Modeling and Analysis of Various Crimes in Louisville

Vinay Vaida

#Calling Louisville Crime Dataset

```
CrimeData <- read.csv('D:\\U albany class Docs\\Applied Stats\\Submitted
final Copy\\dataset 2023\\dataset 2023\\Louisville_Metro_KY_-
_Crime_Data_2023.csv', header = TRUE, stringsAsFactors = FALSE)

summary(CrimeData)
```

```
##  Incident_Number    Date_Reported      Date_Occurred        Badge_ID
##  Length:42487       Length:42487       Length:42487       Length:42487
##  Class  :character  Class  :character  Class  :character  Class  :character
##  Mode   :character  Mode   :character  Mode   :character  Mode   :character
##
##
##
##  Offense_Classification Offense_Code_Name   NIBRS_Code
##  Length:42487           Length:42487       Length:42487
##  Class  :character      Class  :character  Class  :character
##  Mode   :character      Mode   :character  Mode   :character
##
##
##
##  NIBRS_Group        Was_Offense_Completed LMPD_Division       LMPD_Beat
##  Length:42487       Length:42487          Length:42487       Length:42487
##  Class  :character  Class  :character     Class  :character  Class
:character
##  Mode   :character  Mode   :character     Mode   :character  Mode
:character
##
##
##
##  Location_Category  Block_Address          City             Zip_Code
##  Length:42487       Length:42487       Length:42487       Length:42487
##  Class  :character  Class  :character  Class  :character  Class  :character
##  Mode   :character  Mode   :character  Mode   :character  Mode   :character
##
##
##
##     ObjectId
##  Min.   :    1
##  1st Qu.:10622
##  Median :21244
##  Mean   :21244
##  3rd Qu.:31866
##  Max.   :42487
```

#DATA CLEANING for Louisville Crime Dataset

```
library(tidyr)

CrimeData <- separate(CrimeData, Offense_Classification, into =
c("Offense_Classification_Number", "Offense_Classification_Type"), sep = " ",
remove = FALSE)

## Warning: Expected 2 pieces. Additional pieces discarded in 25789 rows [1,
2, 3, 4, 5, 6,
## 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...].

## Warning: Expected 2 pieces. Missing pieces filled with `NA` in 59 rows
[7404, 7406,
## 7414, 7466, 7498, 7499, 7533, 8029, 8107, 8632, 8704, 8768, 9009, 9095,
9706,
## 9824, 9924, 9985, 10322, 10331, ...].

CrimeData <- separate(CrimeData, Date_Occurred, into = c("Occurred_Date",
"Occurred_Time"), sep = " ", remove = FALSE)
CrimeData <- CrimeData[CrimeData$City == "LOUISVILLE", ]
CrimeData$Zip_Code <- gsub(" 0+$", "", CrimeData$Zip_Code)
unique_values <- unique(CrimeData$NIBRS_Code)

head(CrimeData)

##   Incident_Number Date_Reported          Date_Occurred Occurred_Date
## 1     LMPD23073647     8/11/2023 2023/08/11 04:00:00+00    2023/08/11
## 2     LMPD23073911     8/12/2023 2023/08/12 04:00:00+00    2023/08/12
## 3     LMPD23073227     8/10/2023 2023/08/10 04:00:00+00    2023/08/10
## 4     LMPD23072757      8/9/2023 2023/08/09 04:00:00+00    2023/08/09
## 5     LMPD23073019      8/9/2023 2023/08/09 04:00:00+00    2023/08/09
## 6     LMPD23071649      8/6/2023 2023/08/06 04:00:00+00    2023/08/06
##   Occurred_Time Badge_ID Offense_Classification
Offense_Classification_Number
## 1   04:00:00+00     5213     47 FAMILY OFFENSES
47
## 2   04:00:00+00     7365     10 KIDNAPPING ONLY
10
## 3   04:00:00+00     5317     47 FAMILY OFFENSES
47
## 4   04:00:00+00     5240     47 FAMILY OFFENSES
47
## 5   04:00:00+00     5383     10 KIDNAPPING ONLY
10
## 6   04:00:00+00     5516     10 KIDNAPPING ONLY
10
##   Offense_Classification_Type
## 1                      FAMILY
## 2                  KIDNAPPING
## 3                      FAMILY
## 4                      FAMILY
## 5                  KIDNAPPING
## 6                  KIDNAPPING
##                                            Offense_Code_Name NIBRS_Code
## 1           CUSTODIAL INTERFERENCE-FELONY 509.070 10210 100        100
## 2 KIDNAPPING-WITH SERIOUS PHYSICAL INJURY 509.040 10071 100        100
## 3           CUSTODIAL INTERFERENCE-FELONY 509.070 10210 100        100
## 4           CUSTODIAL INTERFERENCE-FELONY 509.070 10210 100        100
## 5        UNLAWFUL IMPRISONMENT-2ND DEGREE 509.030 02606 100        100
## 6        UNLAWFUL IMPRISONMENT-2ND DEGREE 509.030 02606 100        100
##   NIBRS_Group Was_Offense_Completed LMPD_Division LMPD_Beat
## 1           A                   Yes  1st Division       123
## 2           A                   Yes  7th Division      NULL
## 3           A                   Yes  6th Division       625
## 4           A                   Yes  2nd Division       236
## 5           A                   Yes  3rd Division       323
## 6           A                   Yes  4th Division       411
##             Location_Category                Block_Address       City
Zip_Code
## 1 Government/ Public Building           200 BLOCK S 2ND ST LOUISVILLE
40202
## 2              Residence/Home       4900 BLOCK FOREMAN AVE LOUISVILLE
40219
## 3              Residence/Home    4200 BLOCK WALLINGFORD LN LOUISVILLE
40218
## 4              Residence/Home 2500 BLOCK EMMA KATHERINE LN LOUISVILLE
40216
## 5              Residence/Home          9700 BLOCK MASON LN LOUISVILLE
40118
## 6              Residence/Home          1200 BLOCK S 2ND ST LOUISVILLE
40203
##   ObjectId
## 1      452
## 2      528
## 3      771
## 4      895
## 5      943
## 6     1370
```

```
Beat411=CrimeData
```

#Classification of Crime

Serious Crime: Serious crimes typically refer to offenses that are considered more severe or significant in terms of their potential impact on individuals and society. They often result in more severe legal consequences. Serious crimes can include both violent and non-violent offenses, such as murder, robbery, sexual assault, burglary, arson, kidnapping, and terrorism.

Violent Crime: Violent crimes are a subset of serious crimes that specifically involve the use of force or the threat of force against another person, resulting in physical harm or the fear of harm. Examples of violent crimes include assault, battery, homicide, domestic violence, and sexual assault. Classifying Crimes and Identifying Patterns.

#Modeling Crime

Offense Category Offense Code Crime Against

Serious Crimes:-
Counterfeiting/Forgery (Property) 250 Property Destruction/Damage/Vandalism… 290 Property Drug/Narcotic Violations (Society) 35A Society Drug Equipment Violations (Society) 35B Society Embezzlement (Property) 270 Property Extortion/Blackmail (Property) 210 Property Fraud Offenses Property - False Pretenses/Swindle… 26A Property - Credit Card/Automated… 26B Property - Impersonation 26C Property - Welfare Fraud 26D Property - Wire Fraud 26E Property Gambling Offenses (Society) 39A, 39B, 39C, 39D Society Larceny/Theft Offenses Property - Pocket-picking 23A Property - Purse-snatching 23B Property - Shoplifting 23C Property - Theft From Building 23D Property - Theft From Coin-Operated… 23E Property - Theft From Motor Vehicle 23F Property - Theft of Motor Vehicle Parts… 23G Property - All Other Larceny 23H Property Motor Vehicle Theft (Property) 240 Property Pornography/Obscene Material (Society) 370 Society Robbery (Property) 120 Property Stolen Property Offenses (Property) 280 Property

Violent Crimes:-
Homicide Offenses Person (Violent Crime) - Murder & Nonnegligent Manslaughter 09A Person (Violent Crime) - Negligent Manslaughter 09B Person (Violent Crime) - Justifiable Homicide 09C Person (Violent Crime) Kidnapping/Abduction (Person) 100 Person (Violent Crime) Robbery (Property) 120 Property Simple Assault (Person) 13B Person (Violent Crime) Intimidation (Person) 13C Person (Violent Crime) Sex Offenses, Forcible Person (Violent Crime) - Forcible Rape 11A Person (Violent Crime) - Forcible Sodomy 11B Person (Violent Crime) - Sexual Assault With An Object 11C Person (Violent Crime) - Forcible Fondling 11D Person (Violent Crime) Sex Offenses, Nonforcible Person - Incest 36A Person - Statutory Rape 36B Person Prostitution Offenses (Society) 40A, 40B Society Weapon Law Violations (Society) 520 Society

#MODELING OF CRIME

```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
```

```
CrimeData <- CrimeData %>%
  mutate(
    Serious_Crime = ifelse(NIBRS_Code %in% c("250", "290", "35A", "35B",
"270", "210", "64A","720",
                                             "26A", "26B", "26C", "26D",
"26E", "39A", "39B", "39C", "39D",
                                             "23A", "23B", "23C", "23D",
"23E", "23F", "23G", "23H", "240", "370", "280"), 1, 0),
    Violent_Crime = ifelse(NIBRS_Code %in% c("09A", "09B", "09C", "100",
"120", "13A","13B", "13C","90F","90J","90C", "90D","90H",
"90E","510","200","90B",
                                             "11A", "11B", "11C", "11D",
"36A", "36B", "220", "520","90Z","999",
                                             "40A", "40B"), 1, 0)
  )

#Print the modified dataframe
head(CrimeData)

##   Incident_Number Date_Reported          Date_Occurred Occurred_Date
## 1    LMPD23073647     8/11/2023 2023/08/11 04:00:00+00    2023/08/11
## 2    LMPD23073911     8/12/2023 2023/08/12 04:00:00+00    2023/08/12
## 3    LMPD23073227     8/10/2023 2023/08/10 04:00:00+00    2023/08/10
## 4    LMPD23072757      8/9/2023 2023/08/09 04:00:00+00    2023/08/09
## 5    LMPD23073019      8/9/2023 2023/08/09 04:00:00+00    2023/08/09
## 6    LMPD23071649      8/6/2023 2023/08/06 04:00:00+00    2023/08/06
##   Occurred_Time Badge_ID Offense_Classification
Offense_Classification_Number
## 1   04:00:00+00     5213     47 FAMILY OFFENSES
47
## 2   04:00:00+00     7365     10 KIDNAPPING ONLY
10
## 3   04:00:00+00     5317     47 FAMILY OFFENSES
47
## 4   04:00:00+00     5240     47 FAMILY OFFENSES
47
## 5   04:00:00+00     5383     10 KIDNAPPING ONLY
10
## 6   04:00:00+00     5516     10 KIDNAPPING ONLY
10
##   Offense_Classification_Type
## 1                      FAMILY
## 2                  KIDNAPPING
## 3                      FAMILY
## 4                      FAMILY
## 5                  KIDNAPPING
## 6                  KIDNAPPING
##                                              Offense_Code_Name NIBRS_Code
## 1            CUSTODIAL INTERFERENCE-FELONY 509.070 10210 100          100
## 2 KIDNAPPING-WITH SERIOUS PHYSICAL INJURY 509.040 10071 100          100
## 3           CUSTODIAL INTERFERENCE-FELONY 509.070 10210 100        100
## 4           CUSTODIAL INTERFERENCE-FELONY 509.070 10210 100        100
## 5        UNLAWFUL IMPRISONMENT-2ND DEGREE 509.030 02606 100        100
## 6        UNLAWFUL IMPRISONMENT-2ND DEGREE 509.030 02606 100        100
##   NIBRS_Group Was_Offense_Completed LMPD_Division LMPD_Beat
## 1           A                   Yes  1st Division       123
## 2           A                   Yes  7th Division      NULL
## 3           A                   Yes  6th Division       625
## 4           A                   Yes  2nd Division       236
## 5           A                   Yes  3rd Division       323
## 6           A                   Yes  4th Division       411
##              Location_Category                Block_Address       City
Zip_Code
## 1 Government/ Public Building           200 BLOCK S 2ND ST LOUISVILLE
40202
## 2              Residence/Home       4900 BLOCK FOREMAN AVE LOUISVILLE
40219
## 3              Residence/Home    4200 BLOCK WALLINGFORD LN LOUISVILLE
40218
## 4              Residence/Home 2500 BLOCK EMMA KATHERINE LN LOUISVILLE
40216
## 5              Residence/Home          9700 BLOCK MASON LN LOUISVILLE
40118
## 6              Residence/Home          1200 BLOCK S 2ND ST LOUISVILLE
40203
##   ObjectId Serious_Crime Violent_Crime
## 1      452             0             1
## 2      528             0             1
## 3      771             0             1
## 4      895             0             1
## 5      943             0             1
## 6     1370             0             1

Beat411=CrimeData
```

#Dropping Unsed Coloumns in CrimeData DATSET

```
library(dplyr)
columns_to_drop <- c("Incident_Number", "Date_Occurred","Occurred_Time",
                      "Offense_Classification",
"Offense_Classification_Type","Location_Category"
,"Badge_ID","NIBRS_Group","Offense_Classification_Number","ObjectId","LMPD_Be
at",
                      "Offense_Code_Name", "LMPD_Division", "Block_Address",
"City")

CrimeData <- CrimeData %>%
  select(-any_of(columns_to_drop))

CrimeData <- CrimeData %>%
  arrange(Occurred_Date)
head(CrimeData)
```

```
##   Date_Reported Occurred_Date NIBRS_Code Was_Offense_Completed Zip_Code
## 1      1/1/2023    2023/01/01        120                   Yes    40204
## 2      1/1/2023    2023/01/01        120                   Yes    40217
## 3      1/1/2023    2023/01/01        120                   Yes    40208
## 4      1/1/2023    2023/01/01        220                   Yes    40213
## 5      1/1/2023    2023/01/01        220                   Yes    40272
## 6      1/1/2023    2023/01/01        220                   Yes    40258
##   Serious_Crime Violent_Crime
## 1             0             1
## 2             0             1
## 3             0             1
## 4             0             1
## 5             0             1
## 6             0             1
```

#CALLING THE TEMPERATURE DATASET

```
TempData <- read.csv('D:\\U albany class Docs\\Applied Stats\\Submitted final
Copy\\dataset 2023\\dataset 2023\\tempearature DataSet\\Louisville 2023-01-01
to 2023-09-16.csv', header = TRUE, stringsAsFactors = FALSE)



TempData <- TempData %>%
  mutate(datetime = as.Date(datetime, format = "%Y-%m-%d"),
         datetime = format(datetime, "%Y/%m/%d"))

TempData <- TempData %>%
  select(-tempmax, -tempmin, -feelslikemax, -feelslikemin, -feelslike, -dew,
-precipprob, -precipcover,
         -snowdepth, -windgust, -windspeed, -winddir, -sealevelpressure,
         -cloudcover, -solarradiation, -solarenergy, -uvindex,
         -severerisk, -sunrise, -sunset, -moonphase, -conditions,-name,-
stations,-icon,-description)

head(TempData,10)
```

```
##      datetime temp humidity precip preciptype snow visibility
## 1  2023/01/01  8.8     95.6  0.019       rain    0        8.3
## 2  2023/01/02 13.9     89.2  0.554       rain    0       15.1
## 3  2023/01/03 16.3     90.2 62.816       rain    0       13.0
## 4  2023/01/04 12.8     74.2  0.000               0       16.0
## 5  2023/01/05  5.5     65.1  0.000               0       16.0
## 6  2023/01/06  3.8     64.6  0.000               0       16.0
## 7  2023/01/07  5.0     63.1  0.000               0       15.8
## 8  2023/01/08  4.4     70.2  0.293       rain    0       15.6
## 9  2023/01/09  3.6     69.9  0.000               0       15.5
## 10 2023/01/10  5.9     66.2  0.000               0       16.0
```

