SDS 323 Proposal

2024-03-31

R Markdown

```r
getwd()
```

```
## [1] "C:/Users/saman/Downloads"
```

```r
setwd("C:/Users/saman/Downloads")
```

```r
data<- read.csv("Traffic_Cameras_20240330.csv")
```

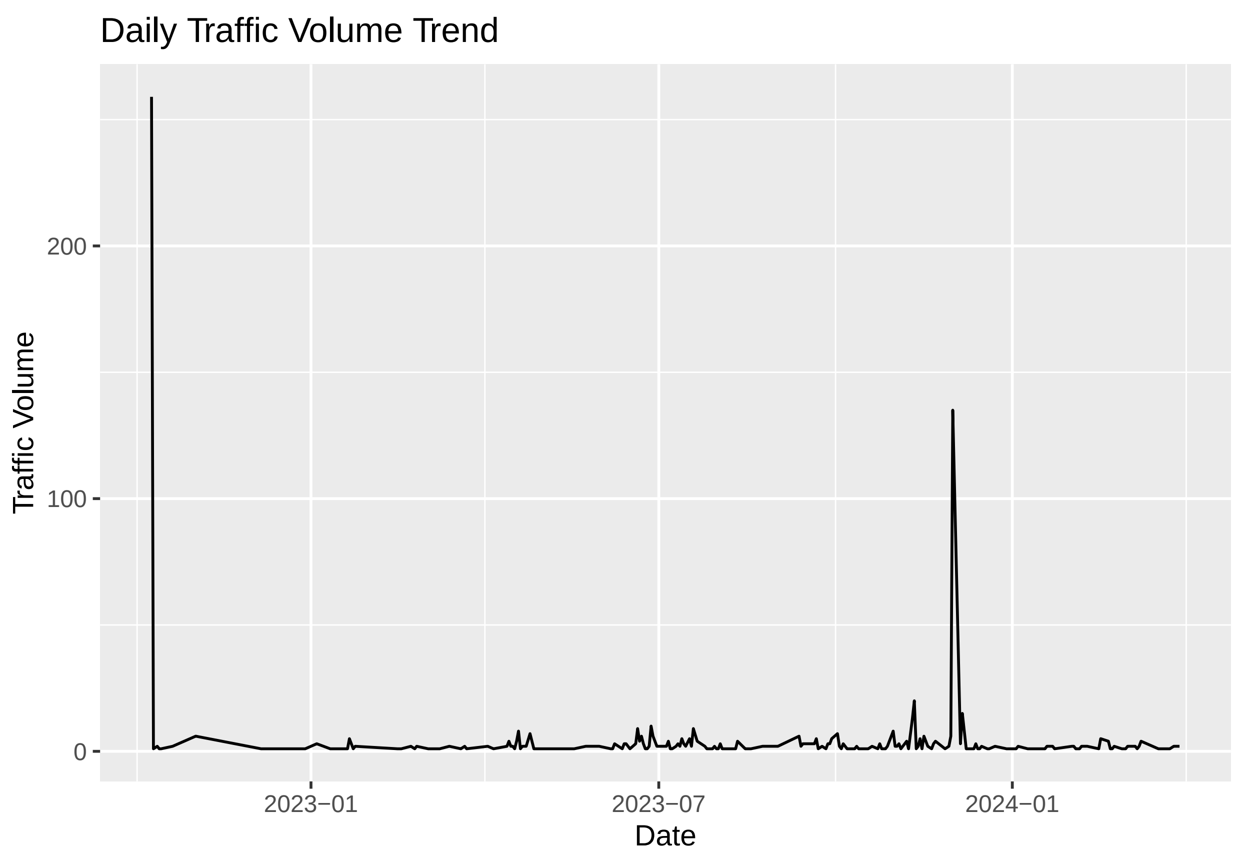
Question 1:

```r
# Load necessary libraries
library(ggplot2)
library(lubridate)
```

