Chicago Crime Rate Prediction

1.Data Collection and Cleaning:

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.3.2
```

```
##
## 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
```

```
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 4.3.2
```

```
library(caret)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```
crime_data <- read.csv("C:/Users/satta/Downloads/crime data set git/Chicago_crime_dataset.csv")
weather_data<-read.csv("C:/Users/satta/Downloads/crime data set git/Temperature_dataset.csv")

crime_data$Date <- as.POSIXct(crime_data$Date, format = "%m/%d/%Y %H:%M")
crime_data$Date <- as.Date(crime_data$Date)
crime_data$Time <- format(crime_data$Date, "%H:%M:%S")
crime_data$Date <- format(crime_data$Date, "%m-%d-%Y")




#summary(crime_data,n=10)
head(weather_data,n=10)
```

	name <chr>	Date <chr>	temp <dbl>	dew <dbl>	humidity <dbl>	precip <dbl>	snow <dbl>
1	Chicago,United States	01-01-2021	-0.7	-3.2	82.8	7.045	0.0
2	Chicago,United States	02-01-2021	0.5	-1.7	85.2	0.000	1.1
3	Chicago,United States	03-01-2021	-0.2	-2.4	85.4	1.054	0.2
4	Chicago,United States	04-01-2021	-2.6	-4.3	88.3	0.000	0.0
5	Chicago,United States	05-01-2021	0.2	-2.4	82.4	0.000	0.0
6	Chicago,United States	06-01-2021	1.3	-2.3	77.3	0.000	0.0
7	Chicago,United States	07-01-2021	2.6	-3.8	63.0	0.000	0.0
8	Chicago,United States	08-01-2021	1.0	-3.8	70.3	0.267	0.0
9	Chicago,United States	09-01-2021	0.0	-5.4	67.4	0.000	0.0
10	Chicago,United States	10-01-2021	-2.0	-6.4	72.1	0.000	0.0

1-10 of 10 rows

```
crime_data <- left_join(crime_data, weather_data, by ="Date")
head(crime_data,n=10)
```

	ID <int>	Case.Number <chr>	Date <chr>	Block <chr>	I… <chr>	Primary.Type <chr>
1	12259050	JE100626	01-01-2021	057XX S DAMEN AVE	1310	CRIMINAL DAMAGE
2	12259424	JE100501	01-01-2021	062XX S MICHIGAN AVE	1320	CRIMINAL DAMAGE
3	12311821	JE164805	01-01-2021	031XX N RACINE AVE	0820	THEFT
4	12260063	JE101649	01-01-2021	031XX W POLK ST	1320	CRIMINAL DAMAGE
5	12259020	JE100698	01-01-2021	075XX S JEFFERY BLVD	141B	WEAPONS VIOLATION
6	12313377	JE166660	01-01-2021	058XX W 55TH ST	0820	THEFT
7	12268897	JE111949	01-01-2021	109XX S EMERALD AVE	1310	CRIMINAL DAMAGE
8	12259086	JE100744	01-01-2021	032XX W FILLMORE ST	0820	THEFT
9	12378360	JE245959	01-01-2021	002XX E ONTARIO ST	0810	THEFT
10	12261931	JE100377	01-01-2021	068XX S PERRY AVE	1310	CRIMINAL DAMAGE

1-10 of 10 rows | 1-7 of 30 columns

```
final <- crime_data %>%
  group_by(Date) %>%
  summarise(

    Temp = first(temp),
    Snow=first(snow),
    Humidity = first(humidity),  # Assuming humidity is constant for a given date
    Precip = first(precip),
    Crime_Count = n()
  )
crime_data <- left_join(crime_data, final, by ="Date")
head(crime_data,n=10)
```

	ID	Case.Number	Date	Block	I…	Primary.Type
	<int>	<chr>	<chr>	<chr>	<chr>	<chr>
1	12259050	JE100626	01-01-2021	057XX S DAMEN AVE	1310	CRIMINAL DAMAGE
2	12259424	JE100501	01-01-2021	062XX S MICHIGAN AVE	1320	CRIMINAL DAMAGE
3	12311821	JE164805	01-01-2021	031XX N RACINE AVE	0820	THEFT
4	12260063	JE101649	01-01-2021	031XX W POLK ST	1320	CRIMINAL DAMAGE
5	12259020	JE100698	01-01-2021	075XX S JEFFERY BLVD	141B	WEAPONS VIOLATION
6	12313377	JE166660	01-01-2021	058XX W 55TH ST	0820	THEFT
7	12268897	JE111949	01-01-2021	109XX S EMERALD AVE	1310	CRIMINAL DAMAGE
8	12259086	JE100744	01-01-2021	032XX W FILLMORE ST	0820	THEFT
9	12378360	JE245959	01-01-2021	002XX E ONTARIO ST	0810	THEFT
10	12261931	JE100377	01-01-2021	068XX S PERRY AVE	1310	CRIMINAL DAMAGE

1-10 of 10 rows | 1-7 of 35 columns

```
# Handle missing values
# For numerical columns, fill NA with the mean or median
crime_data <- crime_data %>% mutate(across(where(is.numeric), ~ ifelse(is.na(.), mean(., na.rm =
TRUE), .)))
missing_values <- is.na(crime_data)
```

2.Exploratory Data Analysis (EDA):

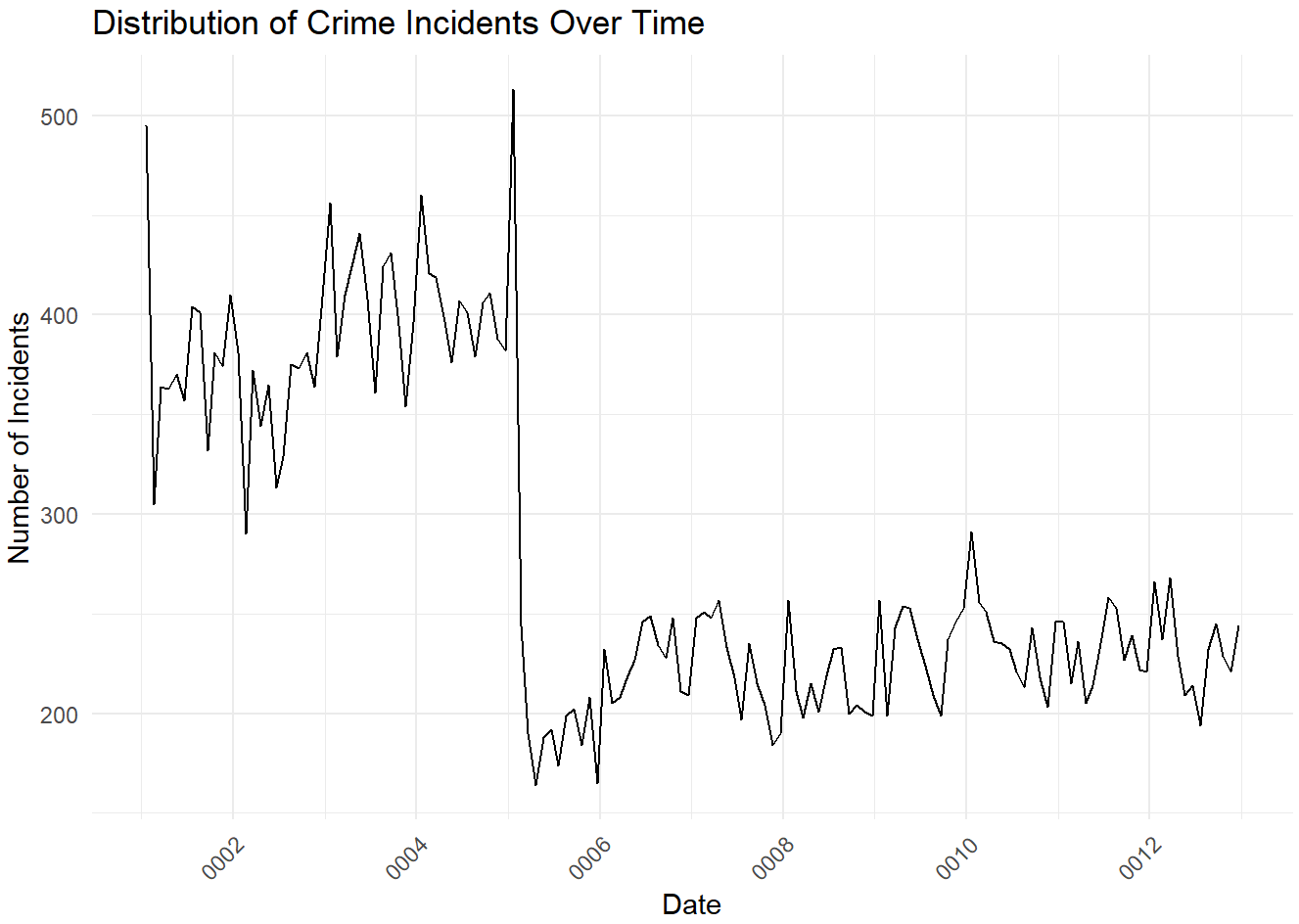
```
# Load the required libraries
library(tidyverse)
```

```
## Warning: package 'readr' was built under R version 4.3.2
```

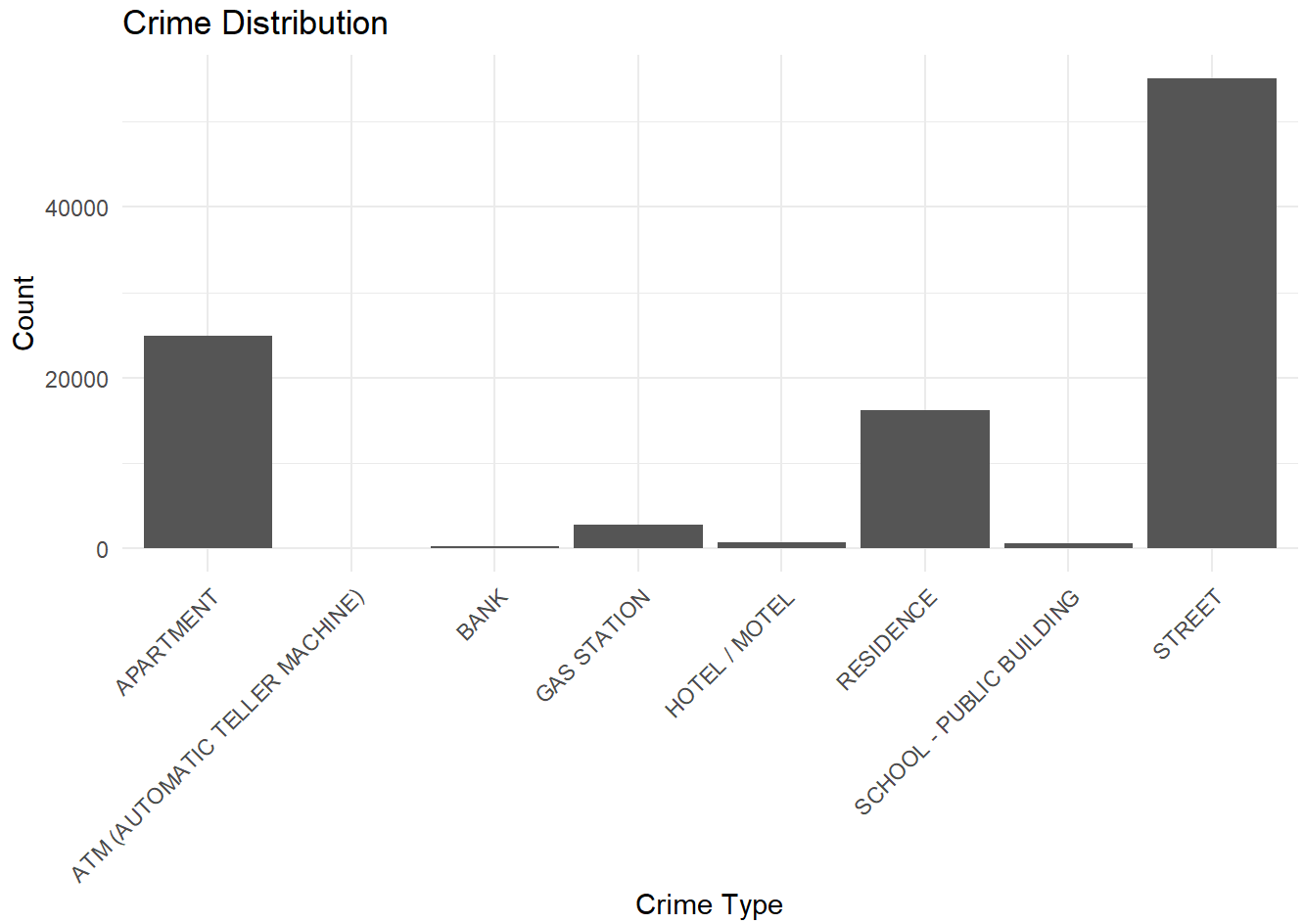
```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ forcats   1.0.0     ✓ readr     2.1.4
## ✓ lubridate 1.9.3     ✓ stringr   1.5.0
## ✓ purrr     1.0.2     ✓ tibble    3.2.1
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ✖ purrr::lift()   masks caret::lift()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
e errors
```