#Merging Datasets based on the crime occurance date using left join

```
TempData_renamed <- TempData %>%
  rename(
    temp.x = temp,
    humidity.x = humidity,
    precip.x = precip,
    visibility.x = visibility
  )

CrimeData$Occurred_Date <- as.Date(CrimeData$Occurred_Date)
TempData$datetime <- as.Date(TempData$datetime)

CrimeData <- left_join(CrimeData, TempData, by = c("Occurred_Date" =
"datetime"))
head(CrimeData)

##   Date_Reported Occurred_Date NIBRS_Code Was_Offense_Completed Zip_Code
## 1      1/1/2023    2023-01-01        120                   Yes    40204
## 2      1/1/2023    2023-01-01        120                   Yes    40217
## 3      1/1/2023    2023-01-01        120                   Yes    40208
## 4      1/1/2023    2023-01-01        220                   Yes    40213
## 5      1/1/2023    2023-01-01        220                   Yes    40272
## 6      1/1/2023    2023-01-01        220                   Yes    40258
##   Serious_Crime Violent_Crime temp humidity precip preciptype snow
visibility
## 1             0             1  8.8     95.6  0.019       rain    0
8.3
## 2             0             1  8.8     95.6  0.019       rain    0
8.3
## 3             0             1  8.8     95.6  0.019       rain    0
8.3
## 4             0             1  8.8     95.6  0.019       rain    0
8.3
## 5             0             1  8.8     95.6  0.019       rain    0
8.3
## 6             0             1  8.8     95.6  0.019       rain    0
8.3
```

#Incorparating public Holiday for better analysis

```
CrimeData$Occurred_Date <- as.Date(CrimeData$Occurred_Date)


public_holidays <- as.Date(c("2023-01-02", "2023-01-16", "2023-04-07", "2023-
05-29", "2023-06-19", "2023-07-04", "2023-09-04"))


CrimeData$Public_Holiday <- ifelse(CrimeData$Occurred_Date %in%
public_holidays, 1, 0)
```

```
#Adding new feature WEEKENDS
CrimeData$Weekend <- ifelse(weekdays(CrimeData$Occurred_Date) %in%
c("Saturday", "Sunday"), 1, 0)

cd1 <- CrimeData
head(CrimeData)

##   Date_Reported Occurred_Date NIBRS_Code Was_Offense_Completed Zip_Code
## 1      1/1/2023    2023-01-01        120                   Yes    40204
## 2      1/1/2023    2023-01-01        120                   Yes    40217
## 3      1/1/2023    2023-01-01        120                   Yes    40208
## 4      1/1/2023    2023-01-01        220                   Yes    40213
## 5      1/1/2023    2023-01-01        220                   Yes    40272
## 6      1/1/2023    2023-01-01        220                   Yes    40258
##   Serious_Crime Violent_Crime temp humidity precip preciptype snow
visibility
## 1             0             1  8.8     95.6  0.019       rain    0
8.3
## 2             0             1  8.8     95.6  0.019       rain    0
8.3
## 3             0             1  8.8     95.6  0.019       rain    0
8.3
## 4             0             1  8.8     95.6  0.019       rain    0
8.3
## 5             0             1  8.8     95.6  0.019       rain    0
8.3
## 6             0             1  8.8     95.6  0.019       rain    0
8.3
##   Public_Holiday Weekend
## 1              0       1
## 2              0       1
## 3              0       1
## 4              0       1
## 5              0       1
## 6              0       1

write.csv(CrimeData, file = "BeforeCrimeData.csv", row.names = FALSE)
```

#TRANSFORMING THE DATASETS

```
summarized_data <- CrimeData %>%
  group_by(Occurred_Date) %>%
  summarise(
    Sum_Serious_Crime = sum(Serious_Crime, na.rm = TRUE),
    Sum_Violent_Crime = sum(Violent_Crime, na.rm = TRUE),
    Temp = first(temp),  # Assuming temperature is constant for a given date
    Preciptype=first(preciptype),
    Snow=first(snow),
    Weekend=max(Weekend),
    Humidity = first(humidity),  # Assuming humidity is constant for a given
date
    Precip = first(precip),  # Assuming precipitation is constant for a given
date
    Visibility = first(visibility),  # Assuming visibility is constant for a
given date
    Public_Holiday = max(Public_Holiday),  # Assuming Public_Holiday is a
binary indicator (0 or 1)
    Was_Offense_Completed_Yes = sum(Was_Offense_Completed == "Yes"),
    Was_Offense_Completed_No = sum(Was_Offense_Completed == "No")
  )
#calculating Total Crime
summarized_data$Total_Crime <- summarized_data$Sum_Serious_Crime +
summarized_data$Sum_Violent_Crime

summary(summarized_data)

##  Occurred_Date        Sum_Serious_Crime Sum_Violent_Crime      Temp
##  Min.   :2023-01-01   Min.   : 57.00    Min.   : 53.00    Min.   :-3.10
##  1st Qu.:2023-02-26   1st Qu.: 82.00    1st Qu.: 78.00    1st Qu.: 9.50
##  Median :2023-04-23   Median : 94.00    Median : 88.00    Median :17.00
##  Mean   :2023-04-23   Mean   : 94.26    Mean   : 90.46    Mean   :15.72
##  3rd Qu.:2023-06-18   3rd Qu.:104.00    3rd Qu.:102.00    3rd Qu.:23.40
##  Max.   :2023-08-13   Max.   :152.00    Max.   :141.00    Max.   :30.90
##   Preciptype              Snow             Weekend          Humidity
##  Length:225          Min.   :0.00000   Min.   :0.0000   Min.   :34.00
##  Class :character   1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.:54.60
##  Mode  :character   Median :0.00000   Median :0.0000   Median :64.70
##                     Mean   :0.05067   Mean   :0.2889   Mean   :63.76
##                     3rd Qu.:0.00000   3rd Qu.:1.0000   3rd Qu.:72.50
##                     Max.   :6.60000   Max.   :1.0000   Max.   :95.60
##      Precip          Visibility     Public_Holiday
Was_Offense_Completed_Yes
##  Min.   : 0.000   Min.   : 5.40   Min.   :0.00000   Min.   : 88.0
##  1st Qu.: 0.000   1st Qu.:15.20   1st Qu.:0.00000   1st Qu.:125.0
##  Median : 0.000   Median :15.90   Median :0.00000   Median :144.0
##  Mean   : 3.360   Mean   :15.24   Mean   :0.02667   Mean   :154.1
##  3rd Qu.: 1.555   3rd Qu.:16.00   3rd Qu.:0.00000   3rd Qu.:182.0
##  Max.   :64.073   Max.   :16.00   Max.   :1.00000   Max.   :265.0
##  Was_Offense_Completed_No  Total_Crime
##  Min.   : 3.00           Min.   :117.0
##  1st Qu.:11.00           1st Qu.:163.0
##  Median :36.00           Median :184.0
##  Mean   :30.61           Mean   :184.7
##  3rd Qu.:43.00           3rd Qu.:204.0
##  Max.   :63.00           Max.   :276.0

#Print the updated data frame
head(summarized_data,20)
```

```
## # A tibble: 20 × 14
##    Occurred_Date Sum_Serious_Crime Sum_Violent_Crime  Temp Preciptype
Snow
##    <date>                    <dbl>             <dbl> <dbl> <chr>
<dbl>
##  1 2023-01-01                   99               106   8.8 "rain"
0
##  2 2023-01-02                  104               117  13.9 "rain"
0
##  3 2023-01-03                   90                95  16.3 "rain"
0
##  4 2023-01-04                   98               112  12.8 ""
0
##  5 2023-01-05                  106               109   5.5 ""
0
##  6 2023-01-06                   90                99   3.8 ""
0
##  7 2023-01-07                   94               101   5   ""
0
##  8 2023-01-08                   88                81   4.4 "rain"
0
##  9 2023-01-09                   72                99   3.6 ""
0
## 10 2023-01-10                  148               105   5.9 ""
0
## 11 2023-01-11                  104               116  10.3 "rain"
0
## 12 2023-01-12                   98                85  11.5 "rain"
0
## 13 2023-01-13                  104               117   2.6 "rain,snow"
0.3
## 14 2023-01-14                  103               114   0.8 "rain,freezing…
0
## 15 2023-01-15                  118               102   1.3 "rain,snow"
0
## 16 2023-01-16                  111                92   5.5 "rain,freezing…
0
## 17 2023-01-17                  132                96  13.8 "rain"
0
## 18 2023-01-18                  133               130   7   "rain"
0
## 19 2023-01-19                  102               102  12.5 "rain"
0
## 20 2023-01-20                  111                94   3   "rain"
0
## # ℹ 8 more variables: Weekend <dbl>, Humidity <dbl>, Precip <dbl>,
## #   Visibility <dbl>, Public_Holiday <dbl>, Was_Offense_Completed_Yes
<int>,
## #   Was_Offense_Completed_No <int>, Total_Crime <dbl>
```

```
sarimadf=summarized_data
RidgeDF=summarized_data

write.csv(summarized_data, file = "CDSum.csv", row.names = FALSE)
```

```
head(Beat411)
```

```
##   Incident_Number Date_Reported          Date_Occurred Occurred_Date
## 1    LMPD23073647     8/11/2023 2023/08/11 04:00:00+00    2023/08/11
## 2    LMPD23073911     8/12/2023 2023/08/12 04:00:00+00    2023/08/12
## 3    LMPD23073227     8/10/2023 2023/08/10 04:00:00+00    2023/08/10
## 4    LMPD23072757      8/9/2023 2023/08/09 04:00:00+00    2023/08/09
## 5    LMPD23073019      8/9/2023 2023/08/09 04:00:00+00    2023/08/09
## 6    LMPD23071649      8/6/2023 2023/08/06 04:00:00+00    2023/08/06
##   Occurred_Time Badge_ID Offense_Classification
Offense_Classification_Number
## 1   04:00:00+00     5213     47 FAMILY OFFENSES
47
## 2   04:00:00+00     7365     10 KIDNAPPING ONLY
10
## 3   04:00:00+00     5317     47 FAMILY OFFENSES
47
## 4   04:00:00+00     5240     47 FAMILY OFFENSES
47
## 5   04:00:00+00     5383     10 KIDNAPPING ONLY
10
## 6   04:00:00+00     5516     10 KIDNAPPING ONLY
10
##   Offense_Classification_Type
## 1                      FAMILY
## 2                  KIDNAPPING
## 3                      FAMILY
## 4                      FAMILY
## 5                  KIDNAPPING
## 6                  KIDNAPPING
##                                           Offense_Code_Name NIBRS_Code
## 1           CUSTODIAL INTERFERENCE-FELONY 509.070 10210 100        100
## 2 KIDNAPPING-WITH SERIOUS PHYSICAL INJURY 509.040 10071 100        100
## 3           CUSTODIAL INTERFERENCE-FELONY 509.070 10210 100        100
## 4           CUSTODIAL INTERFERENCE-FELONY 509.070 10210 100        100
## 5        UNLAWFUL IMPRISONMENT-2ND DEGREE 509.030 02606 100        100
## 6        UNLAWFUL IMPRISONMENT-2ND DEGREE 509.030 02606 100        100
##   NIBRS_Group Was_Offense_Completed LMPD_Division LMPD_Beat
## 1           A                   Yes  1st Division       123
## 2           A                   Yes  7th Division      NULL
## 3           A                   Yes  6th Division       625
## 4           A                   Yes  2nd Division       236
## 5           A                   Yes  3rd Division       323
## 6           A                   Yes  4th Division       411
##             Location_Category                Block_Address       City
```

```
Zip_Code
## 1 Government/ Public Building           200 BLOCK S 2ND ST LOUISVILLE
40202
## 2              Residence/Home       4900 BLOCK FOREMAN AVE LOUISVILLE
40219
## 3              Residence/Home    4200 BLOCK WALLINGFORD LN LOUISVILLE
40218
## 4              Residence/Home 2500 BLOCK EMMA KATHERINE LN LOUISVILLE
40216
## 5              Residence/Home          9700 BLOCK MASON LN LOUISVILLE
40118
## 6              Residence/Home          1200 BLOCK S 2ND ST LOUISVILLE
40203
##   ObjectId Serious_Crime Violent_Crime
## 1      452             0             1
## 2      528             0             1
## 3      771             0             1
## 4      895             0             1
## 5      943             0             1
## 6     1370             0             1
```

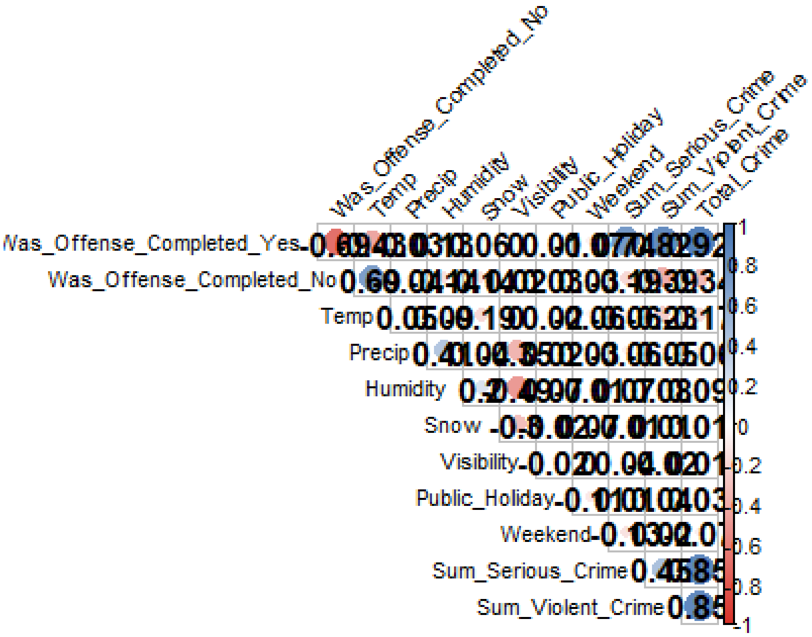
#Creating new database from CrimeData for speific Location Beat 411

```
Beat411$Occurred_Date <- as.Date(Beat411$Occurred_Date)
Beat411 <- Beat411 %>%
  arrange(Occurred_Date)
Beat411 <-  left_join(Beat411, TempData, by = c("Occurred_Date" =
"datetime"))
Beat411 <- Beat411[Beat411$LMPD_Beat == "411", ]
Beat411 <- Beat411 %>%
  select(-Date_Occurred,-Offense_Classification,-Incident_Number,-
Offense_Code_Name,-Block_Address,-City,-ObjectId,-preciptype)
Beat411 <- Beat411 %>%
  group_by(Occurred_Date) %>%
  summarise(
    Sum_Serious_Crime = sum(Serious_Crime, na.rm = TRUE),
    Sum_Violent_Crime = sum(Violent_Crime, na.rm = TRUE),
    Temp = first(temp),  # Corrected spelling here
    Snow = first(snow),
    Humidity = first(humidity),  # Assuming humidity is constant for a given
date
    Precip = first(precip),  # Assuming precipitation is constant for a given
date
    Visibility = first(visibility),  # Assuming Public_Holiday is a binary
indicator (0 or 1)
    Was_Offense_Completed_Yes = sum(Was_Offense_Completed == "Yes"),
    Was_Offense_Completed_No = sum(Was_Offense_Completed == "No")
  )
Beat411$Total_Crime <- Beat411$Sum_Serious_Crime + Beat411$Sum_Violent_Crime
Beat411$Weekend <- ifelse(weekdays(Beat411$Occurred_Date) %in% c("Saturday",
"Sunday"), 1, 0)
head(Beat411)

## # A tibble: 6 × 12
##   Occurred_Date Sum_Serious_Crime Sum_Violent_Crime  Temp  Snow Humidity
Precip
##   <date>                    <dbl>             <dbl> <dbl> <dbl>    <dbl>
<dbl>
## 1 2023-01-01                    0                 3   8.8     0     95.6
0.019
## 2 2023-01-02                    3                 5  13.9     0     89.2
0.554
## 3 2023-01-03                    6                 4  16.3     0     90.2
62.8
## 4 2023-01-04                    7                 4  12.8     0     74.2
0
## 5 2023-01-05                    7                 6   5.5     0     65.1
0
## 6 2023-01-06                    0                 4   3.8     0     64.6
0
## # ℹ 5 more variables: Visibility <dbl>, Was_Offense_Completed_Yes <int>,
## #   Was_Offense_Completed_No <int>, Total_Crime <dbl>, Weekend <dbl>
```

