NYDP Shooting Incident Data

T. A. Meinhold

2025-12-06

Data Report

Introduction

Understanding temporal patterns in criminal activity is essential for interpreting underlying social dynamics and for improving crime-prevention strategies. Crime data often reveal recurring structures across hours, days, and seasons, reflecting human behavior, environmental conditions, and situational opportunities. In this analysis, we examine how crime incidents vary over the course of the week, across different times of day, and throughout the year. By identifying clear peaks, troughs, and periodic structures, we can better understand when crime is most likely to occur and explore plausible explanations for these fluctuations.

Loading the Data

The source data file of the NYDP Shooting Incident Data: https://catalog.data.gov/dataset/nypd-shooting-incident-data-historic. We now import libraries and load the csv-file:

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.6
## v forcats   1.0.1     v stringr   1.6.0
## v ggplot2   4.0.1     v tibble    3.3.0
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.2.0
## -- 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
```

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

url_in <- "https://data.cityofnewyork.us/api/views/833y-fsy8/"

file_names <- c("rows.csv")

urls <- str_c(url_in, file_names)

NY_Cases <- read_csv(urls[1])
```

```
## Rows: 29744 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
```

```
## chr  (12): OCCUR_DATE, BORO, LOC_OF_OCCUR_DESC, LOC_CLASSFCTN_DESC, LOCATION...
## dbl   (5): INCIDENT_KEY, PRECINCT, JURISDICTION_CODE, Latitude, Longitude
## num   (2): X_COORD_CD, Y_COORD_CD
## lgl   (1): STATISTICAL_MURDER_FLAG
## time  (1): OCCUR_TIME
##
## 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.
```

Cleaning the Data

I delete columns not needed here and convert the date in the correct format:

```
NY_Cases <- NY_Cases %>%
  select(
    -JURISDICTION_CODE,
    -X_COORD_CD,
    -Y_COORD_CD,
    -Latitude,
    -Longitude,
    -Lon_Lat,
    -LOC_CLASSFCTN_DESC,
    -LOCATION_DESC,
    -LOC_OF_OCCUR_DESC
  )

NY_Cases <- NY_Cases %>%
  mutate(OCCUR_DATE = mdy(OCCUR_DATE))
```

Now, I look into the data if there are any columns have many `nan` and I factorize the data in order to make everything later more suitable. I also split the date into hour, day, month and year:

```
NY_Cases %>%
  summarise(across(everything(), ~ sum(is.na(.))))
```