```
library(ggplot2)

# Convert the 'Date' column to datetime
crime_data$Date <- as.Date(crime_data$Date)

# Set the plot style
theme_set(theme_minimal())

# Plotting the distribution of incidents over time
ggplot(crime_data, aes(x = Date)) +
  geom_line(stat = 'count') +
  labs(title = 'Distribution of Crime Incidents Over Time',
       x = 'Date',
       y = 'Number of Incidents') +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

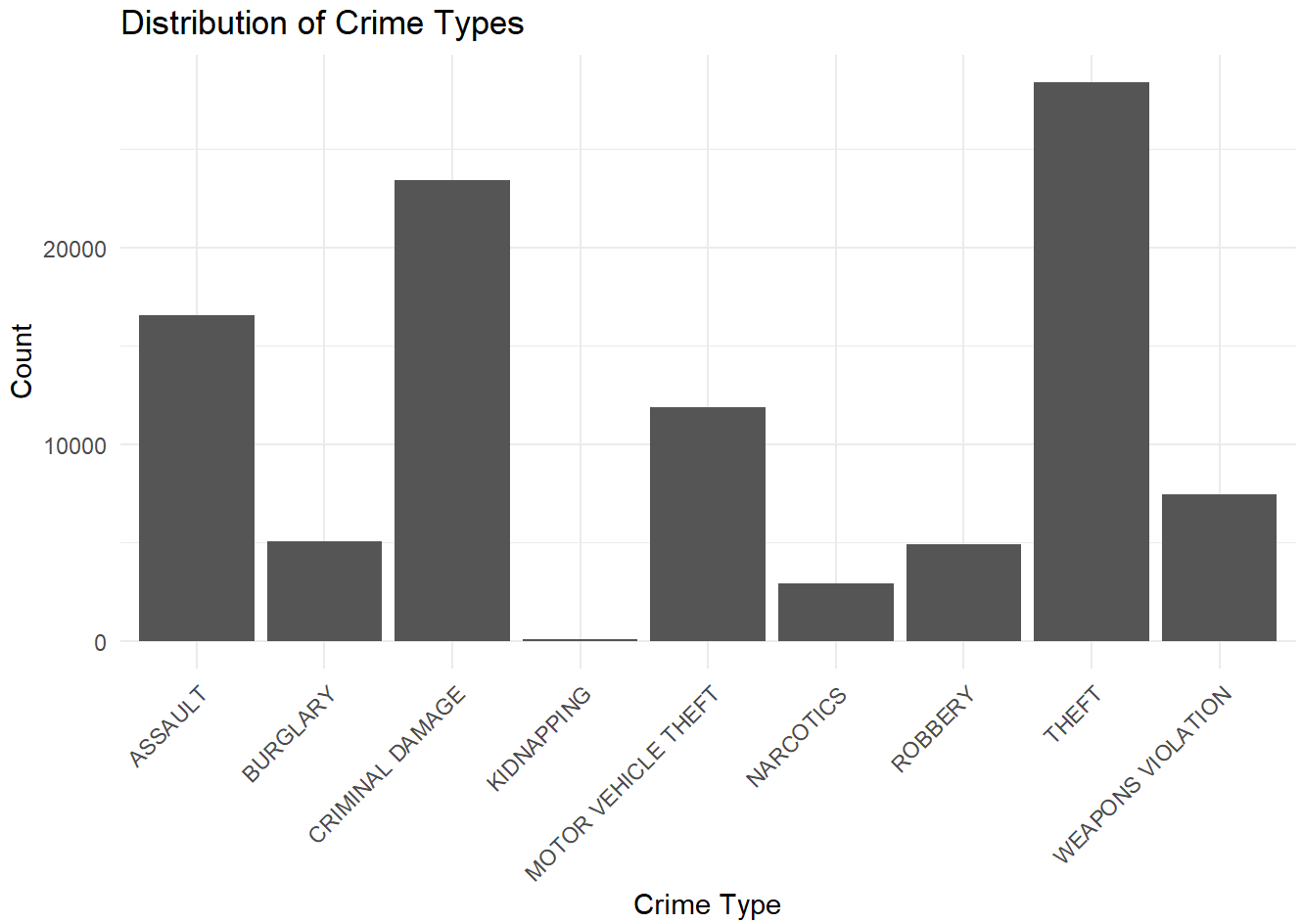
```
## Warning: Removed 60176 rows containing non-finite values (`stat_count()`).
```



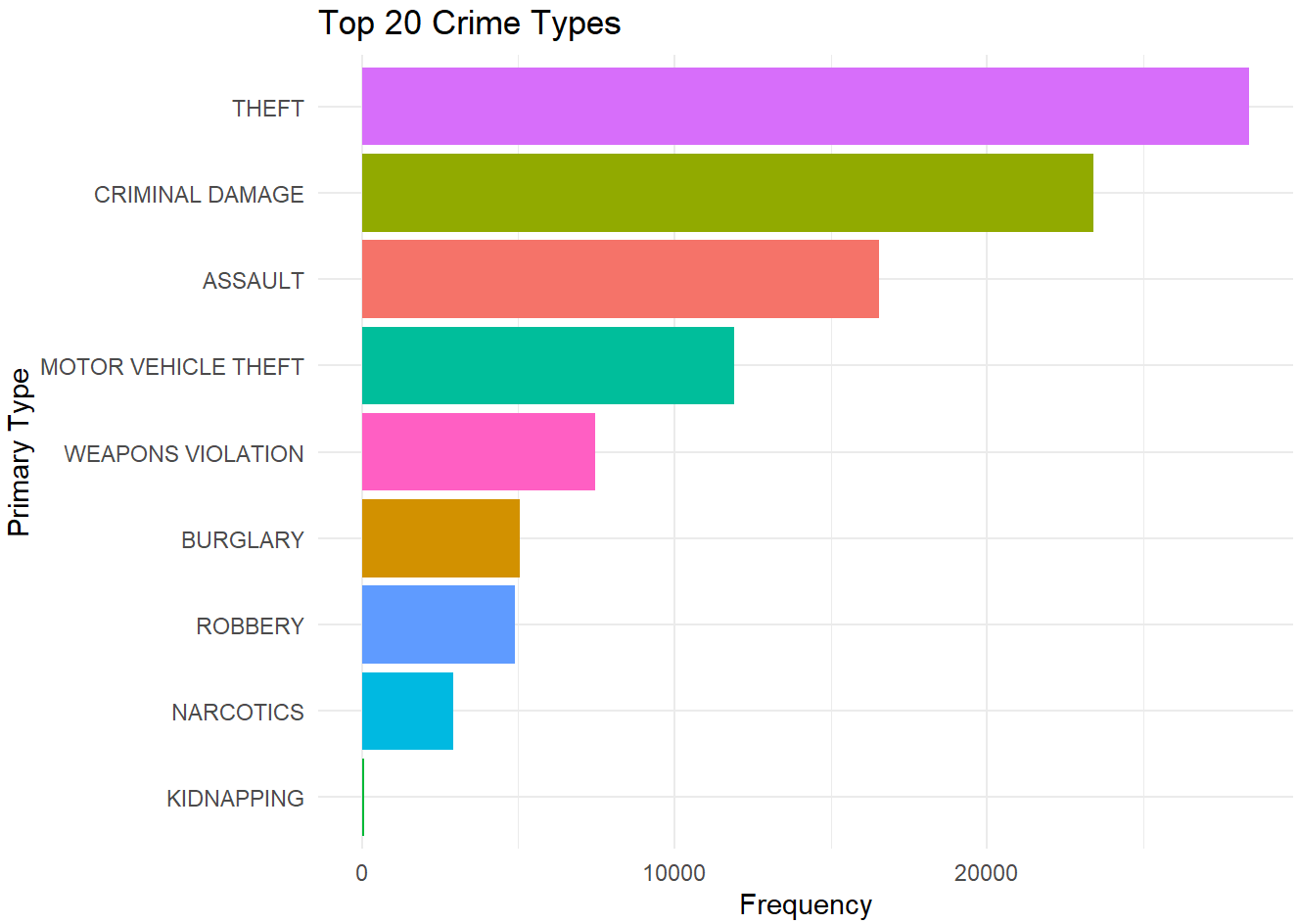

```
# Install and load necessary libraries if not already installed
if (!requireNamespace("ggplot2", quietly = TRUE)) {
  install.packages("ggplot2")
}
library(ggplot2)

# Assuming your dataset is named crime_data
# Create a bar plot of crime distribution
ggplot(crime_data, aes(x = Location.Description)) +
  geom_bar() +
  labs(title = "Crime Distribution",
       x = "Crime Type",
       y = "Count") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))  # Rotate x-axis labels for better re
adability
```