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

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

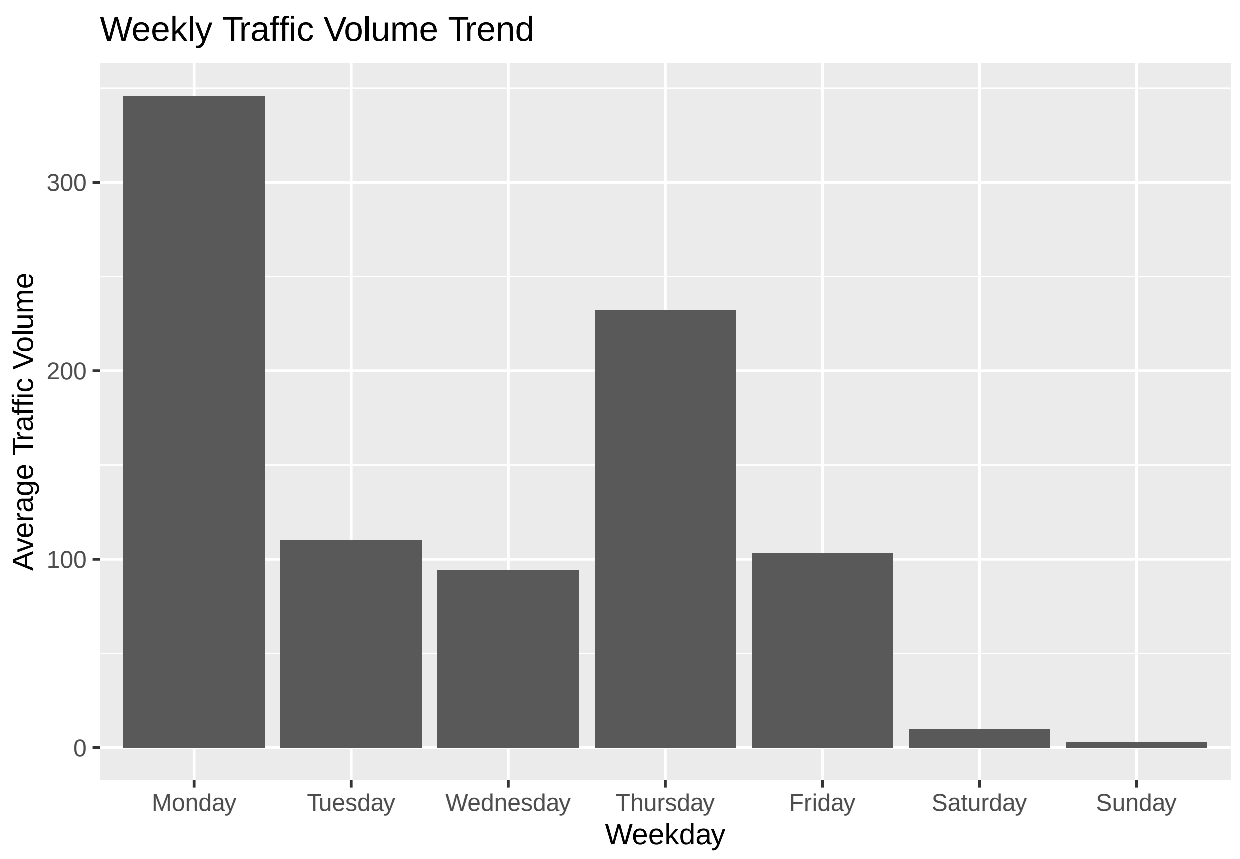
```r
# Convert 'Modified.Date' column to datetime format
data$Modified.Date <- as.POSIXct(data$Modified.Date, format = "%m/%d/%Y %I:%M:%S %p")

# Daily trend
daily_traffic <- aggregate(ID ~ as.Date(Modified.Date), data = data, FUN = length)

# Plot daily traffic volume trend
ggplot(daily_traffic, aes(x = `as.Date(Modified.Date)`, y = ID)) +
