Exploring Crime Trends in Montgomery County

Ray Leika

2025-04-07

Crime, Home Value, and School Exposure: Insights from a Spatial Analysis of Montgomery County

##Exploring crime trends and their effects on communities across Montgomery County

#Introduction This project explores patterns of crime across Montgomery County, Maryland, and investigates how these patterns relate to community-level factors such as housing and public school locations. Understanding where and when crime occurs, and how it may intersect with local resources and neighborhood conditions, is essential for supporting safer, more equitable communities.

Using data from sources including the Montgomery County Open Data Portal and the U.S. Census Bureau, the analysis focuses on identifying crime trends over time, locating geographic hotspots, and comparing crime rates across different cities within the county. Median home value data is used to examine whether areas with higher crime tend to have lower property values, while public school locations are analyzed to understand their proximity to high-crime areas.

```r
getwd()
```

```
## [1] "/Users/leikarayjoseph/Desktop/DATA_205Crime Project"
```

```r
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
setwd("/Users/leikarayjoseph/Desktop/DATA_205Crime Project")
#upload my working directory so I can install my file.
Crime <- read_csv("MC_Crime_DATA.csv")
```

```
## Rows: 439330 Columns: 30
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (23): Offence Code, Dispatch Date / Time, Start_Date_Time, End_Date_Time...
## dbl  (7): Incident ID, CR Number, Victims, Zip Code, Address Number, Latitud...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
Crime
```

```
## # A tibble: 439,330 x 30
##    `Incident ID` `Offence Code` `CR Number` `Dispatch Date / Time`
##            <dbl> <chr>                <dbl> <chr>
##  1     201166610 2308             170548599 <NA>
##  2     201359823 5404             220000965 01/09/2022 01:18:38 AM
##  3     201095140 2303              16043118 <NA>
##  4     201090710 5707              16037677 <NA>
##  5     201362142 2901             220003839 01/28/2022 07:11:29 PM
##  6     201363412 1204             220005400 02/08/2022 05:42:51 AM
##  7     201101176 2303              16050746 <NA>
##  8     201091832 2303              16038909 <NA>
##  9     201296318 2204             200029321 07/28/2020 02:55:00 PM
## 10     201093179 2303              16040649 <NA>
## # i 439,320 more rows
## # i 26 more variables: Start_Date_Time <chr>, End_Date_Time <chr>,
## #   `NIBRS Code` <chr>, Victims <dbl>, `Crime Name1` <chr>,
## #   `Crime Name2` <chr>, `Crime Name3` <chr>, `Police District Name` <chr>,
## #   `Block Address` <chr>, City <chr>, State <chr>, `Zip Code` <dbl>,
## #   Agency <chr>, Place <chr>, Sector <chr>, Beat <chr>, PRA <chr>,
## #   `Address Number` <dbl>, `Street Prefix` <chr>, `Street Name` <chr>, ...
```

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

```
##
## Attaching package: 'plotly'

## The following object is masked from 'package:ggplot2':
##
##     last_plot

## The following object is masked from 'package:stats':
##
##     filter

## The following object is masked from 'package:graphics':
##
##     layout
```

```r
library(leaflet)
library(leaflet.extras)
library(ggmap)
```

```
## i Google's Terms of Service: <https://mapsplatform.google.com>
##   Stadia Maps' Terms of Service: <https://stadiamaps.com/terms-of-service>
##   OpenStreetMap's Tile Usage Policy: <https://operations.osmfoundation.org/policies/tiles>
## i Please cite ggmap if you use it! Use `citation("ggmap")` for details.

##
## Attaching package: 'ggmap'

## The following object is masked from 'package:plotly':
##
```

```
##     wind
```

```r
#putting the headers in lower case
names(Crime) <- tolower(names(Crime))
names(Crime) <- gsub(" ","",names(Crime))
head(Crime)
```

```
## # A tibble: 6 x 30
##   incidentid offencecode  crnumber `dispatchdate/time`    start_date_time
##        <dbl> <chr>           <dbl> <chr>                  <chr>
## 1  201166610 2308        170548599 <NA>                   12/14/2017 04:30:00 PM
## 2  201359823 5404        220000965 01/09/2022 01:18:38 AM 01/09/2022 01:34:00 AM
## 3  201095140 2303         16043118 <NA>                   08/24/2016 09:47:00 PM
## 4  201090710 5707         16037677 <NA>                   07/25/2016 05:31:00 PM
## 5  201362142 2901        220003839 01/28/2022 07:11:29 PM 01/28/2022 02:40:00 PM
## 6  201363412 1204        220005400 02/08/2022 05:42:51 AM 02/08/2022 05:42:00 AM
## # i 25 more variables: end_date_time <chr>, nibrscode <chr>, victims <dbl>,
## #   crimename1 <chr>, crimename2 <chr>, crimename3 <chr>,
## #   policedistrictname <chr>, blockaddress <chr>, city <chr>, state <chr>,
## #   zipcode <dbl>, agency <chr>, place <chr>, sector <chr>, beat <chr>,
## #   pra <chr>, addressnumber <dbl>, streetprefix <chr>, streetname <chr>,
## #   streetsuffix <chr>, streettype <chr>, latitude <dbl>, longitude <dbl>,
## #   policedistrictnumber <chr>, location <chr>
```

```r
Crime$Date <- as.Date(Crime$`dispatchdate/time`, format= "%m/%d/%Y %H:%M:%S %p")

Crime$year <- year(Crime$Date)

head(Crime)
```

```
## # A tibble: 6 x 32
##   incidentid offencecode  crnumber `dispatchdate/time`    start_date_time
##        <dbl> <chr>           <dbl> <chr>                  <chr>
## 1  201166610 2308        170548599 <NA>                   12/14/2017 04:30:00 PM
## 2  201359823 5404        220000965 01/09/2022 01:18:38 AM 01/09/2022 01:34:00 AM
## 3  201095140 2303         16043118 <NA>                   08/24/2016 09:47:00 PM
## 4  201090710 5707         16037677 <NA>                   07/25/2016 05:31:00 PM
## 5  201362142 2901        220003839 01/28/2022 07:11:29 PM 01/28/2022 02:40:00 PM
## 6  201363412 1204        220005400 02/08/2022 05:42:51 AM 02/08/2022 05:42:00 AM
## # i 27 more variables: end_date_time <chr>, nibrscode <chr>, victims <dbl>,
## #   crimename1 <chr>, crimename2 <chr>, crimename3 <chr>,
## #   policedistrictname <chr>, blockaddress <chr>, city <chr>, state <chr>,
## #   zipcode <dbl>, agency <chr>, place <chr>, sector <chr>, beat <chr>,
## #   pra <chr>, addressnumber <dbl>, streetprefix <chr>, streetname <chr>,
## #   streetsuffix <chr>, streettype <chr>, latitude <dbl>, longitude <dbl>,
## #   policedistrictnumber <chr>, location <chr>, Date <date>, year <dbl>
```

```r
Crime$Date <- as.Date(Crime$`dispatchdate/time`, format= "%m/%d/%Y %H:%M:%S %p")

Crime$month <- month(Crime$Date)

head(Crime)
```

```
## # A tibble: 6 x 33
##   incidentid offencecode  crnumber `dispatchdate/time`    start_date_time
##        <dbl> <chr>           <dbl> <chr>                  <chr>
## 1  201166610 2308        170548599 <NA>                   12/14/2017 04:30:00 PM
```

```
## 2  201359823 5404        220000965 01/09/2022 01:18:38 AM 01/09/2022 01:34:00 AM
## 3  201095140 2303         16043118 <NA>                   08/24/2016 09:47:00 PM
## 4  201090710 5707         16037677 <NA>                   07/25/2016 05:31:00 PM
## 5  201362142 2901        220003839 01/28/2022 07:11:29 PM 01/28/2022 02:40:00 PM
## 6  201363412 1204        220005400 02/08/2022 05:42:51 AM 02/08/2022 05:42:00 AM
## # i 28 more variables: end_date_time <chr>, nibrscode <chr>, victims <dbl>,
## #   crimename1 <chr>, crimename2 <chr>, crimename3 <chr>,
## #   policedistrictname <chr>, blockaddress <chr>, city <chr>, state <chr>,
## #   zipcode <dbl>, agency <chr>, place <chr>, sector <chr>, beat <chr>,
## #   pra <chr>, addressnumber <dbl>, streetprefix <chr>, streetname <chr>,
## #   streetsuffix <chr>, streettype <chr>, latitude <dbl>, longitude <dbl>,
## #   policedistrictnumber <chr>, location <chr>, Date <date>, year <dbl>, ...
```

```
# Remove every report from 2025 from my Data
Crime <- subset(Crime, year != 2025)
```

Since the year 2025 is still ongoing, including it in the analysis could lead to misleading results because the data for that year is incomplete. To ensure a more accurate and fair comparison across years, I decided to exclude 2025 from the dataset.

Ruled out all the invalid city names and all the ones that are also not part of Montgomery County

```
Crime <- Crime %>%
  filter(!city %in% c(0, 4, 6, 7, NA))
```

```
# List of cities to exclude
exclude_cities <- c(
  "Mount Rainier", "Alexandria", "Fairfax", "Laurel", "Boyds",
  "Brinklow", "Redland", "District of Columbia", "Hyattsville PG",
  "Riverdale PG", "Washington", "mclean", "Falls Church", "Vienna", "Woodbine","Highland", "Hyattstown"
)

# Clean city names for consistent matching
Crime_cleaned <- Crime %>%
  mutate(city = trimws(tolower(city))) %>%
  filter(!city %in% tolower(trimws(exclude_cities)) & !is.na(city))  # filtering

# View the cleaned dataset
head(Crime_cleaned)
```

```
## # A tibble: 6 x 33
##   incidentid offencecode  crnumber `dispatchdate/time`    start_date_time
##        <dbl> <chr>           <dbl> <chr>                  <chr>
## 1  201359823 5404        220000965 01/09/2022 01:18:38 AM 01/09/2022 01:34:00 AM
## 2  201362142 2901        220003839 01/28/2022 07:11:29 PM 01/28/2022 02:40:00 PM
## 3  201363412 1204        220005400 02/08/2022 05:42:51 AM 02/08/2022 05:42:00 AM
## 4  201296318 2204        200029321 07/28/2020 02:55:00 PM 07/28/2020 02:54:00 PM
## 5  201225360 2204        190004807 01/30/2019 07:49:25 PM 01/30/2019 07:49:00 PM
## 6  201360056 2304        220001336 01/12/2022 02:53:25 AM 01/12/2022 02:53:00 AM
## # i 28 more variables: end_date_time <chr>, nibrscode <chr>, victims <dbl>,
## #   crimename1 <chr>, crimename2 <chr>, crimename3 <chr>,
## #   policedistrictname <chr>, blockaddress <chr>, city <chr>, state <chr>,
## #   zipcode <dbl>, agency <chr>, place <chr>, sector <chr>, beat <chr>,
```

```
## #   pra <chr>, addressnumber <dbl>, streetprefix <chr>, streetname <chr>,
## #   streetsuffix <chr>, streettype <chr>, latitude <dbl>, longitude <dbl>,
## #   policedistrictnumber <chr>, location <chr>, Date <date>, year <dbl>, ...
```