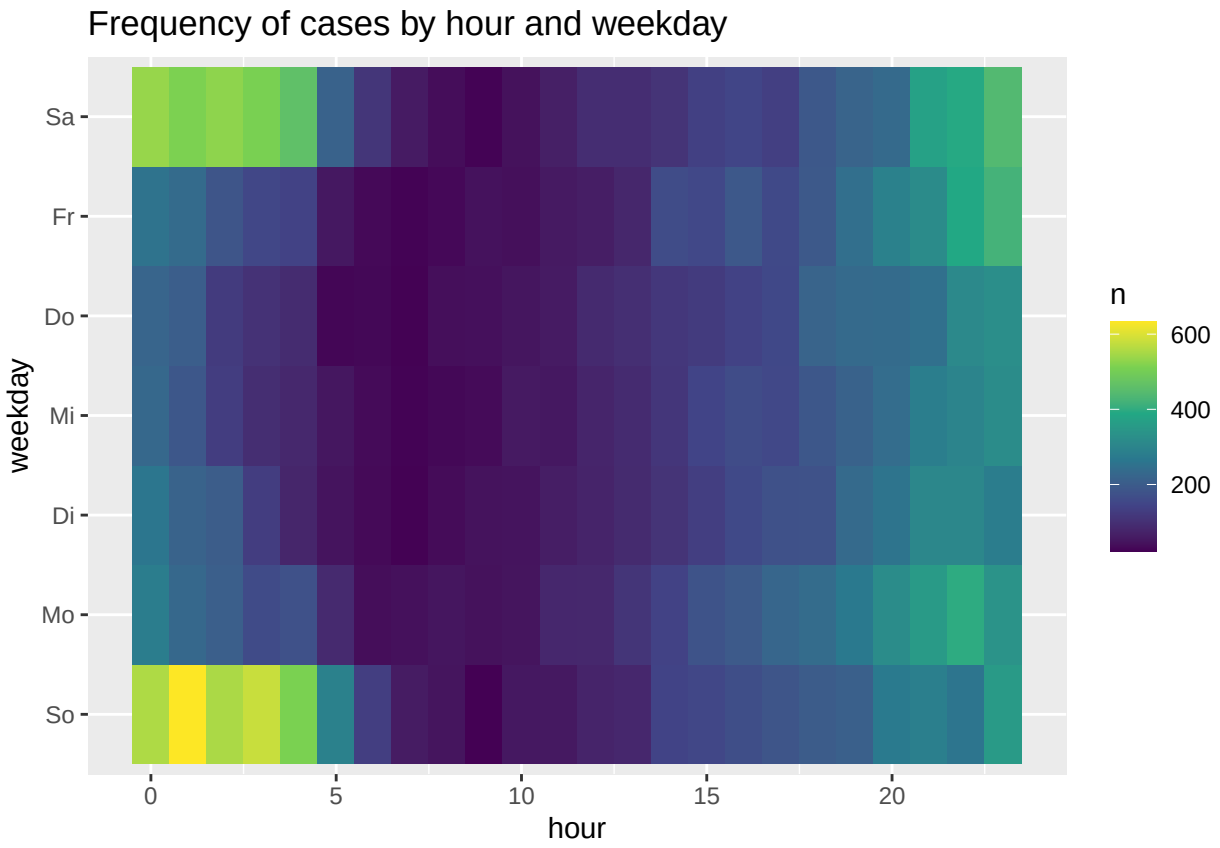
```
## # A tibble: 1 x 12
##   INCIDENT_KEY OCCUR_DATE OCCUR_TIME  BORO PRECINCT STATISTICAL_MURDER_FLAG
##          <int>      <int>      <int> <int>    <int>                   <int>
## 1            0          0          0     0        0                       0
## # i 6 more variables: PERP_AGE_GROUP <int>, PERP_SEX <int>, PERP_RACE <int>,
## #   VIC_AGE_GROUP <int>, VIC_SEX <int>, VIC_RACE <int>
```

```
NY_Cases <- NY_Cases %>%
  mutate(
    BORO           = factor(BORO),
    PERP_SEX       = factor(PERP_SEX),
    PERP_RACE      = factor(PERP_RACE),
    VIC_SEX        = factor(VIC_SEX),
    VIC_RACE       = factor(VIC_RACE),
    PERP_AGE_GROUP = factor(PERP_AGE_GROUP),
    VIC_AGE_GROUP  = factor(VIC_AGE_GROUP)
  )

NY_Cases <- NY_Cases %>%
  mutate(
    year     = year(OCCUR_DATE),
```

```
    month     = month(OCCUR_DATE, label = TRUE, abbr = TRUE),
    weekday   = wday(OCCUR_DATE, label = TRUE, abbr = TRUE),
    hour      = lubridate::hour(OCCUR_TIME)
  )
```

Now let's make a heatmap:

```
NY_Cases %>%
  mutate(
    hour = hour(OCCUR_TIME),
    weekday = wday(OCCUR_DATE, label = TRUE)
  ) %>%
  count(weekday, hour) %>%
  ggplot(aes(hour, weekday, fill = n)) +
  geom_tile() +
  scale_fill_viridis_c() +
  labs(title = "Frequency of cases by hour and weekday")
```



General observations: We can see that on weekends there are significantly more crime cases, especially during the night. This can be explained by the fact that people have more free time on weekends and/or expect fewer witnesses at night.