#Exploratory Data Analysis (EDA):

A)Correlation plot

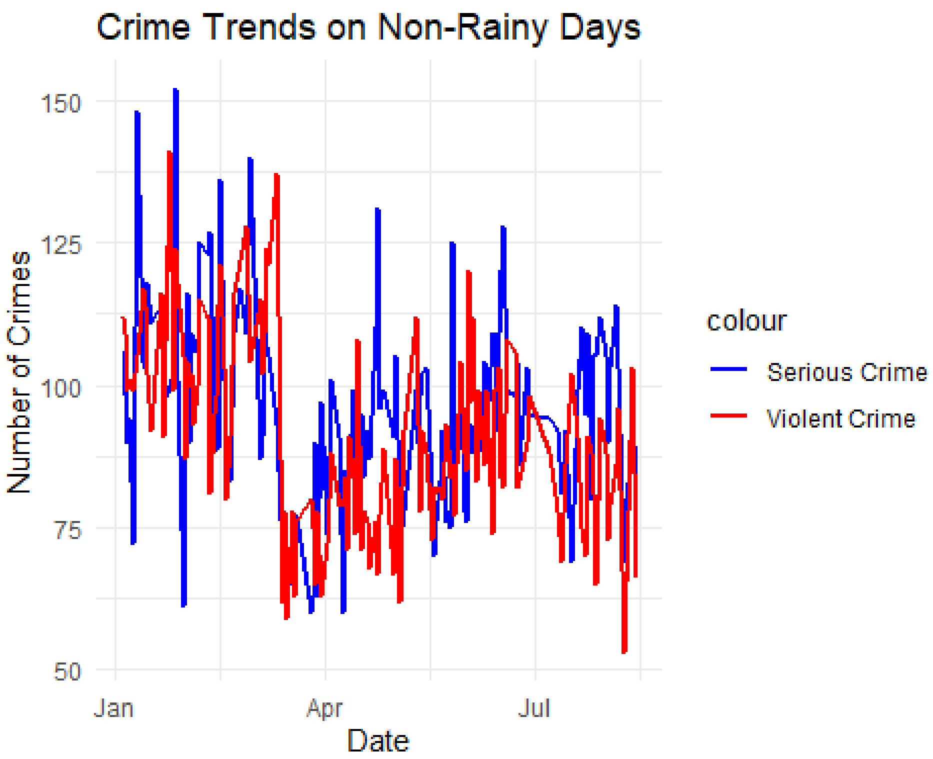
```
library(corrplot)

## Warning: package 'corrplot' was built under R version 4.3.2

## corrplot 0.92 loaded

subset_data <- summarized_data[, c("Was_Offense_Completed_Yes",
                                   "Was_Offense_Completed_No",
                                   "Temp", "Precip", "Humidity","Snow" ,
                                   "Visibility",
"Public_Holiday","Weekend","Sum_Serious_Crime","Sum_Violent_Crime","Total_Cri
me")]

subset_data_numeric <- subset_data[, sapply(subset_data, is.numeric)]

cor_matrix <- cor(subset_data_numeric, use = "complete.obs")

par(mar = c(5, 5, 5, 5))

corrplot(cor_matrix, method = "circle", type = "upper",
         tl.col = "black", tl.srt = 45, tl.cex = 0.7,
         cl.cex = 0.7, cl.ratio = 0.1,
         col = colorRampPalette(c("#D73027", "white", "#4575B4"))(200),
         addCoef.col = "black", diag = FALSE)
```



```
par(mar = c(5, 4, 4, 2) + 0.1)
```

#B.Crime Trends

```
library(ggplot2)

## Warning: package 'ggplot2' was built under R version 4.3.2

library(dplyr)

if ('Weekend' %in% colnames(summarized_data)) {
  summarized_data$Occurred_Date <- as.Date(summarized_data$Occurred_Date)

  conditions <- list(
    weekend = summarized_data[summarized_data$Weekend == 1, ],
    holiday = summarized_data[summarized_data$Public_Holiday == 1, ],
    rain = summarized_data[summarized_data$Preciptype == 'rain', ],
    no_rain = summarized_data[summarized_data$Preciptype != 'rain', ]
  )

  plot_trends <- function(df, title) {
    ggplot(df, aes(x = Occurred_Date)) +
      geom_line(aes(y = Sum_Serious_Crime, color = 'Serious Crime'), size =
1) +
      geom_line(aes(y = Sum_Violent_Crime, color = 'Violent Crime'), size =
1) +
      labs(title = title, x = 'Date', y = 'Number of Crimes') +
      theme_minimal() +
      scale_color_manual(values = c('Serious Crime' = 'blue', 'Violent Crime'
= 'red'))
  }

  plot_trends(conditions$no_rain, 'Crime Trends on Non-Rainy Days')
} else {
  cat("Column 'Weekend' not found in the dataframe.")
}

## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## ℹ Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



#c) CRIME ANALYSIS ON HOLIDAYS

```
CrimeData$Occurred_Date <- as.Date(CrimeData$Occurred_Date)

public_holidays <- as.Date(c("2023-01-02", "2023-01-16", "2023-04-07", "2023-
05-29", "2023-06-19", "2023-07-04", "2023-09-04"))

CrimeData$Public_Holiday <- as.integer(CrimeData$Occurred_Date %in%
public_holidays)

crime_counts <- list(
```

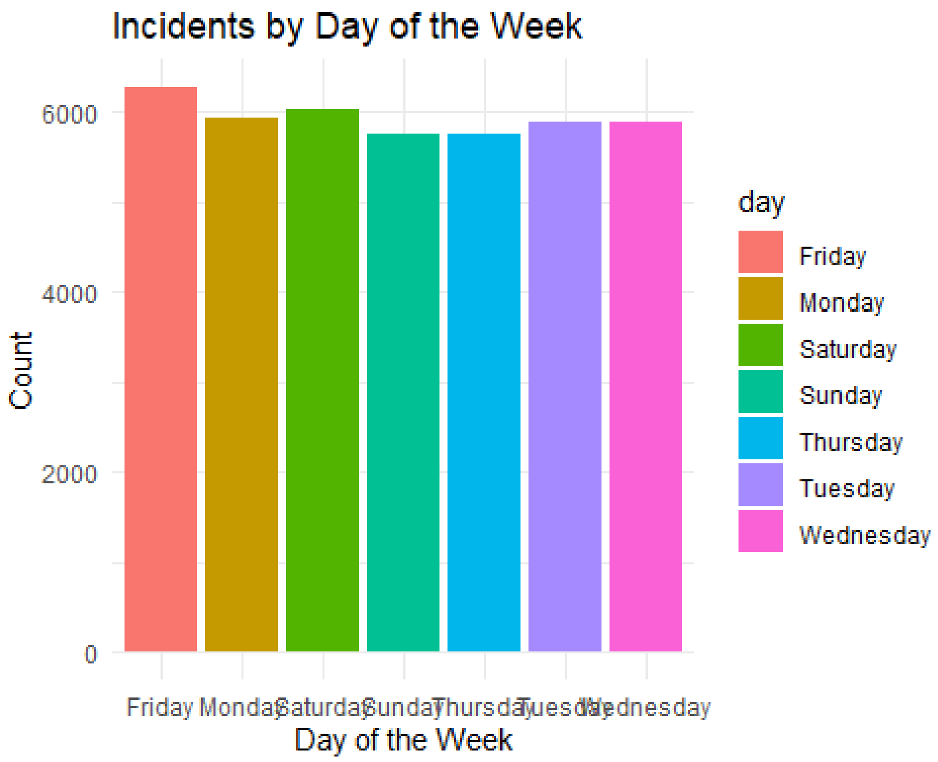
```r
  holidays = colSums(subset(CrimeData, Public_Holiday ==
1)[c("Serious_Crime", "Violent_Crime")]),
  non_holidays = colSums(subset(CrimeData, Public_Holiday ==
0)[c("Serious_Crime", "Violent_Crime")])
)

total_counts <- c(holidays = nrow(subset(CrimeData, Public_Holiday == 1)),
                  non_holidays = nrow(subset(CrimeData, Public_Holiday ==
0)))
library(ggplot2)



plots <- lapply(unique(crime_counts$Holiday_Status), function(status) {
  ggplot(subset(crime_counts, Holiday_Status == status),
         aes(x = Category, y = Count, fill = Category)) +
    geom_bar(stat = "identity", color = "black") +
    labs(title = paste("Crime Counts on", status),
         x = "Crime Category",
         y = "Crime Count",
         fill = "Crime Category") +
    theme_minimal()
})

 # Plot for Non-Holidays
#crime_counts <- data.frame(Category = rep(c("Serious Crime", "Violent
Crime"), each = 2), Holiday_Status = rep(c("Holidays", "Non-Holidays"), times
= 2),Count = c(serious_crime_holidays, serious_crime_non_holidays,
violent_crime_holidays, violent_crime_non_holidays))
cat("Total Crime Counts on Holidays:", total_counts["holidays"], "\n")

## Total Crime Counts on Holidays: 1144

cat("Total Crime Counts on Non-Holidays:", total_counts["non_holidays"],
"\n")

## Total Crime Counts on Non-Holidays: 40418

cat("Serious Crime Counts on Holidays:", crime_counts$holidays[1], "\n")

## Serious Crime Counts on Holidays: 574

cat("Serious Crime Counts on Non-Holidays:", crime_counts$non_holidays[1],
"\n")

## Serious Crime Counts on Non-Holidays: 20635

cat("Violent Crime Counts on Holidays:", crime_counts$holidays[2], "\n")

## Violent Crime Counts on Holidays: 570
```

```r
cat("Violent Crime Counts on Non-Holidays:", crime_counts$non_holidays[2],
"\n")
```

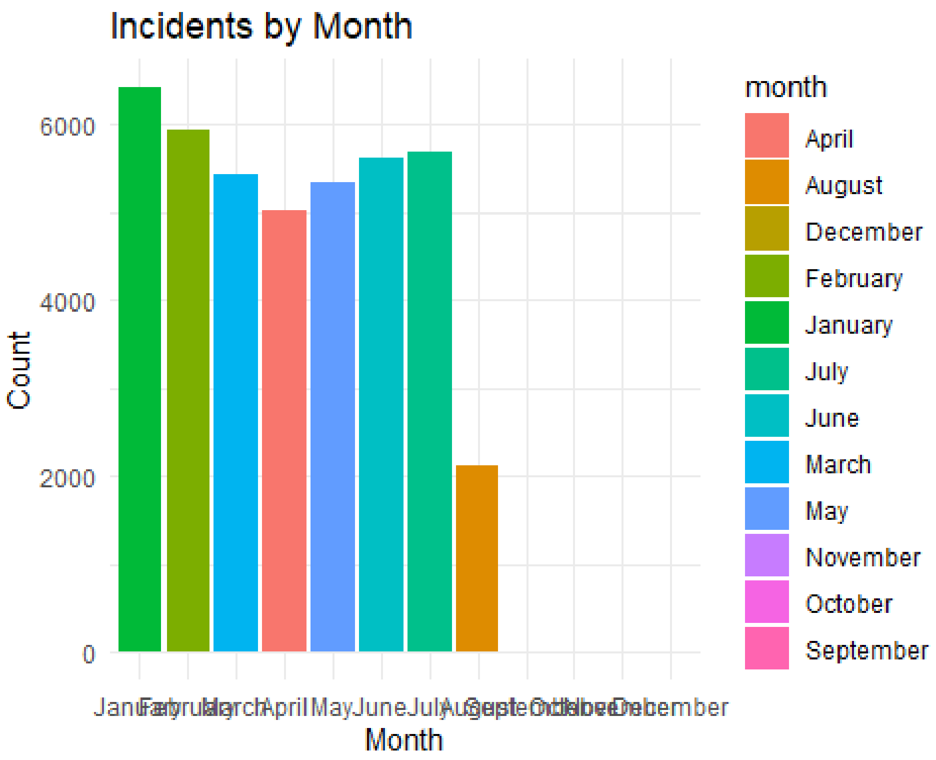
```
## Violent Crime Counts on Non-Holidays: 19783
```

#D) Distribution of Incident Dates

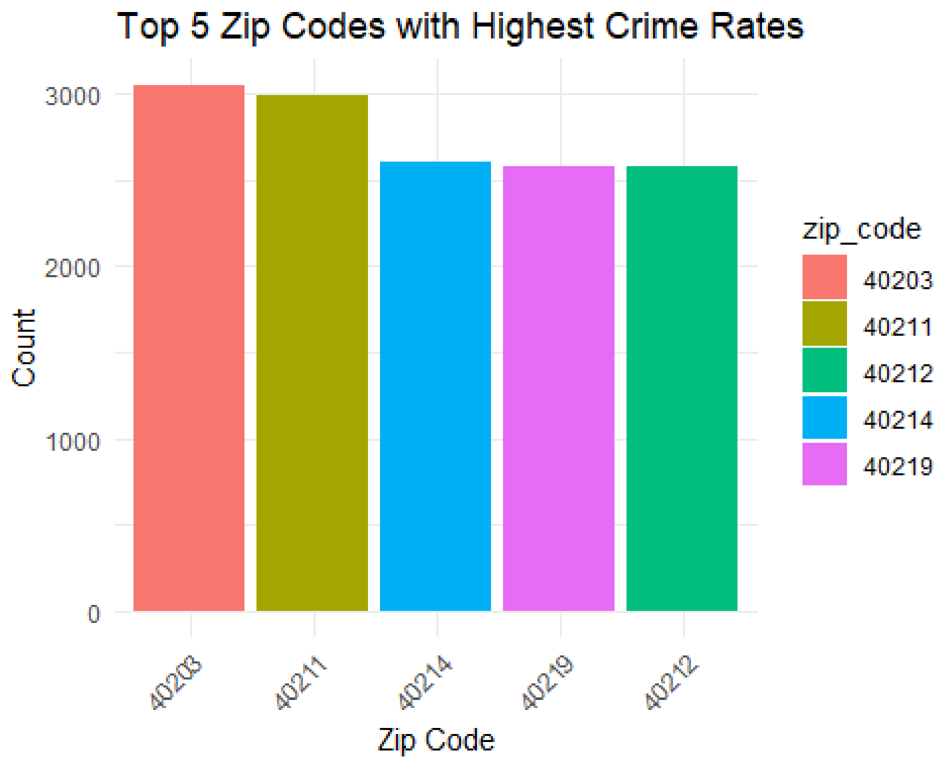
```r
# Load required libraries
library(dplyr)
library(ggplot2)

# 1. Temporal Trends
cd1$Day_of_Week <- factor(weekdays(cd1$Occurred_Date), levels = c("Sunday",
"Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"))
cd1$Month <- factor(months(cd1$Occurred_Date), levels = month.name)

# Count incidents by day of the week and by month
day_of_week_counts <- table(cd1$Day_of_Week)
month_counts <- table(cd1$Month)