The dataset's City column included several locations outside of Montgomery County and some entries with numbers instead of city names. I cleaned the data carefully to ensure the analysis remains focused and accurate.

```
Crime_Select <- Crime_cleaned %>%
  select(`dispatchdate/time`, start_date_time, end_date_time, victims, crimename1, crimename2, crimenam
Crime_Select
```

```
## # A tibble: 362,953 x 19
##    `dispatchdate/time`    start_date_time       end_date_time victims crimename1
##    <chr>                  <chr>                 <chr>           <dbl> <chr>
##  1 01/09/2022 01:18:38 AM 01/09/2022 01:34:00 ~ <NA>                1 Crime Aga~
##  2 01/28/2022 07:11:29 PM 01/28/2022 02:40:00 ~ 01/28/2022 0~       1 Crime Aga~
##  3 02/08/2022 05:42:51 AM 02/08/2022 05:42:00 ~ <NA>                1 Crime Aga~
##  4 07/28/2020 02:55:00 PM 07/28/2020 02:54:00 ~ 07/28/2020 0~       1 Crime Aga~
##  5 01/30/2019 07:49:25 PM 01/30/2019 07:49:00 ~ <NA>                1 Crime Aga~
##  6 01/12/2022 02:53:25 AM 01/12/2022 02:53:00 ~ 01/12/2022 0~       1 Crime Aga~
##  7 04/29/2017 04:47:44 PM 04/22/2017 03:00:00 ~ 04/25/2017 1~       1 Crime Aga~
##  8 02/09/2022 11:28:51 AM 02/09/2022 11:28:00 ~ 02/09/2022 1~       1 Crime Aga~
##  9 02/16/2022 07:33:27 AM 02/15/2022 10:30:00 ~ 02/16/2022 0~       1 Crime Aga~
## 10 06/05/2021 08:41:55 PM 06/05/2021 08:58:00 ~ <NA>                1 Crime Aga~
## # i 362,943 more rows
## # i 14 more variables: crimename2 <chr>, crimename3 <chr>,
## #   policedistrictname <chr>, city <chr>, zipcode <dbl>, agency <chr>,
## #   place <chr>, latitude <dbl>, longitude <dbl>, policedistrictnumber <chr>,
## #   location <chr>, Date <date>, year <dbl>, month <dbl>
```

```
# Count the variable "Crimename1" by year.
Year_count <- Crime_Select |>
  group_by(crimename1, year) |>
  count() |> # The number of crime for each crimename1 by year.
  arrange(n) # Arrange in ascending order.

  Year_count
```

```
## # A tibble: 32 x 3
## # Groups:   crimename1, year [32]
##    crimename1                 year     n
##    <chr>                     <dbl> <int>
##  1 Crime Against Not a Crime  2021   431
##  2 Crime Against Not a Crime  2017   459
##  3 Crime Against Not a Crime  2023   493
##  4 Crime Against Not a Crime  2020   511
##  5 Crime Against Not a Crime  2024   532
##  6 Crime Against Not a Crime  2018   591
##  7 Crime Against Not a Crime  2022   617
##  8 Crime Against Not a Crime  2019   622
##  9 Crime Against Person       2017  3428
## 10 Crime Against Person       2020  4298
## # i 22 more rows
```

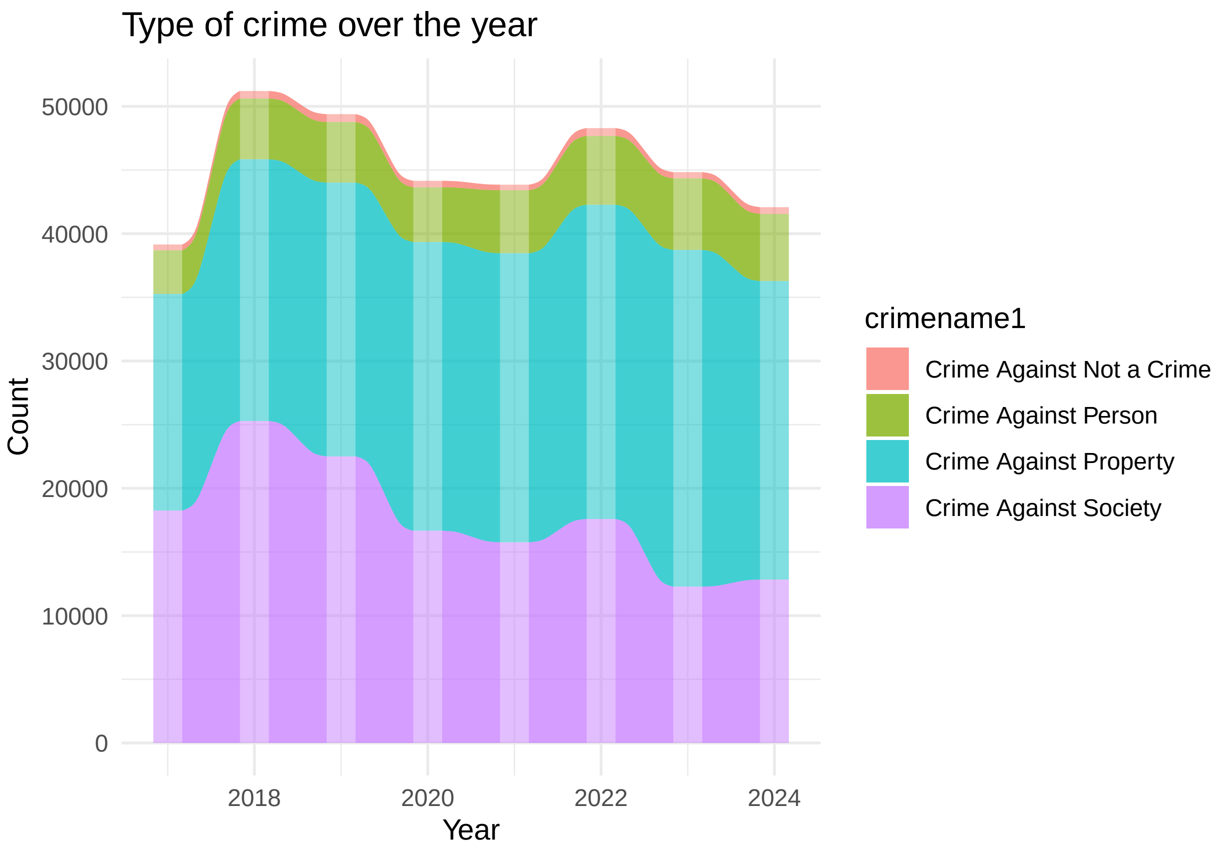
```
# Count the variable "crimename1" to see wich type of crime happend the most.
Crime_count1 <- Crime_Select |>
```

```
 group_by(crimename1) |>
 count() |>
  # The variable crimename1 for each type of crime.
 arrange(n)
# Arrange in ascending order.

 Crime_count1
```

```
## # A tibble: 4 x 2
## # Groups:   crimename1 [4]
##   crimename1                      n
##   <chr>                       <int>
## 1 Crime Against Not a Crime    4256
## 2 Crime Against Person        38502
## 3 Crime Against Society      141175
## 4 Crime Against Property     179020
```

Looking at the dataset, most crimes are related to property with 179,020 incidents, followed by crimes that impact society with 141,175 incidents, like drug offenses and public disturbances. There are also quite a few crimes against people with 38,502 incidents, such as assaults and robberies. The "Not a Crime" category with 4,256 incidents includes cases that were misclassified or don't actually fit the definition of a crime.

```
plot1 <- ggplot(data = Year_count, aes(x = year,
           y = n,
           alluvium= crimename1,
           fill = crimename1, label = crimename1)) +
  geom_alluvium() +
  geom_flow() +
  #geom_stratum(alpha = 0.5) +
   labs(x= "Year",
          y= "Count",
          title = "Type of crime over the year") +
          #caption = "source: DATA MONTGOMERY") +
  theme_minimal()
 #ggtitle("Type of Crime over the Year")

plot1
```



Crime against property is the most common type of crime and has been growing over the years.

See what is part of the not a crime category:

```r
# Filter for 'Not a Crime' category
not_a_crime <- subset(Crime_Select, crimename1 == "Crime Against Not a Crime")

# View first few rows
head(not_a_crime)
```

```
## # A tibble: 6 x 19
##   `dispatchdate/time`    start_date_time        end_date_time victims crimename1
##   <chr>                  <chr>                  <chr>           <dbl> <chr>
## 1 11/05/2024 09:40:43 PM 11/05/2024 07:23:00 PM <NA>                1 Crime Aga~
## 2 07/31/2020 04:38:50 PM 07/31/2020 04:38:00 PM <NA>                1 Crime Aga~
## 3 07/07/2024 03:15:38 PM 07/07/2024 03:15:00 PM <NA>                1 Crime Aga~
## 4 07/10/2024 03:13:31 AM 07/10/2024 03:13:00 AM <NA>                1 Crime Aga~
## 5 11/18/2020 09:26:50 PM 11/18/2020 09:26:00 PM <NA>                1 Crime Aga~
## 6 11/05/2024 09:54:56 PM 11/05/2024 09:54:00 PM <NA>                1 Crime Aga~
## # i 14 more variables: crimename2 <chr>, crimename3 <chr>,
## #   policedistrictname <chr>, city <chr>, zipcode <dbl>, agency <chr>,
## #   place <chr>, latitude <dbl>, longitude <dbl>, policedistrictnumber <chr>,
## #   location <chr>, Date <date>, year <dbl>, month <dbl>
```

Where does most of the Crime Againts Property happend:

```r
total_property_crime <- Crime_Select |>
  filter(crimename1 == "Crime Against Property") |>
  nrow()

Crime_Place <- Crime_Select |>
  filter(crimename1 == "Crime Against Property") |>
  count(place, sort = TRUE) |>
  slice_max(n, n = 10) |>
  mutate(percentage = (n / total_property_crime) * 100,
         percentage = sprintf("%.1f%%", percentage))
Crime_Place
```

```
## # A tibble: 10 x 3
##    place                                  n percentage
##    <chr>                              <int> <chr>
##  1 Parking Lot - Residential          18939 10.6%
##  2 Other/Unknown                      18904 10.6%
##  3 Street - Residential               16802 9.4%
##  4 Residence - Single Family          12160 6.8%
##  5 Residence - Driveway               11095 6.2%
##  6 Residence - Apartment/Condo        10427 5.8%
##  7 Retail - Department/Discount Store  9128 5.1%
##  8 Parking Lot - Commercial            6843 3.8%
##  9 Grocery/Supermarket                 5200 2.9%
## 10 Retail - Mall                       4390 2.5%
```

```r
ggplot(Crime_Place, aes(x = reorder(place, n), y = n)) +
  geom_col(fill = "steelblue") +
  geom_text(aes(label = percentage),
            hjust = -0.1,  # pushes text slightly outside the bar
            size = 2) +
  coord_flip() +
  labs(
    title = "Top 10 Places Where Crimes Against Property Occur",
    x = "Place",
    y = "Number of Crimes"
  ) +
  theme_minimal()
```

Most crimes against property happen in residential parking lots.This may be because these areas are often less monitored, especially at night. People might also leave valuables in their cars, making them easy targets for theft.

see the most commun crime in the crimename2 column

```
# Count the variable "crimename1" to see wich type of crime happend the most.
Crime_count_xx <- Crime_Select |>
 group_by(crimename2) |>
 count() |>
  # The variable crimename1 for each type of crime.
 arrange(n)
# Arrange in ascending order.

 Crime_count_xx
```

```
## # A tibble: 54 x 2
## # Groups:   crimename2 [54]
##    crimename2                                      n
##    <chr>                                       <int>
##  1 "Bribery"                                       1
##  2 "Operating/Promoting/Assisting Gambling"        1
##  3 "\"Human Trafficking, Involuntary Servitude\""  2
##  4 "Negligent Manslaughter"                        4
##  5 "Incest"                                        9
##  6 "Justifiable Homicide"                         13
```

```
##  7 "Curfew/Loitering/Vagrancy Violations"           33
##  8 "Welfare Fraud"                                   44
##  9 "Hacking/Computer Invasion"                       45
## 10 "\"Human Trafficking, Commercial Sex Acts\""      65
## # i 44 more rows
```

Use the top 10 most common type of crime and visualize the trend over the years.