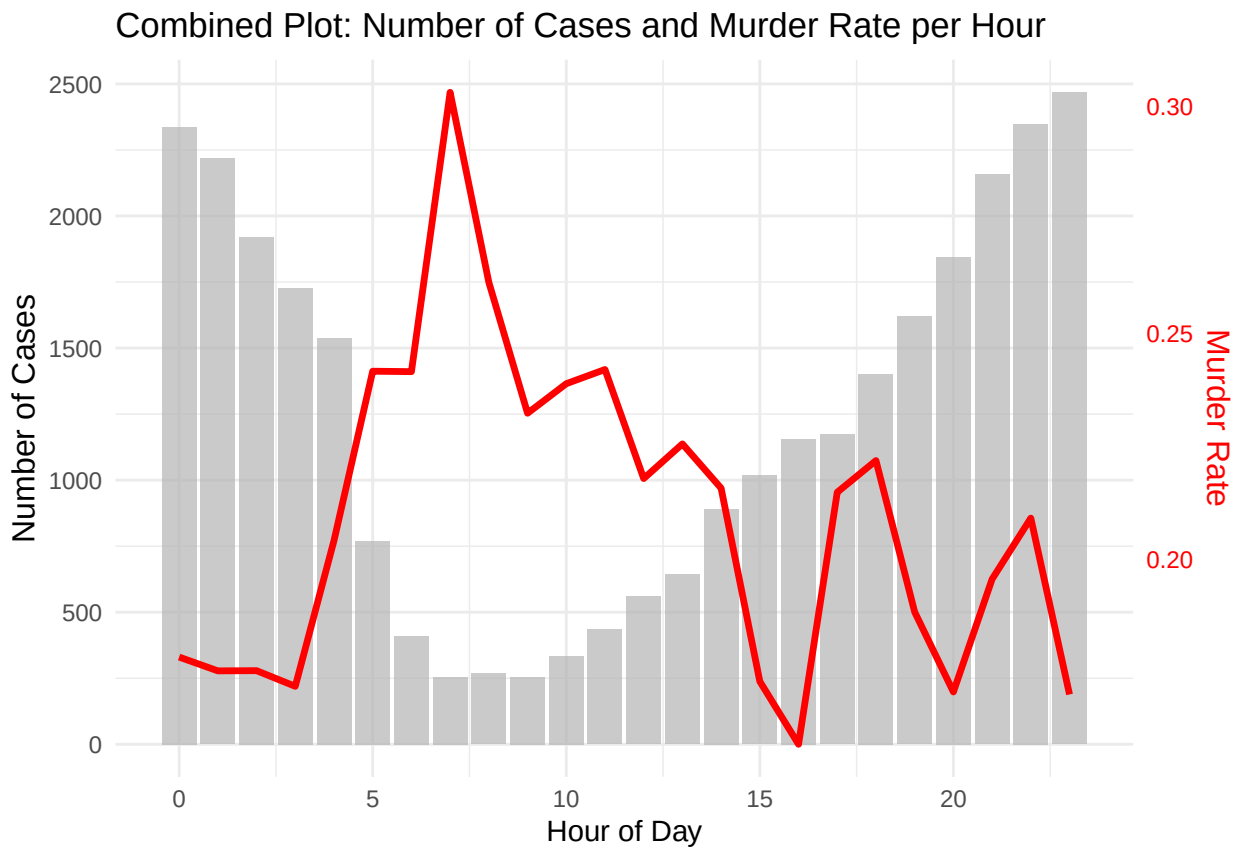
Detailed observations: Throughout the entire week, the number of crime incidents in the evening and at night is fairly similar. This is probably the time when more "*crime professionals*" are active. However, during the weekend, more crimes occur in the early morning hours. This could be explained by the fact that far more people are out and have more time. At night, observations are generally more difficult, and fewer people are awake, which makes unobserved interactions and criminal activity more likely.