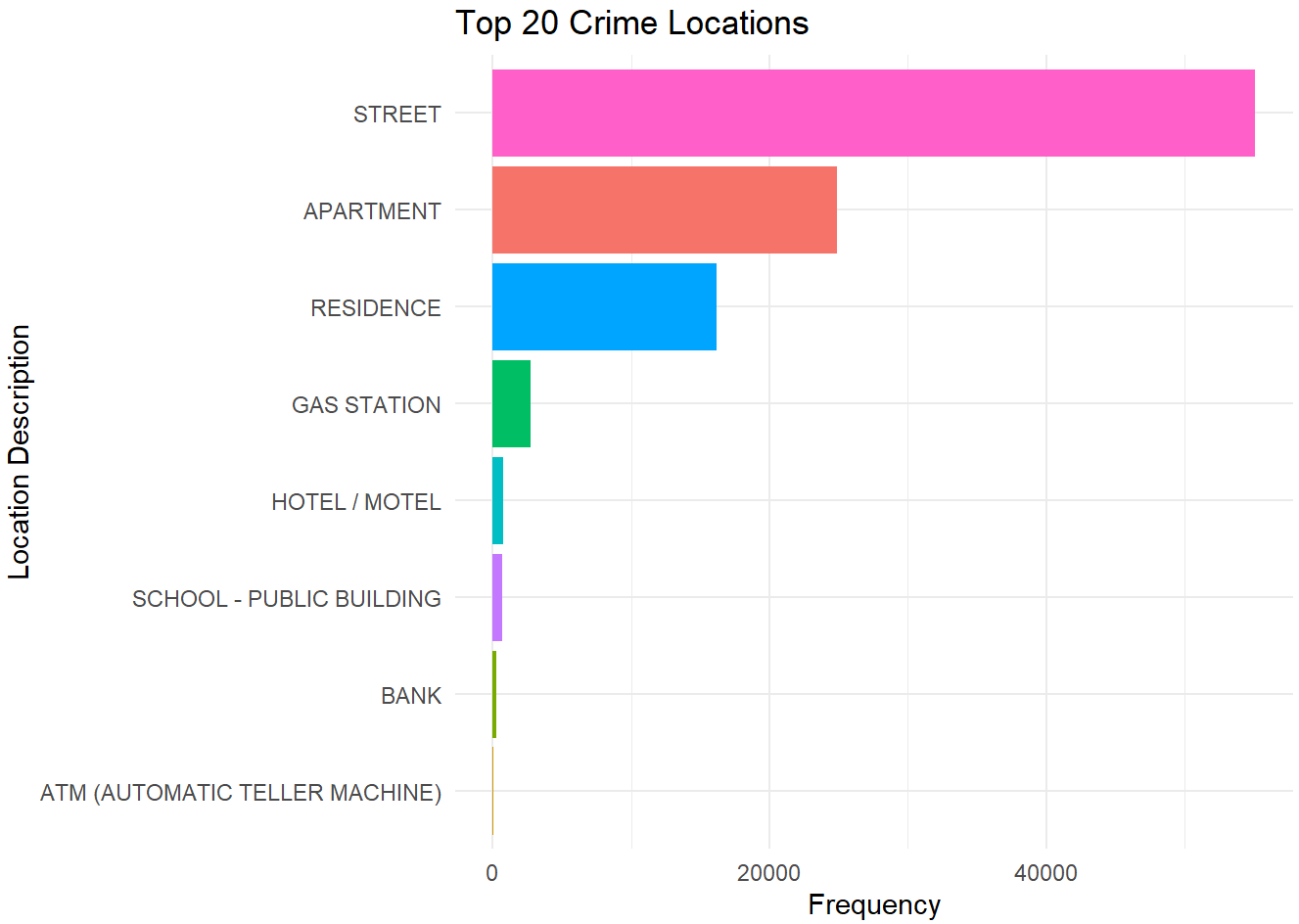

```
# Distribution of crime types
ggplot(crime_data, aes(x = Primary.Type)) +
  geom_bar() +
  labs(title = "Distribution of Crime Types",
       x = "Crime Type",
       y = "Count") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




```
# Load the required libraries
library(tidyverse)

# Plotting the distribution of Primary Type and Location Description
par(mfrow=c(2,1), mar=c(4,5,4,2))

# Plot for Primary Type
crime_data %>%
  count(Primary.Type) %>%
  arrange(desc(n)) %>%
  slice_head(n = 20) %>%
  ggplot(aes(y = n, x = reorder(Primary.Type, n), fill = Primary.Type)) +
  geom_bar(stat = 'identity') +
  labs(title = 'Top 20 Crime Types',
       y = 'Frequency',
       x = 'Primary Type') +
  theme_minimal() +
  theme(legend.position = 'none') +
  coord_flip()
```

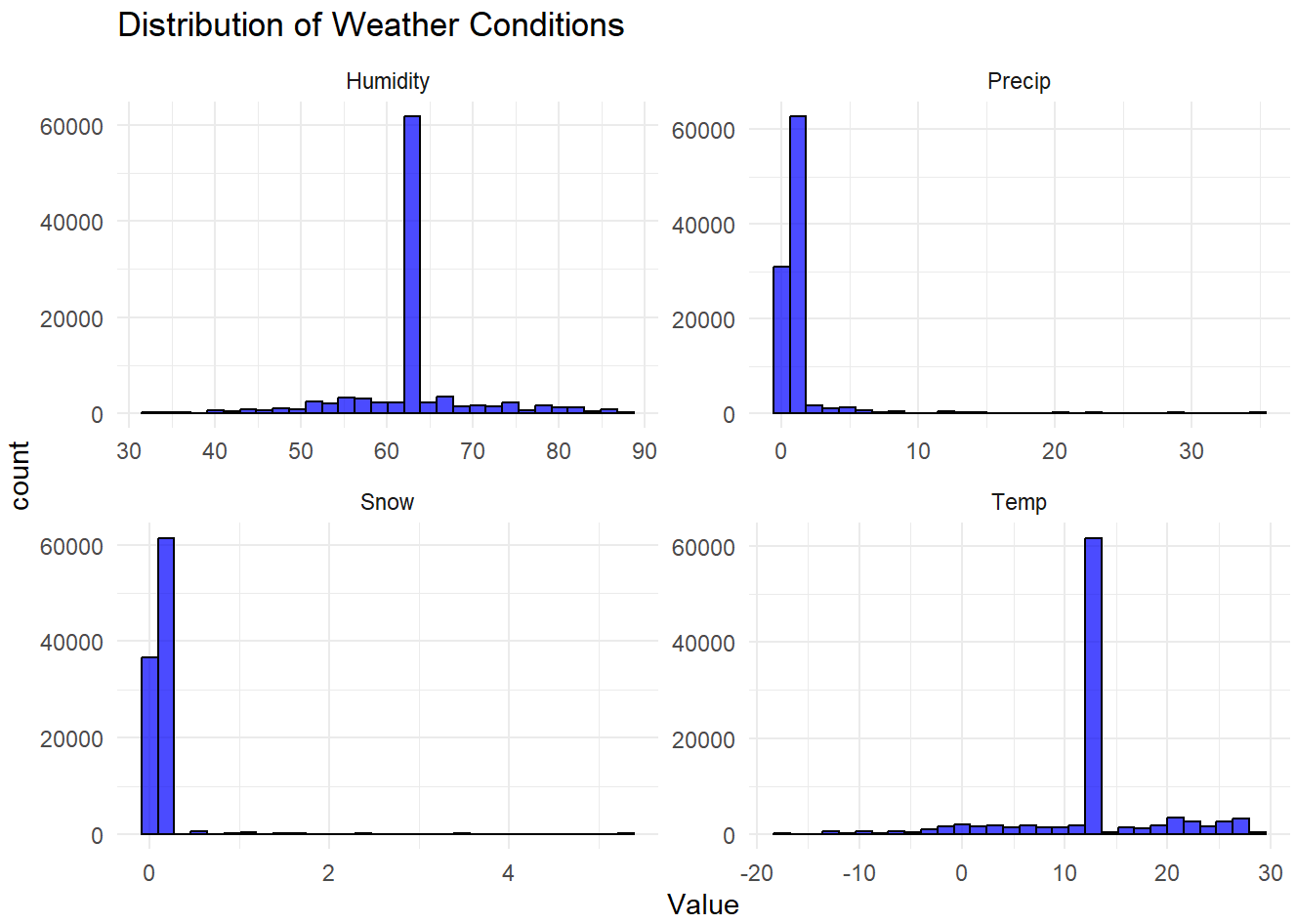



```
# Plot for Location Description
crime_data %>%
  count(Location.Description) %>%
  arrange(desc(n)) %>%
  slice_head(n = 20) %>%
  ggplot(aes(y = n, x = reorder(Location.Description, n), fill = Location.Description)) +
  geom_bar(stat = 'identity') +
  labs(title = 'Top 20 Crime Locations',
       y = 'Frequency',
       x = 'Location Description') +
  theme_minimal() +
  theme(legend.position = 'none') +
  coord_flip()
```




```
# Weather conditions
weather_cols <- c('Temp', 'Snow', 'Humidity', 'Precip')

crime_data %>%
  gather(key = 'Weather', value = 'Value', weather_cols) %>%
  ggplot(aes(x = Value)) +
  geom_histogram(bins = 30, fill = 'blue', color = 'black', alpha = 0.7) +
  facet_wrap(~Weather, scales = 'free') +
  labs(title = 'Distribution of Weather Conditions')
```

