorning Off-Peak Hours (10-13) 3.55e-09 ***
## Time_PeriodLate Night Hours (1-6)            < 2e-16 ***
## Time_PeriodEvening Hours (19-22)             < 2e-16 ***
## Time_PeriodMidnight Hours (22-1)             < 2e-16 ***
## Route                                        < 2e-16 ***
## DayMonday                                 0.020975 *
## DaySaturday                               0.000202 ***
## DaySunday                                    < 2e-16 ***
## DayThursday                               0.267828
## DayTuesday                                0.410753
## DayWednesday                              0.031192 *
## Direction                                 1.46e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.887 on 150719 degrees of freedom
## Multiple R-squared:  0.07227,    Adjusted R-squared:  0.07216
## F-statistic: 690.6 on 17 and 150719 DF,  p-value: < 2.2e-16
```

#FORWARD SELECTION

```
library(MASS) # stepwise regression
```

```
## Warning: package 'MASS' was built under R version 4.3.1
```

```
##
## Attaching package: 'MASS'
```

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

```
library(leaps) # all subsets regression
full <- lm(Min_Delay_Winsorized ~ Incident + Time_Period + Route + Day + Direction , data = data2_fille
null <- lm(Min_Delay_Winsorized~1,data=data2_filled_updated2)
stepF <- stepAIC(null, scope=list(lower=null, upper=full),
direction= "forward", trace=TRUE)
```

```
## Start:  AIC=633918
## Min_Delay_Winsorized ~ 1
##
##               Df Sum of Sq      RSS    AIC
## + Incident     3    406042  9700927 627743
## + Route        1    249332  9857636 630155
## + Time_Period  6     93300 10013669 632532
## + Day          6     18715 10088253 633651
## + Direction    1      2762 10104207 633879
## <none>                     10106968 633918
##
## Step:  AIC=627743.2
## Min_Delay_Winsorized ~ Incident
##
##               Df Sum of Sq     RSS    AIC
## + Route        1    223472 9477455 624232
## + Time_Period  6     99186 9601740 626206
## + Day          6     22634 9678293 627403
## + Direction    1      1399 9699528 627723
## <none>                     9700927 627743
##
## Step:  AIC=624232.2
## Min_Delay_Winsorized ~ Incident + Route
##
##               Df Sum of Sq     RSS    AIC
## + Time_Period  6     85697 9391758 622875
## + Day          6     14737 9462718 624010
## + Direction    1      3097 9474358 624185
## <none>                     9477455 624232
##
## Step:  AIC=622875
## Min_Delay_Winsorized ~ Incident + Route + Time_Period
##
##             Df Sum of Sq     RSS    AIC
## + Day        6   12070.2 9379688 622693
## + Direction  1    2996.3 9388762 622829
## <none>                   9391758 622875
##
## Step:  AIC=622693.2
## Min_Delay_Winsorized ~ Incident + Route + Time_Period + Day
##
##             Df Sum of Sq     RSS    AIC
## + Direction  1    3117.2 9376571 622645
## <none>                   9379688 622693
##
## Step:  AIC=622645
## Min_Delay_Winsorized ~ Incident + Route + Time_Period + Day +
##     Direction
```

```
summary(stepF)
```

```
##
## Call:
## lm(formula = Min_Delay_Winsorized ~ Incident + Route + Time_Period +
```