Correlation Analysis

We now look into how overall crime frequency relates to crime severity across the day, using a combined visualization of hourly case counts and murder rates.

```
NY_cases_corr <- NY_Cases %>%
  mutate(hour = hour(OCCUR_TIME)) %>%
  group_by(hour) %>%
  summarise(n_cases = n(), murder_rate = mean(STATISTICAL_MURDER_FLAG))

max_cases <- max(NY_cases_corr$n_cases)
min_rate  <- min(NY_cases_corr$murder_rate)
max_rate  <- max(NY_cases_corr$murder_rate)

scale_factor <- max_cases / (max_rate - min_rate)

ggplot(NY_cases_corr, aes(hour, murder_rate)) +
  geom_col(aes(y = n_cases), fill = "grey70", alpha = 0.7) +
  geom_line(aes(y = (murder_rate - min_rate) * scale_factor),
            color = "red", size = 1.2) +
  scale_y_continuous(
    name = "Number of Cases",
    sec.axis = sec_axis(transform = ~ . / scale_factor + min_rate,
                        name = "Murder Rate")
  ) +
  labs(
    title = "Combined Plot: Number of Cases and Murder Rate per Hour",
    x = "Hour of Day"
  ) +
  theme_minimal() +
  theme(
    axis.title.y = element_text(color = "black", size = 12),
    axis.title.y.right = element_text(color = "red", size = 12),
    axis.text.y.right  = element_text(color = "red")
  )
```

```
## 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.
```



The plot shows two patterns that together reveal how crime severity and crime frequency differ across the day:

1. Total number of cases (grey bars): Crime frequency is strongly concentrated during late-night and early-morning hours. The highest number of incidents occurs between 22:00 and 02:00, particularly around midnight, which aligns with typical nightlife activity. The number of cases then drops sharply during the early morning (around 5–7 a.m.) and gradually increases again throughout the afternoon and evening.
2. Murder rate (red line): The murder rate does not follow the same pattern as the general crime frequency. Instead, the rate peaks sharply during the early morning hours (around 6–7 a.m.), even though total crime is relatively low at that time. This suggests that incidents occurring in these low-activity hours tend to be more severe or more targeted. In contrast, the murder rate is somewhat lower during the late-night peak (0–2 a.m.), when many incidents occur but a larger share are non-lethal.
3. Combined interpretation:

- High-volume hours (evening and night):
  - many incidents, but
  - lower proportion are murders.
- Low-volume hours (early morning):
  - fewer incidents, but
  - a relatively high proportion are fatal.
- Evening to late night (17:00–02:00):
  - moderate murder rate with
  - steadily rising incident numbers.

This pattern could indicate that crime during high-activity periods is driven more by opportunity and social interaction, while early-morning crimes may be more intentional or involve different offender–victim dynamics.

Model

We will now make a model in order to understand the dynamics better:

```
model_f <- glm(
  STATISTICAL_MURDER_FLAG ~ factor(hour) + weekday + BORO + PRECINCT,
  family = binomial,
  data = NY_Cases
)
summary(model_f)
```

