Plot_Visuals

Sai

2025-07-06

```
library(tidyverse)
library(ranger)
```

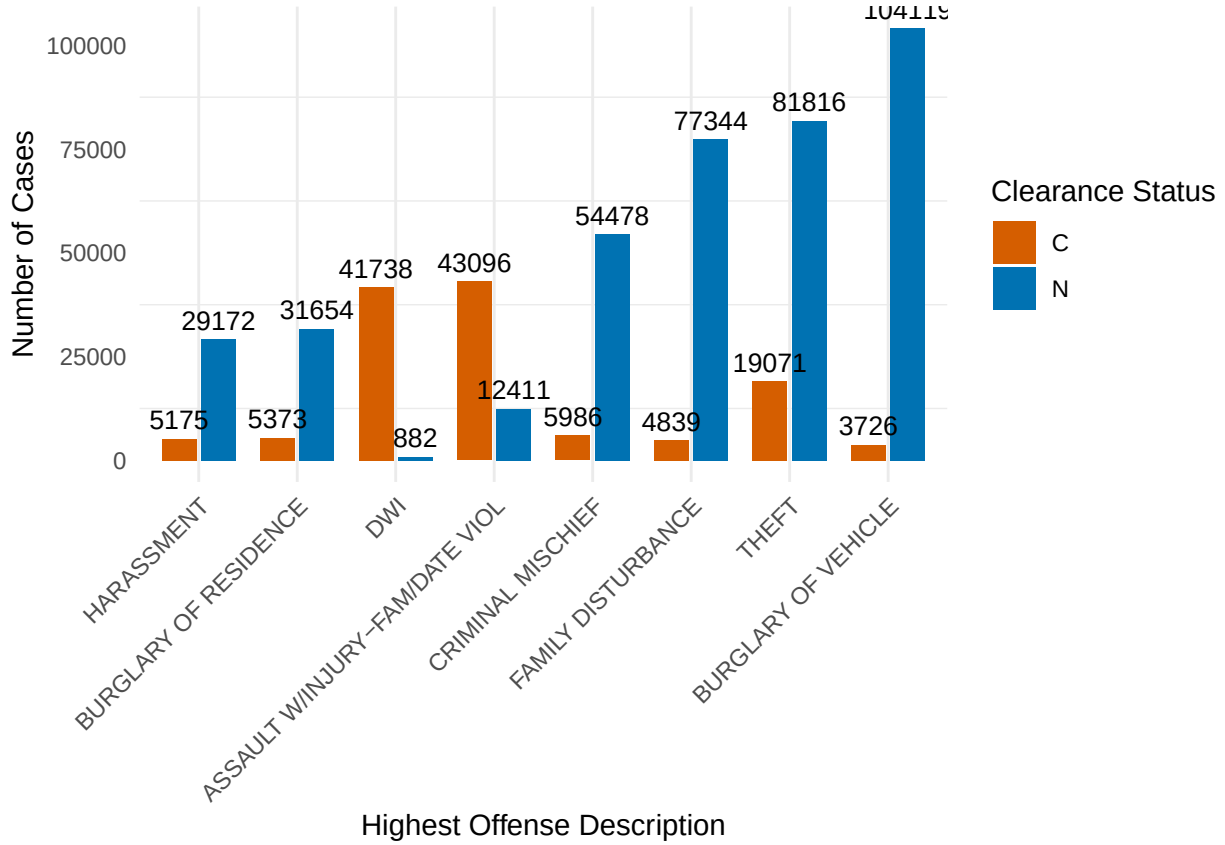
```
crime <- read_csv("./data/crime_sample.csv")
```

```
crime <- crime %>%
  mutate(
    highest_offense_description = as.factor(highest_offense_description),
    highest_offense_code        = as.factor(highest_offense_code),
    family_violence             = as.factor(family_violence),
    occurred_date_time_month    = as.ordered(occurred_date_time_month),
    occurred_date_time_day_of_week = as.ordered(occurred_date_time_day_of_week),
    location_type               = as.factor(location_type),
    apd_sector                  = as.factor(apd_sector),
    apd_district                = as.factor(apd_district)
  )

crime
```

```
## # A tibble: 1,174,559 x 21
##    highest_offense_description    highest_offense_code family_violence
##    <fct>                          <fct>                <fct>
##  1 ASSAULT W/INJURY-FAM/DATE VIOL 900                  Y
##  2 POSS OF ALCOHOL - AGE 17 TO 20 2209                 N
##  3 POSS OF FIREARM BY FELON       1502                 N
##  4 CRIMINAL MISCHIEF              1400                 N
##  5 HARASSMENT                     2703                 N
##  6 CRIMINAL TRESPASS              2716                 N
##  7 THEFT                          600                  N
##  8 THEFT                          600                  N
##  9 MISAPPLY FIDUCIARY PROP        1201                 N
## 10 CRIMINAL TRESPASS              2716                 N
## # i 1,174,549 more rows
## # i 18 more variables: occurred_date_time <dttm>,
## #   occurred_date_time_year <dbl>, occurred_date_time_month <ord>,
## #   occurred_date_time_week_of_year <dbl>, occurred_date_time_day <dbl>,
## #   occurred_date_time_day_of_week <ord>, occurred_date_time_hour <dbl>,
## #   occurred_date_time_minute <dbl>, occurred_date <dttm>, occurred_time <dbl>,
## #   report_date_time <dttm>, report_date <dttm>, report_time <dbl>, ...
```