  geom_line() +
  labs(x = "Date", y = "Traffic Volume", title = "Daily Traffic Volume Trend")
```



```r
# Weekly trend
data$Weekday <- factor(weekdays(data$Modified.Date), levels = c("Monday", "Tuesday", "Wednesday", "Thurs
weekly_traffic <- aggregate(ID ~ Weekday, data = data, FUN = mean)
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

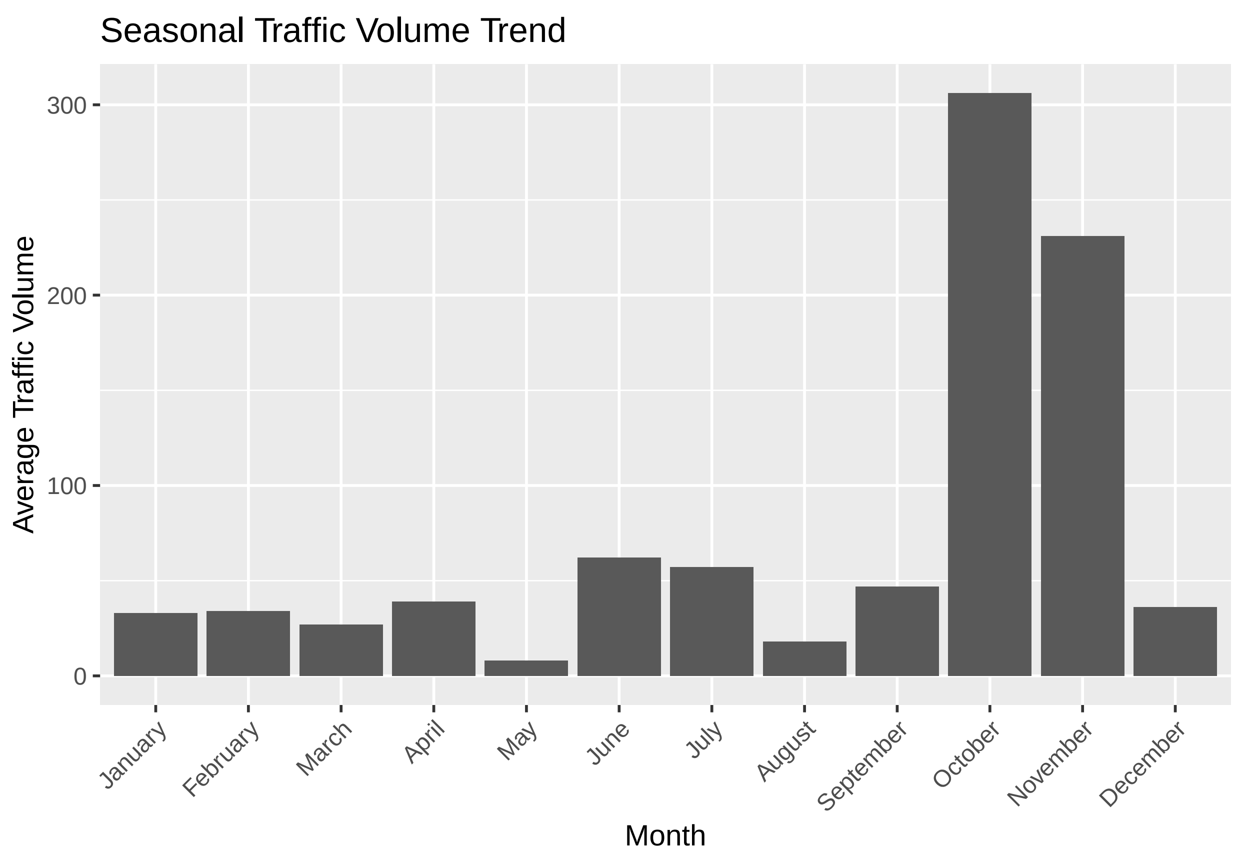
```r
weekly_traffic <- aggregate(ID ~ Weekday, data = data, FUN = function(x) length(unique(x)))