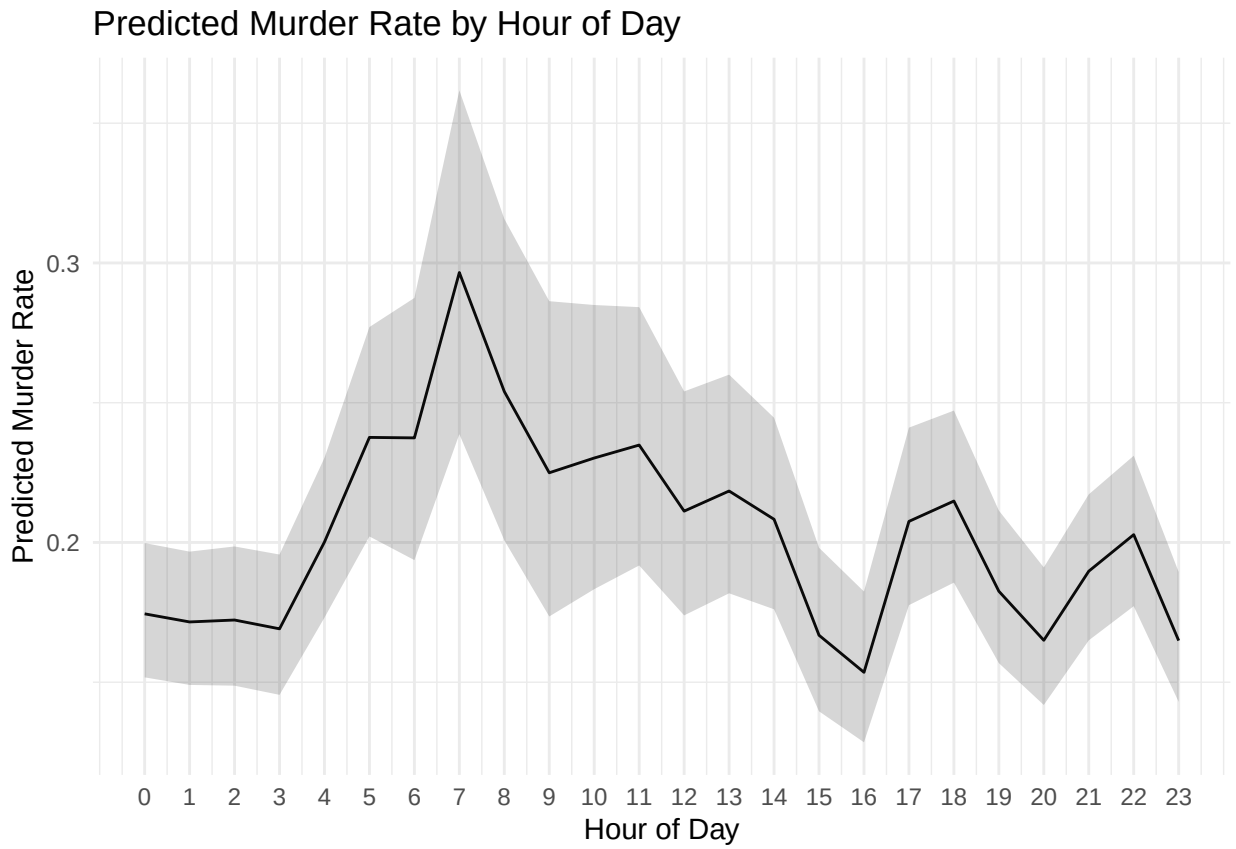
```
##
## Call:
## glm(formula = STATISTICAL_MURDER_FLAG ~ factor(hour) + weekday +
##     BORO + PRECINCT, family = binomial, data = NY_Cases)
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -1.4218314  0.1270108 -11.195  < 2e-16 ***
## factor(hour)1     -0.0202161  0.0777608  -0.260 0.794881
## factor(hour)2     -0.0154476  0.0808475  -0.191 0.848470
## factor(hour)3     -0.0376736  0.0838238  -0.449 0.653116
## factor(hour)4      0.1690526  0.0834692   2.025 0.042834 *
## factor(hour)5      0.3883762  0.1003006   3.872 0.000108 ***
## factor(hour)6      0.3874252  0.1275405   3.038 0.002384 **
## factor(hour)7      0.6907164  0.1469231   4.701 2.59e-06 ***
## factor(hour)8      0.4769719  0.1493513   3.194 0.001405 **
## factor(hour)9      0.3172177  0.1584597   2.002 0.045297 *
## factor(hour)10     0.3472252  0.1393586   2.492 0.012717 *
## factor(hour)11     0.3731905  0.1246498   2.994 0.002754 **
## factor(hour)12     0.2366610  0.1159503   2.041 0.041245 *
## factor(hour)13     0.2794012  0.1089912   2.564 0.010362 *
## factor(hour)14     0.2190705  0.0980317   2.235 0.025438 *
## factor(hour)15    -0.0541022  0.0991746  -0.546 0.585393
## factor(hour)16    -0.1527518  0.0971104  -1.573 0.115726
## factor(hour)17     0.2145262  0.0895994   2.394 0.016653 *
## factor(hour)18     0.2581718  0.0842373   3.065 0.002178 **
## factor(hour)19     0.0551839  0.0836946   0.659 0.509672
## factor(hour)20    -0.0671484  0.0824002  -0.815 0.415126
## factor(hour)21     0.1024525  0.0768037   1.334 0.182219
## factor(hour)22     0.1853741  0.0744166   2.491 0.012737 *
## factor(hour)23    -0.0679739  0.0762769  -0.891 0.372850
## weekday.L         -0.0074570  0.0363526  -0.205 0.837470
## weekday.Q         -0.0682940  0.0394268  -1.732 0.083243 .