```
## Warning: Using an external vector in selections was deprecated in tidyselect 1.1.0.
## ℹ Please use `all_of()` or `any_of()` instead.
##   # Was:
##   data %>% select(weather_cols)
##
##   # Now:
##   data %>% select(all_of(weather_cols))
##
## See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```




```
# Install and load necessary libraries if not already installed
if (!requireNamespace("tidyverse", quietly = TRUE)) {
  install.packages("tidyverse")
}
library(tidyverse)



# Check the structure of the dataset
str(crime_data)
```

```
## 'data.frame':    100562 obs. of  34 variables:
##  $ ID                  : int  12259050 12259424 12311821 12260063 12259020 12313377 12268897
12259086 12378360 12261931 ...
##  $ Case.Number         : chr  "JE100626" "JE100501" "JE164805" "JE101649" ...
##  $ Date                : Date, format: "0001-01-20" "0001-01-20" ...
##  $ Block               : chr  "057XX S DAMEN AVE" "062XX S MICHIGAN AVE" "031XX N RACINE AVE"
"031XX W POLK ST" ...
##  $ IUCR                : chr  "1310" "1320" "0820" "1320" ...
##  $ Primary.Type        : chr  "CRIMINAL DAMAGE" "CRIMINAL DAMAGE" "THEFT" "CRIMINAL DAMAGE"
...
##  $ Description         : chr  "TO PROPERTY" "TO VEHICLE" "$500 AND UNDER" "TO VEHICLE" ...
##  $ Location.Description: chr  "APARTMENT" "STREET" "APARTMENT" "STREET" ...
##  $ Arrest              : chr  "false" "false" "false" "false" ...
##  $ Domestic            : chr  "false" "false" "false" "false" ...
##  $ Beat                : int  715 311 1933 1134 414 811 2233 1134 1834 722 ...
##  $ District            : int  7 3 19 11 4 8 22 11 18 7 ...
##  $ Ward                : num  15 20 32 24 8 23 34 24 42 6 ...
##  $ Community.Area      : int  67 40 6 27 43 56 49 29 8 69 ...
##  $ FBI.Code            : chr  "14" "14" "06" "14" ...
##  $ X.Coordinate        : num  1164009 1178263 1167730 1155633 1190847 ...
##  $ Y.Coordinate        : num  1866506 1863570 1921189 1896202 1855361 ...
##  $ Year                : int  2021 2021 2021 2021 2021 2021 2021 2021 2021 2021 ...
##  $ Updated.On          : chr  "01/16/2021 03:39:23 PM" "01/16/2021 03:39:23 PM" "03/12/2021 0
3:39:32 PM" "01/16/2021 03:39:23 PM" ...
##  $ Latitude            : num  41.8 41.8 41.9 41.9 41.8 ...
##  $ Longitude           : num  -87.7 -87.6 -87.7 -87.7 -87.6 ...
##  $ Location            : chr  "(41.789314851, -87.674170888)" "(41.780946398, -87.621995369)"
"(41.939290467, -87.658952119)" "(41.870976478, -87.70408564)" ...
##  $ Time                : chr  "00:00:00" "00:00:00" "00:00:00" "00:00:00" ...
##  $ name                : chr  "Chicago,United States" "Chicago,United States" "Chicago,United
States" "Chicago,United States" ...
##  $ temp                : num  -0.7 -0.7 -0.7 -0.7 -0.7 -0.7 -0.7 -0.7 -0.7 -0.7 ...
##  $ dew                 : num  -3.2 -3.2 -3.2 -3.2 -3.2 -3.2 -3.2 -3.2 -3.2 -3.2 ...
##  $ humidity            : num  82.8 82.8 82.8 82.8 82.8 82.8 82.8 82.8 82.8 82.8 ...
##  $ precip              : num  7.04 7.04 7.04 7.04 7.04 ...
##  $ snow                : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Temp                : num  -0.7 -0.7 -0.7 -0.7 -0.7 -0.7 -0.7 -0.7 -0.7 -0.7 ...
##  $ Snow                : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Humidity            : num  82.8 82.8 82.8 82.8 82.8 82.8 82.8 82.8 82.8 82.8 ...
##  $ Precip              : num  7.04 7.04 7.04 7.04 7.04 ...
##  $ Crime_Count         : int  228 228 228 228 228 228 228 228 228 228 ...
```

```
# Summary statistics
summary(crime_data)
```

```
##        ID           Case.Number             Date                Block
##  Min.   :12258517   Length:100562      Min.   :0001-01-20   Length:100562
##  1st Qu.:12370192   Class :character   1st Qu.:0003-03-27   Class :character
##  Median :12479306   Mode  :character   Median :0005-07-20   Mode  :character
##  Mean   :12477579                      Mean   :0006-04-14
##  3rd Qu.:12582888                      3rd Qu.:0009-05-20
##  Max.   :13275887                      Max.   :0012-12-20
##                                        NA's   :60176
##      IUCR           Primary.Type       Description        Location.Description
##  Length:100562      Length:100562      Length:100562      Length:100562
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##    Arrest            Domestic              Beat         District
##  Length:100562      Length:100562      Min.   : 111   Min.   : 1.00
##  Class :character   Class :character   1st Qu.: 532   1st Qu.: 5.00
##  Mode  :character   Mode  :character   Median :1011   Median :10.00
##                                        Mean   :1112   Mean   :10.89
##                                        3rd Qu.:1622   3rd Qu.:16.00
##                                        Max.   :2535   Max.   :31.00
##
##       Ward       Community.Area   FBI.Code          X.Coordinate
##  Min.   : 1.00   Min.   : 1.0   Length:100562      Min.   :1095509
##  1st Qu.: 9.00   1st Qu.:24.0   Class :character   1st Qu.:1153969
##  Median :21.00   Median :37.0   Mode  :character   Median :1166956
##  Mean   :21.94   Mean   :38.6                      Mean   :1165896
##  3rd Qu.:32.00   3rd Qu.:58.0                      3rd Qu.:1177433
##  Max.   :50.00   Max.   :77.0                      Max.   :1205119
##
##   Y.Coordinate          Year       Updated.On           Latitude
##  Min.   :1813909   Min.   :2021   Length:100562      Min.   :41.64
##  1st Qu.:1856623   1st Qu.:2021   Class :character   1st Qu.:41.76
##  Median :1882922   Median :2021   Mode  :character   Median :41.83
##  Mean   :1882922   Mean   :2021                      Mean   :41.83
##  3rd Qu.:1906649   3rd Qu.:2021                      3rd Qu.:41.90
##  Max.   :1951493   Max.   :2022                      Max.   :42.02
##
##    Longitude        Location             Time               name
##  Min.   :-87.92   Length:100562      Length:100562      Length:100562
##  1st Qu.:-87.71   Class :character   Class :character   Class :character
##  Median :-87.66   Mode  :character   Mode  :character   Mode  :character
##  Mean   :-87.67
##  3rd Qu.:-87.62
##  Max.   :-87.52
##
##       temp             dew            humidity         precip
##  Min.   :-17.60   Min.   :-24.40   Min.   :33.00   Min.   : 0.000
##  1st Qu.: 12.06   1st Qu.:  4.63   1st Qu.:62.87   1st Qu.: 0.000
##  Median : 12.06   Median :  4.63   Median :62.87   Median : 1.566
```

```
##  Mean   : 12.06   Mean   :  4.63   Mean   :62.87   Mean   : 1.566
##  3rd Qu.: 12.06   3rd Qu.:  4.63   3rd Qu.:62.87   3rd Qu.: 1.566
##  Max.   : 28.70   Max.   : 22.50   Max.   :88.30   Max.   :34.776
##
##       snow             Temp             Snow           Humidity
##  Min.   :0.0000   Min.   :-17.60   Min.   :0.0000   Min.   :33.00
##  1st Qu.:0.0000   1st Qu.: 12.06   1st Qu.:0.0000   1st Qu.:62.87
##  Median :0.1068   Median : 12.06   Median :0.1068   Median :62.87
##  Mean   :0.1068   Mean   : 12.06   Mean   :0.1068   Mean   :62.87
##  3rd Qu.:0.1068   3rd Qu.: 12.06   3rd Qu.:0.1068   3rd Qu.:62.87
##  Max.   :5.3000   Max.   : 28.70   Max.   :5.3000   Max.   :88.30
##
##      Precip        Crime_Count
##  Min.   : 0.000   Min.   : 12.0
##  1st Qu.: 0.000   1st Qu.:192.0
##  Median : 1.566   Median :212.0
##  Mean   : 1.566   Mean   :210.9
##  3rd Qu.: 1.566   3rd Qu.:232.0
##  Max.   :34.776   Max.   :291.0
##
```

```
# Missing values
missing_values <- colSums(is.na(crime_data))
print("Missing Values:")
```

```
## [1] "Missing Values:"
```

```
print(missing_values[missing_values > 0])
```

```
##  Date  Time  name
## 60176    12 60176
```

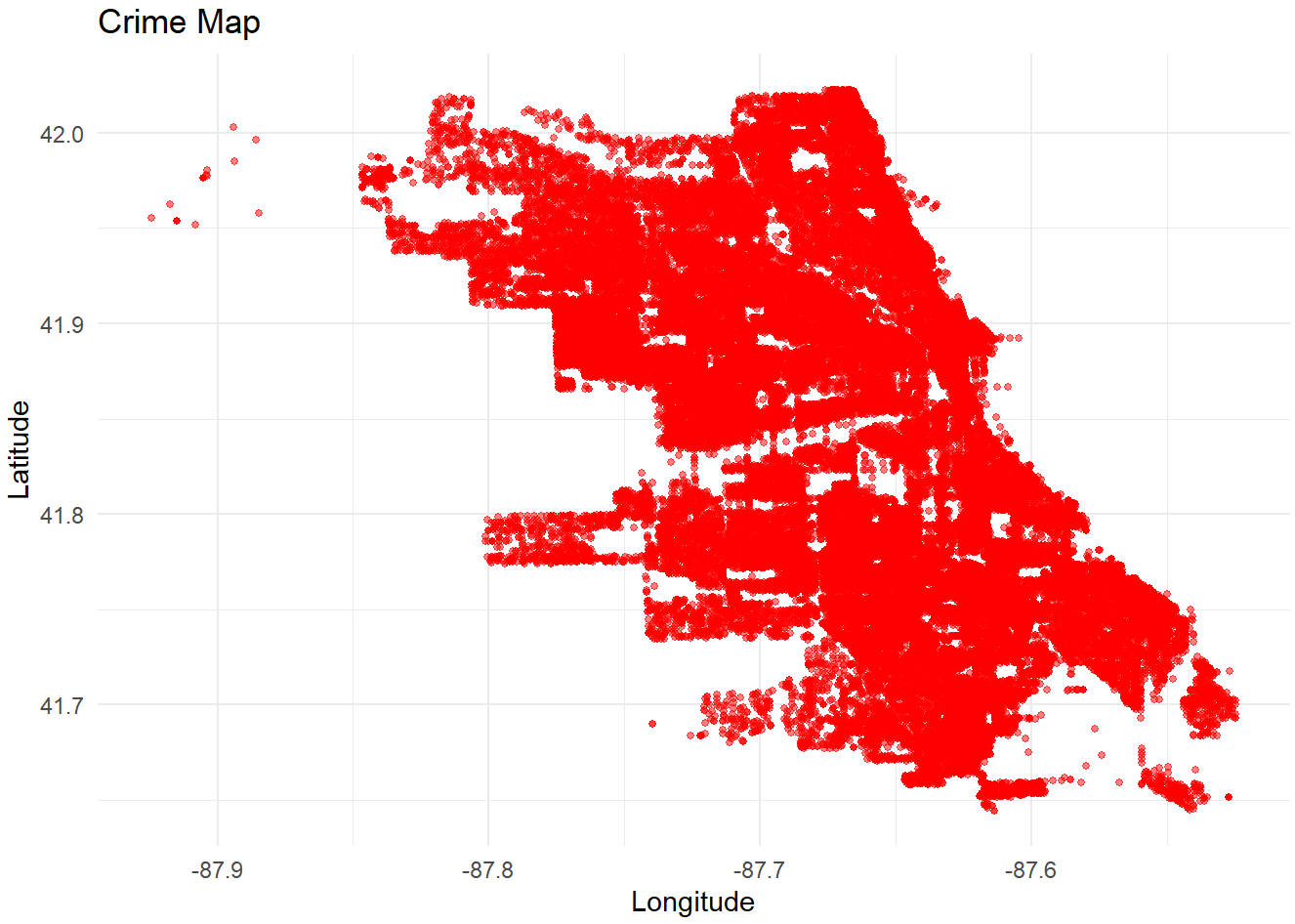
```
# Unique values in categorical columns
print("Unique Values in Categorical Columns:")
```

```
## [1] "Unique Values in Categorical Columns:"
```

```
sapply(crime_data[, sapply(crime_data, is.factor)], function(x) length(unique(x)))
```

```
## named list()
```

```
# Distribution of crime counts by month
crime_data$Date <- as.Date(crime_data$Date, format = "%m-%d-%Y")
crime_data$Month <- format(crime_data$Date, "%Y-%m")



# Visualize geographic patterns
crime_map <- ggplot(crime_data, aes(x = Longitude, y = Latitude)) +
  geom_point(alpha = 0.5, size = 1, color = "red") +
  ggtitle("Crime Map")

print(crime_map)
```