# Plot weekly traffic volume trend
ggplot(weekly_traffic, aes(x = Weekday, y = ID)) +
  geom_bar(stat = "identity") +
  labs(x = "Weekday", y = "Average Traffic Volume", title = "Weekly Traffic Volume Trend")
```



```r
# Seasonal trend
data$Month <- factor(months(data$Modified.Date), levels = month.name)
seasonal_traffic <- aggregate(ID ~ Month, data = data, FUN = mean)
```

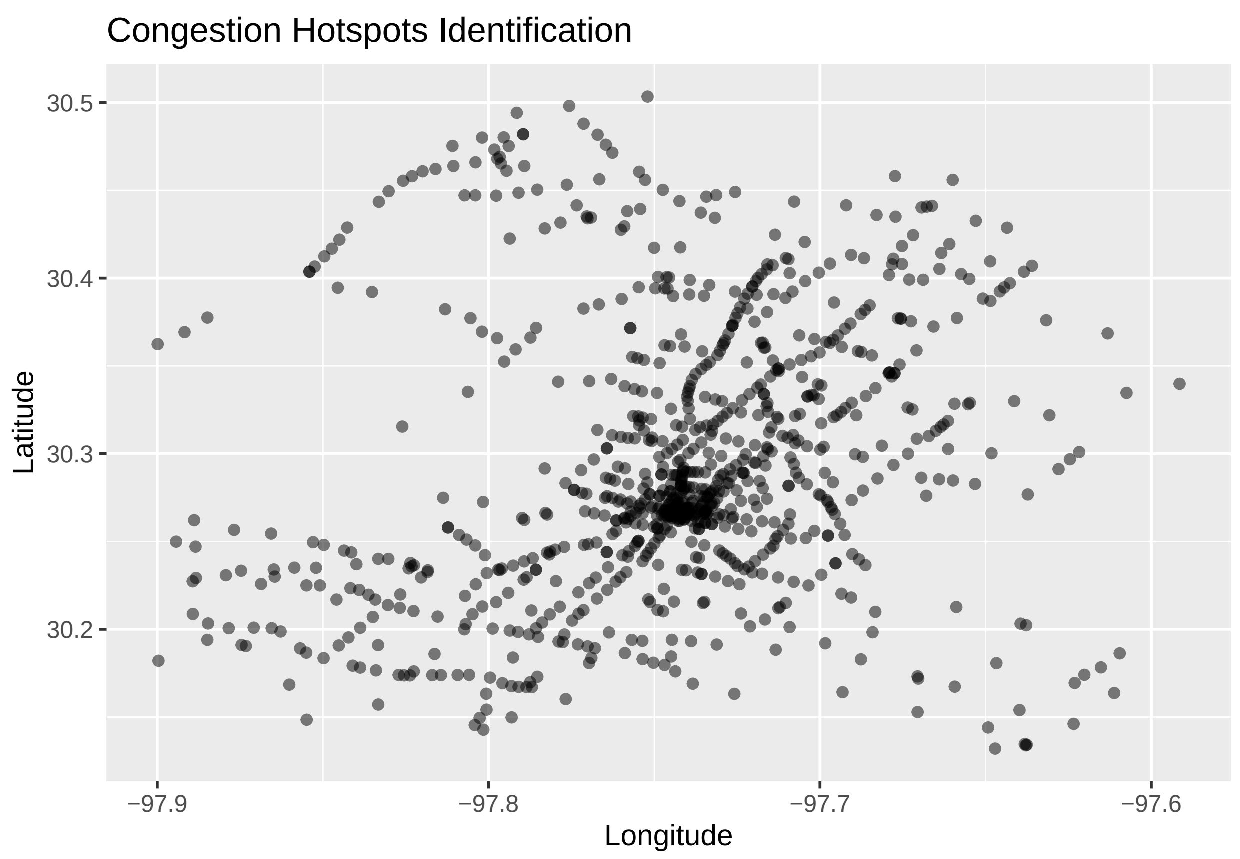
```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```r
seasonal_traffic <- aggregate(ID ~ Month, data = data, FUN = function(x) length(unique(x)))

# Plot seasonal traffic volume trend
ggplot(seasonal_traffic, aes(x = Month, y = ID)) +
  geom_bar(stat = "identity") +
  labs(x = "Month", y = "Average Traffic Volume", title = "Seasonal Traffic Volume Trend" ) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Question 2:

```
library(stringr)
data$longitude <- as.numeric(str_extract(data$Location, "-?[0-9]+\\.[0-9]+"))
data$latitude <- as.numeric(str_extract(data$Location, "\\s-?[0-9]+\\.[0-9]+"))

# Plot congestion hotspots using ggplot2
ggplot(data, aes(x = longitude, y = latitude)) +
  geom_point(alpha = 0.5) +
  labs(x = "Longitude", y = "Latitude", title = "Congestion Hotspots Identification")
```



Question 3:

```r
# Load necessary libraries
library(ggplot2)
library(lubridate)
library(tidyr)
library(dplyr)
```