# Plotting day of the week counts
ggplot(data = data.frame(day = names(day_of_week_counts), count =
as.numeric(day_of_week_counts)),
       aes(x = day, y = count, fill = day)) +
  geom_bar(stat = "identity") +
  labs(title = "Incidents by Day of the Week", x = "Day of the Week", y =
"Count") +
  theme_minimal()
```



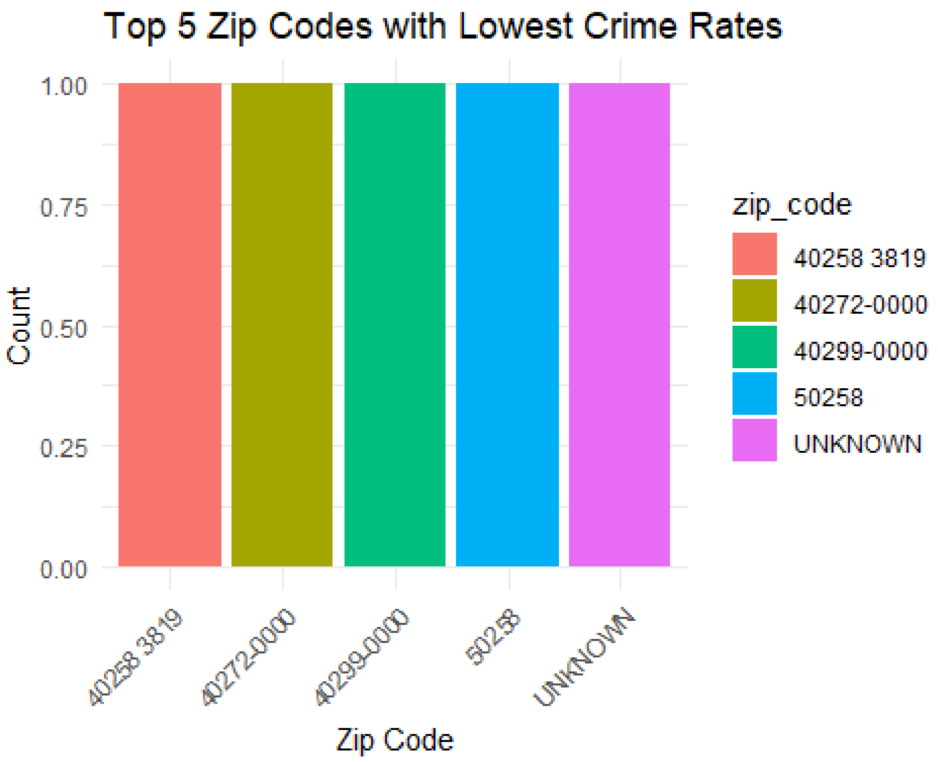
```
# Plotting month counts
ggplot(data = data.frame(month = names(month_counts), count =
as.numeric(month_counts)),
       aes(x = month, y = count, fill = month)) +
  geom_bar(stat = "identity") +
  labs(title = "Incidents by Month", x = "Month", y = "Count") +
  scale_x_discrete(limits = month.name) +  # Ensure months are in order
  theme_minimal()
```



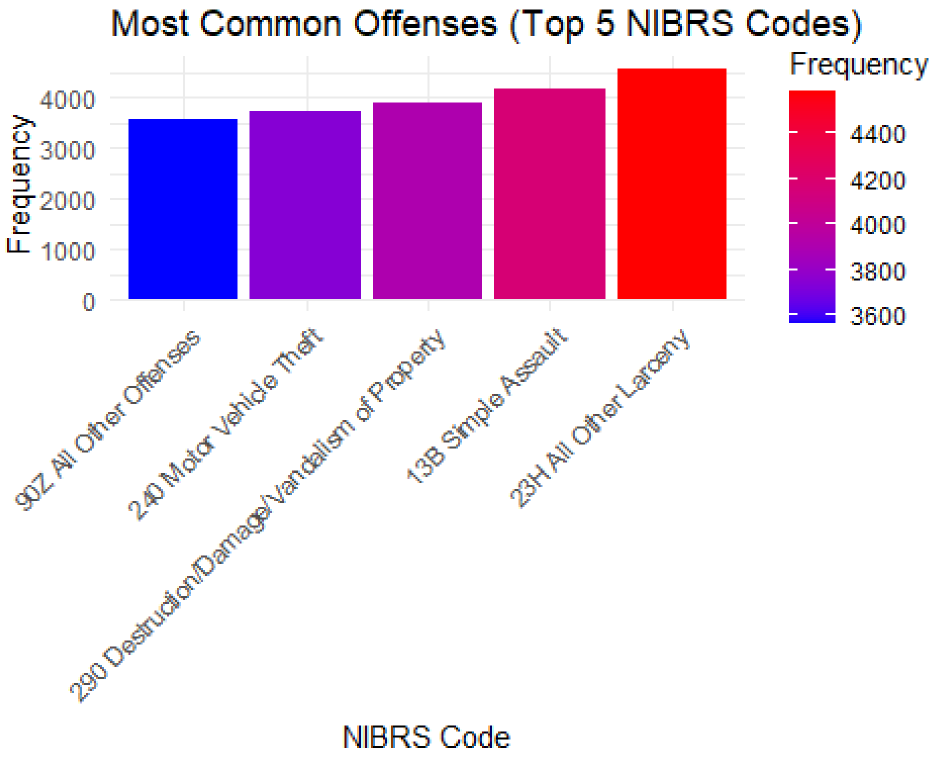
```
# 2. Most and Least Common Offenses
most_common_offenses <- head(sort(table(cd1$NIBRS_Code), decreasing = TRUE),
5)
least_common_offenses <- tail(sort(table(cd1$NIBRS_Code), decreasing = TRUE),
5)

# 3. Proportion of Serious and Violent Crimes
serious_crime_proportion <- table(cd1$Serious_Crime) /
sum(table(cd1$Serious_Crime))
violent_crime_proportion <- table(cd1$Violent_Crime) /
sum(table(cd1$Violent_Crime))

# 4. Geographical Hotspots
top_zip_codes <- head(sort(table(cd1$Zip_Code), decreasing = TRUE), 5)
bottom_zip_codes <- tail(sort(table(cd1$Zip_Code), decreasing = TRUE), 5)

# Plotting top zip codes
ggplot(data = data.frame(zip_code = names(top_zip_codes), count =
as.numeric(top_zip_codes)),
       aes(x = reorder(zip_code, -count), y = count, fill = zip_code)) +
  geom_bar(stat = "identity") +
  labs(title = "Top 5 Zip Codes with Highest Crime Rates", x = "Zip Code", y
= "Count") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



```r
# Plotting bottom zip codes
ggplot(data = data.frame(zip_code = names(bottom_zip_codes), count =
as.numeric(bottom_zip_codes)),
       aes(x = reorder(zip_code, count), y = count, fill = zip_code)) +
  geom_bar(stat = "identity") +
  labs(title = "Top 5 Zip Codes with Lowest Crime Rates", x = "Zip Code", y =
"Count") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



```r
# 5. Correlations with Weather Conditions
weather_columns <- c('temp.x', 'humidity.x', 'precip.x', 'visibility.x')

# 6. Impact of Public Holidays on Crime Rates
public_holiday_crime_count <- cd1 %>% group_by(Public_Holiday) %>%
summarize(Count = n())

# Define the labels
labels <- c(
  "90Z All Other Offenses",
  "240 Motor Vehicle Theft",
  "290 Destruction/Damage/Vandalism of Property",
  "13B Simple Assault",
  "23H All Other Larceny"
)

#7. Most Common Offenses
most_common_plot <- ggplot(data = data.frame(NIBRS_Code =
names(most_common_offenses), Frequency = as.numeric(most_common_offenses)),
                           aes(x = reorder(NIBRS_Code, Frequency), y =
Frequency, fill = Frequency)) +
  geom_bar(stat = "identity") +
  scale_fill_gradient(low = "blue", high = "red") +
  labs(title = 'Most Common Offenses (Top 5 NIBRS Codes)',
       x = 'NIBRS Code', y = 'Frequency') +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_x_discrete(labels = labels)

print(most_common_plot)
```



```
if (exists("correlations_with_weather")) {
  findings_summary <- list(
    "Temporal Trends" = list(
      "Day of Week Counts" = day_of_week_counts,
      "Month Counts" = month_counts
    ),
    "Common Offenses" = list(
      "Most Common" = most_common_offenses,
      "Least Common" = least_common_offenses
    ),
    "Crime Proportions" = list(
      "Serious Crime" = serious_crime_proportion,
      "Violent Crime" = violent_crime_proportion
    ),
    "Geographical Hotspots" = list(
      "Top Zip Codes" = top_zip_codes,
      "Bottom Zip Codes" = bottom_zip_codes
    ),
    "Weather Correlations" = correlations_with_weather,
    "Public Holiday Impact" = public_holiday_crime_count$Count
  )
} else {
  # 'correlations_with_weather' doesn't exist
  findings_summary <- list(
    "Temporal Trends" = list(
      "Day of Week Counts" = day_of_week_counts,
      "Month Counts" = month_counts
    ),
    "Common Offenses" = list(
      "Most Common" = most_common_offenses,
      "Least Common" = least_common_offenses
    ),
    "Crime Proportions" = list(
      "Serious Crime" = serious_crime_proportion,
      "Violent Crime" = violent_crime_proportion
    ),
    "Geographical Hotspots" = list(
      "Top Zip Codes" = top_zip_codes,
      "Bottom Zip Codes" = bottom_zip_codes
    ),
    "Public Holiday Impact" = public_holiday_crime_count$Count
  )
}

# Displaying findings
findings_summary
```

```
## $`Temporal Trends`
## $`Temporal Trends`$`Day of Week Counts`
##
##    Sunday    Monday   Tuesday Wednesday  Thursday    Friday  Saturday
##      5763      5938      5891      5896      5763      6280      6031
##
## $`Temporal Trends`$`Month Counts`
##
##   January  February     March     April       May      June      July
August
##      6413      5941      5431      5023      5330      5621      5683
2120
## September   October  November  December
##         0         0         0         0
##
##
## $`Common Offenses`
## $`Common Offenses`$`Most Common`
##
##  90Z  240  290  13B  23H
## 4580 4164 3903 3742 3561
##
## $`Common Offenses`$`Least Common`
##
## 09C 39B 40B 64A 90B
##   1   1   1   1   1
## 
## 
## $`Crime Proportions`
## $`Crime Proportions`$`Serious Crime`
## 
##         0         1 
## 0.4897021 0.5102979 
## 
## $`Crime Proportions`$`Violent Crime`
## 
##         0         1 
## 0.5102979 0.4897021 
## 
## 
## $`Geographical Hotspots`
## $`Geographical Hotspots`$`Top Zip Codes`
## 
## 40203 40211 40214 40219 40212 
##  3049  2994  2606  2586  2585 
## 
## $`Geographical Hotspots`$`Bottom Zip Codes`
## 
## 40258 3819 40272-0000 40299-0000      50258    UNKNOWN 
##          1          1          1          1          1 
## 
## 
## $`Public Holiday Impact`
## [1] 40418  1144
```

#MACHINE LEARNING MODELS

LINEAR REGRESSION

#Serious Crime linear model

```
library(olsrr)
```

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

```
## 
## Attaching package: 'olsrr'
```

```
## The following object is masked from 'package:datasets':
## 
##     rivers
```

```
summarized_data$Days_Since_Start <-
as.numeric(difftime(summarized_data$Occurred_Date,
min(summarized_data$Occurred_Date), units = "days"))
```

```r
# Split the dataset into training and testing sets
set.seed(42)  # Set seed for reproducibility
train_indices <- sample(1:nrow(summarized_data), 0.8 * nrow(summarized_data))
train_data <- summarized_data[train_indices, ]
test_data <- summarized_data[-train_indices, ]

# Create and train the linear regression model
model <- lm(formula = paste('Sum_Serious_Crime', "~", 'Days_Since_Start +
Temp + Humidity + Precip + Visibility + Public_Holiday + Snow +
Was_Offense_Completed_Yes + Was_Offense_Completed_No'), data = train_data)


# Predictions on the test set
y_pred <- predict(model, newdata = test_data)

# Calculate Mean Squared Error (MSE)
mse <- mean((y_pred - test_data$Sum_Serious_Crime)^2)

# Calculate R-squared (R2)
residuals <- y_pred - test_data$Sum_Serious_Crime
ss_residual <- sum(residuals^2)
ss_total <- sum((test_data$Sum_Serious_Crime -
mean(test_data$Sum_Serious_Crime))^2)
r2 <- 1 - (ss_residual / ss_total)

cat("Mean Squared Error (MSE):", mse, "\n")
```

```
## Mean Squared Error (MSE): 77.57485
```

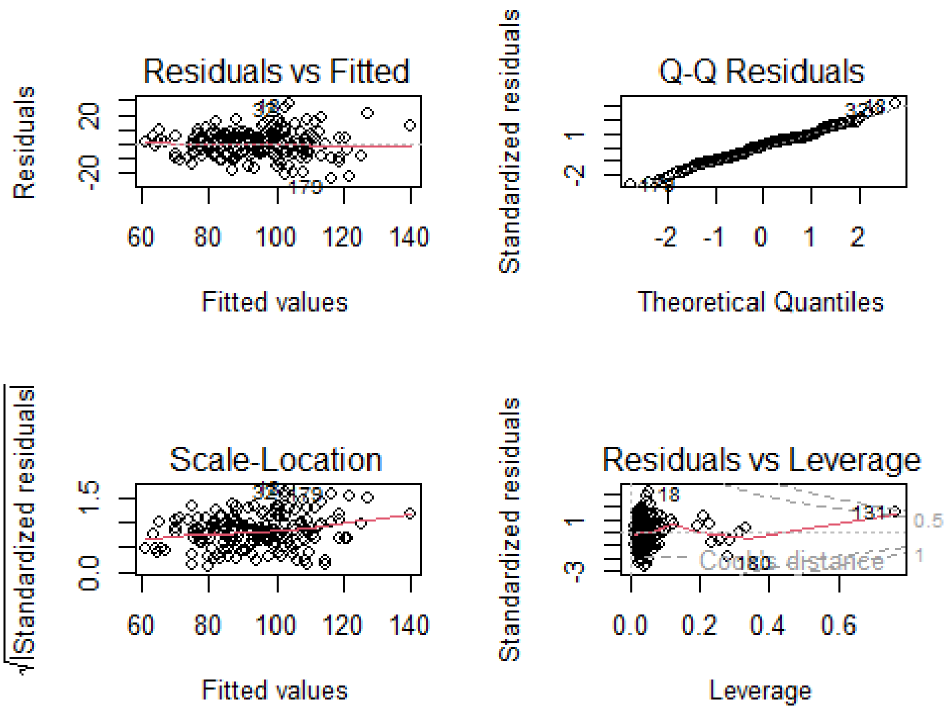
```r
cat("R-squared (R2):", r2, "\n")
```

```
## R-squared (R2): 0.8144813
```

```r
summary(model)
```

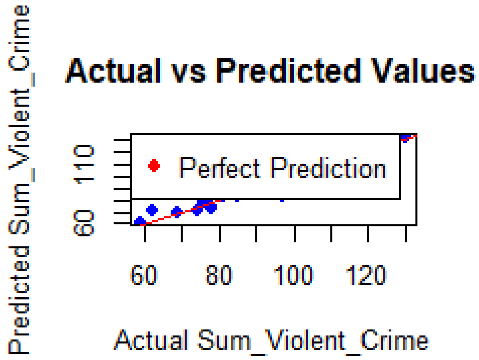
```
##
## Call:
## lm(formula = paste("Sum_Serious_Crime", "~", "Days_Since_Start + Temp +
Humidity + Precip + Visibility + Public_Holiday + Snow +
Was_Offense_Completed_Yes + Was_Offense_Completed_No"),
##     data = train_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -23.5058  -6.3008   0.7098   5.8692  27.2897
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 -22.52361   12.23105  -1.842   0.0673 .
## Days_Since_Start           0.03400    0.02495   1.363   0.1748
## Temp                      -0.16486    0.17690  -0.932   0.3527
## Humidity                   0.04629    0.07071   0.655   0.5135
## Precip                     0.12037    0.08840   1.362   0.1751
## Visibility                 0.84957    0.53244   1.596   0.1124
## Public_Holiday            -1.26576    4.24732  -0.298   0.7661
## Snow                      -0.88760    3.53138  -0.251   0.8018
## Was_Offense_Completed_Yes  0.52470    0.02696  19.459  < 2e-16 ***
## Was_Offense_Completed_No   0.60875    0.07927   7.680  1.2e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.23 on 170 degrees of freedom
## Multiple R-squared:  0.7144, Adjusted R-squared:  0.6993
## F-statistic: 47.24 on 9 and 170 DF,  p-value: < 2.2e-16
```

```r
layout(matrix(c(1, 2, 3, 4), nrow = 2, byrow = TRUE))
plot(model)
```



```r
plot(test_data$Sum_Violent_Crime, y_pred, main = "Actual vs Predicted
Values",
     xlab = "Actual Sum_Violent_Crime", ylab = "Predicted Sum_Violent_Crime",
pch = 16, col = "blue")