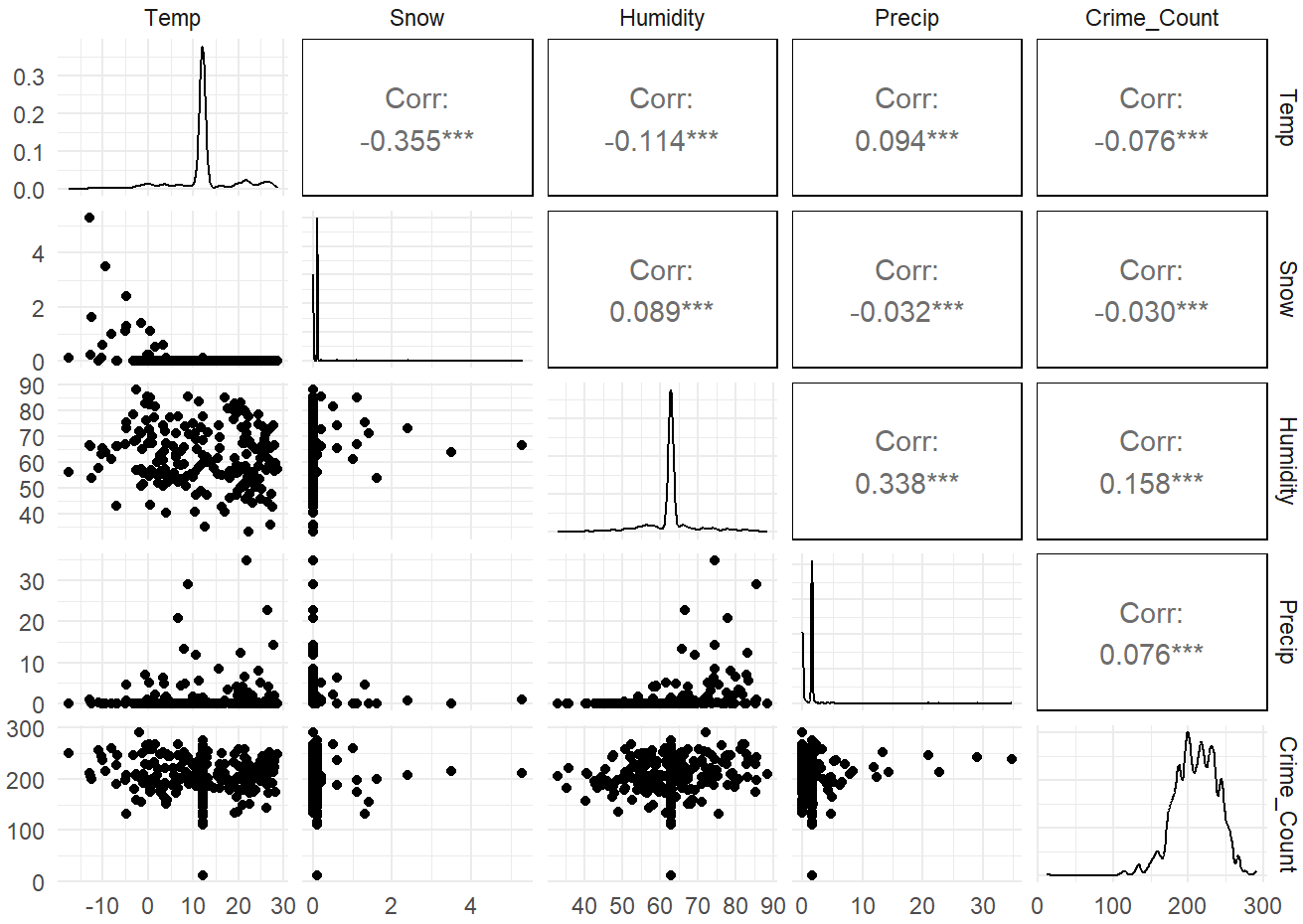

```
# Correlation matrix
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
weather_data <- crime_data[, c("Temp", "Snow", "Humidity", "Precip", "Crime_Count")]
cor(weather_data)
```

```
##                    Temp        Snow    Humidity      Precip Crime_Count
## Temp         1.00000000 -0.35520188 -0.11443576  0.09357598 -0.07634935
## Snow        -0.35520188  1.00000000  0.08861045 -0.03233319 -0.03049506
## Humidity    -0.11443576  0.08861045  1.00000000  0.33806329  0.15774452
## Precip       0.09357598 -0.03233319  0.33806329  1.00000000  0.07587110
## Crime_Count -0.07634935 -0.03049506  0.15774452  0.07587110  1.00000000
```

```
ggpairs(crime_data,columns=c("Temp", "Snow", "Humidity", "Precip", "Crime_Count"))
```

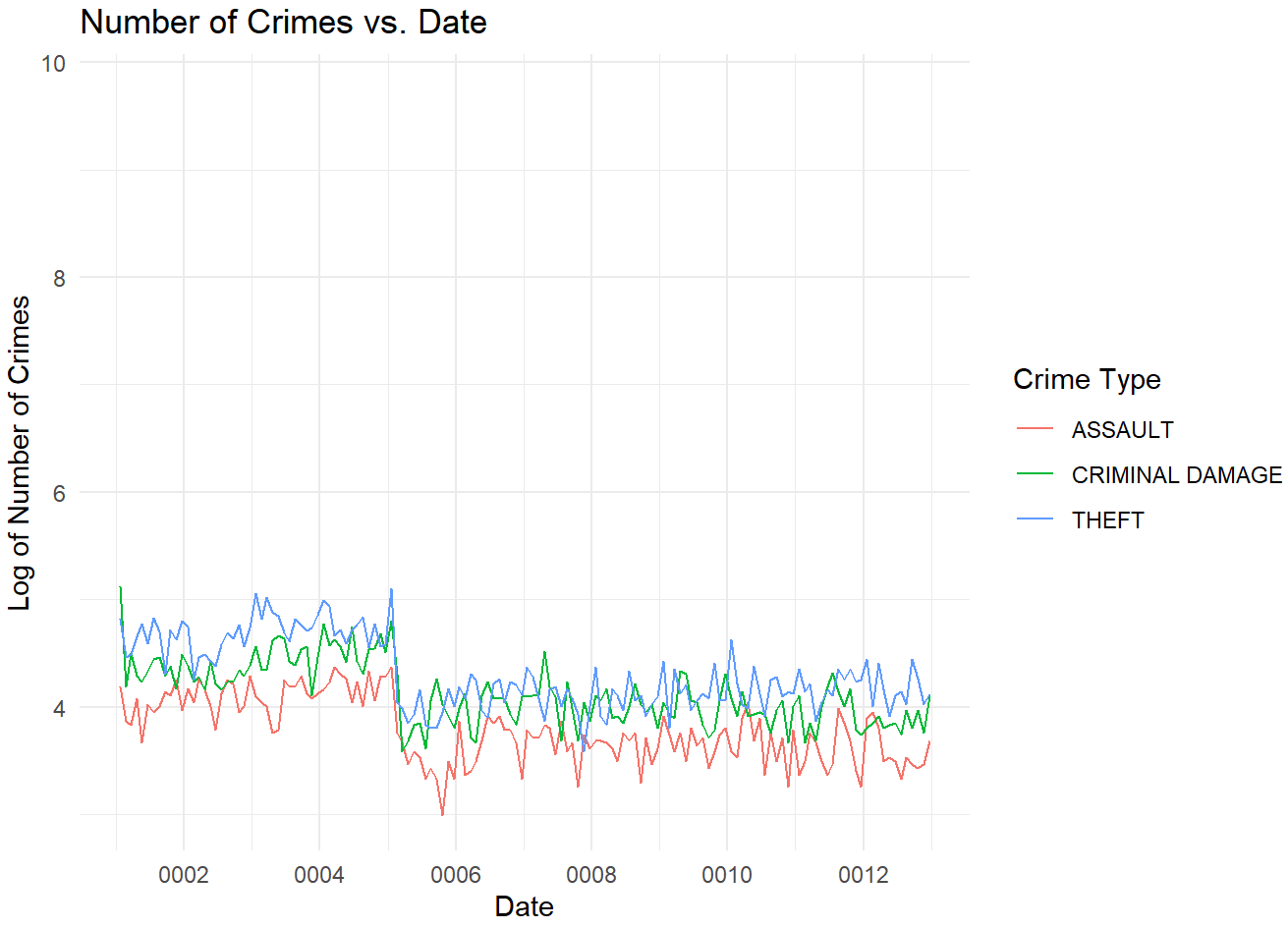



```
# Load necessary libraries
library(tidyverse)

# Convert the 'Date' column to a Date type
crime_data$Date <- as.Date(crime_data$Date, format = "%m-%d-%Y")

# Filter for specific crime types (theft, criminal damage, assault)
filtered_data <- crime_data %>%
  filter(Primary.Type %in% c("THEFT", "CRIMINAL DAMAGE", "ASSAULT"))

# Aggregate the data by date and crime type
crime_counts <- filtered_data %>%
  group_by(Date, Primary.Type) %>%
  summarise(CrimeCount = n())
```

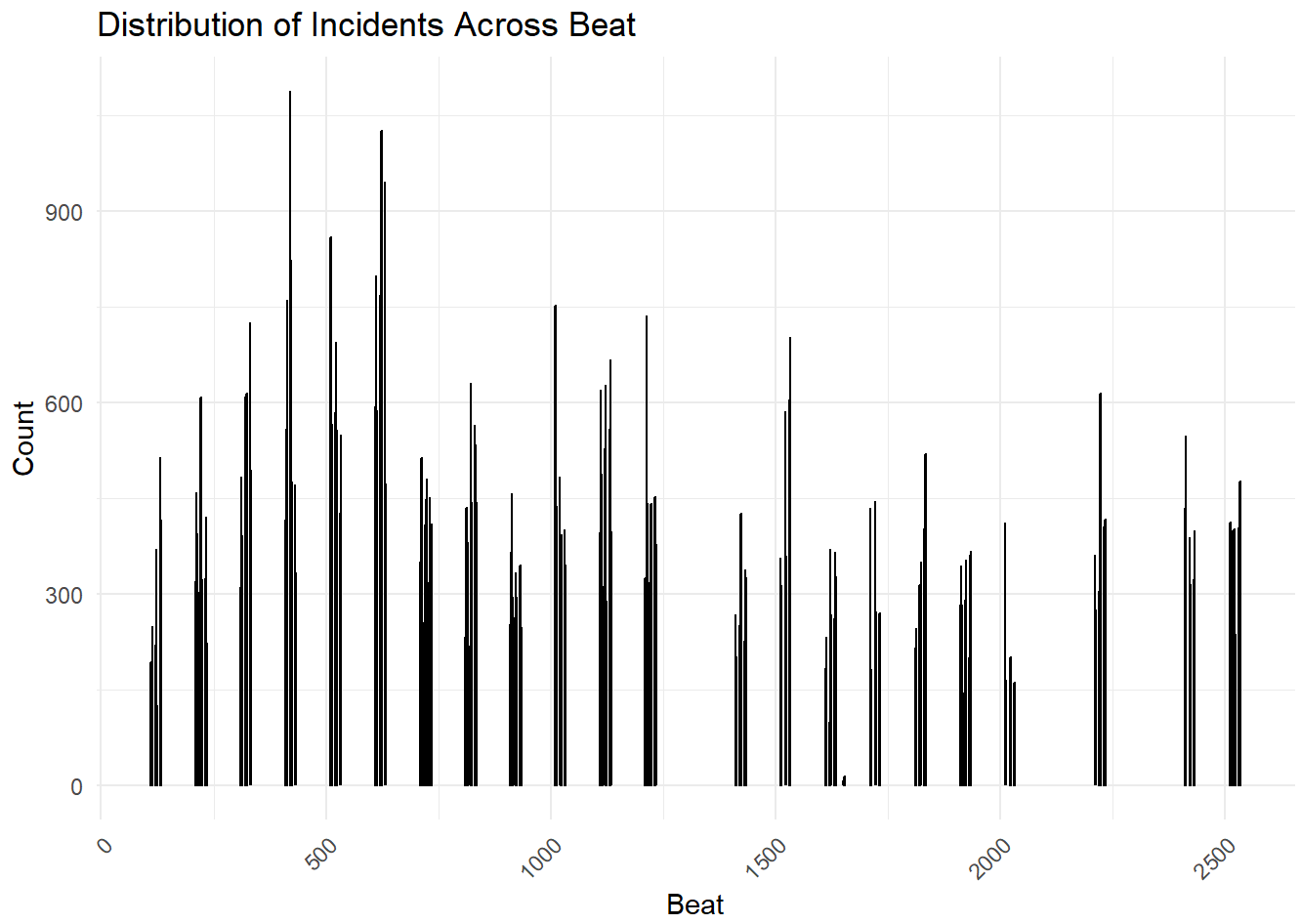
```
## `summarise()` has grouped output by 'Date'. You can override using the
## `.groups` argument.
```

```
# Plot the graph with log scale on the y-axis
ggplot(crime_counts, aes(x = Date, y = log(CrimeCount), color = Primary.Type)) +
  geom_line() +
  labs(title = "Number of Crimes vs. Date",
       x = "Date",
       y = "Log of Number of Crimes",
       color = "Crime Type") +
  scale_y_continuous(labels = scales::comma) +  # Add comma separator for y-axis labels
  theme_minimal()
```

```
## Warning: Removed 3 rows containing missing values (`geom_line()`).
```




```
# Plot the distribution of incidents across Beat
ggplot(crime_data, aes(x = Beat)) +
  geom_bar(fill = "skyblue", color = "black") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Distribution of Incidents Across Beat", x = "Beat", y = "Count")
```