```
##     Day + Direction, data = data2_filled_updated2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -19.774  -5.336  -1.669   5.184  22.260
##
## Coefficients:
##                                          Estimate Std. Error t value
## (Intercept)                              1.707e+01  9.720e-02 175.660
## IncidentAccidents n' Emergencies        -4.585e+00  6.158e-02 -74.453
## IncidentMechanical issue                -3.300e+00  5.560e-02 -59.343
## IncidentMaintenance Operations          -2.442e+00  5.732e-02 -42.600
## Route                                   -3.953e-03  6.883e-05 -57.429
## Time_PeriodAfternoon Peak Hours (16-19) -8.563e-02  6.355e-02  -1.347
## Time_PeriodMorning Peak Hours (6-10)     2.467e-02  6.609e-02   0.373
## Time_PeriodMorning Off-Peak Hours (10-13) 4.114e-01  6.968e-02   5.904
## Time_PeriodLate Night Hours (1-6)        2.121e+00  7.250e-02  29.250
## Time_PeriodEvening Hours (19-22)         7.490e-01  7.518e-02   9.963
## Time_PeriodMidnight Hours (22-1)         2.032e+00  1.427e-01  14.241
## DayMonday                                1.709e-01  7.405e-02   2.308
## DaySaturday                              2.825e-01  7.601e-02   3.717
## DaySunday                                8.565e-01  8.236e-02  10.399
## DayThursday                             -7.899e-02  7.129e-02  -1.108
## DayTuesday                              -5.955e-02  7.239e-02  -0.823
## DayWednesday                            -1.533e-01  7.117e-02  -2.155
## Direction                               -1.301e-01  1.839e-02  -7.079
##                                         Pr(>|t|)
## (Intercept)                              < 2e-16 ***
## IncidentAccidents n' Emergencies         < 2e-16 ***
## IncidentMechanical issue                 < 2e-16 ***
## IncidentMaintenance Operations           < 2e-16 ***
## Route                                    < 2e-16 ***
## Time_PeriodAfternoon Peak Hours (16-19)  0.177866
## Time_PeriodMorning Peak Hours (6-10)     0.708883
## Time_PeriodMorning Off-Peak Hours (10-13) 3.55e-09 ***
## Time_PeriodLate Night Hours (1-6)        < 2e-16 ***
## Time_PeriodEvening Hours (19-22)         < 2e-16 ***
## Time_PeriodMidnight Hours (22-1)         < 2e-16 ***
## DayMonday                                0.020975 *
## DaySaturday                              0.000202 ***
## DaySunday                                < 2e-16 ***
## DayThursday                              0.267828
## DayTuesday                               0.410753
## DayWednesday                             0.031192 *
## Direction                                1.46e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.887 on 150719 degrees of freedom
## Multiple R-squared:  0.07227,    Adjusted R-squared:  0.07216
## F-statistic: 690.6 on 17 and 150719 DF,  p-value: < 2.2e-16
```

BACKWARD ELIMINATION

```
full <- lm(Min_Delay_Winsorized ~ Incident + Time_Period + Route + Day + Direction , data = data2_fille
stepB <- stepAIC(full, direction= "backward", trace=TRUE)
```

```
## Start:  AIC=622645
## Min_Delay_Winsorized ~ Incident + Time_Period + Route + Day +
##     Direction
##
##               Df Sum of Sq     RSS    AIC
## <none>                     9376571 622645
## - Direction    1      3117 9379688 622693
## - Day          6     12191 9388762 622829
## - Time_Period  6     82914 9459484 623960
## - Route        1    205181 9581752 625906
## - Incident     3    389090 9765661 628768
```

```
summary(stepB)
```

```
##
## Call:
## lm(formula = Min_Delay_Winsorized ~ Incident + Time_Period +
##     Route + Day + Direction, data = data2_filled_updated2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -19.774  -5.336  -1.669   5.184  22.260
##
## Coefficients:
##                                             Estimate Std. Error t value
## (Intercept)                                1.707e+01  9.720e-02 175.660
## IncidentAccidents n' Emergencies          -4.585e+00  6.158e-02 -74.453
## IncidentMechanical issue                  -3.300e+00  5.560e-02 -59.343
## IncidentMaintenance Operations            -2.442e+00  5.732e-02 -42.600
## Time_PeriodAfternoon Peak Hours (16-19)   -8.563e-02  6.355e-02  -1.347
## Time_PeriodMorning Peak Hours (6-10)       2.467e-02  6.609e-02   0.373
## Time_PeriodMorning Off-Peak Hours (10-13)  4.114e-01  6.968e-02   5.904
## Time_PeriodLate Night Hours (1-6)          2.121e+00  7.250e-02  29.250
## Time_PeriodEvening Hours (19-22)           7.490e-01  7.518e-02   9.963
## Time_PeriodMidnight Hours (22-1)           2.032e+00  1.427e-01  14.241
## Route                                     -3.953e-03  6.883e-05 -57.429
## DayMonday                                  1.709e-01  7.405e-02   2.308
## DaySaturday                                2.825e-01  7.601e-02   3.717
## DaySunday                                  8.565e-01  8.236e-02  10.399
## DayThursday                               -7.899e-02  7.129e-02  -1.108
## DayTuesday                                -5.955e-02  7.239e-02  -0.823
## DayWednesday                              -1.533e-01  7.117e-02  -2.155
## Direction                                 -1.301e-01  1.839e-02  -7.079
##                                            Pr(>|t|)
## (Intercept)                                 < 2e-16 ***
## IncidentAccidents n' Emergencies            < 2e-16 ***
```