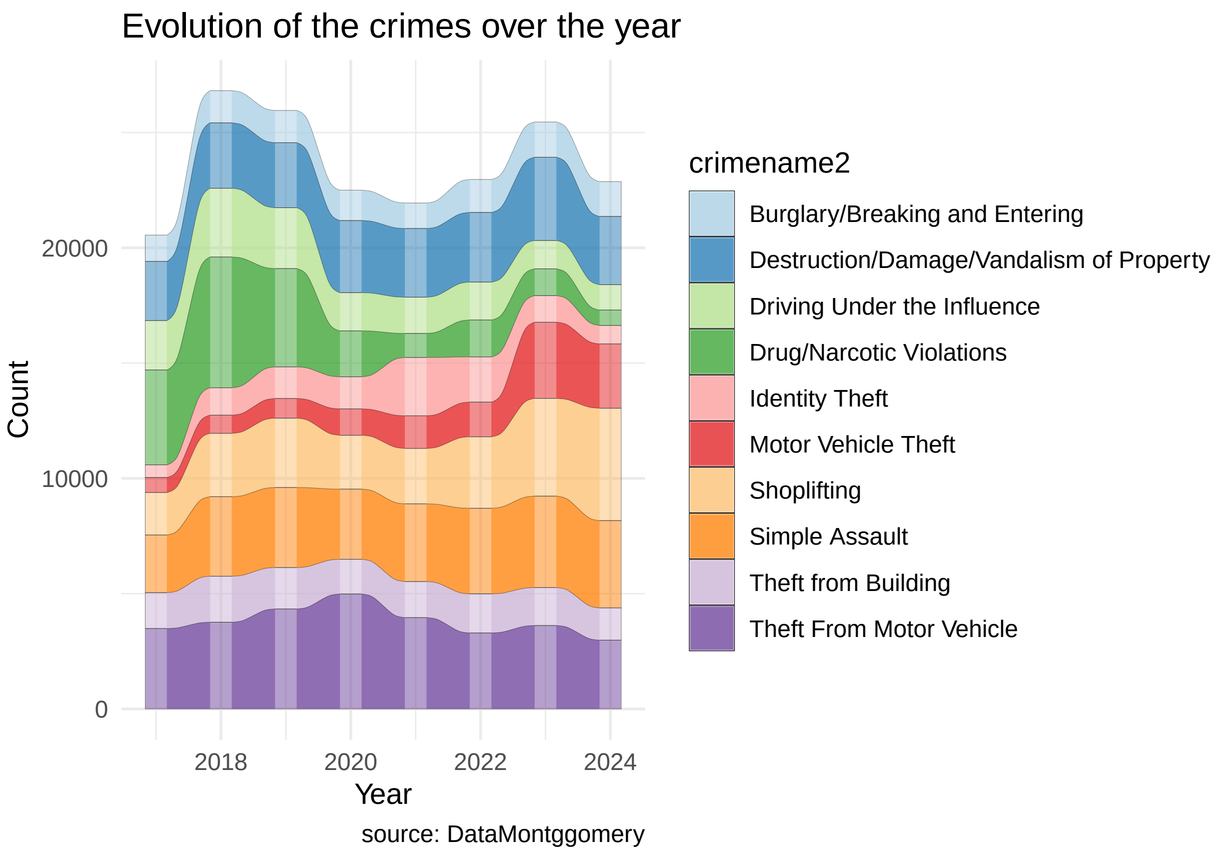
```
# Count the variable "crimename2" by year.
Year_count2 <- Crime |>
filter(crimename2 %in% c("Theft From Motor Vehicle",
                          "Simple Assault",
                          "Shoplifting",
                          "Destruction/Damage/Vandalism of Property",
                          "Drug/Narcotic Violations",
                          "Driving Under the Influence",
                         "Theft from Building",
                         "Motor Vehicle Theft",
                         "Identity Theft",
                         "Burglary/Breaking and Entering")) |>
  group_by(crimename2, year) |>
  count() |> # The variable crimename2 for each year.
  arrange(n) # Arrange in ascending order.

  Year_count2
```

```
## # A tibble: 80 x 3
## # Groups:   crimename2, year [80]
##    crimename2                      year     n
##    <chr>                          <dbl> <int>
##  1 Identity Theft                  2017   550
##  2 Motor Vehicle Theft             2017   651
##  3 Drug/Narcotic Violations        2024   668
##  4 Motor Vehicle Theft             2018   789
##  5 Identity Theft                  2024   801
##  6 Motor Vehicle Theft             2019   854
##  7 Drug/Narcotic Violations        2021  1043
##  8 Driving Under the Influence     2024  1100
##  9 Burglary/Breaking and Entering  2021  1106
## 10 Burglary/Breaking and Entering  2017  1140
## # i 70 more rows
```

```
plot2 <- ggplot(data = Year_count2, aes(x = year,
           y = n,
           alluvium= crimename2,
           fill = crimename2, label = crimename2)) +
  geom_alluvium(color= "black", size= 0.1) +
  geom_flow() +
  #geom_stratum(alpha = 0.5) +
  scale_fill_brewer(palette = "Paired")+ # add color palette
   labs(x= "Year",
          y= "Count",
          title = "Evolution of the crimes over the year",
          caption = "source: DataMontggomery") +
  theme_minimal()
```

```
## 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.
```

```
plot2
```



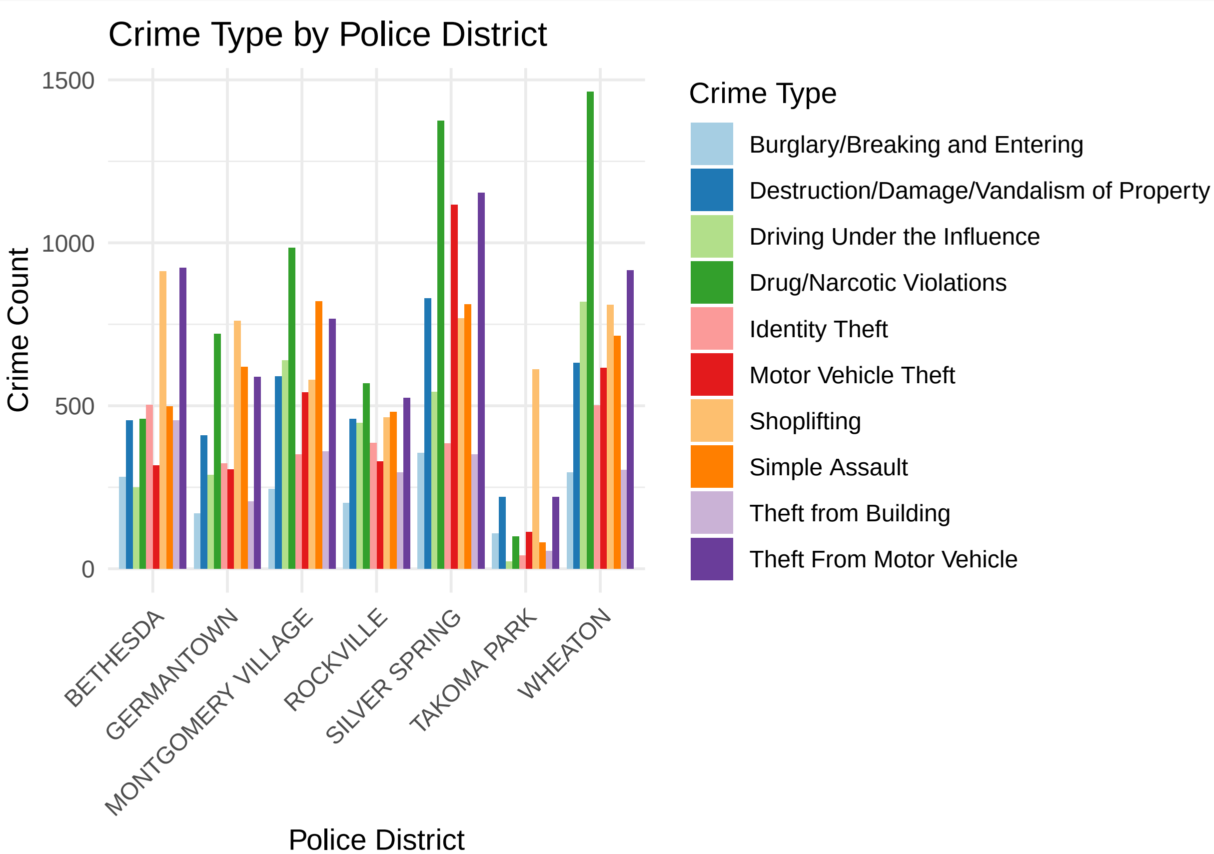
Some crimes increased while others decreased over the years. In both plots, we can see that shoplifting kept rising each year. Drug and narcotic violations dropped a lot compared to before. Robbery went up a little, but the number is still low compared to other crimes.Inflation,post pandemic. comparing commun value

```
# Count some of the types of crime per year by police district name
Yearly_crime <- Crime_Select |>
filter(crimename2 %in% c("Theft From Motor Vehicle",
                          "Simple Assault",
                          "Shoplifting",
                          "Destruction/Damage/Vandalism of Property",
                          "Drug/Narcotic Violations",
                          "Driving Under the Influence",
                         "Theft from Building",
                         "Motor Vehicle Theft",
                         "Identity Theft",
                         "Burglary/Breaking and Entering"),
         policedistrictname != "OTHER") |>
  group_by(crimename2, year, policedistrictname) |>
  count() |>
  arrange(n) # Arrange in ascending order
```

```r
Yearly_crime
```

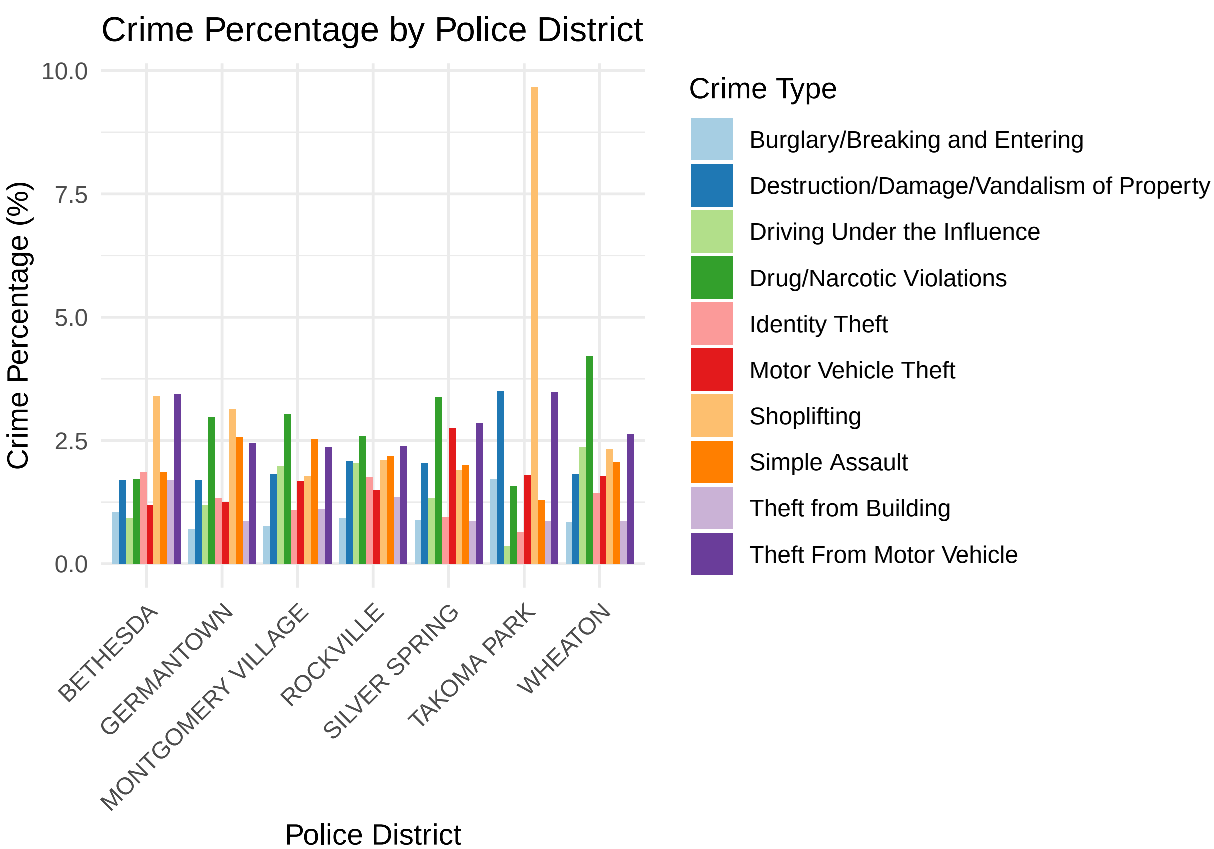
```
## # A tibble: 560 x 4
## # Groups:   crimename2, year, policedistrictname [560]
##    crimename2                   year policedistrictname     n
##    <chr>                       <dbl> <chr>              <int>
##  1 Driving Under the Influence  2024 TAKOMA PARK            6
##  2 Driving Under the Influence  2023 TAKOMA PARK            7
##  3 Identity Theft               2017 TAKOMA PARK            7
##  4 Identity Theft               2018 TAKOMA PARK            9
##  5 Driving Under the Influence  2022 TAKOMA PARK           10
##  6 Drug/Narcotic Violations     2021 TAKOMA PARK           13
##  7 Drug/Narcotic Violations     2024 TAKOMA PARK           13
##  8 Driving Under the Influence  2020 TAKOMA PARK           15
##  9 Driving Under the Influence  2021 TAKOMA PARK           15
## 10 Driving Under the Influence  2019 TAKOMA PARK           18
## # i 550 more rows
```

```r
ggplot(Yearly_crime, aes(x = policedistrictname, y = n, fill = crimename2)) +
  geom_bar(stat = "identity", position = "dodge") +
  scale_fill_brewer(palette = "Paired")+ # add color palette
  labs(x = "Police District", y = "Crime Count", fill = "Crime Type",
       title = "Crime Type by Police District") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



This plot shows that Wheaton and Silver Spring experience the highest number of crimes among all police districts, with shoplifting and theft from motor vehicles being particularly common. Takoma Park, on the other hand, consistently reports much lower crime counts. While certain crime types like shoplifting remain high across most districts, other types such as drug/narcotic violations and motor vehicle theft are more prominent in specific areas like Silver Spring. Overall, the distribution of crime types varies slightly by district, but the most heavily affected areas are clearly identifiable.

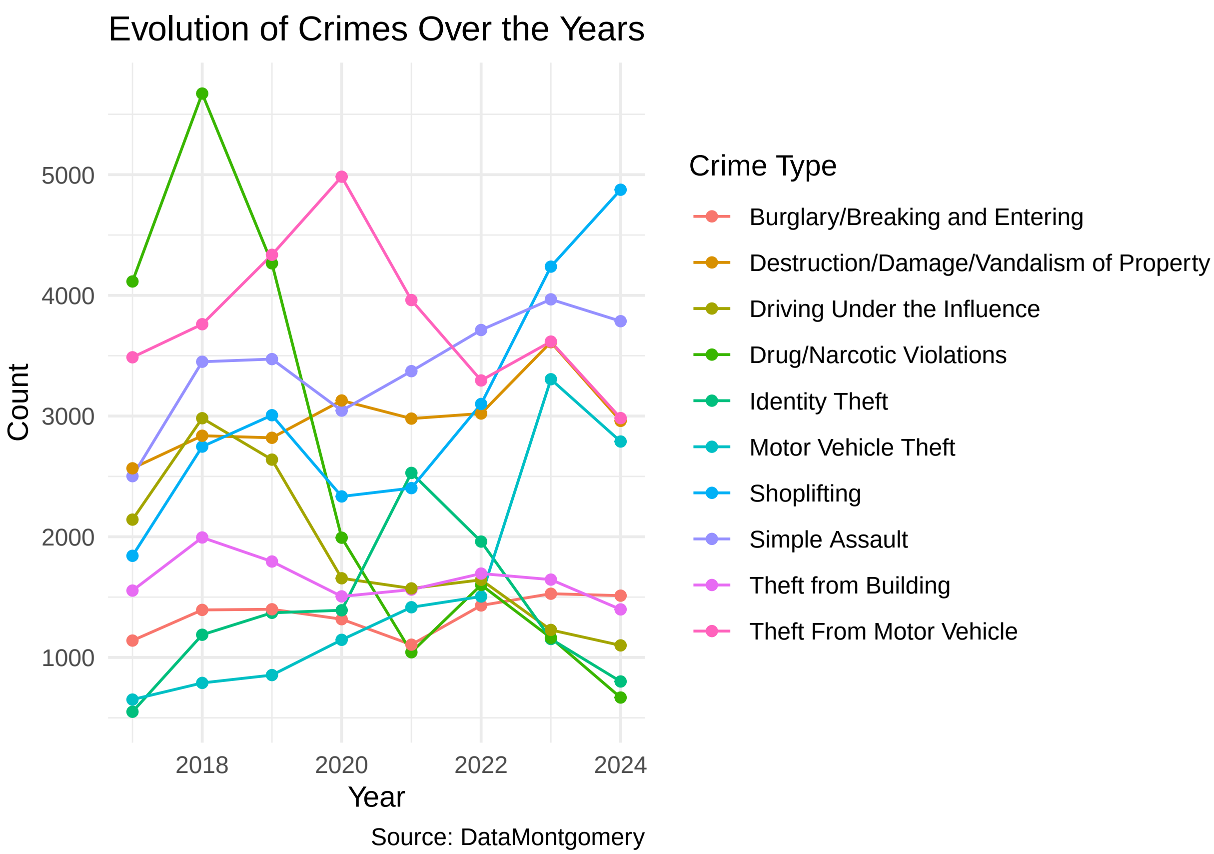
```
Yearly_crime1 <- Yearly_crime |>
  group_by(policedistrictname) |>
  mutate(total_crimes_in_district = sum(n),  # Total crimes in each district
         crime_percentage = (n / total_crimes_in_district) * 100)  # Crime percentage