# Add a diagonal line for reference (perfect predictions)
abline(a = 0, b = 1, col = "red")
```

```
# Add a legend
legend("topleft", legend = "Perfect Prediction", col = "red", pch = 16)
```

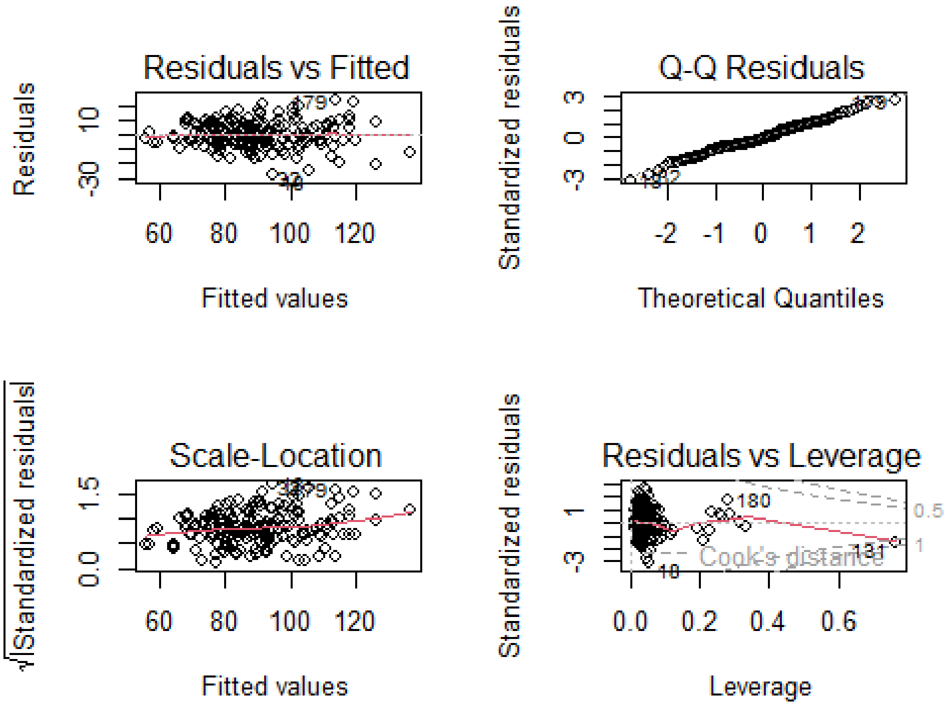


#Violent Crime Linear regression model

```
# Create and train the linear regression model
model <- lm(formula = paste('Sum_Violent_Crime', "~", 'Days_Since_Start +
Temp + Humidity + Precip + Visibility + Public_Holiday + Snow +
Was_Offense_Completed_Yes + Was_Offense_Completed_No'), data = train_data)


# Predictions on the test set
y_pred <- predict(model, newdata = test_data)

# Calculate Mean Squared Error (MSE)
mse <- mean((y_pred - test_data$Sum_Violent_Crime)^2)

# Calculate R-squared (R2)
residuals <- y_pred - test_data$Sum_Violent_Crime
ss_residual <- sum(residuals^2)
ss_total <- sum((test_data$Sum_Violent_Crime -
mean(test_data$Sum_Violent_Crime))^2)
r2 <- 1 - (ss_residual / ss_total)

cat("Mean Squared Error (MSE):", mse, "\n")
```

```
## Mean Squared Error (MSE): 77.57485
```

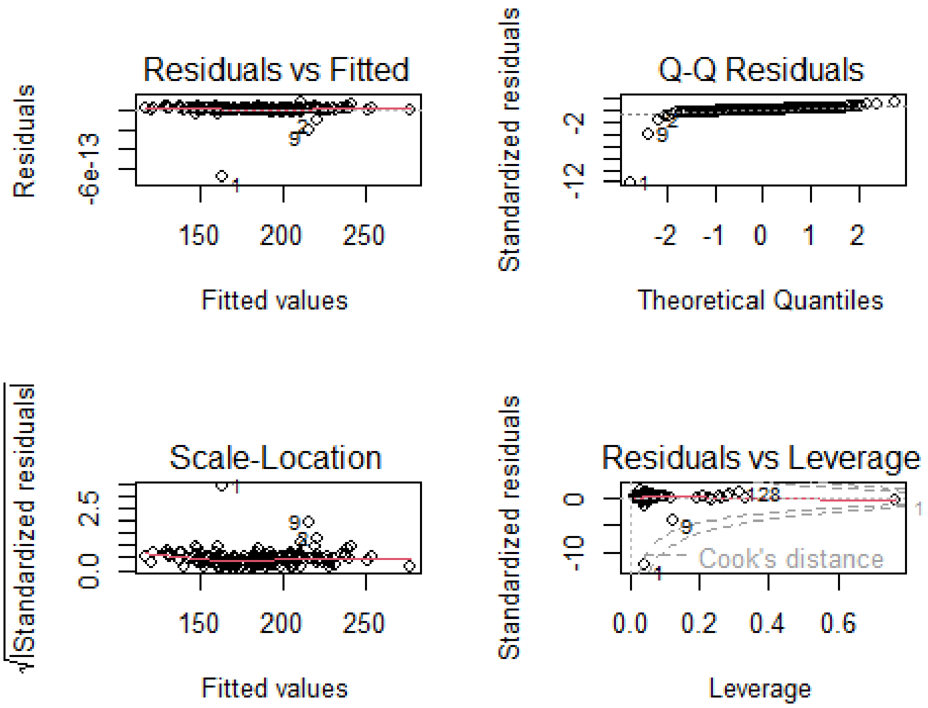
```
cat("R-squared (R2):", r2, "\n")

## R-squared (R2): 0.7250919

summary(model)

##
## Call:
## lm(formula = paste("Sum_Violent_Crime", "~", "Days_Since_Start + Temp +
Humidity + Precip + Visibility + Public_Holiday + Snow +
Was_Offense_Completed_Yes + Was_Offense_Completed_No"),
##     data = train_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -27.2897  -5.8692  -0.7098   6.3008  23.5058
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)               22.52361   12.23105   1.842   0.0673 .
## Days_Since_Start          -0.03400    0.02495  -1.363   0.1748
## Temp                       0.16486    0.17690   0.932   0.3527
## Humidity                  -0.04629    0.07071  -0.655   0.5135
## Precip                    -0.12037    0.08840  -1.362   0.1751
## Visibility                -0.84957    0.53244  -1.596   0.1124
## Public_Holiday             1.26576    4.24732   0.298   0.7661
## Snow                       0.88760    3.53138   0.251   0.8018
## Was_Offense_Completed_Yes  0.47530    0.02696  17.627  < 2e-16 ***
## Was_Offense_Completed_No   0.39125    0.07927   4.936 1.89e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.23 on 170 degrees of freedom
## Multiple R-squared:  0.7301, Adjusted R-squared:  0.7159
## F-statistic: 51.11 on 9 and 170 DF,  p-value: < 2.2e-16

layout(matrix(c(1, 2, 3, 4), nrow = 2, byrow = TRUE))
plot(model)
```



#LINEAR REGRESSION FOR TOTAL CRIME

```
library(olsrr)
summarized_data$Days_Since_Start <-
as.numeric(difftime(summarized_data$Occurred_Date,
min(summarized_data$Occurred_Date), units = "days"))


# Split the dataset into training and testing sets
set.seed(42)  # Set seed for reproducibility
train_indices <- sample(1:nrow(summarized_data), 0.8 * nrow(summarized_data))
train_data <- summarized_data[train_indices, ]
test_data <- summarized_data[-train_indices, ]

# Create and train the linear regression model
model <- lm(formula = paste('Total_Crime', "~", 'Days_Since_Start + Temp +
Humidity + Precip + Visibility + Public_Holiday + Snow +
Was_Offense_Completed_Yes + Was_Offense_Completed_No'), data = train_data)


# Predictions on the test set
y_pred <- predict(model, newdata = test_data)

# Calculate Mean Squared Error (MSE)
mse <- mean((y_pred - test_data$Total_Crime)^2)
```

```
# Calculate R-squared (R2)
residuals <- y_pred - test_data$Total_Crime
ss_residual <- sum(residuals^2)
ss_total <- sum((test_data$Total_Crime - mean(test_data$Total_Crime))^2)
r2 <- 1 - (ss_residual / ss_total)

cat("Mean Squared Error (MSE):", mse, "\n")

## Mean Squared Error (MSE): 3.863946e-27

cat("R-squared (R2):", r2, "\n")

## R-squared (R2): 1

summary(model)

## Warning in summary.lm(model): essentially perfect fit: summary may be
## unreliable

##
## Call:
## lm(formula = paste("Total_Crime", "~", "Days_Since_Start + Temp + Humidity
+ Precip + Visibility + Public_Holiday + Snow + Was_Offense_Completed_Yes +
Was_Offense_Completed_No"),
##     data = train_data)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -6.674e-13 -3.620e-15  3.690e-15  1.521e-14  6.928e-14
##
## Coefficients:
##                              Estimate Std. Error    t value Pr(>|t|)
## (Intercept)               -1.331e-13  7.501e-14 -1.774e+00   0.0779 .
## Days_Since_Start          -3.377e-16  1.530e-16 -2.207e+00   0.0287 *
## Temp                       1.847e-15  1.085e-15  1.703e+00   0.0904 .
## Humidity                   8.986e-16  4.336e-16  2.072e+00   0.0398 *
## Precip                    -3.526e-16  5.421e-16 -6.500e-01   0.5163
## Visibility                 3.415e-15  3.265e-15  1.046e+00   0.2971
## Public_Holiday            -1.844e-14  2.605e-14 -7.080e-01   0.4798
## Snow                       2.512e-14  2.166e-14  1.160e+00   0.2477
## Was_Offense_Completed_Yes  1.000e+00  1.654e-16  6.047e+15   <2e-16 ***
## Was_Offense_Completed_No   1.000e+00  4.861e-16  2.057e+15   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.661e-14 on 170 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 5.084e+30 on 9 and 170 DF,  p-value: < 2.2e-16

layout(matrix(c(1, 2, 3, 4), nrow = 2, byrow = TRUE))
plot(model)
```



#USING LINEAR REGRESSION PREDICTING THE CRIME COUNT ON SPECIFIC DATE

```
summarized_data$Days_Since_Start <-
as.numeric(difftime(summarized_data$Occurred_Date,
min(summarized_data$Occurred_Date), units = "days"))

# Selecting independent variables for the model
independent_vars <- c("Days_Since_Start", "Temp", "Humidity", "Precip",
"Visibility", "Public_Holiday", "Snow", "Was_Offense_Completed_Yes",
"Was_Offense_Completed_No")

# Dependent variables
dependent_vars <- c('Sum_Serious_Crime', 'Sum_Violent_Crime')

# Split the dataset into training and testing sets
set.seed(42)  # Set seed for reproducibility
train_indices <- sample(1:nrow(summarized_data), 0.8 * nrow(summarized_data))
train_data <- summarized_data[train_indices, ]
test_data <- summarized_data[-train_indices, ]


# Create and train the linear regression models
models <- lapply(dependent_vars, function(dep_var) {
  model <- lm(formula(paste(dep_var, "~", paste(independent_vars, collapse =
" + "))), data = train_data)
  return(model)
```

```r
})

prediction_date <- as.Date('2023-01-01')
# Calculate the number of days since the start of the dataset for the
prediction date
days_since_start <- as.numeric(difftime(prediction_date,
min(summarized_data$Occurred_Date), units = "days"))

# Since we don't have future weather data or other variables, we'll have to
make assumptions.
# One approach is to use the mean values of the weather variables and other
conditions from the dataset.
mean_values <- colMeans(subset(summarized_data, select = independent_vars),
na.rm = TRUE)

# Update the 'Days_Since_Start' for the prediction date
mean_values['Days_Since_Start'] <- days_since_start

# Reshape for a single prediction
prediction_input <- as.data.frame(t(mean_values))

# Predicting the crime for the specified date
predicted_crime <- lapply(models, function(model) {
  predict(model, newdata = prediction_input)
})

# Display the predictions
for (i in seq_along(dependent_vars)) {
  cat(paste("Predicted",dependent_vars[i], "on",prediction_date , ":",
predicted_crime[[i]], "\n"))
}
```

```
## Predicted Sum_Serious_Crime on 2023-01-01 : 90.6079366793486
## Predicted Sum_Violent_Crime on 2023-01-01 : 94.1120633206513
```

```r
cat("\n")

# Assuming you have already loaded required libraries like ggplot2

# Add the Days_Since_Start variable to the dataset
summarized_data$Days_Since_Start <-
as.numeric(difftime(summarized_data$Occurred_Date,
min(summarized_data$Occurred_Date), units = "days"))

# Selecting independent variables for the model
independent_vars <- c("Days_Since_Start", "Temp", "Humidity", "Precip",
"Visibility", "Public_Holiday", "Snow", "Was_Offense_Completed_Yes",
"Was_Offense_Completed_No")

# Dependent variables
```

```r
dependent_vars <- c('Sum_Serious_Crime', 'Sum_Violent_Crime')

# Split the dataset into training and testing sets
set.seed(42)  # Set seed for reproducibility
train_indices <- sample(1:nrow(summarized_data), 0.8 * nrow(summarized_data))
train_data <- summarized_data[train_indices, ]
test_data <- summarized_data[-train_indices, ]

# Create and train the linear regression models
models <- lapply(dependent_vars, function(dep_var) {
  model <- lm(formula(paste(dep_var, "~", paste(independent_vars, collapse =
" + "))), data = train_data)
  return(model)
})

# Specify the prediction date
prediction_date <- as.Date('2023-01-01')

# Calculate the number of days since the start of the dataset for the
prediction date
days_since_start <- as.numeric(difftime(prediction_date,
min(summarized_data$Occurred_Date), units = "days"))

# Since we don't have future weather data or other variables, we'll have to
make assumptions.
# One approach is to use the mean values of the weather variables and other
conditions from the dataset.
mean_values <- colMeans(subset(summarized_data, select = independent_vars),
na.rm = TRUE)

# Update the 'Days_Since_Start' for the prediction date
mean_values['Days_Since_Start'] <- days_since_start

# Reshape for a single prediction
prediction_input <- as.data.frame(t(mean_values))

# Predicting the crime for the specified date
predicted_crime <- lapply(models, function(model) {
  predict(model, newdata = prediction_input)
})

# Display the predictions
for (i in seq_along(dependent_vars)) {
  cat(paste("Predicted", dependent_vars[i], "on", prediction_date, ":",
predicted_crime[[i]], "\n"))
}

## Predicted Sum_Serious_Crime on 2023-01-01 : 90.6079366793486
## Predicted Sum_Violent_Crime on 2023-01-01 : 94.1120633206513
```

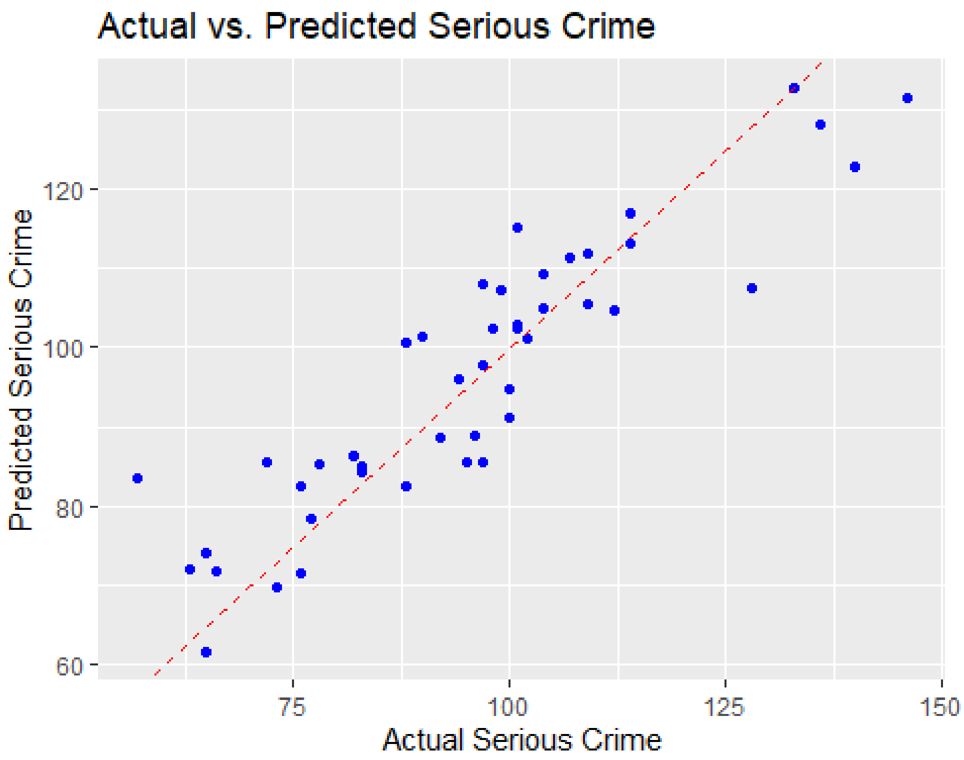
```r
# Make predictions on the test dataset
test_predictions <- lapply(models, function(model) {
  predict(model, newdata = test_data)
})