## weekday.C         -0.0618465  0.0392986  -1.574 0.115544
## weekday^4         -0.0003927  0.0398603  -0.010 0.992140
## weekday^5          0.0226245  0.0420614   0.538 0.590651
## weekday^6         -0.1102291  0.0431332  -2.556 0.010602 *
## BOROBROOKLYN       0.0400758  0.0815328   0.492 0.623052
## BOROMANHATTAN     -0.1249478  0.0699557  -1.786 0.074083 .
## BOROQUEENS         0.1189058  0.1622022   0.733 0.463515
## BOROSTATEN ISLAND  0.1950085  0.2091181   0.933 0.351064
## PRECINCT          -0.0017693  0.0024985  -0.708 0.478853
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 29251  on 29743  degrees of freedom
## Residual deviance: 29110  on 29709  degrees of freedom
## AIC: 29180
##
## Number of Fisher Scoring iterations: 4
```

```r
newdata <- data.frame(
  hour = factor(0:23, levels = levels(NY_Cases$hour)),
  weekday = factor("Mo", levels = levels(NY_Cases$weekday)),
  BORO = factor("BRONX", levels = levels(NY_Cases$BORO)),
  PRECINCT = 40
)

predict_hourly <- predict(model_f, newdata, type = "response")

ref_weekday <- levels(NY_Cases$weekday)[1]
ref_boro    <- levels(NY_Cases$BORO)[1]
ref_precinct <- median(NY_Cases$PRECINCT)

newdata <- data.frame(
  hour    = 0:23,
  weekday = factor(ref_weekday, levels = levels(NY_Cases$weekday)),
  BORO    = factor(ref_boro,    levels = levels(NY_Cases$BORO)),
  PRECINCT = ref_precinct
)

pred <- predict(model_f, newdata, type = "link", se.fit = TRUE)

newdata$fit   <- pred$fit
newdata$se    <- pred$se.fit
newdata$prob  <- plogis(newdata$fit)
newdata$lower <- plogis(newdata$fit - 1.96 * newdata$se)
newdata$upper <- plogis(newdata$fit + 1.96 * newdata$se)

summary(newdata$prob)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.1535  0.1721  0.2052  0.2037  0.2263  0.2966
```

```r
ggplot(newdata, aes(x = hour, y = prob)) +
  geom_line() +
  geom_ribbon(aes(ymin = lower, ymax = upper), alpha = 0.2) +
  scale_x_continuous(breaks = 0:23) +
  labs(
    title = "Predicted Murder Rate by Hour of Day",
    x = "Hour of Day",
    y = "Predicted Murder Rate"
  ) +
  theme_minimal()
```