# Step 2: Plot the data with crime percentages
ggplot(Yearly_crime1, aes(x = policedistrictname, y = crime_percentage, fill = crimename2)) +
  geom_bar(stat = "identity", position = "dodge") +
  scale_fill_brewer(palette = "Paired")+ # add color palette
  labs(x = "Police District", y = "Crime Percentage (%)", fill = "Crime Type",
       title = "Crime Percentage by Police District") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



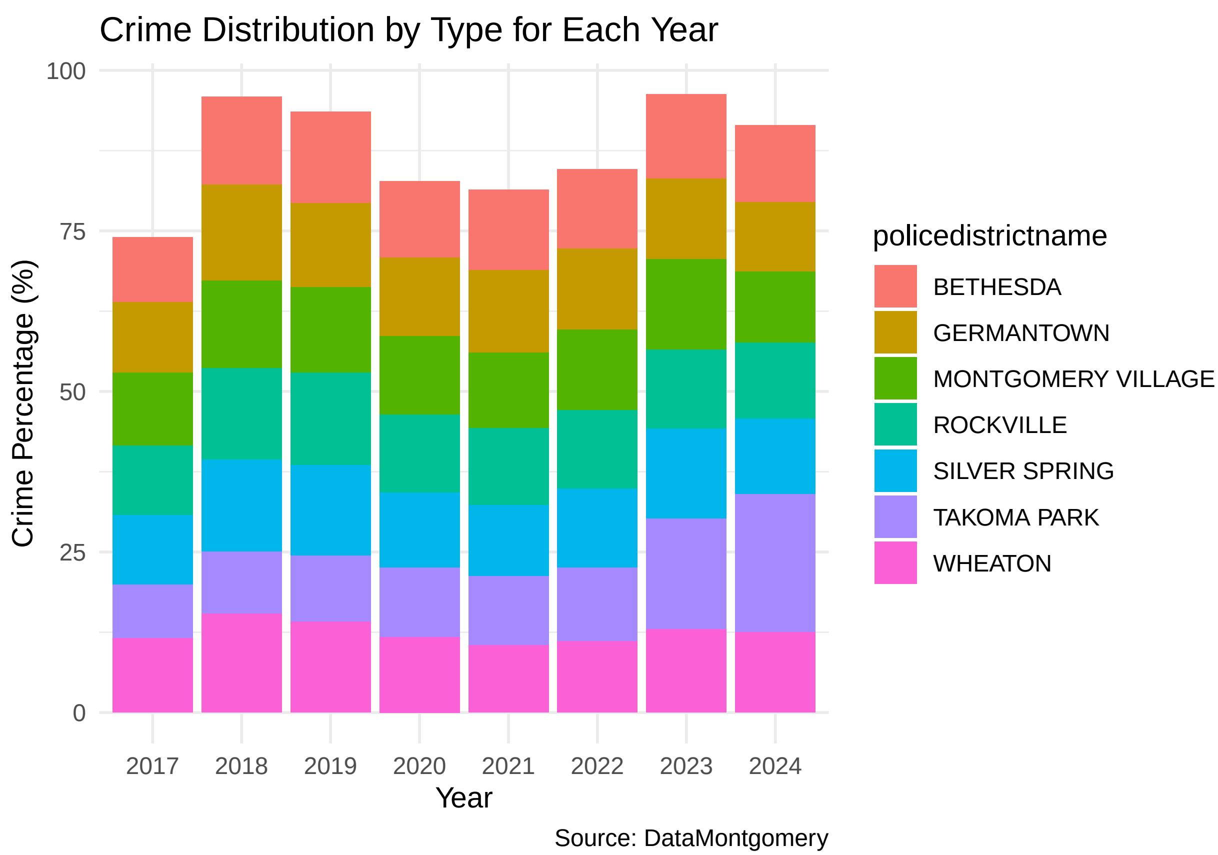
Compared to the previous plot showing the raw number of crimes, this percentage-based plot highlights the relative importance of each crime type within each district. Even though Silver Spring and Wheaton had higher absolute crime counts before, when adjusting for the total number of crimes, the differences between districts are less dramatic. One standout observation is that shoplifting represents a much larger share of total crime in Takoma Park, even though the district has fewer crimes overall.

How has crime changed over the years?

```
plot2.0 <- ggplot(data = Year_count2, aes(x = year, y = n, color = crimename2)) +
  geom_line() +
  geom_point() +
  labs(
    x = "Year",
    y = "Count",
    title = "Evolution of Crimes Over the Years",
    caption = "Source: DataMontgomery",
    color = "Crime Type"
  ) +
  theme_minimal()
plot2.0
```

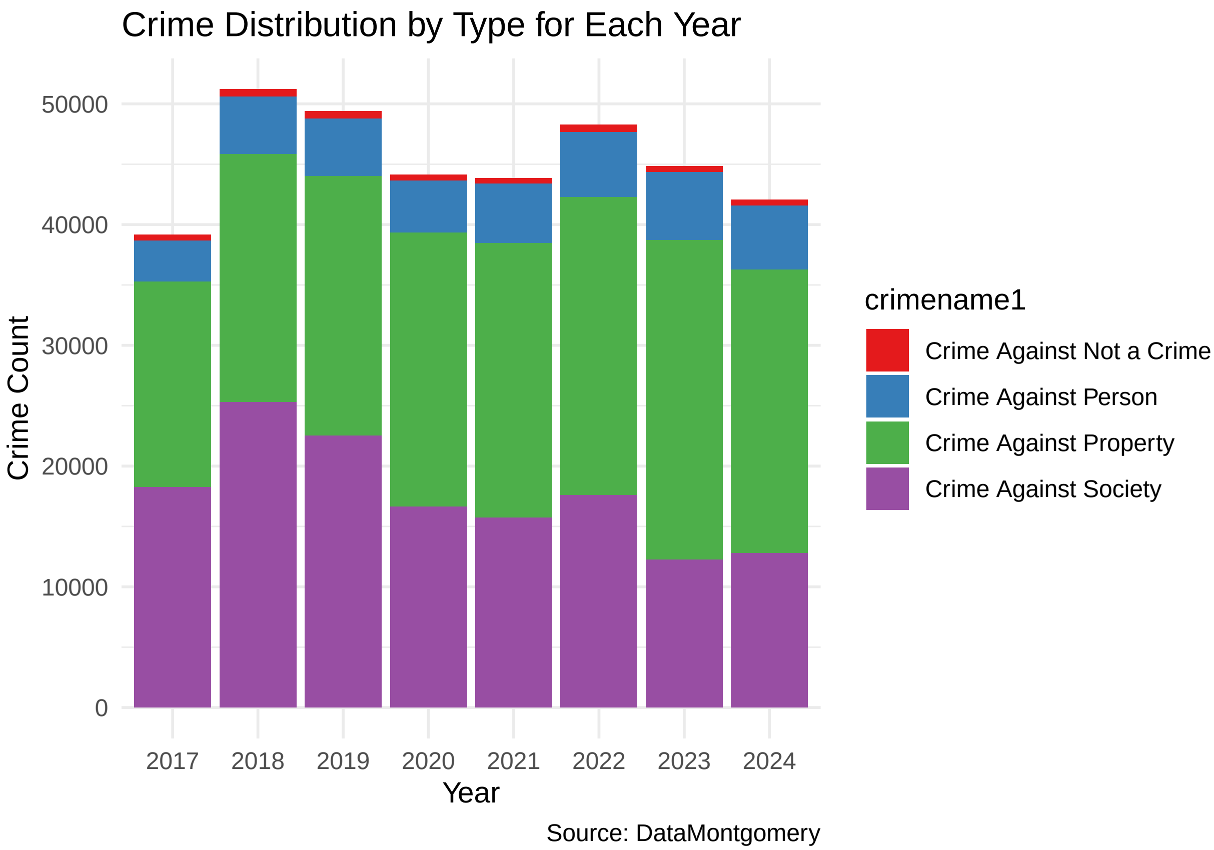



```
# Crime Report Distribution by Police District for Each Year
ggplot(data = Yearly_crime1, aes(x = factor(year), y = crime_percentage, fill =policedistrictname )) +
  geom_bar(stat = "identity", position = "stack") +
  labs(x = "Year",
       y = "Crime Percentage (%)",
       title = "Crime Distribution by Type for Each Year",
       caption = "Source: DataMontgomery") +
  theme_minimal()
```




```
  #scale_fill_brewer(palette = "Set1")
```

```
# Crime Distribution by Type for Each Year
plot3 <- ggplot(data = Year_count, aes(x = factor(year), y = n, fill = crimename1)) +
  geom_bar(stat = "identity", position = "stack") +
  labs(x = "Year",
       y = "Crime Count",
       title = "Crime Distribution by Type for Each Year",
       caption = "Source: DataMontgomery") +
  theme_minimal() +
  scale_fill_brewer(palette = "Set1")

plot3
```