```
## IncidentMechanical issue                    < 2e-16 ***
## IncidentMaintenance Operations              < 2e-16 ***
## Time_PeriodAfternoon Peak Hours (16-19)    0.177866
## Time_PeriodMorning Peak Hours (6-10)       0.708883
## Time_PeriodMorning Off-Peak Hours (10-13) 3.55e-09 ***
## Time_PeriodLate Night Hours (1-6)           < 2e-16 ***
## Time_PeriodEvening Hours (19-22)            < 2e-16 ***
## Time_PeriodMidnight Hours (22-1)            < 2e-16 ***
## Route                                       < 2e-16 ***
## DayMonday                                  0.020975 *
## DaySaturday                                0.000202 ***
## DaySunday                                   < 2e-16 ***
## DayThursday                                0.267828
## DayTuesday                                 0.410753
## DayWednesday                               0.031192 *
## Direction                                  1.46e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.887 on 150719 degrees of freedom
## Multiple R-squared:  0.07227,    Adjusted R-squared:  0.07216
## F-statistic: 690.6 on 17 and 150719 DF,  p-value: < 2.2e-16
```

#UNDERSTAND BEST COMBINATION OF ATTRIBUTES

```
subsets<-regsubsets(Min_Delay_Winsorized ~ Incident + Time_Period + Route + Day + Direction , data = da
sub.sum <- summary(subsets)
as.data.frame(sub.sum$outmat)
```

```
##          IncidentAccidents n' Emergencies IncidentMechanical issue
## 1  ( 1 )
## 2  ( 1 )                                *
## 3  ( 1 )                                *                        *
## 4  ( 1 )                                *                        *
## 5  ( 1 )                                *                        *
## 6  ( 1 )                                *                        *
## 7  ( 1 )                                *                        *
## 8  ( 1 )                                *                        *
##          IncidentMaintenance Operations Time_PeriodAfternoon Peak Hours (16-19)
## 1  ( 1 )
## 2  ( 1 )
## 3  ( 1 )
## 4  ( 1 )                               *
## 5  ( 1 )                               *
## 6  ( 1 )                               *
## 7  ( 1 )                               *
## 8  ( 1 )                               *
##          Time_PeriodMorning Peak Hours (6-10)
## 1  ( 1 )
## 2  ( 1 )
## 3  ( 1 )
## 4  ( 1 )
## 5  ( 1 )
```

```
## 6  ( 1 )
## 7  ( 1 )
## 8  ( 1 )
##          Time_PeriodMorning Off-Peak Hours (10-13)
## 1  ( 1 )
## 2  ( 1 )
## 3  ( 1 )
## 4  ( 1 )
## 5  ( 1 )
## 6  ( 1 )
## 7  ( 1 )
## 8  ( 1 )
##          Time_PeriodLate Night Hours (1-6) Time_PeriodEvening Hours (19-22)
## 1  ( 1 )
## 2  ( 1 )
## 3  ( 1 )
## 4  ( 1 )
## 5  ( 1 )                                *
## 6  ( 1 )                                *
## 7  ( 1 )                                *
## 8  ( 1 )                                *                                  *
##          Time_PeriodMidnight Hours (22-1) Route DayMonday DaySaturday DaySunday
## 1  ( 1 )                                    *
## 2  ( 1 )                                    *
## 3  ( 1 )                                    *
## 4  ( 1 )                                    *
## 5  ( 1 )                                    *
## 6  ( 1 )                               *    *
## 7  ( 1 )                               *    *                                *
## 8  ( 1 )                               *    *                                *
##          DayThursday DayTuesday DayWednesday Direction
## 1  ( 1 )
## 2  ( 1 )
## 3  ( 1 )
## 4  ( 1 )
## 5  ( 1 )
## 6  ( 1 )
## 7  ( 1 )
## 8  ( 1 )
```