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

# Define the public holidays
public_holidays_data <- tibble(
  Date = as.Date(c(
    "2021-01-01", "2021-01-18", "2021-02-15", "2021-05-31", "2021-07-05",
    "2021-09-06", "2021-10-11", "2021-11-11", "2021-11-25", "2021-12-25",
    "2022-01-01", "2022-01-17", "2022-02-21", "2022-05-30", "2022-07-04",
    "2022-09-05", "2022-10-10", "2022-11-11", "2022-11-24", "2022-12-26"
  )),
  Holiday = c(
    "New Year's Day", "Martin Luther King Jr. Day", "Presidents' Day",
    "Memorial Day", "Independence Day", "Labor Day", "Columbus Day",
    "Veterans Day", "Thanksgiving Day", "Christmas Day",
    "New Year's Day", "Martin Luther King Jr. Day", "Presidents' Day",
    "Memorial Day", "Independence Day", "Labor Day", "Columbus Day",
    "Veterans Day", "Thanksgiving Day", "Christmas Day (Observed)"
  )
)

# Display the public holidays dataframe
print(public_holidays_data)
```

```
## # A tibble: 20 × 2
##    Date       Holiday
##    <date>     <chr>
##  1 2021-01-01 New Year's Day
##  2 2021-01-18 Martin Luther King Jr. Day
##  3 2021-02-15 Presidents' Day
##  4 2021-05-31 Memorial Day
##  5 2021-07-05 Independence Day
##  6 2021-09-06 Labor Day
##  7 2021-10-11 Columbus Day
##  8 2021-11-11 Veterans Day
##  9 2021-11-25 Thanksgiving Day
## 10 2021-12-25 Christmas Day
## 11 2022-01-01 New Year's Day
## 12 2022-01-17 Martin Luther King Jr. Day
## 13 2022-02-21 Presidents' Day
## 14 2022-05-30 Memorial Day
## 15 2022-07-04 Independence Day
## 16 2022-09-05 Labor Day
## 17 2022-10-10 Columbus Day
## 18 2022-11-11 Veterans Day
## 19 2022-11-24 Thanksgiving Day
## 20 2022-12-26 Christmas Day (Observed)
```

3.MODELING: Given the nature of your project, I assume you might be interested in predicting the Crime_Count based on other variables like Temp, Snow, Humidity, Precip, and possibly time-related variables (like the date or year).

Linear Regression: To model the relationship between Crime_Count and other independent variables using a linear approach. Random Forest Regression: To model the same relationship but using a non-linear, ensemble-based approach.

a. Linear Regression in R For linear regression, you can use the lm() function in R. Here's an example of how you might set up a linear regression model to predict Crime_Count based on certain variables:

```
library(readr)
head(crime_data)
```

	ID	Case.Number	Date	Block	IU…	Primary.Type
	<int>	<chr>	<date>	<chr>	<chr>	<chr>
1	12259050	JE100626	0001-01-20	057XX S DAMEN AVE	1310	CRIMINAL DAMAGE
2	12259424	JE100501	0001-01-20	062XX S MICHIGAN AVE	1320	CRIMINAL DAMAGE
3	12311821	JE164805	0001-01-20	031XX N RACINE AVE	0820	THEFT
4	12260063	JE101649	0001-01-20	031XX W POLK ST	1320	CRIMINAL DAMAGE
5	12259020	JE100698	0001-01-20	075XX S JEFFERY BLVD	141B	WEAPONS VIOLATION
6	12313377	JE166660	0001-01-20	058XX W 55TH ST	0820	THEFT

6 rows | 1-7 of 36 columns

```
set.seed(42) # For reproducibility
trainIndex <- createDataPartition(crime_data$Crime_Count, p = 0.8,
                                  list = FALSE,
                                  times = 1)
dataTrain <- crime_data[trainIndex, ]
dataTest <- crime_data[-trainIndex, ]

# Model training
model <- lm(lag(Crime_Count) ~ Temp + Snow + Humidity + Precip, data = dataTrain)
summary(model)
```

```
##
## Call:
## lm(formula = lag(Crime_Count) ~ Temp + Snow + Humidity + Precip,
##     data = dataTrain)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -198.856  -19.299    2.144   20.185   74.649
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 179.92903    0.93613 192.206   <2e-16 ***
## Temp         -0.34989    0.01452 -24.100   <2e-16 ***
## Snow         -6.08057    0.30672 -19.825   <2e-16 ***
## Humidity      0.56160    0.01448  38.791   <2e-16 ***
## Precip        0.31075    0.03506   8.863   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 27.45 on 80445 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.03478,    Adjusted R-squared:  0.03473
## F-statistic: 724.7 on 4 and 80445 DF,  p-value: < 2.2e-16
```

```
lr_predictions <- predict(model, dataTest)
mse <- mean((lr_predictions - dataTest$Crime_Count)^2)
rsq <- summary(model)$r.squared

# Output the MSE and R-squared
print(paste("Mean Squared Error:", mse))
```

```
## [1] "Mean Squared Error: 749.621826698263"
```

```
print(paste("R-squared:", rsq))
```

```
## [1] "R-squared: 0.0347806691210506"
```

Mean Absolute Error (MAE): It measures the average absolute differences between the predicted values and the actual values. Smaller MAE values indicate better model accuracy.

Root Mean Squared Error (RMSE): It is similar to MAE but gives more weight to large errors. RMSE is the square root of the mean of the squared differences between predicted and actual values. Like MAE, lower RMSE values indicate better model accuracy.

```
# Ensure the Date column is correctly formatted and free of NAs
crime_data <- crime_data %>%
  mutate(Date = as.Date(Date)) %>%
  filter(!is.na(Date), !is.na(Crime_Count))

# Checking and printing minimum date values
min_year <- year(min(crime_data$Date))
min_month <- month(min(crime_data$Date))
print(paste("Using start year:", min_year, "and start month:", min_month))
```

```
## [1] "Using start year: 1 and start month: 1"
```

```
# Assuming monthly data
ts_data <- ts(crime_data$Crime_Count, start = c(min_year, min_month), frequency = 12)
```

b. ARIMA Model

```
library(readr)
library(dplyr)
library(lubridate)
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 4.3.2
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(ggplot2)

# Ensure the Date column is a Date type
crime_data$Date <- as.Date(crime_data$Date)

# Check for any NA in Date or Crime_Count
sum(is.na(crime_data$Date))
```

```
## [1] 0
```

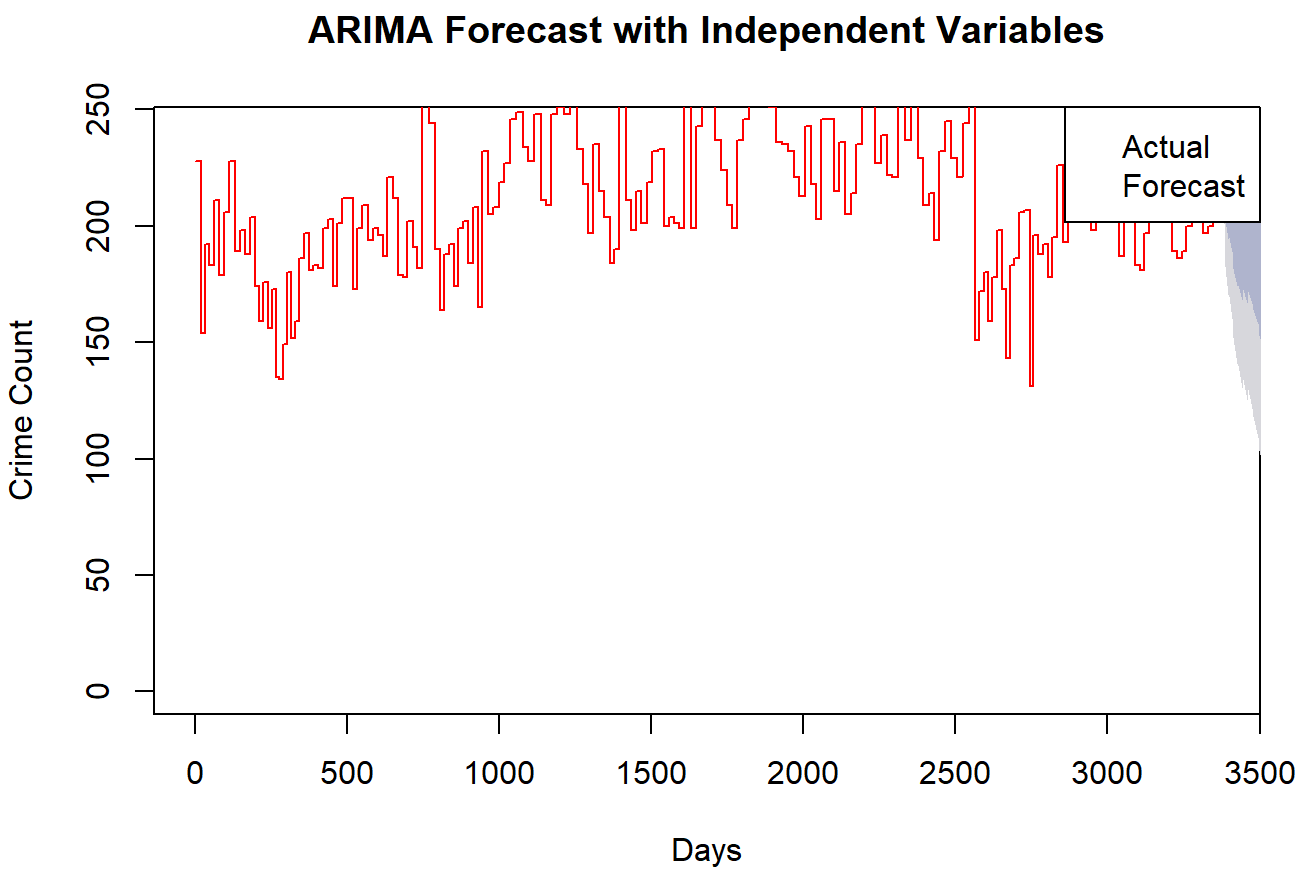
```
sum(is.na(crime_data$Crime_Count))
```

```
## [1] 0
```

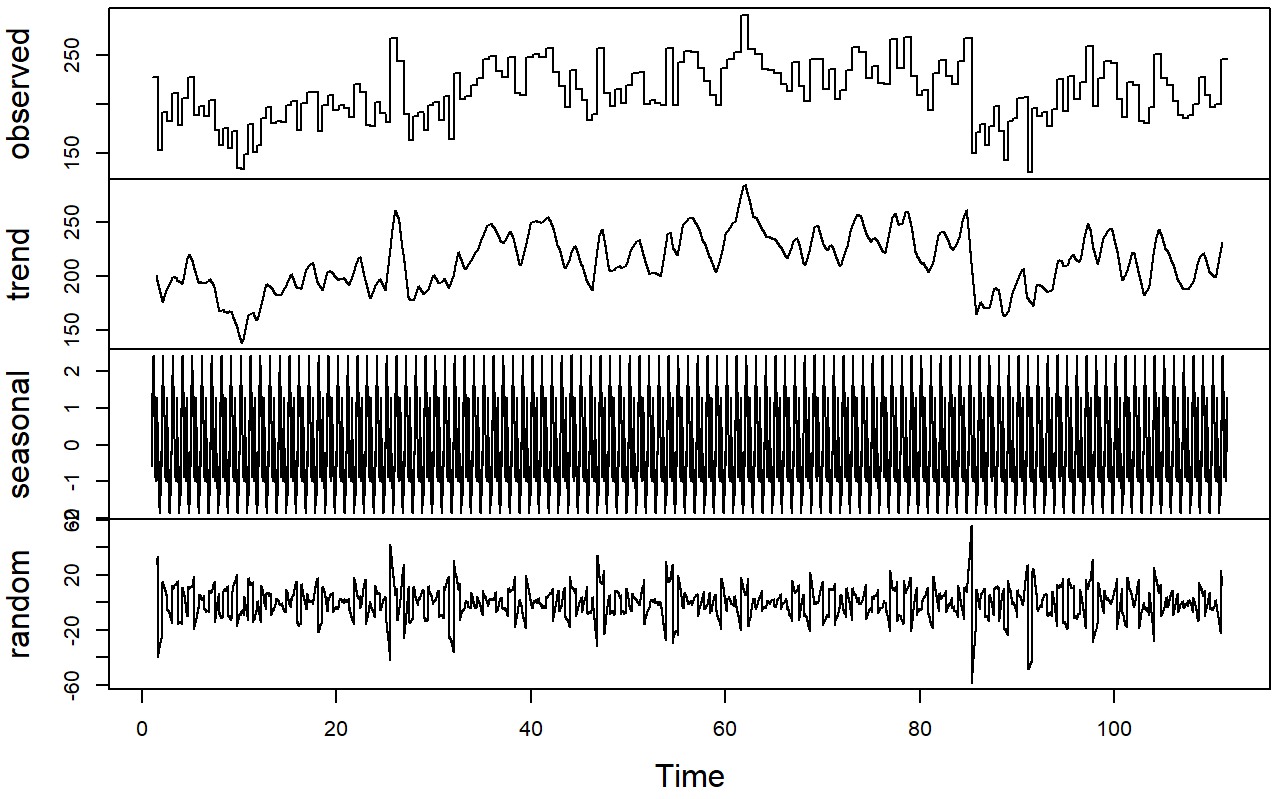
```
# Assuming monthly data, adjust frequency to 12
ts_data <- ts(crime_data$Crime_Count, start = c(year(min(crime_data$Date)), month(min(crime_data
$Date))), frequency = 12)

regressor_columns <- c("Temp","Snow","Humidity","Precip")