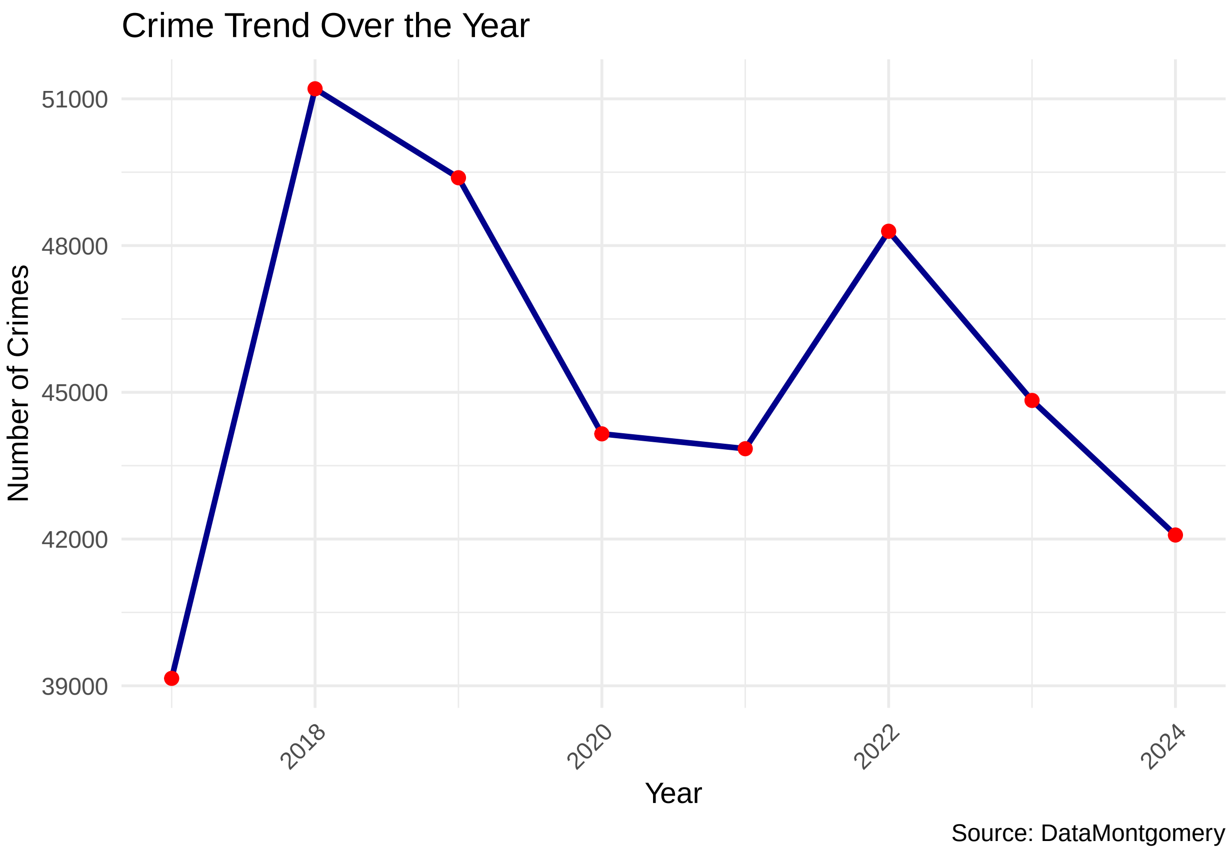



```r
# Crime count for each city
Crimecount_per_city <- Crime_Select |>
 group_by(city) |>
 count()
Crimecount_per_city
```

```
## # A tibble: 32 x 2
## # Groups:   city [32]
##    city             n
##    <chr>        <int>
##  1 aspen hill      13
##  2 barnesville     62
##  3 bethesda     25509
##  4 brookeville   1127
##  5 burtonsville  4667
##  6 cabin john     255
##  7 chevy chase   7469
##  8 clarksburg    5125
##  9 colesville       2
## 10 damascus      2864
## # i 22 more rows
```

```r
# Count crimes per year
Crime_trend <- Crime_Select |>
  group_by(year) |>
  count()

# Create the line chart for crime trend over the year
```

```r
ggplot(Crime_trend, aes(x = year, y = n, group = 1)) +
  geom_line(color = "darkblue", size = 1) +
  geom_point(color = "red", size = 2) +
  labs(x = "Year", y = "Number of Crimes", title = "Crime Trend Over the Year", caption = "Source: DataM
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



What times of day see the most crime?

Did the time when crimes happen change over the years? Are crimes happening later at night now compared to before?

```r
Crime_Select <- Crime_Select %>%
  mutate(`start_date_time` = parse_date_time(`start_date_time`, orders = "mdY IMS p"),
         hour = hour(`start_date_time`))
Crime_Select
```

```
## # A tibble: 362,953 x 20
##    `dispatchdate/time`    start_date_time     end_date_time   victims crimename1
##    <chr>                  <dttm>              <chr>             <dbl> <chr>
##  1 01/09/2022 01:18:38 AM 2022-01-09 01:34:00 <NA>                  1 Crime Aga~
##  2 01/28/2022 07:11:29 PM 2022-01-28 14:40:00 01/28/2022 02:~       1 Crime Aga~
##  3 02/08/2022 05:42:51 AM 2022-02-08 05:42:00 <NA>                  1 Crime Aga~
##  4 07/28/2020 02:55:00 PM 2020-07-28 14:54:00 07/28/2020 03:~       1 Crime Aga~
##  5 01/30/2019 07:49:25 PM 2019-01-30 19:49:00 <NA>                  1 Crime Aga~
```

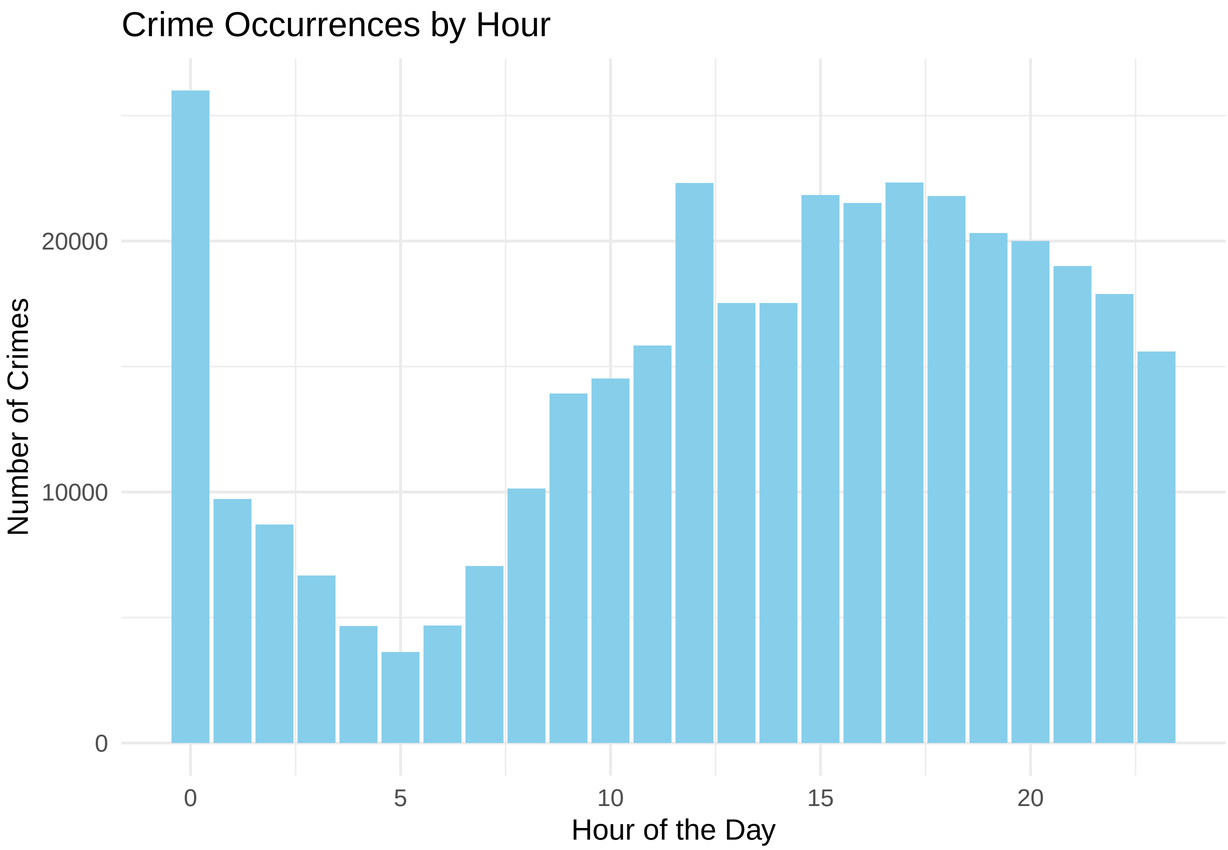
```
##  6 01/12/2022 02:53:25 AM 2022-01-12 02:53:00 01/12/2022 03:~       1 Crime Aga~
##  7 04/29/2017 04:47:44 PM 2017-04-22 15:00:00 04/25/2017 10:~       1 Crime Aga~
##  8 02/09/2022 11:28:51 AM 2022-02-09 11:28:00 02/09/2022 10:~       1 Crime Aga~
##  9 02/16/2022 07:33:27 AM 2022-02-15 22:30:00 02/16/2022 09:~       1 Crime Aga~
## 10 06/05/2021 08:41:55 PM 2021-06-05 20:58:00 <NA>                  1 Crime Aga~
## # i 362,943 more rows
## # i 15 more variables: crimename2 <chr>, crimename3 <chr>,
## #   policedistrictname <chr>, city <chr>, zipcode <dbl>, agency <chr>,
## #   place <chr>, latitude <dbl>, longitude <dbl>, policedistrictnumber <chr>,
## #   location <chr>, Date <date>, year <dbl>, month <dbl>, hour <int>
```

Group crime by hour and make a plot:

```r
Crime_hour <- Crime_Select |>
  group_by(hour) |>
  summarise(crime_count = n())
Crime_hour
```

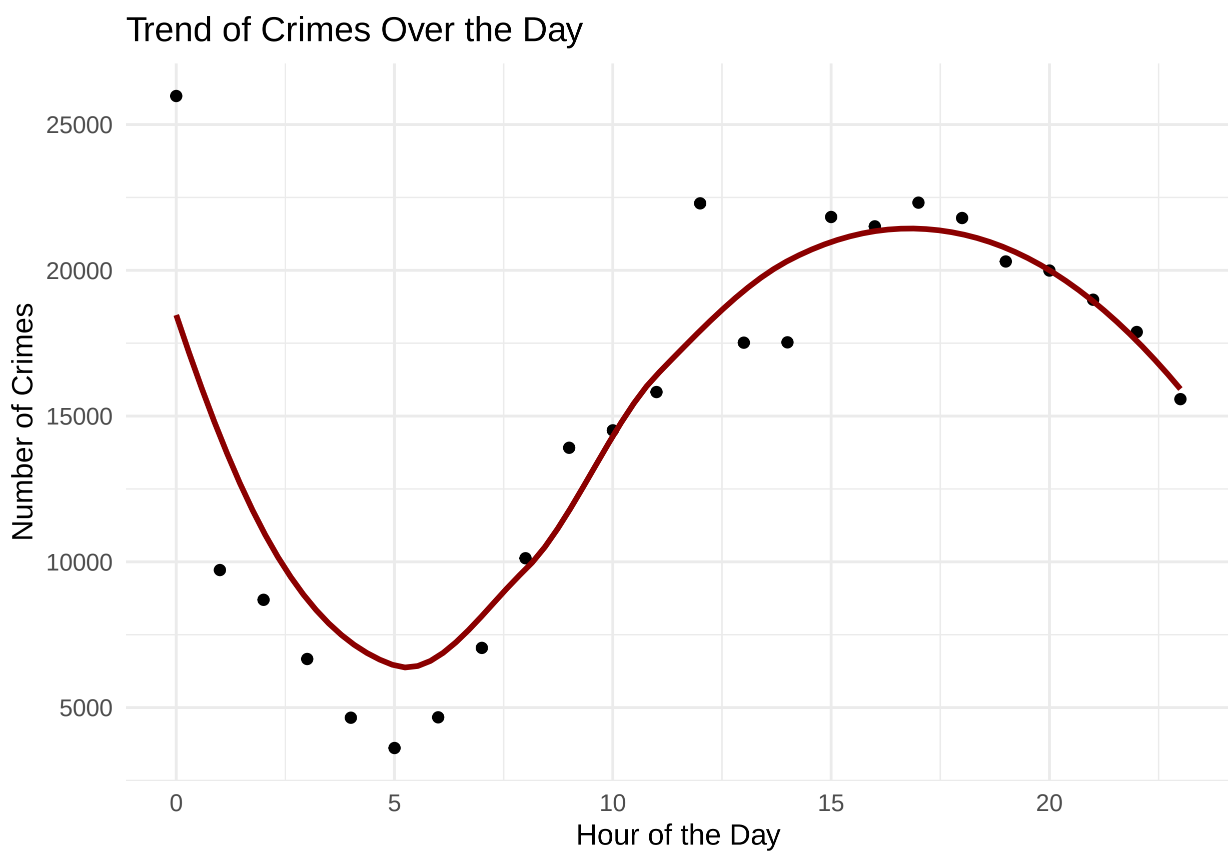
```
## # A tibble: 24 x 2
##     hour crime_count
##    <int>       <int>
##  1     0       25979
##  2     1        9718
##  3     2        8697
##  4     3        6667
##  5     4        4652
##  6     5        3613
##  7     6        4666
##  8     7        7047
##  9     8       10123
## 10     9       13912
## # i 14 more rows
```