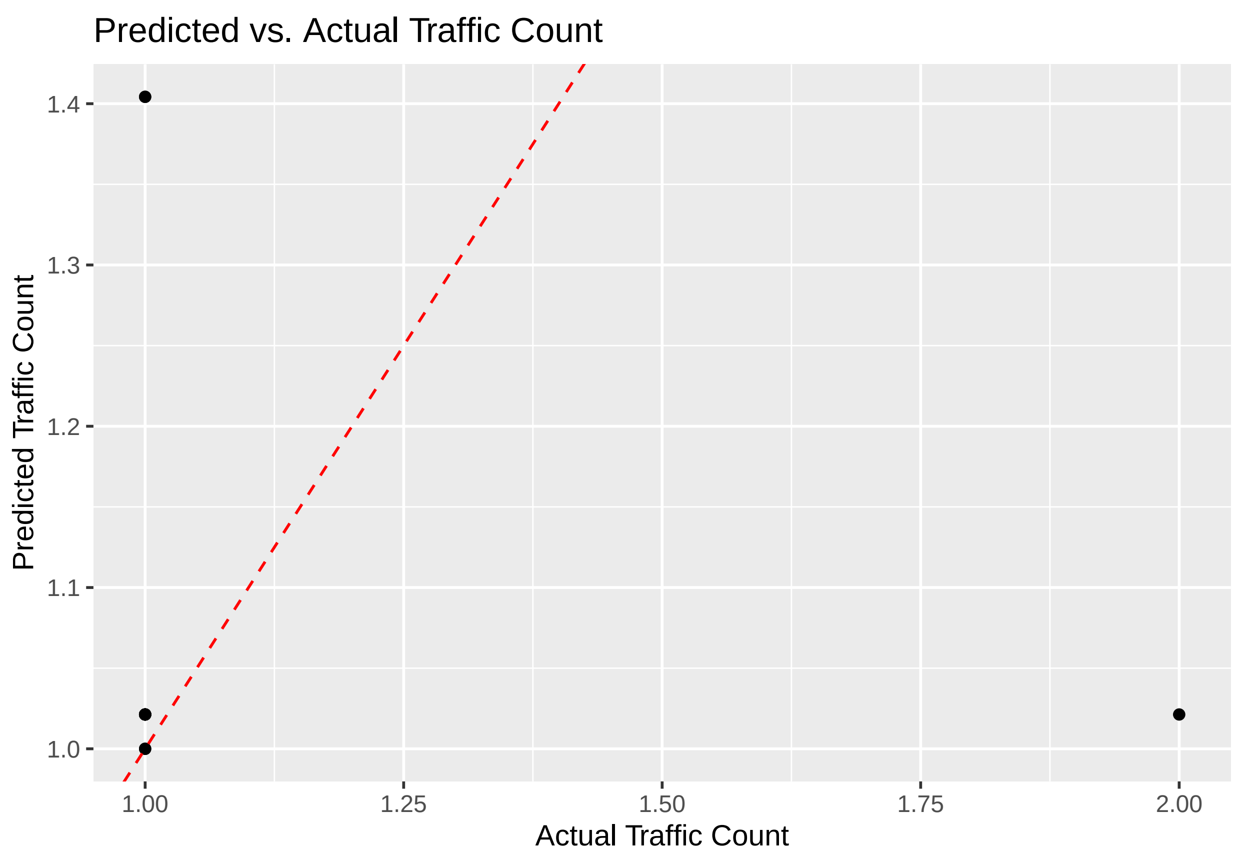
```
##
## Attaching package: 'dplyr'

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

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

```r
# Convert 'Modified Date' column to datetime format
data$Modified.Date <- as.POSIXct(data$Modified.Date, format = "%m/%d/%Y %I:%M:%S %p")

# Extract date and time components
data <- data %>%
  mutate(Date = as.Date(Modified.Date),
         Time = format(Modified.Date, format = "%H:%M:%S"))

# Aggregate traffic volume by intersection ID, date, and time
traffic_volume <- data %>%
  group_by(COA.Intersection.ID, Date, Time) %>%
  summarise(Traffic_Count = n()) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'COA.Intersection.ID', 'Date'. You can
## override using the `.groups` argument.
```

```r
# Create lagged variables for predictive modeling
traffic_volume <- traffic_volume %>%
  arrange(Date, Time) %>%
  group_by(COA.Intersection.ID) %>%
  mutate(Lag_Traffic_Count_1 = lag(Traffic_Count, 1),
         Lag_Traffic_Count_2 = lag(Traffic_Count, 2)) %>%
  ungroup()

# Split data into training and testing sets
train_data <- traffic_volume %>%
  filter(Date < "2024-01-01")
test_data <- traffic_volume %>%
  filter(Date >= "2024-01-01")

# Train predictive model (e.g., linear regression)
model <- lm(Traffic_Count ~ Lag_Traffic_Count_1 + Lag_Traffic_Count_2, data = train_data)

# Predict traffic count on test set
test_data$Predicted_Traffic_Count <- predict(model, newdata = test_data)

# Evaluate model performance (optional)
model_performance <- summary(model)

# Plot predicted vs. actual traffic count
ggplot(test_data, aes(x = Traffic_Count, y = Predicted_Traffic_Count)) +
  geom_point() +
  geom_abline(intercept = 0, slope = 1, color = "red", linetype = "dashed") +
  labs(x = "Actual Traffic Count", y = "Predicted Traffic Count", title = "Predicted vs. Actual Traffic
```

```
## Warning: Removed 50 rows containing missing values (`geom_point()`).
```


Predicted vs. Actual Traffic Count
Predicted Traffic Count
Actual Traffic Count
1.4
1.3
1.2
1.1
1.0
1.00
1.25
1.50
1.75
2.00