# Create a data frame for actual vs. predicted values
comparison_data <- data.frame(
  Actual_Serious_Crime = test_data$Sum_Serious_Crime,
  Actual_Violent_Crime = test_data$Sum_Violent_Crime,
  Predicted_Serious_Crime = test_predictions[[1]],
  Predicted_Violent_Crime = test_predictions[[2]]
)

# Print actual vs. predicted values
print("Actual vs. Predicted Values:")
```

```
## [1] "Actual vs. Predicted Values:"
```

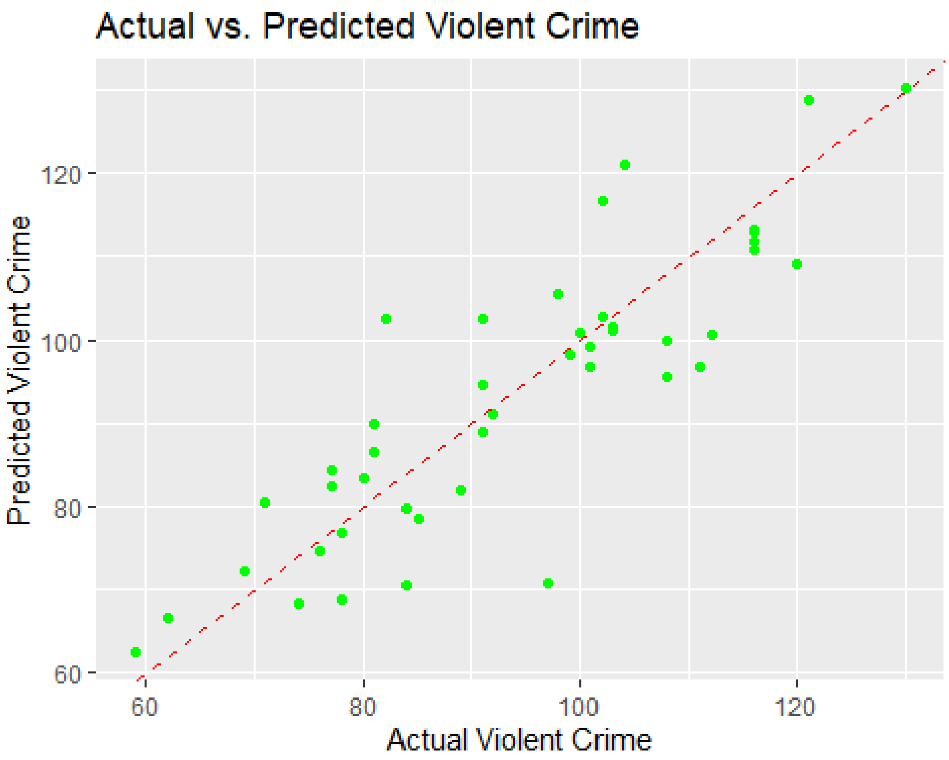
```r
print(head(comparison_data))
```

```
##   Actual_Serious_Crime Actual_Violent_Crime Predicted_Serious_Crime
## 1                   94                  101                95.93807
## 2                   88                   81                82.45461
## 3                  104                  116               109.28602
## 4                  133                  130               132.74704
## 5                  102                  102               101.18559
## 6                  114                  116               116.84176
##   Predicted_Violent_Crime
## 1                99.06193
## 2                86.54539
## 3               110.71398
## 4               130.25296
## 5               102.81441
## 6               113.15824
```

```r
# Plot actual vs. predicted values
library(ggplot2)

ggplot(comparison_data, aes(x = Actual_Serious_Crime, y =
Predicted_Serious_Crime)) +
  geom_point(color = "blue") +
  geom_abline(intercept = 0, slope = 1, linetype = "dashed", color = "red") +
  labs(title = "Actual vs. Predicted Serious Crime",
       x = "Actual Serious Crime",
       y = "Predicted Serious Crime")
```



```
ggplot(comparison_data, aes(x = Actual_Violent_Crime, y =
Predicted_Violent_Crime)) +
  geom_point(color = "green") +
  geom_abline(intercept = 0, slope = 1, linetype = "dashed", color = "red") +
  labs(title = "Actual vs. Predicted Violent Crime",
       x = "Actual Violent Crime",
       y = "Predicted Violent Crime")
```



#USING LINEAR REGRESSION PREDICTING THE CRIME COUNT FOR WHOLE WEEK

```
predict_for_date <- function(prediction_date, models, independent_vars,
summarized_data) {
  days_since_start <- as.numeric(difftime(prediction_date,
min(summarized_data$Occurred_Date), units = "days"))
  mean_values <- colMeans(subset(summarized_data, select = independent_vars),
na.rm = TRUE)
  mean_values['Days_Since_Start'] <- days_since_start
  prediction_input <- as.data.frame(t(mean_values))
  predicted_crime <- lapply(models, function(model) predict(model, newdata =
prediction_input))
  return(predicted_crime)
}

# Dates for prediction
prediction_dates <- as.Date(c('2023-07-22', '2023-07-23', '2023-07-24',
'2023-07-25', '2023-07-26', '2023-07-27', '2023-07-28', '2023-07-29', '2023-
07-30'))

# Loop over prediction dates and display predictions
for (i in seq_along(prediction_dates)) {
  formatted_date <- format(prediction_dates[i], "%Y-%m-%d")
  cat("Predicted Sum_Serious_Crime on", formatted_date, ":",
predict_for_date(prediction_dates[i], models, independent_vars,
summarized_data)[[1]], "\n")
  cat("Predicted Sum_Violent_Crime on", formatted_date, ":",
predict_for_date(prediction_dates[i], models, independent_vars,
summarized_data)[[2]], "\n")
  cat("\n")
}
```

```
## Predicted Sum_Serious_Crime on 2023-07-22 : 97.47695
## Predicted Sum_Violent_Crime on 2023-07-22 : 87.24305
##
## Predicted Sum_Serious_Crime on 2023-07-23 : 97.51095
## Predicted Sum_Violent_Crime on 2023-07-23 : 87.20905
##
## Predicted Sum_Serious_Crime on 2023-07-24 : 97.54496
## Predicted Sum_Violent_Crime on 2023-07-24 : 87.17504
##
## Predicted Sum_Serious_Crime on 2023-07-25 : 97.57896
## Predicted Sum_Violent_Crime on 2023-07-25 : 87.14104
##
## Predicted Sum_Serious_Crime on 2023-07-26 : 97.61297
## Predicted Sum_Violent_Crime on 2023-07-26 : 87.10703
##
## Predicted Sum_Serious_Crime on 2023-07-27 : 97.64697
## Predicted Sum_Violent_Crime on 2023-07-27 : 87.07303
##
## Predicted Sum_Serious_Crime on 2023-07-28 : 97.68098
## Predicted Sum_Violent_Crime on 2023-07-28 : 87.03902
##
## Predicted Sum_Serious_Crime on 2023-07-29 : 97.71498
## Predicted Sum_Violent_Crime on 2023-07-29 : 87.00502
##
## Predicted Sum_Serious_Crime on 2023-07-30 : 97.74899
## Predicted Sum_Violent_Crime on 2023-07-30 : 86.97101
```

#Random Forest Regression for Serious Crime Model

```
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
```

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

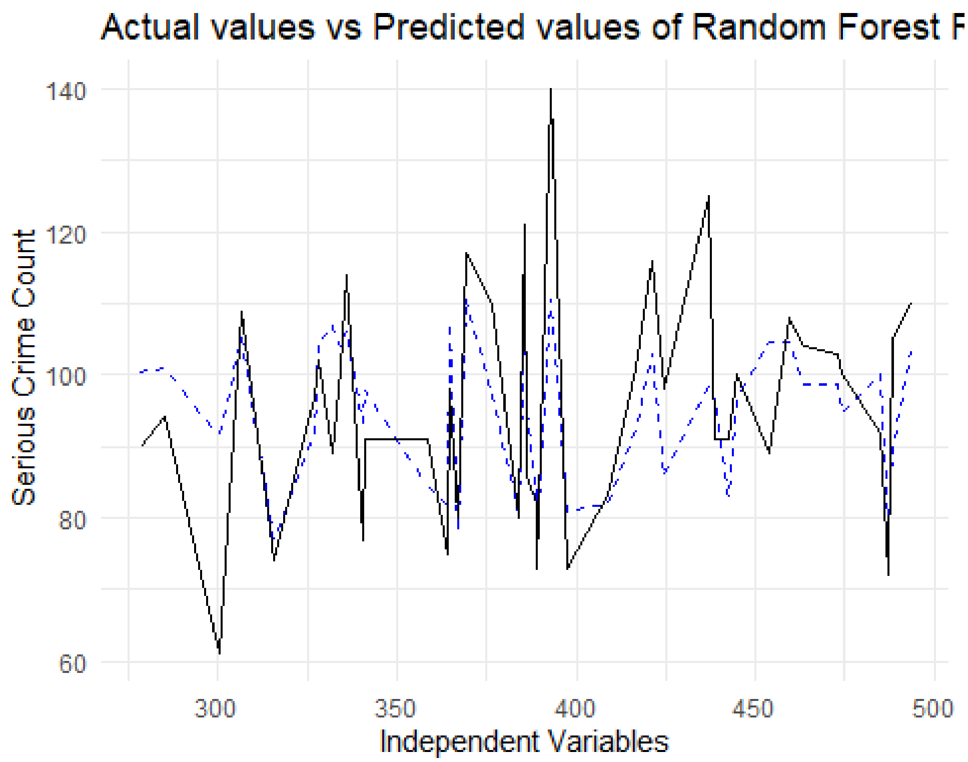
```
##
## Attaching package: 'lubridate'
```

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

```
library(randomForest)
library(caret)
```

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

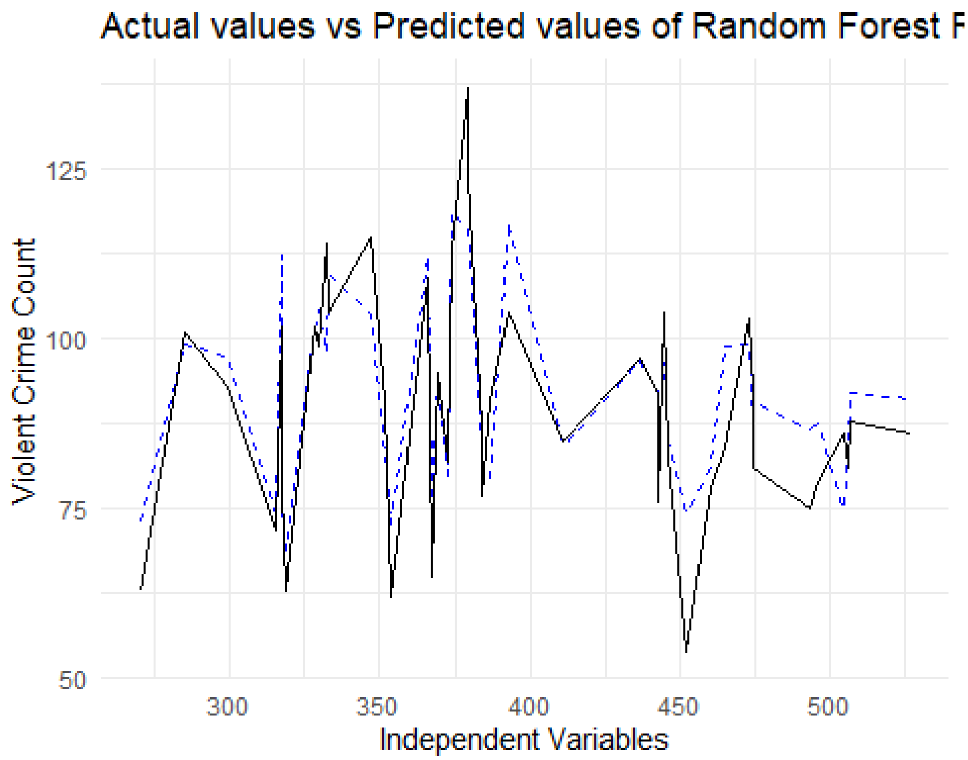
```
crime_data <- summarized_data

# Set seed for reproducibility
crime_data=summarized_data
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, ]
trainIndex <- createDataPartition(crime_data$Sum_Serious_Crime, p = 0.8,
                                  list = FALSE,
                                  times = 1)
dataTrain <- crime_data[trainIndex, ]
dataTest <- crime_data[-trainIndex, ]



rf_model <- randomForest(Sum_Serious_Crime ~ Temp + Snow + Humidity + Precip
+ Was_Offense_Completed_Yes + Was_Offense_Completed_No + Days_Since_Start +
Visibility, data = dataTrain, ntree = 100)
rf_predictions <- predict(rf_model, newdata = dataTest)

# Plotting

ggplot(dataTest) +
  geom_line(aes(x = Temp + Snow + Humidity + Precip +
Was_Offense_Completed_Yes + Was_Offense_Completed_No + Days_Since_Start +
Visibility, y = rf_predictions), color = "blue", linetype = "dashed") +
  geom_line(aes(x = Temp + Snow + Humidity + Precip +
Was_Offense_Completed_Yes + Was_Offense_Completed_No + Days_Since_Start +
Visibility, y = dataTest$Sum_Serious_Crime), color = "black") +
  labs(title = "Actual values vs Predicted values of Random Forest Regression
model Serious Crime model", x = "Independent Variables", y = " Serious Crime
Count") +
  theme_minimal()

## Warning: Use of `dataTest$Sum_Serious_Crime` is discouraged.
## ℹ Use `Sum_Serious_Crime` instead.
```



```
rf_predictions <- predict(rf_model, testingSet)
mse <- mean((rf_predictions - testingSet$Sum_Serious_Crime)^2)
rsq <- cor(rf_predictions, testingSet$Sum_Serious_Crime)^2

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

## [1] "Mean Squared Error: 56.3811823612398"

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

## [1] "R-squared: 0.908896466724577"
```

#Random Forest Regression for Violent Crime Model

```r
library(randomForest)

# Load necessary libraries
library(readr)
library(dplyr)
library(lubridate)
library(randomForest)
library(caret)
crime_data <- summarized_data
trainIndex <- createDataPartition(crime_data$Sum_Violent_Crime, p = 0.8,
                                  list = FALSE,
                                  times = 1)
dataTrain <- crime_data[trainIndex, ]
dataTest <- crime_data[-trainIndex, ]



rf_model <- randomForest(Sum_Violent_Crime ~ Temp + Snow + Humidity + Precip
+ Was_Offense_Completed_Yes + Was_Offense_Completed_No + Days_Since_Start +
Visibility, data = dataTrain, ntree = 100)
rf_predictions <- predict(rf_model, newdata = dataTest)

# Plotting

ggplot(dataTest) +
  geom_line(aes(x = Temp + Snow + Humidity + Precip +
Was_Offense_Completed_Yes + Was_Offense_Completed_No + Days_Since_Start +
Visibility, y = rf_predictions), color = "blue", linetype = "dashed") +
  geom_line(aes(x = Temp + Snow + Humidity + Precip +
Was_Offense_Completed_Yes + Was_Offense_Completed_No + Days_Since_Start +
Visibility, y = dataTest$Sum_Violent_Crime), color = "black") +
  labs(title = "Actual values vs Predicted values of Random Forest Regression
model Violent Crime model", x = "Independent Variables", y = "Violent Crime
Count") +
  theme_minimal()