```r
ggplot(Crime_hour, aes(x = hour, y = crime_count)) +
  geom_col(fill = "skyblue") +
  labs(x = "Hour of the Day", y = "Number of Crimes",
       title = "Crime Occurrences by Hour") +
  theme_minimal()
```




```r
ggplot(Crime_hour, aes(x = hour, y = crime_count)) +
  geom_point() +
  geom_smooth(method = "loess", se = FALSE, color = "darkred") +
  labs(x = "Hour of the Day", y = "Number of Crimes",
       title = "Trend of Crimes Over the Day") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

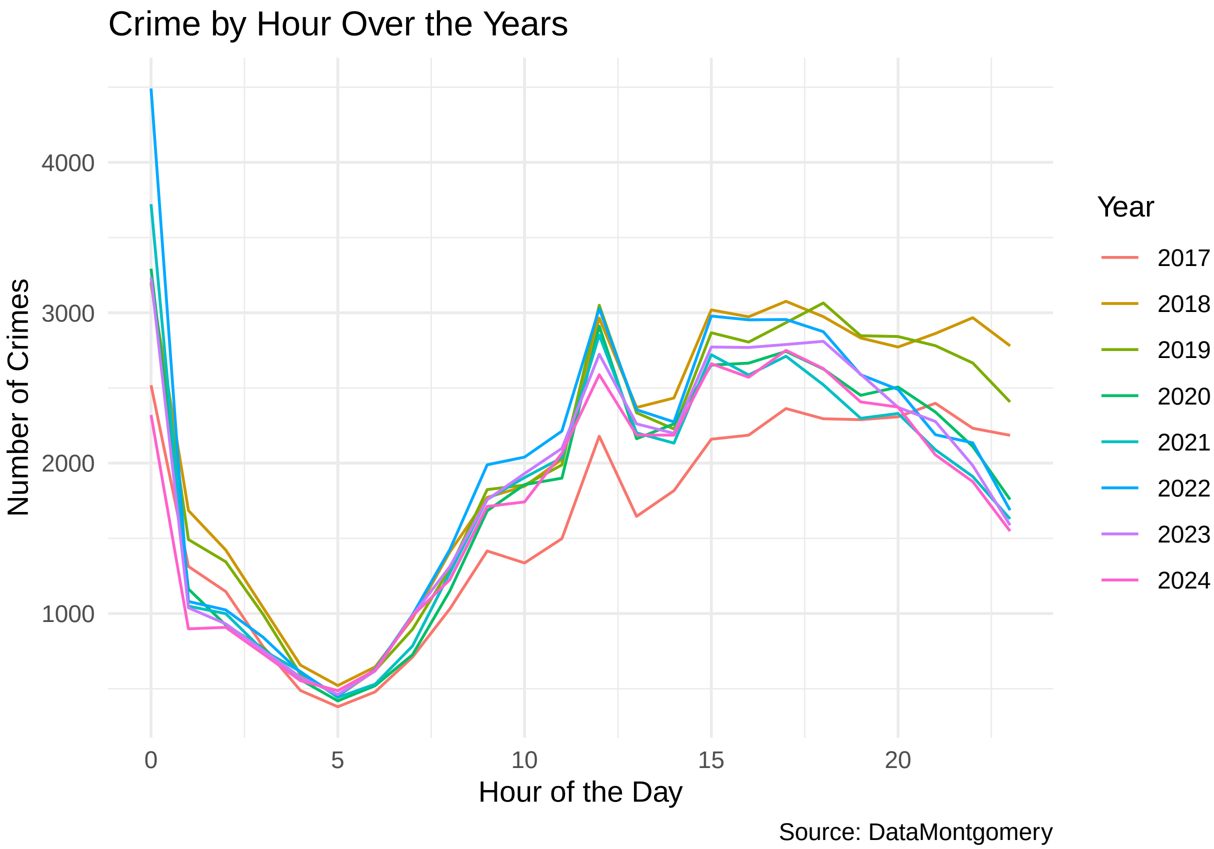



```r
Crime_hour_year <- Crime_Select |>
  group_by(year, hour) %>%
  summarise(total_crimes = n(), .groups = "drop")
Crime_hour_year
```

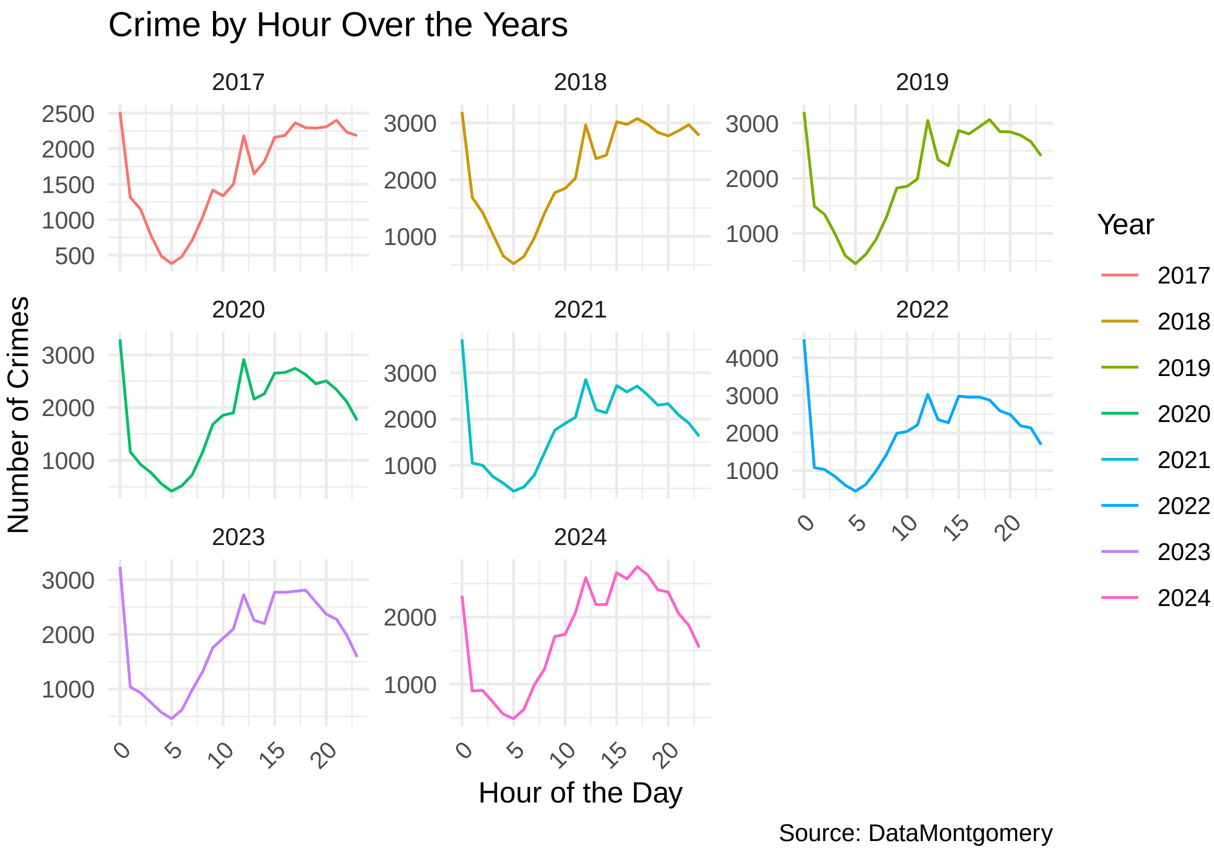
```
## # A tibble: 192 x 3
##     year  hour total_crimes
##    <dbl> <int>        <int>
##  1  2017     0         2518
##  2  2017     1         1313
##  3  2017     2         1146
##  4  2017     3          780
##  5  2017     4          488
##  6  2017     5          380
##  7  2017     6          479
##  8  2017     7          710
##  9  2017     8         1031
## 10  2017     9         1416
## # i 182 more rows
```

```r
ggplot(Crime_hour_year, aes(x = hour, y = total_crimes, color = factor(year))) +
  geom_line() +
  labs(
    title = "Crime by Hour Over the Years",
    x = "Hour of the Day",
    y = "Number of Crimes",
    color = "Year",
    caption = "Source: DataMontgomery"
```

```
  ) +
  theme_minimal()
```



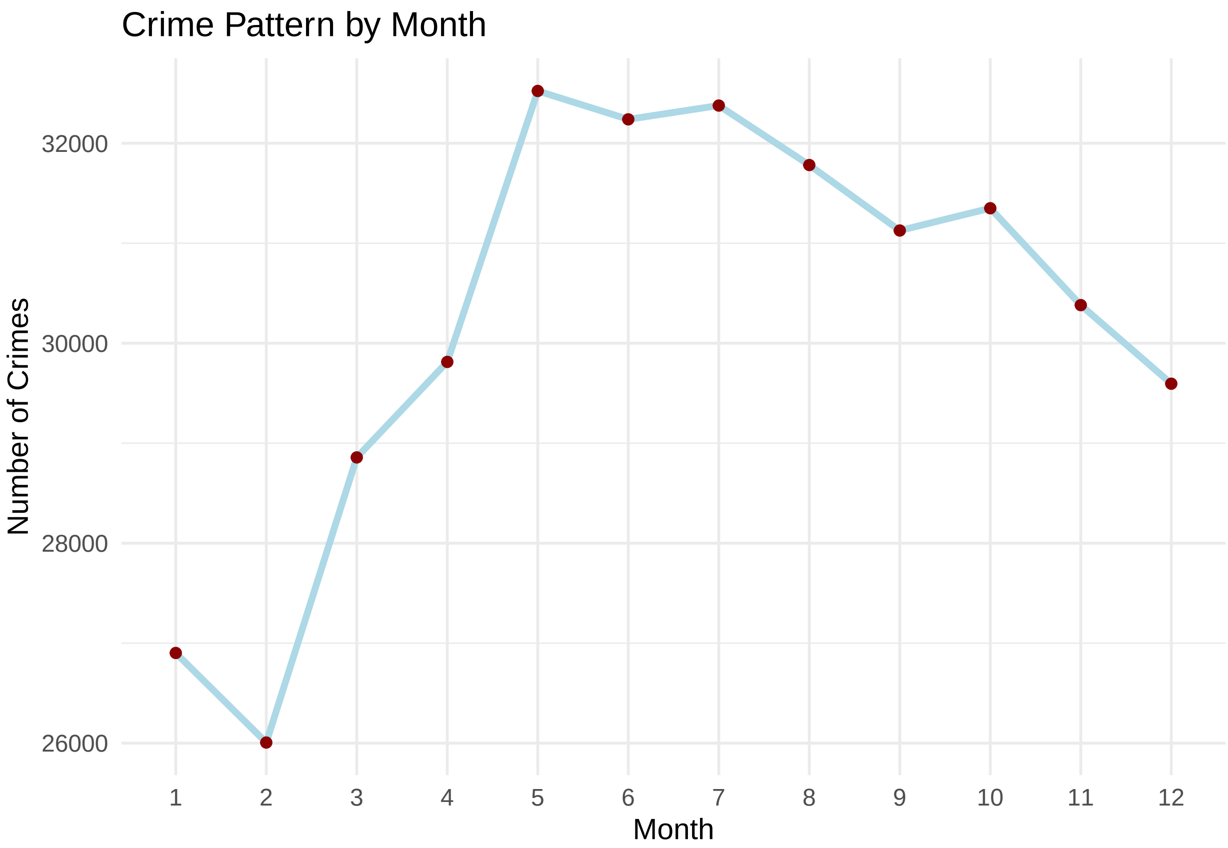

```
ggplot(Crime_hour_year, aes(x = hour, y = total_crimes)) +
  geom_line(aes(color = factor(year))) +  # Make a line for each year
  labs(
    title = "Crime by Hour Over the Years",
    x = "Hour of the Day",
    y = "Number of Crimes",
    color = "Year",
    caption = "Source: DataMontgomery"
  ) +
  facet_wrap(~ year, scales = "free_y") +  # Facet by year, reset each year
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))  # Rotate x-axis labels for clarity
```



In these plots, we observe that despite fluctuations in the overall number of crimes, the pattern of when crimes occur remains consistent across the years. We see a recurring peak around midnight, indicating more crimes happening during this time. Additionally, there is a noticeable decline in crimes around 5 AM each year, with a gradual increase as the day progresses, peaking again around noon before decreasing once more. Although the total number of crimes may fluctuate from year to year, the timing of when crimes occur remains largely unchanged, with the peak at midnight and the decrease around 5 AM persisting. Over the years, there is no significant shift towards crimes happening later at night. The overall pattern has stayed relatively stable, suggesting that the time of day when crimes happen has not changed significantly over time.