Count of Crime Clearances according to top-8 crime types .

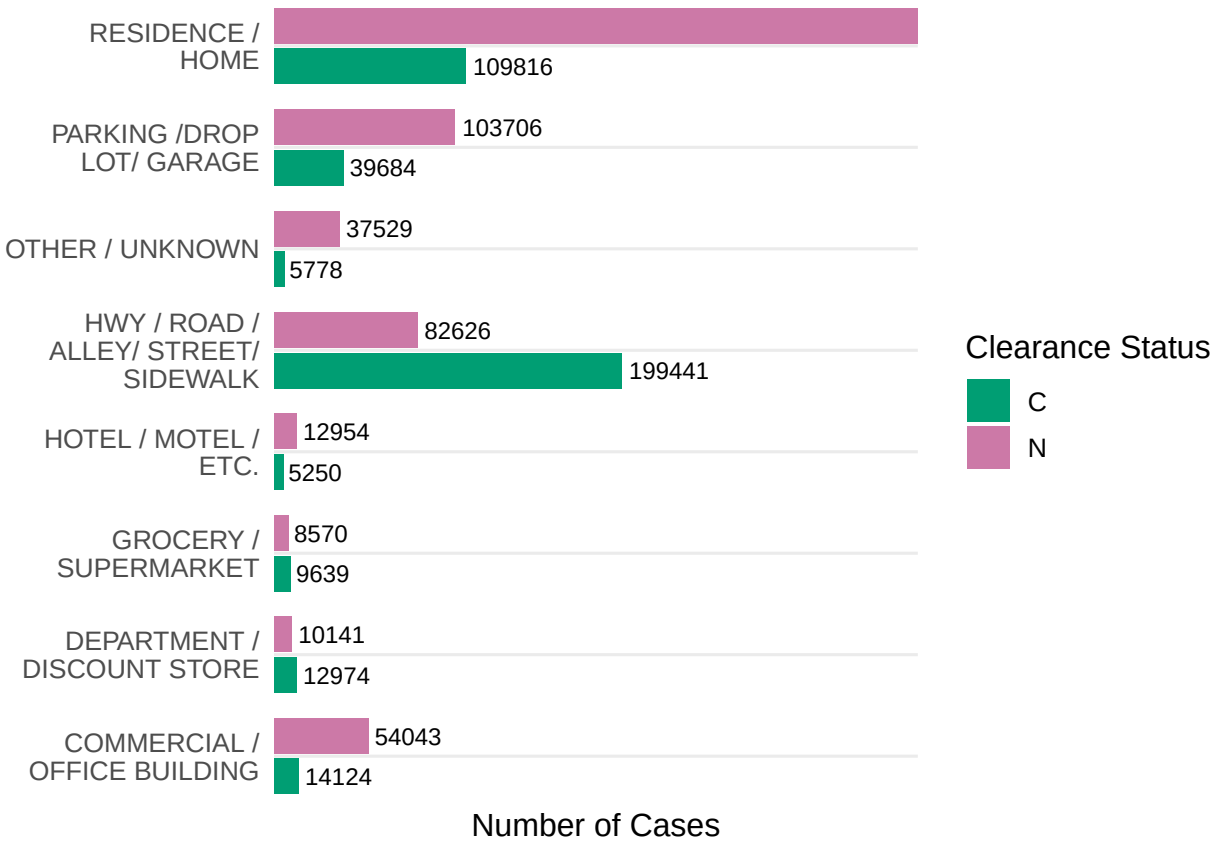
```r
library(dplyr)
library(forcats)
library(ggplot2)

# 1. Identify the top-8 offenses by total count
top8 <- crime %>%
  count(highest_offense_description, sort = TRUE) %>%
  slice_head(n = 8) %>%
  pull(highest_offense_description)

# 2. Build a summary table of counts by offense × clearance_status
plot_data <- crime %>%
  filter(highest_offense_description %in% top8) %>%
  count(highest_offense_description, clearance_status) %>%
  # lock in the ordering of the factor so it follows the overall ranking
  mutate(highest_offense_description = factor(highest_offense_description, levels = rev(top8)))

# 3. Create the grouped bar chart
ggplot(plot_data, aes(x = highest_offense_description, y = n, fill = clearance_status)) +
  geom_col(position = position_dodge(width = 0.8), width = 0.7) +
  geom_text(aes(label = n),
            position = position_dodge(width = 0.8),
            vjust = -0.5,
            size = 3.5) +
  scale_fill_manual(
    name   = "Clearance Status",
    values = c("C" = "#D55E00", "N" = "#0072B2")
  ) +
  labs(
    x = "Highest Offense Description",
    y = "Number of Cases"
  ) +
  theme_minimal() +
  theme(
    axis.text.x        = element_text(angle = 45, hjust = 1),
    panel.grid.major.y = element_blank()
  )
```



Count of crime clearances according to top-8 locations.

```
library(dplyr)
library(forcats)
library(stringr)
library(ggplot2)

# 1. Identify the top-8 location types
top8_loc <- crime %>%
  count(location_type, sort = TRUE) %>%
  slice_head(n = 8) %>%
  pull(location_type)

# 2. Summarise counts & wrap the labels to a fixed width
plot_loc <- crime %>%
  filter(location_type %in% top8_loc) %>%
  count(location_type, clearance_status) %>%
  # preserve the original ranking, but wrap for display
  mutate(
    location_type = factor(location_type, levels = top8_loc),
    location_wrapped = str_wrap(location_type, width = 15)
  )

# 3. Plot horizontal grouped bars with inside labels
ggplot(plot_loc, aes(
    x    = location_wrapped,
    y    = n,
```