To analyze how the likelihood of a murder varies across the day, we fitted a logistic regression model using the binary variable *STATISTICAL_MURDER_FLAG* as the outcome.

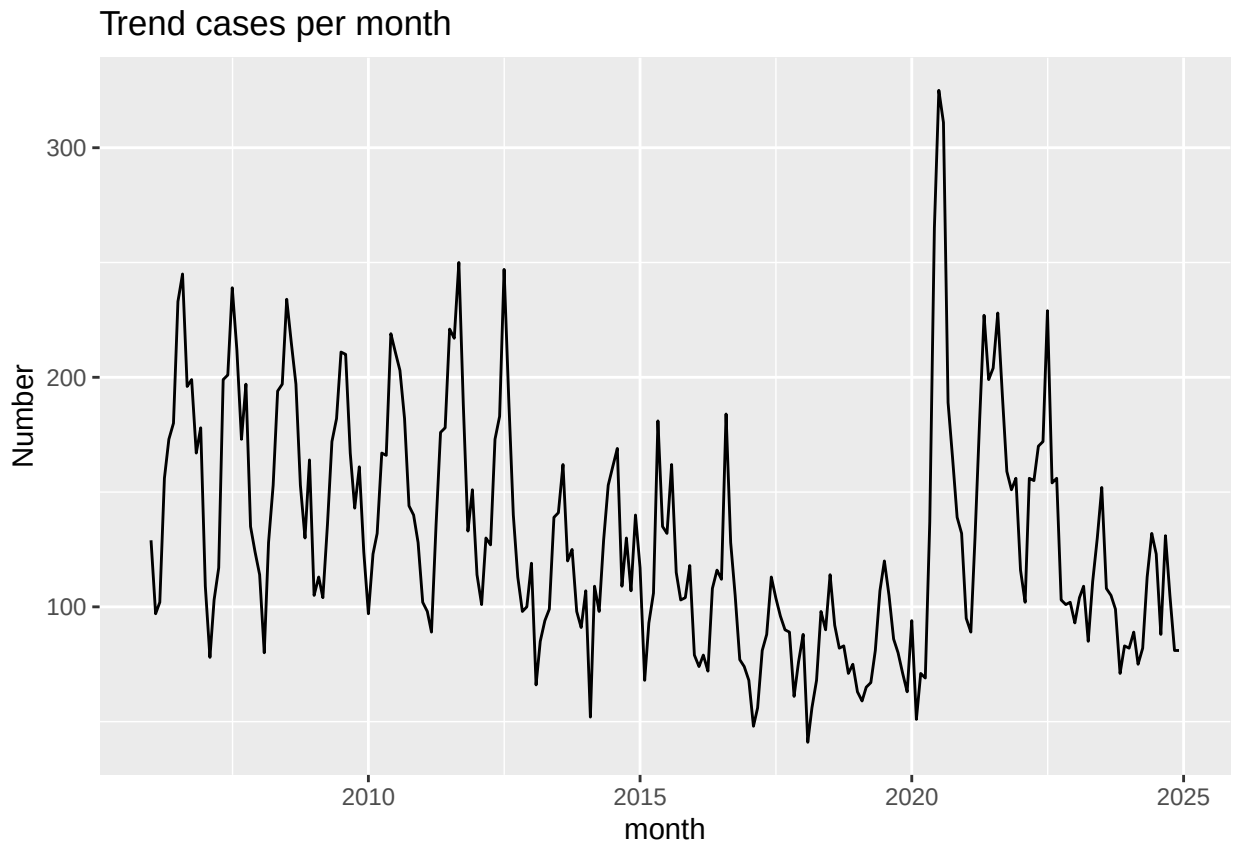
To properly capture these non-linear patterns, we replaced the numeric "hour" variable with a categorical factor, allowing the model to estimate a separate murder probability for each hour of the day without forcing a linear trend. This dramatically improved model fit (lower AIC) and revealed several hours with significantly higher or lower murder rates compared to midnight (the reference category).

We then created a new data frame representing each hour of the day and generated predicted murder probabilities using the fitted factor-based model. These predictions were visualized in a line plot with 95% confidence intervals, showing how the estimated murder rate changes throughout the day according to the model.

This approach provides a statistically grounded way to confirm and quantify the temporal patterns already visible in the raw data and exploratory plots.

Outlook: Seasonal Analysis

We now look into a time series and plot the crime cases per month:



Here we can see a periodic structure: one peak generally corresponds to one year. All peaks reach their highest values in summer, while the lowest points appear at the end or the beginning of a year. But why do we observe more crime in summer than in winter?

One possible explanation could be that nights in summer are less dark than in winter, allowing law enforcement to see better. However, this does not make much sense, as modern technology should be sufficient to detect and catch criminals regardless of season.

Conclusion

Our analysis reveals that crime is shaped by both short-term (hourly) and long-term (seasonal) temporal patterns. On the daily level, total crime incidents peak during late-night hours, yet the fitted logistic model shows that the highest predicted murder probabilities occur in the early morning hours (around 5–7 a.m.), when overall crime frequency is relatively low. This suggests that incidents in these hours tend to be more severe or more targeted, while late-night peaks are driven more by high volumes of less lethal offenses.