See the crime trend by month over the years:

```
Crime_Select |>
  count(month) |>
  ggplot(aes(x = factor(month), y = n, group = 1)) +
  geom_line(color = "lightblue", linewidth = 1.2) +
  geom_point(color = "darkred") +
  labs(
    title = "Crime Pattern by Month",
    x = "Month",
    y = "Number of Crimes"
  ) +
  theme_minimal()
```



We can see that there is an increase in the amount of crime in May. As the weather gets warmer, more people go outside. This creates more chances for crimes like theft, fights, or property damage to happen.

Using API for Montgomery county home value

```
library(httr)
```

```
##
## Attaching package: 'httr'

## The following object is masked from 'package:plotly':
##
##     config
```

```
library(jsonlite)
```

```
##
## Attaching package: 'jsonlite'

## The following object is masked from 'package:purrr':
##
##     flatten
```

```
library(dplyr)
library(stringr)
```

```
# Define the API endpoint and parameters
url <- "https://api.census.gov/data/2022/acs/acs5"
params <- list(
```

```r
  get = "NAME,B25077_001E",  # NAME = City Name, B25077_001E = Median Home Value
  `for` = "place:*",         # Get all places (cities/towns)
  `in` = "state:24"          # Maryland (state:24)
)
```

```r
# Send GET request
response <- GET(url, query = params)

# Check if request succeeded
if (status_code(response) == 200) {
  # Parse JSON response
  data <- fromJSON(content(response, "text"))

  # Convert to DataFrame (first row is column names)
  df <- as.data.frame(data[-1, ], stringsAsFactors = FALSE)
  names(df) <- data[1, ]

  # Convert median home value to numeric
  df$B25077_001E <- as.numeric(df$B25077_001E)

  # Define Montgomery County cities
  montgomery_cities <- c(
    "Chevy Chase",  "Aspen Hill",
    "Damascus", "Gaithersburg", "Clarksburg",
    "Olney", "Garrett Park", "Glen Echo", "Kensington", "Laytonsville", "Martins Additions", "Colesvill
    "North Chevy Chase", "Poolesville", "Rockville", "Takoma Park", "Seneca Valley", "Montgomery Villag
    "Derwood", "White Oak", "Washington Grove", "Burtonsville", "darnestown", "brookeville", "sandy spr
    "Glenmont", "Wheaton", "Silver Spring", "Bethesda", "Potomac", "Germantown"
  )

  # Filter rows for Montgomery County cities (case-insensitive match)
  df_filtered <- df %>%
    filter(str_detect(tolower(NAME), paste(tolower(montgomery_cities), collapse = "|"))) %>%
    rename(City = NAME, Median_Home_Value = B25077_001E) %>%
    select(City, Median_Home_Value)

  # Show result
  print(df_filtered)

} else {
  # Print error if request failed
  cat("Error:", status_code(response), content(response, "text"))
}
```

```
##                                          City Median_Home_Value
## 1            Ashton-Sandy Spring CDP, Maryland            783400
## 2                     Aspen Hill CDP, Maryland            495800
## 3                 Barnesville town, Maryland            690800
## 4                       Bethesda CDP, Maryland           1088000
## 5                 Brookeville town, Maryland            647100
## 6                 Burtonsville CDP, Maryland            475800
## 7                     Cabin John CDP, Maryland           1117300
## 8                 Chevy Chase town, Maryland           1616900
## 9                    Chevy Chase CDP, Maryland           1159700
```

```
## 10  Chevy Chase Section Five village, Maryland            1542900
## 11 Chevy Chase Section Three village, Maryland            1555400
## 12             Chevy Chase View town, Maryland            1181800
## 13          Chevy Chase Village town, Maryland            2000001
## 14                    Clarksburg CDP, Maryland             604000
## 15                    Colesville CDP, Maryland             560700
## 16                      Damascus CDP, Maryland             474800
## 17                    Darnestown CDP, Maryland             902500
## 18                       Derwood CDP, Maryland             558000
## 19                 Gaithersburg city, Maryland             472800
## 20                 Garrett Park town, Maryland             909300
## 21                    Germantown CDP, Maryland             393700
## 22                    Glen Echo town, Maryland             962800
## 23                      Glenmont CDP, Maryland             516700
## 24                   Kensington town, Maryland             882900
## 25                 Laytonsville town, Maryland            1019900
## 26            Montgomery Village CDP, Maryland             354800
## 27                North Bethesda CDP, Maryland             714500
## 28           North Chevy Chase village, Maryland          1063500
## 29             North Kensington CDP, Maryland             553000
## 30                North Potomac CDP, Maryland             770000
## 31                         Olney CDP, Maryland             615700
## 32                 Poolesville town, Maryland             606300
## 33                       Potomac CDP, Maryland            1044900
## 34               Potomac Heights CDP, Maryland              75000
## 35                  Potomac Park CDP, Maryland             120000
## 36                    Rockville city, Maryland             623800
## 37                 Silver Spring CDP, Maryland             606100
## 38             South Kensington CDP, Maryland             891200
## 39                  Takoma Park city, Maryland             685000
## 40            Washington Grove town, Maryland             569100
## 41                       Wheaton CDP, Maryland             454600
## 42                     White Oak CDP, Maryland             475000
```

EDA: I have some more cities that I don't have in my crime data, I'm will clean them so I can have the same number of cities but I'm also going to put the cities name in lower case so they can match the crime one and like that I will not have a problem to combine them.

```r
#putting the headers in lower case
names(df_filtered) <- tolower(names(df_filtered))
names(df_filtered) <- gsub(" ","",names(df_filtered))
head(df_filtered)
```

```
##                                  city median_home_value
## 1 Ashton-Sandy Spring CDP, Maryland            783400
## 2          Aspen Hill CDP, Maryland            495800
## 3        Barnesville town, Maryland            690800
## 4            Bethesda CDP, Maryland           1088000
## 5        Brookeville town, Maryland            647100
## 6        Burtonsville CDP, Maryland            475800
```

```r
df_filtered <- df_filtered %>%
  mutate(city = tolower(city),                           # make lowercase
         city = str_remove(city, "\\s*,\\s*maryland$"),   # remove ", Maryland" with optional spaces
         city = str_remove(city, "\\s+cdp$"),             # remove trailing "CDP"
         city = str_remove(city, "\\city$"),
```

```r
         city = str_trim(city))
df_filtered
```

```
##                                 city median_home_value
## 1                ashton-sandy spring            783400
## 2                         aspen hill            495800
## 3                   barnesville town            690800
## 4                           bethesda           1088000
## 5                   brookeville town            647100
## 6                       burtonsville            475800
## 7                         cabin john           1117300
## 8                   chevy chase town           1616900
## 9                        chevy chase           1159700
## 10  chevy chase section five village           1542900
## 11 chevy chase section three village           1555400
## 12             chevy chase view town           1181800
## 13          chevy chase village town           2000001
## 14                        clarksburg            604000
## 15                        colesville            560700
## 16                          damascus            474800
## 17                        darnestown            902500
## 18                           derwood            558000
## 19                 gaithersburg city            472800
## 20                 garrett park town            909300
## 21                        germantown            393700
## 22                    glen echo town            962800
## 23                          glenmont            516700
## 24                   kensington town            882900
## 25                 laytonsville town           1019900
## 26                montgomery village            354800
## 27                    north bethesda            714500
## 28          north chevy chase village           1063500
## 29                  north kensington            553000
## 30                     north potomac            770000
## 31                             olney            615700
## 32                  poolesville town            606300
## 33                           potomac           1044900
## 34                   potomac heights             75000
## 35                      potomac park            120000
## 36                     rockville city            623800
## 37                     silver spring            606100
## 38                  south kensington            891200
## 39                  takoma park city            685000
## 40             washington grove town            569100
## 41                           wheaton            454600
## 42                         white oak            475000
```

```r
remove_cities <- c(
  "chevy chase section five village",
  "chevy chase section three village",
  "chevy chase view town",
  "chevy chase town",
  "chevy chase village town",
  "north chevy chase village",
```

```r
  "north potomac",
  "potomac heights",
  "south kensington",
  "north kensington",
  "potomac park"
)

df_filtered <- df_filtered |>
  filter(!city %in% remove_cities)
df_filtered
```

```
##                      city median_home_value
## 1    ashton-sandy spring            783400
## 2             aspen hill            495800
## 3       barnesville town            690800
## 4               bethesda           1088000
## 5       brookeville town            647100
## 6           burtonsville            475800
## 7             cabin john           1117300
## 8            chevy chase           1159700
## 9             clarksburg            604000
## 10            colesville            560700
## 11              damascus            474800
## 12            darnestown            902500
## 13               derwood            558000
## 14     gaithersburg city            472800
## 15     garrett park town            909300
## 16            germantown            393700
## 17        glen echo town            962800
## 18              glenmont            516700
## 19       kensington town            882900
## 20     laytonsville town           1019900
## 21    montgomery village            354800
## 22        north bethesda            714500
## 23                 olney            615700
## 24      poolesville town            606300
## 25               potomac           1044900
## 26        rockville city            623800
## 27          silver spring            606100
## 28      takoma park city            685000
## 29 washington grove town            569100
## 30               wheaton            454600
## 31             white oak            475000
```

```r
df_filtered <- df_filtered |>
  mutate(
    city = str_replace_all(city, " town| city", ""),  # remove ' town' or ' city'
    city = str_replace(city, "ashton-sandy spring", "sandy spring"),  # rename ashton_sandy spring to m
    city = str_trim(city)  # remove extra spaces
  )
df_filtered
```

```
##                 city median_home_value
## 1       sandy spring            783400
## 2         aspen hill            495800
```

```
## 3          barnesville              690800
## 4             bethesda             1088000
## 5          brookeville              647100
## 6         burtonsville              475800
## 7           cabin john             1117300
## 8          chevy chase             1159700
## 9           clarksburg              604000
## 10          colesville              560700
## 11            damascus              474800
## 12          darnestown              902500
## 13             derwood              558000
## 14        gaithersburg              472800
## 15        garrett park              909300
## 16          germantown              393700
## 17           glen echo              962800
## 18            glenmont              516700
## 19          kensington              882900
## 20        laytonsville             1019900
## 21 montgomery village              354800
## 22      north bethesda              714500
## 23               olney              615700
## 24         poolesville              606300
## 25             potomac             1044900
## 26           rockville              623800
## 27        silver spring             606100
## 28         takoma park              685000
## 29    washington grove              569100
## 30             wheaton              454600
## 31           white oak              475000
```

Looking for correlation

Is crime related to property values? Do cities with higher home prices have lower crime rates? Understanding this relationship can help reveal whether economic factors play a role in local crime patterns.

```r
MC_crime_Combine <- left_join(Crimecount_per_city,df_filtered, by = "city")

MC_crime_Combine
```

```
## # A tibble: 32 x 3
## # Groups:   city [32]
##    city             n median_home_value
##    <chr>        <int>             <dbl>
##  1 aspen hill      13            495800
##  2 barnesville     62            690800
##  3 bethesda     25509           1088000
##  4 brookeville   1127            647100
##  5 burtonsville  4667            475800
##  6 cabin john     255           1117300
##  7 chevy chase   7469           1159700
##  8 clarksburg    5125            604000
##  9 colesville       2            560700
## 10 damascus      2864            474800
## # i 22 more rows
```

```r
correlation <- cor(MC_crime_Combine$n, MC_crime_Combine$median_home_value, use = "complete.obs")
print(correlation)
```

```
## [1] -0.1463264
```

```r
# Find the statistical information for my model
Eq <- lm(n ~ median_home_value, data= MC_crime_Combine)
summary(Eq)
```

```
##
## Call:
## lm(formula = n ~ median_home_value, data = MC_crime_Combine)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -15467 -11859  -8024   -846 115217
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        2.300e+04  1.492e+04   1.542    0.134
## median_home_value -1.631e-02  2.047e-02  -0.797    0.432
##
## Residual standard error: 25900 on 29 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.02141,    Adjusted R-squared:  -0.01233
## F-statistic: 0.6345 on 1 and 29 DF,  p-value: 0.4322
```