# Ensure no NAs in regressors
crime_data <- crime_data %>%
  filter(complete.cases(.[regressor_columns]))

model <- auto.arima(ts_data, xreg = as.matrix(crime_data[, regressor_columns, drop = FALSE]))
summary(model)
```

```
## Series: ts_data
## Regression with ARIMA(0,1,0) errors
##
## Coefficients:
##         Temp     Snow  Humidity   Precip
##       0.1809  -6.4801    0.4327  -0.0687
## s.e.  0.0146   0.1885    0.0089   0.0231
##
## sigma^2 = 3.317:  log likelihood = -81514.77
## AIC=163039.5   AICc=163039.5   BIC=163082.6
##
## Training set error measures:
##                        ME     RMSE        MAE          MPE       MAPE
## Training set 0.0006415215 1.821191 0.09683261 -0.003957288 0.04813143
##                     MASE          ACF1
## Training set 0.07942304 -1.231625e-07
```

```
forecast_values <- forecast(model, xreg = as.matrix(crime_data[, regressor_columns, drop = FALS
E]), h = nrow(crime_data))
plot(forecast_values, main = "ARIMA Forecast with Independent Variables",
     ylab = "Crime Count", xlab = "Days", xlim = c(min(time(ts_data)), max(time(ts_data))),
     ylim = c(0, max(forecast_values$upper[,2], ts_data))/4)

lines(ts_data, col = "red")
legend("topright", legend = c("Actual", "Forecast"), col = c("red", "blue"))
```




```
crime_ts <- ts(crime_data$Crime_Count, frequency = 365) # Assuming daily data

decomposition <- decompose(crime_ts)
plot(decomposition)
```

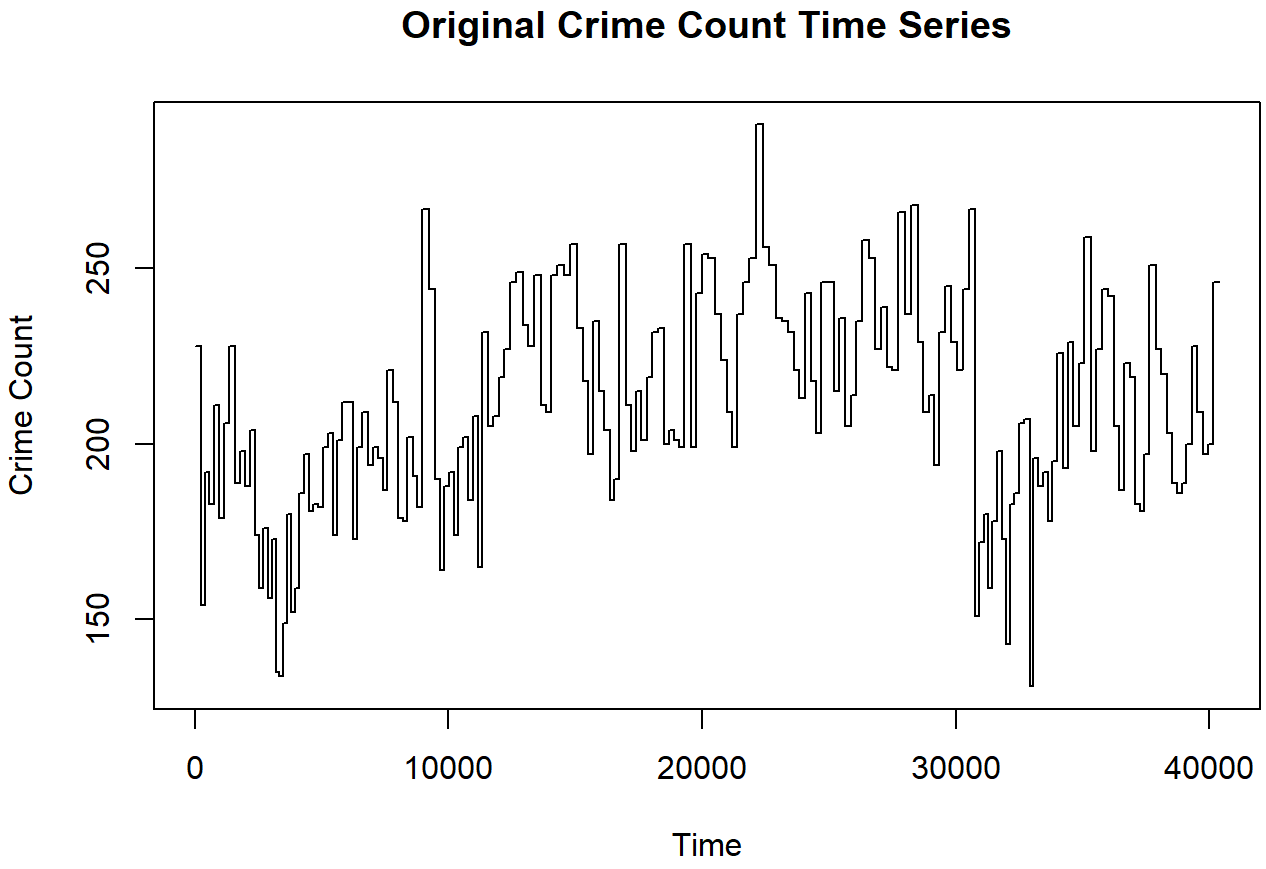


ARIMA RELATED GRAPHS:

```
# Load necessary libraries
library(readr)
library(dplyr)
library(lubridate)
library(tseries)
library(forecast)
library(ggplot2)

crime_ts <- ts(crime_data$Crime_Count, start = c(year(min(crime_data$Date)), month(min(crime_dat
a$Date))), frequency = 1)

# Augmented Dickey-Fuller Test
adf_test <- adf.test(crime_ts, alternative = "stationary")
```

```
## Warning in adf.test(crime_ts, alternative = "stationary"): p-value smaller than
## printed p-value
```

```
# Differencing the series if not stationary
if (adf_test$p.value > 0.05) {
  crime_ts_diff <- diff(crime_ts)
  adf_test_diff <- adf.test(crime_ts_diff, alternative = "stationary")
}

# Plotting the original and differenced series
ts.plot(crime_ts, main="Original Crime Count Time Series", ylab="Crime Count", xlab="Time")
```




```
if (exists("crime_ts_diff")) {
  ts.plot(crime_ts_diff, main="Differenced Crime Count Time Series", ylab="Differenced Crime Cou
nt", xlab="Time")
}

# ACF and PACF plots
#Acf(crime_ts_diff, main="ACF of Differenced Series")
#Pacf(crime_ts_diff, main="PACF of Differenced Series")
```

c. RANDOM FOREST REGRESSION MODEL:

```
# Load necessary libraries
library(readr)
library(dplyr)
library(lubridate)
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 4.3.2
```

```
## randomForest 4.7-1.1
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

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

```
library(caret)

# Set seed for reproducibility
set.seed(42)

# Calculate the size of the training set (80% of the dataset)
training_size <- floor(0.8 * nrow(crime_data))

# Randomly sample row indices for the training set
training_indices <- sample(seq_len(nrow(crime_data)), size = training_size)

# Create training and testing sets
trainingSet <- crime_data[training_indices, ]
testingSet <- crime_data[-training_indices, ]


# Ensure that Crime_Count and other predictors are numeric

trainingSet$Temp <- as.numeric(trainingSet$Temp)
trainingSet$Snow <- as.numeric(trainingSet$Snow)
trainingSet$Humidity <- as.numeric(trainingSet$Humidity)
trainingSet$Precip <- as.numeric(trainingSet$Precip)


# Random Forest model training
rf_model <- randomForest(Crime_Count ~Temp + Snow + Humidity + Precip , data = trainingSet, ntre
e = 100)

# Model prediction and evaluation
rf_predictions <- predict(rf_model, testingSet)
mse <- mean((rf_predictions - testingSet$Crime_Count)^2)
rsq <- cor(rf_predictions, testingSet$Crime_Count)^2

# Output the MSE and R-squared
print(paste("Mean Squared Error:", mse))
```

```
## [1] "Mean Squared Error: 331.795424181371"
```

```
print(paste("R-squared:", rsq))
```

```
## [1] "R-squared: 0.67447459774457"
```