## Warning: Use of `dataTest$Sum_Violent_Crime` is discouraged.
## ℹ Use `Sum_Violent_Crime` instead.
```



```
rf_predictions <- predict(rf_model, testingSet)
mse <- mean((rf_predictions - testingSet$Sum_Violent_Crime)^2)
rsq <- cor(rf_predictions, testingSet$Sum_Violent_Crime)^2

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

## [1] "Mean Squared Error: 25.3462900254642"

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

## [1] "R-squared: 0.920380237510544"
```

Random Forest Model for Total Crime

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

# Set seed for reproducibility
crime_data=summarized_data
set.seed(42)

# Calculate the size of the training set (80% of the dataset)
```

```r
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(Total_Crime ~Temp + Snow + Humidity +
Precip+Was_Offense_Completed_Yes+Was_Offense_Completed_No+Days_Since_Start+Vi
sibility, data = trainingSet, ntree = 100)

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

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

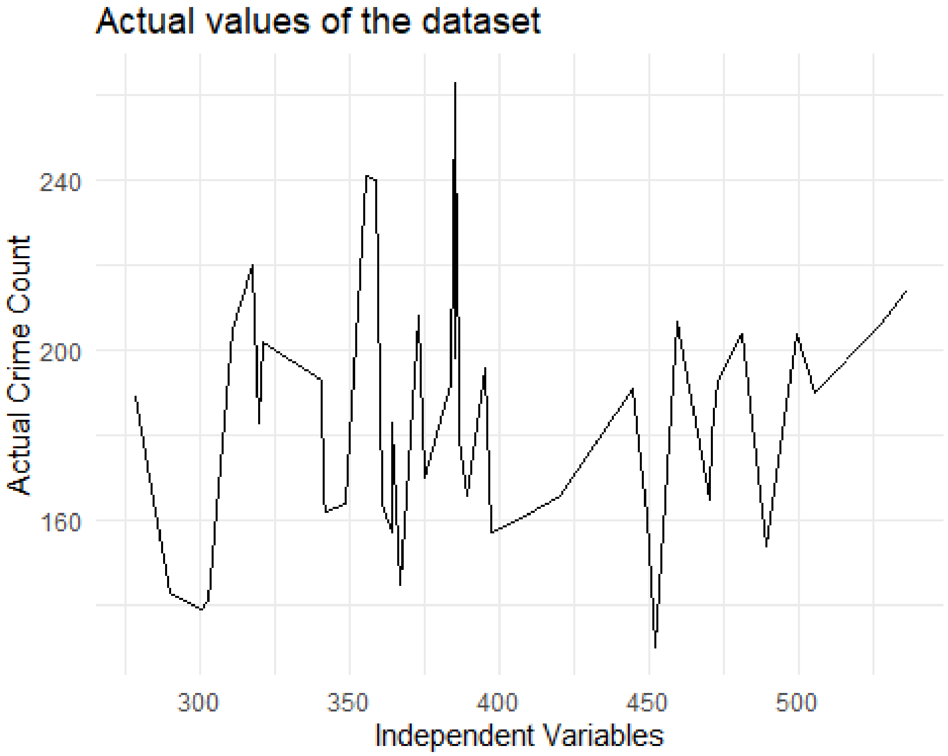
```
## [1] "Mean Squared Error: 200.718061858294"
```

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

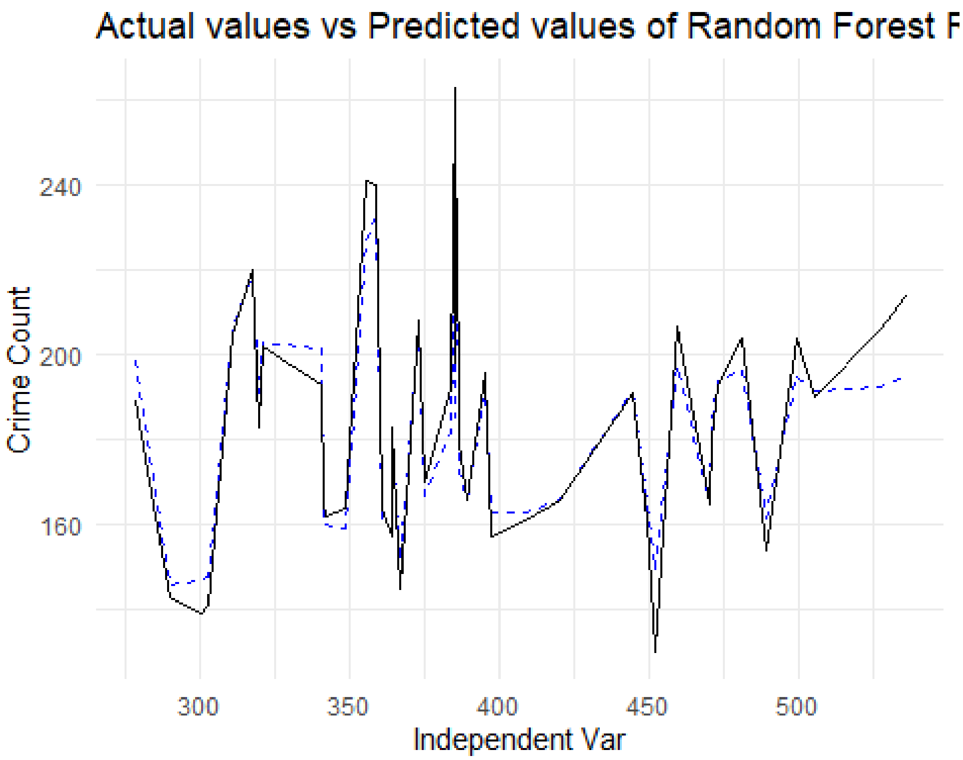
```
## [1] "R-squared: 0.871362040544095"
```

```r
crime_data <- summarized_data
trainIndex <- createDataPartition(crime_data$Total_Crime, p = 0.8,
                                  list = FALSE,
                                  times = 1)
dataTrain <- crime_data[trainIndex, ]
dataTest <- crime_data[-trainIndex, ]



rf_model <- randomForest(Total_Crime ~ Temp + Snow + Humidity + Precip +
Was_Offense_Completed_Yes + Was_Offense_Completed_No + Days_Since_Start +
Visibility, data = dataTrain, ntree = 100)
rf_predictions <- predict(rf_model, newdata = dataTest)
```

```r
# Plotting
ggplot(dataTest) +
  geom_line(aes(x = Temp + Snow + Humidity + Precip +
Was_Offense_Completed_Yes + Was_Offense_Completed_No + Days_Since_Start +
Visibility, y = Total_Crime), color = "black") +
  labs(title = "Actual values of the dataset", x = "Independent Variables", y
= "Actual Crime Count") +
  theme_minimal()
```



```r
ggplot(dataTest) +
  geom_line(aes(x = Temp + Snow + Humidity + Precip +
Was_Offense_Completed_Yes + Was_Offense_Completed_No + Days_Since_Start +
Visibility, y = rf_predictions), color = "blue", linetype = "dashed") +
  geom_line(aes(x = Temp + Snow + Humidity + Precip +
Was_Offense_Completed_Yes + Was_Offense_Completed_No + Days_Since_Start +
Visibility, y = dataTest$Total_Crime), color = "black") +
  labs(title = "Actual values vs Predicted values of Random Forest Regression
Total Crime model", x = "Independent Var", y = "Crime Count") +
  theme_minimal()

## Warning: Use of `dataTest$Total_Crime` is discouraged.
## ℹ Use `Total_Crime` instead.
```



```
#install.packages("knitr")
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(tibble)

# Load the dataset
data <- sarimadf

# Load required libraries
library(readr)
library(forecast)
library(dplyr)
library(knitr)


data$Occurred_Date <- as.Date(data$Occurred_Date)

# Set the index to Occurred_Date
```

```r
data <- data %>%
  as_tibble() %>%
  mutate(Occurred_Date = lubridate::ymd(Occurred_Date)) %>%
  select(Occurred_Date, Total_Crime) %>%
  arrange(Occurred_Date)

# Splitting the dataset into training and testing sets
train_end <- as.Date('2023-07-21')
test_start <- as.Date('2023-07-22')
test_end <- as.Date('2023-07-30')

train_data <- data %>%
  filter(Occurred_Date <= train_end) %>%
  pull(Total_Crime)

test_data <- data %>%
  filter(Occurred_Date >= test_start & Occurred_Date <= test_end) %>%
  pull(Total_Crime)

# Fitting the SARIMA model with the specified parameters (1, 1, 2)(1, 1, 1,
7)
model <- forecast::Arima(train_data, order=c(1, 1, 2), seasonal=c(1, 1, 1,
7))

# Forecasting the Total Crime from July 22, 2023, to July 30, 2023
forecast_values <- forecast::forecast(model, h = length(test_data))
forecast_df <- data.frame(
  Date = seq(test_start, test_end, by = "days"),
  Forecast = as.numeric(forecast_values$mean),
  Lower_CI = as.numeric(forecast_values$lower),
  Upper_CI = as.numeric(forecast_values$upper),
  check.names = FALSE
)

# Creating a DataFrame for visualization
comparison_df <- data.frame(
  Date = seq(test_start, test_end, by = "days"),
  Actual = test_data,
  Forecasted = forecast_df$Forecast
)

# Plotting the forecasts along with the actual data
library(ggplot2)

plot <- ggplot(comparison_df, aes(x = Date)) +
  geom_line(aes(y = Actual), color = 'blue', linetype = 'solid', size = 1,
label = "Actual") +
  geom_line(aes(y = Forecasted), color = 'green', linetype = 'solid', size =
1, label = "Forecasted") +
  labs(title = 'Total Crime Forecast vs Actuals', x = 'Date', y = 'Total
Crime') +
  theme_minimal() +
  theme(legend.position = "bottom")
```

```
## Warning in geom_line(aes(y = Actual), color = "blue", linetype = "solid",
:
## Ignoring unknown parameters: `label`
```

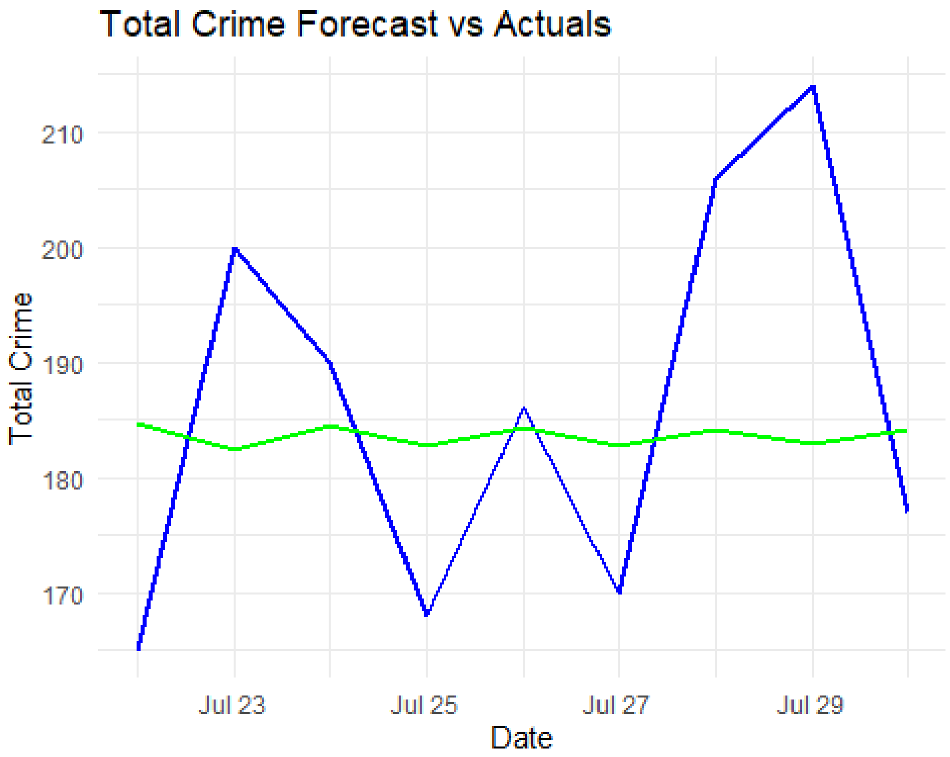
```
## Warning in geom_line(aes(y = Forecasted), color = "green", linetype =
"solid",
## : Ignoring unknown parameters: `label`
```

```r
table_df <- data.frame(
  Date = comparison_df$Date,
  Actual = comparison_df$Actual,
  Predicted = comparison_df$Forecasted
)

# Display the table
knitr::kable(table_df, caption = "Actual vs Predicted Values")
```

Actual vs Predicted Values

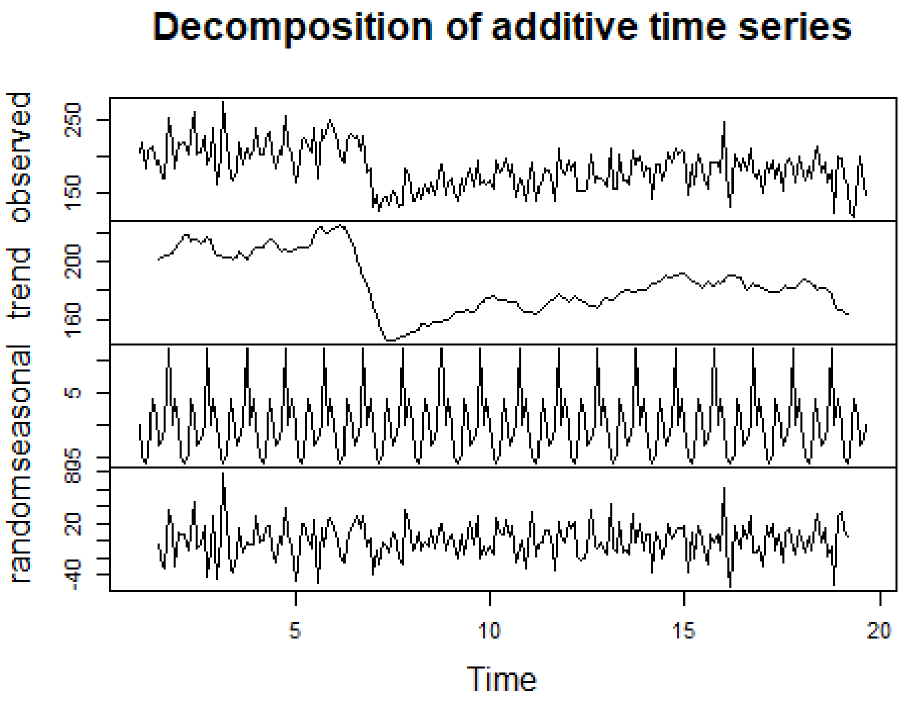
Date	Actual	Predicted
2023-07-22	165	184.7350
2023-07-23	200	182.4717
2023-07-24	190	184.5202
2023-07-25	168	182.6661
2023-07-26	186	184.3442
2023-07-27	170	182.8254
2023-07-28	206	184.2000
2023-07-29	214	182.9559
2023-07-30	177	184.0819
2023-07-22	165	184.7350
2023-07-23	200	182.4717
2023-07-24	190	184.5202
2023-07-25	168	182.6661
2023-07-26	186	184.3442
2023-07-27	170	182.8254
2023-07-28	206	184.2000
2023-07-29	214	182.9559
2023-07-30	177	184.0819