The results showed that there is no meaningful relationship between the two. The p-value was 0.43, which means the connection we see in the data is likely just due to chance. The R-squared value was about 2%, meaning home values explain only a tiny part of the differences in crime numbers between cities. In short, in this dataset, higher or lower home prices do not seem to be linked to how many crimes happen.

```r
# Perform Pearson correlation test
cor_test <- cor.test(MC_crime_Combine$n, MC_crime_Combine$median_home_value, use = "complete.obs")

# Print the result
print(cor_test)
```

```
##
##  Pearson's product-moment correlation
##
## data:  MC_crime_Combine$n and MC_crime_Combine$median_home_value
## t = -0.79657, df = 29, p-value = 0.4322
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.4759869  0.2193889
## sample estimates:
##        cor
## -0.1463264
```

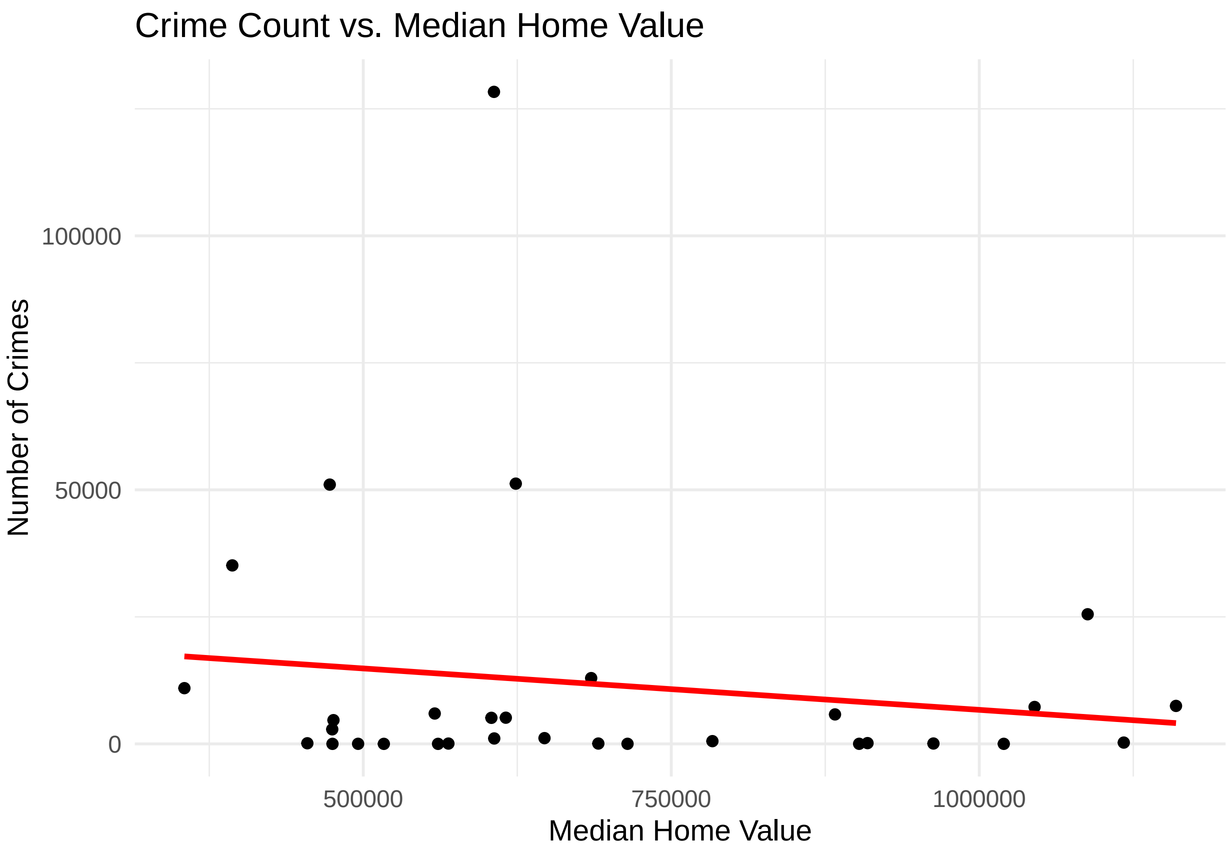
The plot below shows how the number of crimes in each city compares to its median home value.

```r
# Remove scientific notation
options(scipen = 999)
# Scatter plot with a trend line
ggplot(MC_crime_Combine, aes(x = median_home_value, y = n)) +
  geom_point() + # scatter plot points
```

```r
  labs(
    title = "Crime Count vs. Median Home Value",
    x = "Median Home Value",
    y = "Number of Crimes"
  ) +
  theme_minimal() +
  #xlim(0, 1000000) +
  #ylim(0,20000) +
  geom_smooth(method = "lm", formula= y~x, se = FALSE, color = "red") # linear trend line
```

```
## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```




```r
  #method = 'lm', formula= y~x, se = FALSE
```

Mapping Crime Across Cities

```r
# Calculate quantile bounds on Crime_Select
lon_q <- quantile(Crime_Select$longitude, c(0.01, 0.99))
lat_q <- quantile(Crime_Select$latitude, c(0.01, 0.99))

# Filter outliers directly into a temporary object
Crime_Select_NoOutliers <- Crime_Select |>
  filter(
```

```
    !is.na(longitude), !is.na(latitude),
    longitude != 0, latitude != 0,
    longitude >= lon_q[1], longitude <= lon_q[2],
    latitude >= lat_q[1], latitude <= lat_q[2]
  )

# Plot the filtered data
leaflet(data = Crime_Select_NoOutliers) |>
  addTiles() |>
  addCircleMarkers(
    ~longitude, ~latitude,
    radius = 3,
    color = "red",
    stroke = FALSE,
    fillOpacity = 0.5,
    clusterOptions = markerClusterOptions(),
    popup = ~paste("City:", city, "<br>", "Crime:", crimename1)
  )
```

Green circle - 10 or less crime. Yellow circle - 100-10 crimes. Orange circle - more than 100 crimes and more.

```
Crimecount_per_city
```

```
## # A tibble: 32 x 2
## # Groups:   city [32]
##    city              n
##    <chr>         <int>
##  1 aspen hill       13
##  2 barnesville      62
##  3 bethesda      25509
##  4 brookeville    1127
##  5 burtonsville   4667
##  6 cabin john      255
##  7 chevy chase    7469
##  8 clarksburg     5125
##  9 colesville        2
## 10 damascus       2864
## # i 22 more rows
```

Exploring Crime Around Schools

Understanding distribution of crime near schools.Identify whether schools are surrounded by higher levels of crime and highlight potential areas of concern.

```
#upload my working directory so I can install my file.
School_Info <- read_csv("Public_Schools_20250324.csv")
```

```
## Rows: 200 Columns: 10
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (7): CATEGORY, SCHOOL NAME, ADDRESS, CITY, PHONE, URL, LOCATION
## dbl (3): ZIP CODE, LONGITUDE, LATITUDE
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
School_Info
```

```
## # A tibble: 200 x 10
##    CATEGORY          `SCHOOL NAME` ADDRESS CITY  `ZIP CODE` PHONE URL   LONGITUDE
##    <chr>             <chr>         <chr>   <chr>      <dbl> <chr> <chr>     <dbl>
##  1 HIGH SCHOOLS      Damascus HS   25921 ~ Dama~      20872 301-~ http~     -77.2
##  2 ELEMENTARY SCHO~ Clearspring ~ 9930 M~ Dama~      20872 301-~ http~     -77.2
##  3 ELEMENTARY SCHO~ Sherwood ES   1401 O~ Sand~      20860 301-~ http~     -77.0
##  4 ELEMENTARY SCHO~ Pine Crest ES 201 Wo~ Silv~      20901 301-~ http~     -77.0
##  5 MIDDLE SCHOOLS    Earle B. Woo~ 14615 ~ Rock~      20852 301-~ http~     -77.1
##  6 ELEMENTARY SCHO~ Seven Locks ~ 7000 R~ Beth~      20817 301-~ http~     -77.1
##  7 ELEMENTARY SCHO~ Roscoe R Nix~ 1100 C~ Silv~      20903 301-~ http~     -77.0
##  8 ELEMENTARY SCHO~ Georgian For~ 3100 R~ Silv~      20906 301-~ http~     -77.1
##  9 MIDDLE SCHOOLS    Argyle MS     2400 B~ Silv~      20906 301-~ http~     -77.0
## 10 ELEMENTARY SCHO~ Burnt Mills ~ 11211 ~ Silv~      20901 301-~ http~     -77.0
## # i 190 more rows
## # i 2 more variables: LATITUDE <dbl>, LOCATION <chr>
```

```r
#putting the headers in lower case
names(School_Info) <- tolower(names(School_Info))
names(School_Info) <- gsub(" ","",names(School_Info))
head(School_Info)
```

```
## # A tibble: 6 x 10
##   category       schoolname address city  zipcode phone url   longitude latitude
##   <chr>          <chr>      <chr>   <chr>   <dbl> <chr> <chr>     <dbl>    <dbl>
## 1 HIGH SCHOOLS   Damascus ~ 25921 ~ Dama~   20872 301-~ http~     -77.2     39.3
## 2 ELEMENTARY SC~ Clearspri~ 9930 M~ Dama~   20872 301-~ http~     -77.2     39.3
## 3 ELEMENTARY SC~ Sherwood ~ 1401 O~ Sand~   20860 301-~ http~     -77.0     39.1
## 4 ELEMENTARY SC~ Pine Cres~ 201 Wo~ Silv~   20901 301-~ http~     -77.0     39.0
## 5 MIDDLE SCHOOLS Earle B. ~ 14615 ~ Rock~   20852 301-~ http~     -77.1     39.1
## 6 ELEMENTARY SC~ Seven Loc~ 7000 R~ Beth~   20817 301-~ http~     -77.1     39.0
## # i 1 more variable: location <chr>
```

```r
library(sf)
```

```
## Linking to GEOS 3.13.0, GDAL 3.8.5, PROJ 9.5.1; sf_use_s2() is TRUE
```

```r
# Remove rows with missing or invalid coordinates
School_Info <- School_Info[is.finite(School_Info$longitude) & is.finite(School_Info$latitude), ]

# Convert to sf object
schools_sf <- st_as_sf(School_Info, coords = c("longitude", "latitude"), crs = 4326)
crime_sf <- st_as_sf(Crime_Select, coords = c("longitude", "latitude"), crs = 4326)
```

```r
# Summarize crime count and average location by city
city_sf <- Crime_Select %>%
  group_by(city) %>%
  summarise(
    crime_count = n(),
    longitude = mean(longitude, na.rm = TRUE),
    latitude = mean(latitude, na.rm = TRUE)
  ) %>%
  ungroup() %>%
  st_as_sf(coords = c("longitude", "latitude"), crs = 4326)
```

Explore whether schools are situated in areas with high or low crime density and to identify any visible patterns of clustering.

```r
library(dplyr)
library(sf)
library(leaflet)
library(leaflet.extras)
library(htmltools)

# Step 1: Filter invalid coordinates FIRST
Crime_Select_clean <- Crime_Select %>%
  filter(is.finite(longitude), is.finite(latitude),
         longitude != 0, latitude != 0)

# Step 2: Convert to sf
crime_sf <- st_as_sf(Crime_Select_clean, coords = c("longitude", "latitude"), crs = 4326)

# Step 3: Use sf object directly in leaflet
leaflet() %>%
  addTiles() %>%
  addHeatmap(data = crime_sf,
             intensity = ~1,   # Optional: uniform intensity
             radius = 15,
             blur = 10,
             max = 0.05,
             group = "Crimes") %>%
  addCircleMarkers(data = schools_sf,
                   radius = 5,
                   color = "blue",
                   stroke = TRUE,
                   weight = 1,
                   fillOpacity = 1,
                   popup = ~schoolname,
                   group = "Schools") %>%
  addLayersControl(
    overlayGroups = c("Crimes", "Schools"),
    options = layersControlOptions(collapsed = FALSE)
  )
```

```r
write.csv(Crime_Select, "crime_select.csv", row.names = FALSE)
```

```r
write.csv(School_Info, "School_Info.csv", row.names = FALSE)
```

```r
write.csv(df_filtered, "Median_home_value_filtered.csv", row.names = FALSE)
```

The R portion of this project focused on uncovering initial trends and relationships within the data through visual and exploratory analysis. Using RStudio, I created a range of charts including bar plots, alluvial diagrams, and correlation plots that revealed which cities had the highest crime counts, how crime types shifted over time, and how crime frequency aligned with property values.

Key findings included the dominance of property crime across the county, particularly in cities like Silver Spring and Rockville. The bar plots made it clear that crime was not evenly distributed, while alluvial diagrams helped visualize how different types of crime were connected to city locations. I also used R to examine the relationship between the number of crimes and the median home value by city. While this analysis showed no strong correlation overall, it raised important questions that shaped the more targeted statistical tests conducted later in Python.