```
Cleaned_Data<-data2_filled_updated2
Cleaned_Data$Time_Period<- as.integer(Cleaned_Data$Time_Period)
Cleaned_Data$Incident<- as.numeric(Cleaned_Data$Incident)
Cleaned_Data$Day<- as.numeric(Cleaned_Data$Day)
str(Cleaned_Data)
```

```
## tibble [150,737 x 17] (S3: tbl_df/tbl/data.frame)
##  $ Date                  : POSIXct[1:150737], format: "2023-01-01" "2023-01-01" ...
##  $ Mode of Transportation: chr [1:150737] "Bus" "Bus" "Bus" "Bus" ...
##  $ Route                 : num [1:150737] 91 69 35 900 85 40 336 52 24 36 ...
##  $ Time                  : chr [1:150737] "02:30" "02:34" "03:06" "03:14" ...
##  $ Day                   : num [1:150737] 4 4 4 4 4 4 4 4 4 4 ...
##  $ Location              : chr [1:150737] "WOODBINE AND MORTIMER" "WARDEN STATION" "JANE STATION" "K
##  $ Incident              : num [1:150737] 1 2 1 2 2 2 1 2 1 1 ...
```

```
##  $ Min Delay             : num [1:150737] 81 22 30 17 1 0 138 30 20 334 ...
##  $ Min Gap               : num [1:150737] 111 44 60 17 1 0 168 60 40 344 ...
##  $ Direction             : int [1:150737] 5 4 3 3 3 5 3 2 5 5 ...
##  $ Vehicle               : num [1:150737] 8772 8407 1051 3334 1559 ...
##  $ Min_Delay_Winsorized  : num [1:150737] 30 22 30 17 1 0 30 30 20 30 ...
##  $ Delay_Severity        : Factor w/ 3 levels "Borderline Late (<10 Min)",..: 3 3 3 3 1 1 3 3 3 3 ..
##  $ Time_Period           : int [1:150737] 5 5 5 5 5 5 5 5 5 5 ...
##  $ Month_Name            : Ord.factor w/ 12 levels "January"<"February"<..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Month                 : chr [1:150737] "01" "01" "01" "01" ...
##  $ Min_Gap_Winsorized    : num [1:150737] 60 44 60 17 1 0 60 60 40 60 ...
```

#DATA NORMALIZATION BEFORE APPLYING PREDICTIVE MODELING

```
library(class)
library(gmodels)
```

```
## Warning: package 'gmodels' was built under R version 4.3.1
```

```
## Registered S3 method overwritten by 'gdata':
##   method         from
##   reorder.factor DescTools
```

```
round(prop.table(table(Cleaned_Data$Incident))*100,digits = 1)
```

```
##
##    1    2    3    4
## 24.9 19.7 29.3 26.1
```

```
normalize<-function(x){return((x-min(x))/(max(x)-min(x)))}
```

```
normalized<-as.data.frame(lapply(Cleaned_Data[c(3,5,7,10,11,14)],normalize))
```

```
Preprocessed_dataset<-cbind(Cleaned_Data$Delay_Severity,normalized)
str(Preprocessed_dataset)
```

```
## 'data.frame':    150737 obs. of  7 variables:
##  $ Cleaned_Data$Delay_Severity: Factor w/ 3 levels "Borderline Late (<10 Min)",..: 3 3 3 3 1 1 3 3 3
##  $ Route                      : num  0.0901 0.0681 0.034 0.8999 0.0841 ...
##  $ Day                        : num  0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 ...
##  $ Incident                   : num  0 0.333 0 0.333 0.333 ...
##  $ Direction                  : num  1 0.75 0.5 0.5 0.5 1 0.5 0.25 1 1 ...
##  $ Vehicle                    : num  0.0886 0.0849 0.0106 0.0337 0.0157 ...
##  $ Time_Period                : num  0.667 0.667 0.667 0.667 0.667 ...
```