On a broader time scale, the data exhibit a clear seasonal cycle, with one major peak per year: crime counts are highest during the summer months and reach their minimum at the turn of the year. This seasonal pattern is consistent with increased outdoor and social activity, higher mobility, and more interaction in warmer periods, which create additional opportunities for crime, whereas winter conditions and year-end periods may suppress such activity.

By combining non-linear modeling of the hourly effect (via a factor-based logistic regression) with aggregated seasonal analysis, we obtain a more nuanced picture: crime severity and crime frequency do not move in lockstep. High-frequency periods are not always the most lethal, and long-term seasonal peaks are driven primarily by exposure and opportunity rather than by changes in law-enforcement capacity alone.

As before, these findings must be interpreted in light of potential **biases**: underreporting, changes in policing

strategies, temporal variation in patrol patterns, and unobserved social or environmental factors may all influence the observed trends. These findings may be influenced also by analytical biases. Modeling choices—such as treating "hour" as a factor, aggregating data by month, or ignoring temporal dependence—may shape the results and potentially over- or understate certain patterns. Nevertheless, the combined hourly and seasonal analysis provides a useful framework for identifying when crime, especially lethal crime, is most likely to occur.

Here all Interpretations in our Table:

Table 1: Summary table

Time Scale	Observed Pattern	Evidence	Interpretation
Hourly (all crimes)	Peak at 22:00–02:00	Descriptive plots	Nightlife-driven exposure.
Hourly (murder probability)	Peak at 05:00–07:00	Logistic model (factor hour)	Early-morning incidents more severe.
Afternoon hours	Moderate murder risk	Model coefficients (10–14)	Serious incidents despite lower volume.
Evening hours	High volume, moderate severity	Model (17–18)	Many incidents, but less lethal.
Weekday effects	Small differences	Weak significance	Hour > weekday.
Borough effects	No major differences	BORO not significant	Geography less relevant.
Seasonal (all crimes)	Summer peaks, winter lows	Monthly aggregation	More activity/exposure in summer.

```
sessionInfo()
```

```
## R version 4.5.2 (2025-10-31 ucrt)
## Platform: x86_64-w64-mingw32/x64
## Running under: Windows 11 x64 (build 26100)
##
## Matrix products: default
##   LAPACK version 3.12.1
##
## locale:
## [1] LC_COLLATE=German_Germany.utf8  LC_CTYPE=German_Germany.utf8
## [3] LC_MONETARY=German_Germany.utf8 LC_NUMERIC=C
## [5] LC_TIME=German_Germany.utf8
##
## time zone: Europe/Berlin
## tzcode source: internal
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] lubridate_1.9.4 forcats_1.0.1   stringr_1.6.0   dplyr_1.1.4
##  [5] purrr_1.2.0     readr_2.1.6     tidyr_1.3.1     tibble_3.3.0
##  [9] ggplot2_4.0.1   tidyverse_2.0.0
##
## loaded via a namespace (and not attached):
##  [1] bit_4.6.0          gtable_0.3.6       crayon_1.5.3       compiler_4.5.2
##  [5] tidyselect_1.2.1   parallel_4.5.2     scales_1.4.0       yaml_2.3.11
##  [9] fastmap_1.2.0      R6_2.6.1           labeling_0.4.3     generics_0.1.4
## [13] curl_7.0.0         knitr_1.50         pillar_1.11.1      RColorBrewer_1.1-3
```

```
## [17] tzdb_0.5.0         rlang_1.1.6        stringi_1.8.7      xfun_0.54
## [21] S7_0.2.1           bit64_4.6.0-1      viridisLite_0.4.2  timechange_0.3.0
## [25] cli_3.6.5          withr_3.0.2        magrittr_2.0.4     digest_0.6.39
## [29] grid_4.5.2         vroom_1.6.7        rstudioapi_0.17.1  hms_1.1.4
## [33] lifecycle_1.0.4    vctrs_0.6.5        evaluate_1.0.5     glue_1.8.0
## [37] farver_2.1.2       rmarkdown_2.30     tools_4.5.2        pkgconfig_2.0.3
## [41] htmltools_0.5.8.1
```