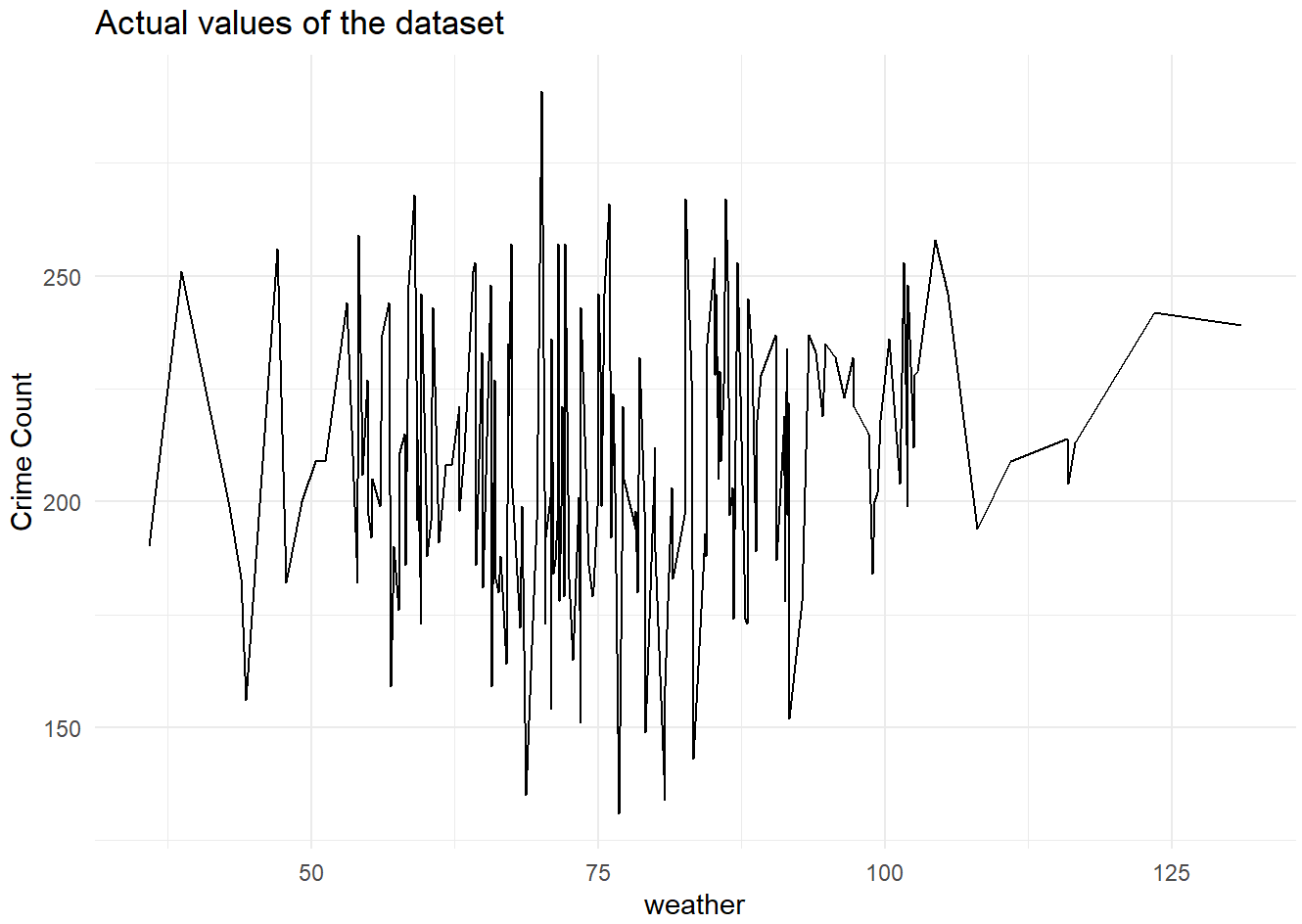
PREDICTION:

```
# Now, let's say you have a new data point for which you want to make predictions:
new_data <- data.frame(Temp=-0.7,Snow=0
,Humidity=82.8
,Precip=7.045

)  # Replace with your actual values

# Predict the target variable for the new data point
predicted_value <- predict(rf_model, new_data)

# Print the predicted value
print(predicted_value)
```

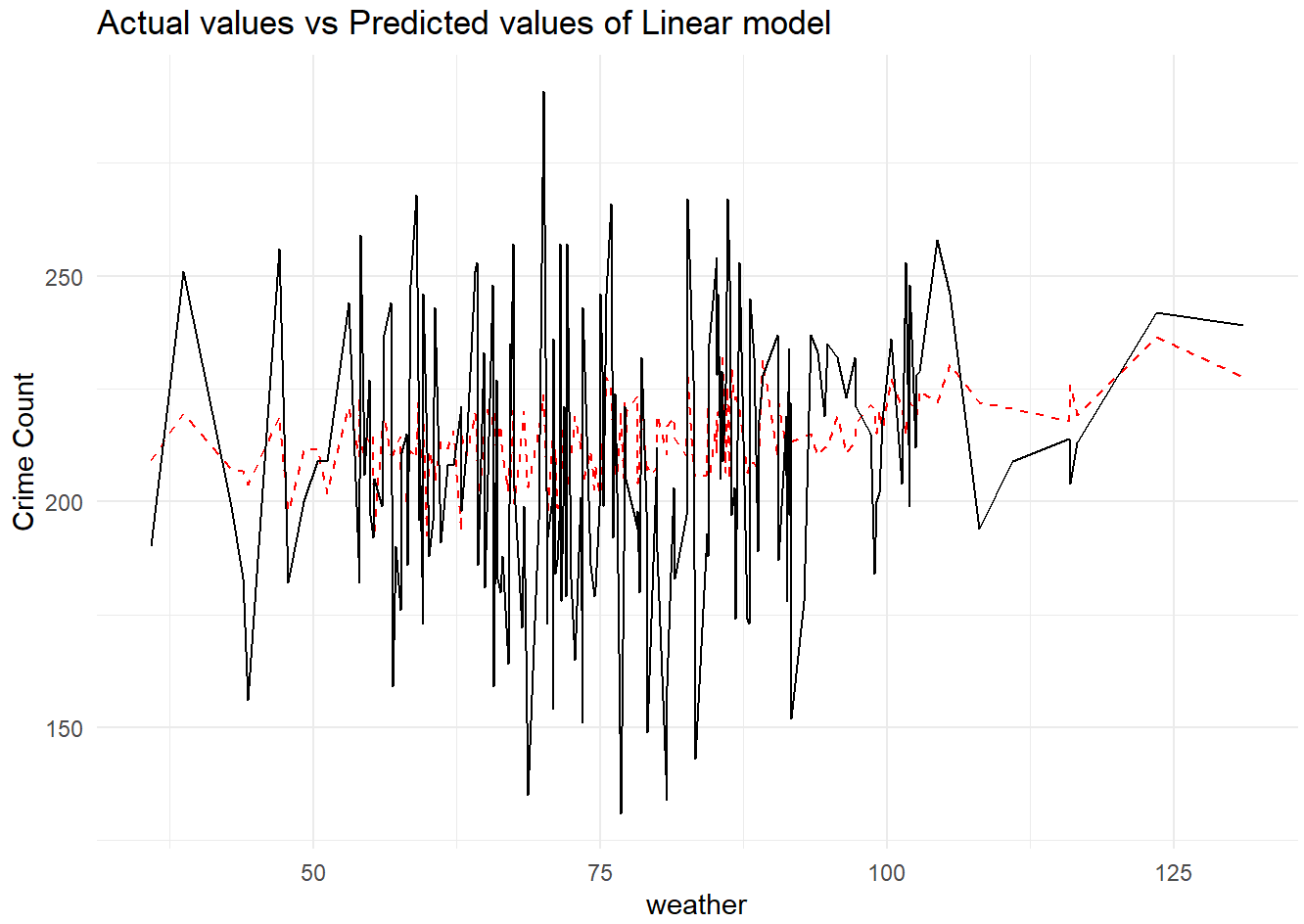
```
##      1
## 228.66
```

PREDICTION GRAPHS OF DIFFERENT MODELS:

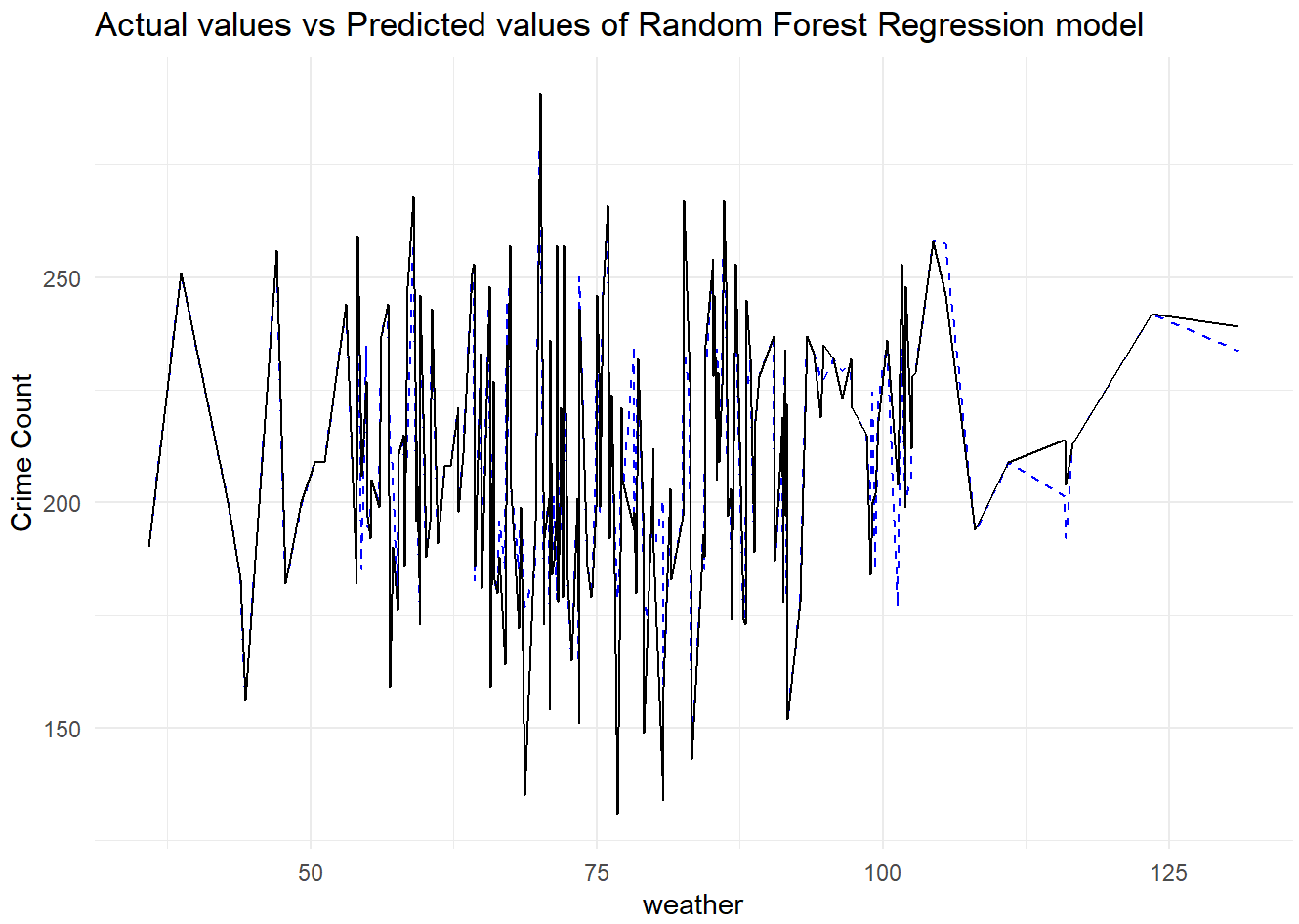
```
library(randomForest)
trainIndex <- createDataPartition(crime_data$Crime_Count, p = 0.8,
                                  list = FALSE,
                                  times = 1)
dataTrain <- crime_data[trainIndex, ]
dataTest <- crime_data[-trainIndex, ]
# Model training
model_lm <- lm(Crime_Count ~ Temp + Snow + Humidity + Precip, data = dataTrain)
lr_predictions <- predict(model_lm, dataTest)
rf_model <- randomForest(Crime_Count ~Temp , data = dataTrain, ntree = 100)
rf_predictions <- predict(rf_model, dataTest)

ggplot(dataTest) +
geom_line(aes(x = Temp + Snow + Humidity + Precip, y = Crime_Count), color = "black")+
  labs(title = "Actual values of the dataset", x = "weather", y = "Crime Count") +
  theme_minimal()
```




```
ggplot(dataTest) +
  geom_line(aes(x = Temp + Snow + Humidity + Precip, y = lr_predictions), color = "red", linetyp
e = "dashed") +
  geom_line(aes(x = Temp + Snow + Humidity + Precip, y = Crime_Count), color = "black")+
  labs(title = "Actual values vs Predicted values of Linear model", x = "weather", y = "Crime Co
unt") +
  theme_minimal()
```




```
ggplot(dataTest) +
  geom_line(aes(x = Temp + Snow + Humidity + Precip, y = rf_predictions), color = "blue", linety
pe = "dashed" )+
  geom_line(aes(x = Temp + Snow + Humidity + Precip, y = Crime_Count), color = "black")+
  labs(title = "Actual values vs Predicted values of Random Forest Regression model", x = "weath
er", y = "Crime Count") +
  theme_minimal()
```


Actual values vs Predicted values of Random Forest Regression model
Crime Count
250
200
150
50
75
100
125
weather