```
# Display the plot
print(plot)
```



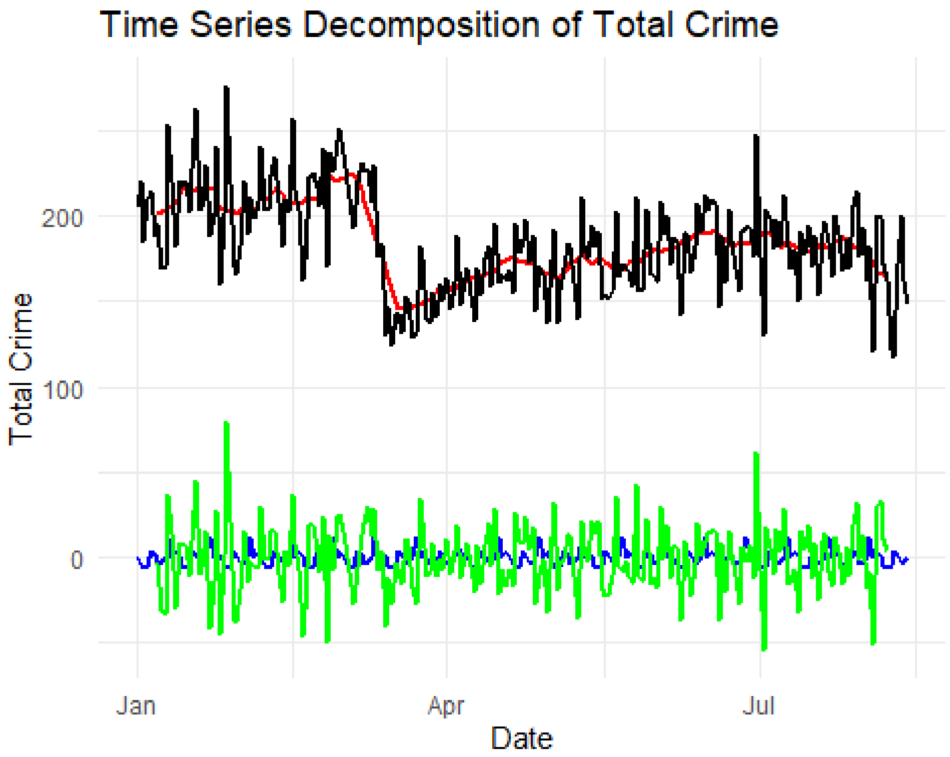
#Time series Decomposition

```
# Load necessary libraries

library(ggplot2)

# Assuming your dataset is already loaded as 'tc' and 'Total_Crime' is the
target variable

data <- sarimadf

# Convert 'Occurred_Date' to Date if it's not already in Date format
data$Occurred_Date <- as.Date(data$Occurred_Date)

# Create a time series object
crime_ts <- ts(data$Total_Crime, frequency = 12)  # Assuming daily data

# Perform time series decomposition
decomposition <- decompose(crime_ts)

# Plot the decomposition
plot(decomposition)
```



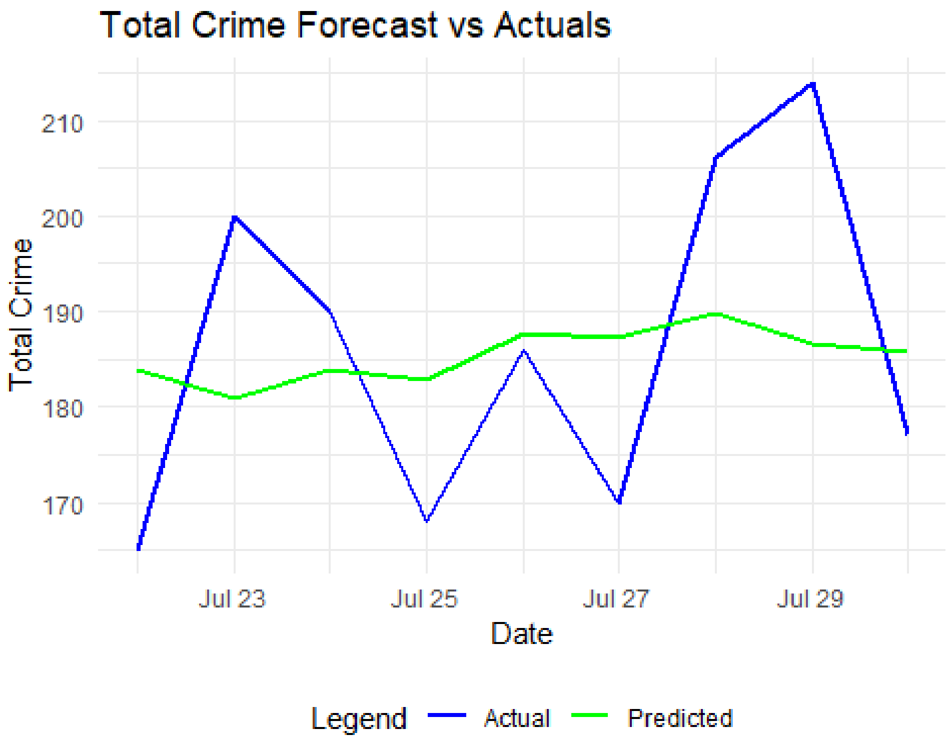
```
# Plot individual components using ggplot2
ggplot() +
  geom_line(aes(x = data$Occurred_Date, y = decomposition$seasonal), color =
'blue', linetype = 'solid', size = 1, label = "Seasonal") +
  geom_line(aes(x = data$Occurred_Date, y = decomposition$trend), color =
'red', linetype = 'solid', size = 1, label = "Trend") +
  geom_line(aes(x = data$Occurred_Date, y = decomposition$random), color =
'green', linetype = 'solid', size = 1, label = "Residual") +
  geom_line(aes(x = data$Occurred_Date, y = data$Total_Crime), color =
'black', linetype = 'solid', size = 1, label = "Original") +
  labs(title = 'Time Series Decomposition of Total Crime', x = 'Date', y =
'Total Crime') +
  theme_minimal() +
  theme(legend.position = "bottom")
```

```
## Warning in geom_line(aes(x = data$Occurred_Date, y =
decomposition$seasonal), :
## Ignoring unknown parameters: `label`

## Warning in geom_line(aes(x = data$Occurred_Date, y = decomposition$trend),
:
## Ignoring unknown parameters: `label`

## Warning in geom_line(aes(x = data$Occurred_Date, y =
decomposition$random), :
## Ignoring unknown parameters: `label`
## Warning in geom_line(aes(x = data$Occurred_Date, y = data$Total_Crime), :
## Ignoring unknown parameters: `label`

## Don't know how to automatically pick scale for object of type <ts>.
Defaulting
## to continuous.

## Warning: Removed 12 rows containing missing values (`geom_line()`).

## Warning: Removed 12 rows containing missing values (`geom_line()`).
```



#ARIMA analysis for whole dataset

```
# Load libraries
library(forecast)
library(tibble)
library(lubridate)
library(ggplot2)
library(dplyr)


# Load the dataset
data <- sarimadf
data$Occurred_Date <- as.Date(data$Occurred_Date)

# Set the index to Occurred_Date
data <- data %>%
  mutate(Occurred_Date = lubridate::ymd(Occurred_Date)) %>%
  select(Occurred_Date, Total_Crime, Temp) %>%
  arrange(Occurred_Date)

# Splitting the dataset into training and testing sets
train_end <- as.Date('2023-07-21')
test_start <- as.Date('2023-07-22')
test_end <- as.Date('2023-07-30')

train_data <- data %>%
  filter(Occurred_Date <= train_end) %>%
  select(Total_Crime, Temp)

test_data <- data %>%
  filter(Occurred_Date %in% seq(test_start, test_end, by = "days")) %>%
  select(Total_Crime, Temp)

# Fitting the ARIMA model with the specified parameters (1, 1, 2)
model <- forecast::Arima(train_data$Total_Crime, order=c(1, 1, 2), xreg =
train_data$Temp)

# Forecasting
forecast_values <- forecast::forecast(model, h = nrow(test_data), xreg =
test_data$Temp)

# Create DataFrames
forecast_df <- data.frame(
  Date = seq(test_start, test_end, by = "days"),
  Forecast = as.numeric(forecast_values$mean),
  Lower_CI = as.numeric(forecast_values$lower),
  Upper_CI = as.numeric(forecast_values$upper),
  check.names = FALSE
)

comparison_df <- data.frame(
  Date = forecast_df$Date,
  Actual = test_data$Total_Crime,
  Forecasted = forecast_df$Forecast
)

# Plotting
plot <- ggplot(comparison_df, aes(x = Date)) +
  geom_line(aes(y = Actual, color = 'Actual'), linetype = 'solid', size = 1)
+
  geom_line(aes(y = Forecasted, color = 'Predicted'), linetype = 'solid',
size = 1) +
  labs(title = 'Total Crime Forecast vs Actuals', x = 'Date', y = 'Total
Crime') +
  theme_minimal() +
  theme(legend.position = "bottom") +
  scale_color_manual(values = c('Actual' = 'blue', 'Predicted' = 'green')) +
  guides(color = guide_legend(title = "Legend", override.aes = list(linetype
= 'solid', size = 2)))

# Display the plot
print(plot)
```



```
table_df <- data.frame(
  Date = comparison_df$Date,
  Actual = comparison_df$Actual,
  Predicted = comparison_df$Forecasted
)

# Print the table without kableExtra
print(table_df)
```

```
##          Date Actual Predicted
## 1  2023-07-22    165  183.8315
## 2  2023-07-23    200  180.9831
## 3  2023-07-24    190  183.8070
## 4  2023-07-25    168  182.7872
## 5  2023-07-26    186  187.6331
## 6  2023-07-27    170  187.3087
## 7  2023-07-28    206  189.9036
## 8  2023-07-29    214  186.7181
## 9  2023-07-30    177  185.7425
## 10 2023-07-22    165  183.8315
## 11 2023-07-23    200  180.9831
## 12 2023-07-24    190  183.8070
## 13 2023-07-25    168  182.7872
## 14 2023-07-26    186  187.6331
## 15 2023-07-27    170  187.3087
## 16 2023-07-28    206  189.9036
## 17 2023-07-29    214  186.7181
## 18 2023-07-30    177  185.7425
```

#ARIMA analysis for Beat411

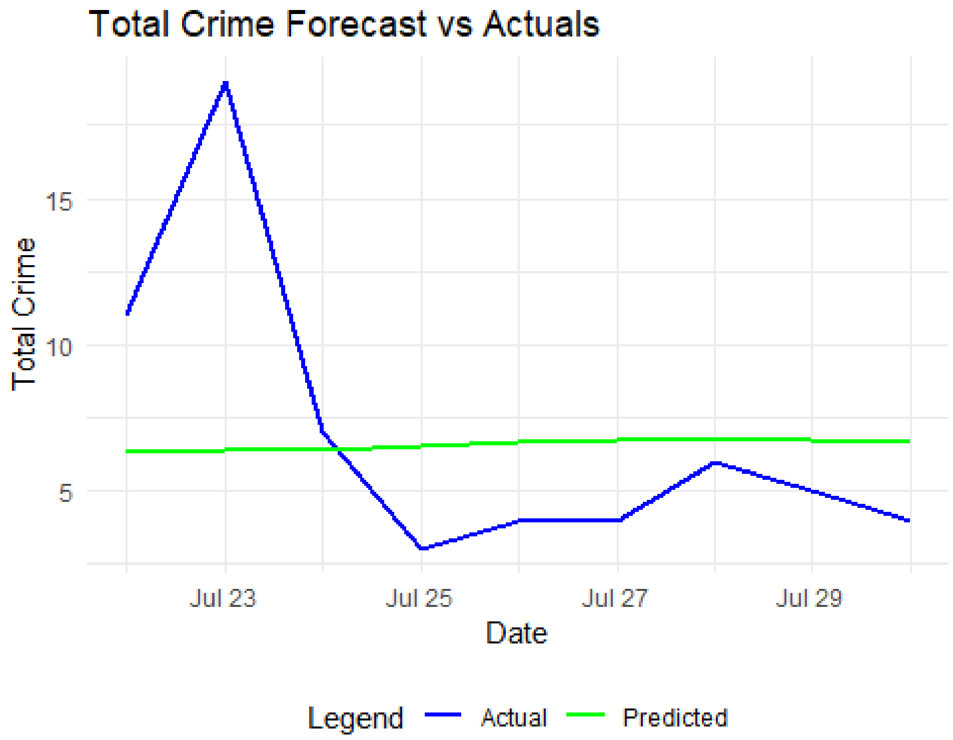
```r
library(forecast)
library(tibble)
library(lubridate)
library(ggplot2)
library(dplyr)

# Load the dataset
data <- Beat411

data$Occurred_Date <- as.Date(data$Occurred_Date)

# Set the index to Occurred_Date
data <- data %>%
  as_tibble() %>%
  mutate(Occurred_Date = lubridate::ymd(Occurred_Date)) %>%
  select(Occurred_Date, Total_Crime, Temp) %>%  # Use the correct column name (Temp)
  arrange(Occurred_Date)

# Splitting the dataset into training and testing sets
train_end <- as.Date('2023-07-21')
test_start <- as.Date('2023-07-22')
test_end <- as.Date('2023-07-30')

train_data <- data %>%
  filter(Occurred_Date <= train_end) %>%
  select(Total_Crime, Temp)  # Use the correct column name (Temp)

test_data <- data %>%
  filter(Occurred_Date >= test_start & Occurred_Date <= test_end) %>%
  select(Total_Crime, Temp)  # Use the correct column name (Temp)

# Fitting the ARIMA model with the specified parameters (1, 1, 2)
model <- forecast::Arima(train_data$Total_Crime, order=c(1, 1, 2), xreg = train_data$Temp)

# Print the summary of the ARIMA model
cat("ARIMA Model Summary:\n")
```

```
## ARIMA Model Summary:
```

```r
print(summary(model))
```

```
## Series: train_data$Total_Crime
## Regression with ARIMA(1,1,2) errors
##
## Coefficients:
##          ar1      ma1     ma2    xreg
##       0.9293  -1.8208  0.8209  0.0392
## s.e.  0.0648   0.0944  0.0938  0.0515
##
## sigma^2 = 11.65:  log likelihood = -523.9
## AIC=1057.8   AICc=1058.11   BIC=1074.24
##
## Training set error measures:
##                       ME     RMSE      MAE      MPE     MAPE      MASE
## Training set -0.1918451 3.370498 2.743232 -42.3081 65.64167 0.7715341
##                     ACF1
## Training set 0.01172366
```

```r
# Forecasting the Total Crime from July 22, 2023, to July 30, 2023
forecast_values <- forecast::forecast(model, h = nrow(test_data), xreg =
test_data$Temp)
forecast_df <- data.frame(
  Date = seq(test_start, test_end, by = "days"),
  Forecast = as.numeric(forecast_values$mean),
  Lower_CI = as.numeric(forecast_values$lower),
  Upper_CI = as.numeric(forecast_values$upper),
  check.names = FALSE
)

# Creating a DataFrame for visualization
comparison_df <- data.frame(
  Date = seq(test_start, test_end, by = "days"),
  Actual = test_data$Total_Crime,
  Forecasted = forecast_df$Forecast
)

# Plotting the forecasts along with the actual data
plot <- ggplot(comparison_df, aes(x = Date)) +
  geom_line(aes(y = Actual, color = 'Actual'), linetype = 'solid', size = 1)
+
  geom_line(aes(y = Forecasted, color = 'Predicted'), linetype = 'solid',
size = 1) +
  labs(title = 'Total Crime Forecast vs Actuals', x = 'Date', y = 'Total
Crime') +
  theme_minimal() +
  theme(legend.position = "bottom") +
  scale_color_manual(values = c('Actual' = 'blue', 'Predicted' = 'green')) +
  guides(color = guide_legend(title = "Legend", override.aes = list(linetype
= 'solid', size = 2)))

# Display the plot
print(plot)
```



```
table_df <- data.frame(
  Date = comparison_df$Date,
  Actual = comparison_df$Actual,
  Predicted = comparison_df$Forecasted
)

# Print the table without kableExtra
print(table_df)

##          Date Actual Predicted
## 1  2023-07-22     11  6.324266
## 2  2023-07-23     19  6.364702
## 3  2023-07-24      7  6.417672
## 4  2023-07-25      3  6.502986
## 5  2023-07-26      4  6.640441
## 6  2023-07-27      4  6.732283
## 7  2023-07-28      6  6.794361
## 8  2023-07-29      5  6.764156
## 9  2023-07-30      4  6.700589
## 10 2023-07-22     11  6.324266
## 11 2023-07-23     19  6.364702
## 12 2023-07-24      7  6.417672
## 13 2023-07-25      3  6.502986
## 14 2023-07-26      4  6.640441
## 15 2023-07-27      4  6.732283
## 16 2023-07-28      6  6.794361
## 17 2023-07-29      5  6.764156
## 18 2023-07-30      4  6.700589
```