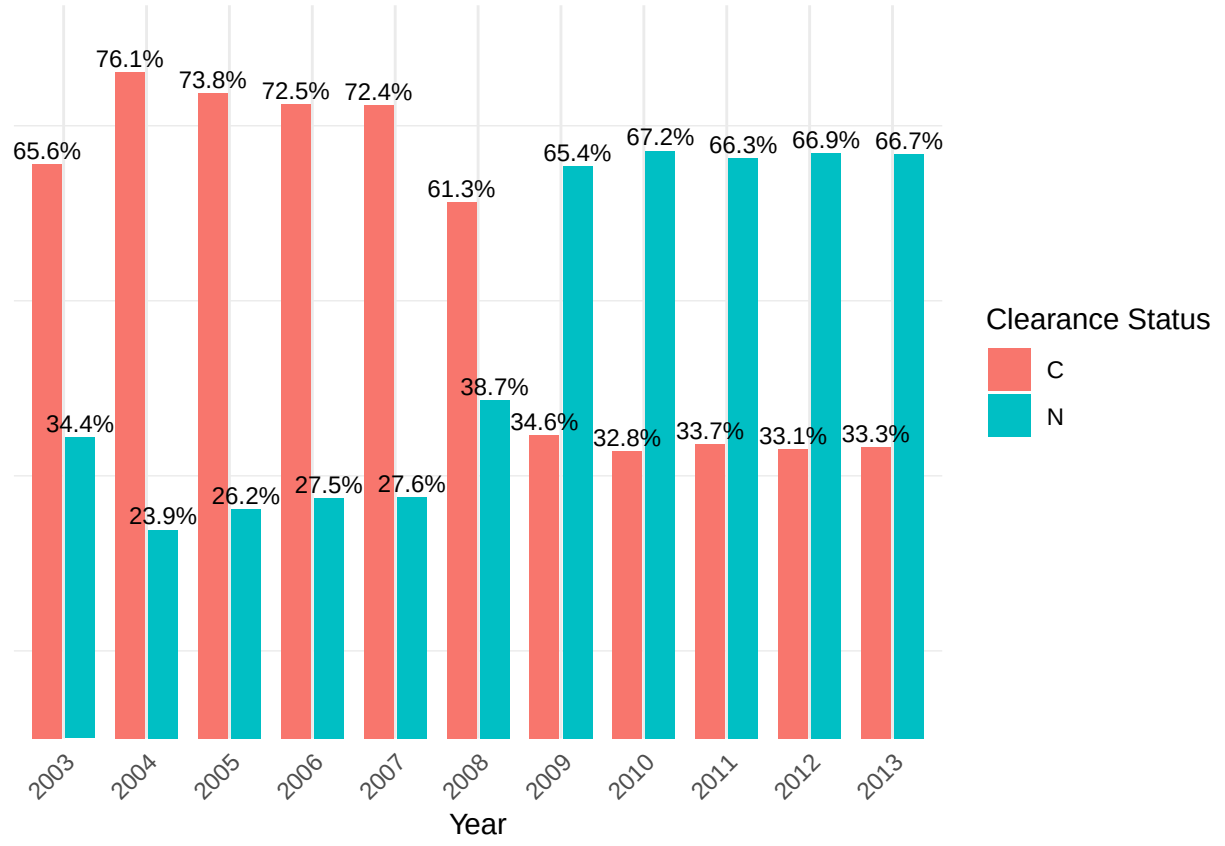
```
  fill = clearance_status
)) +
geom_col(position = position_dodge(width = 0.8), width = 0.7) +
geom_text(aes(label = n),
          position = position_dodge(width = 0.8),
          # center labels vertically and nudge them slightly to the right
          vjust = 0.5,
          hjust = -0.1,
          size = 3) +
coord_flip(expand = FALSE) +
scale_fill_manual(
  name   = "Clearance Status",
  values = c("C" = "#009E73", "N" = "#CC79A7")
) +
scale_y_continuous(expand = expansion(add = c(0, 0))) +
labs(x = NULL, y = "Number of Cases") +
theme_minimal(base_size = 12) +
theme(
  # remove the x-axis text (we've flipped)
  axis.text.x        = element_blank(),
  axis.ticks.x       = element_blank(),
  panel.grid.major.x = element_blank(),
  panel.grid.minor   = element_blank(),
  # tighten up margins so labels fit
  plot.margin        = margin(5, 5, 5, 5)
)
```



Crime clearance rates (2003-2013), (2014-2024)

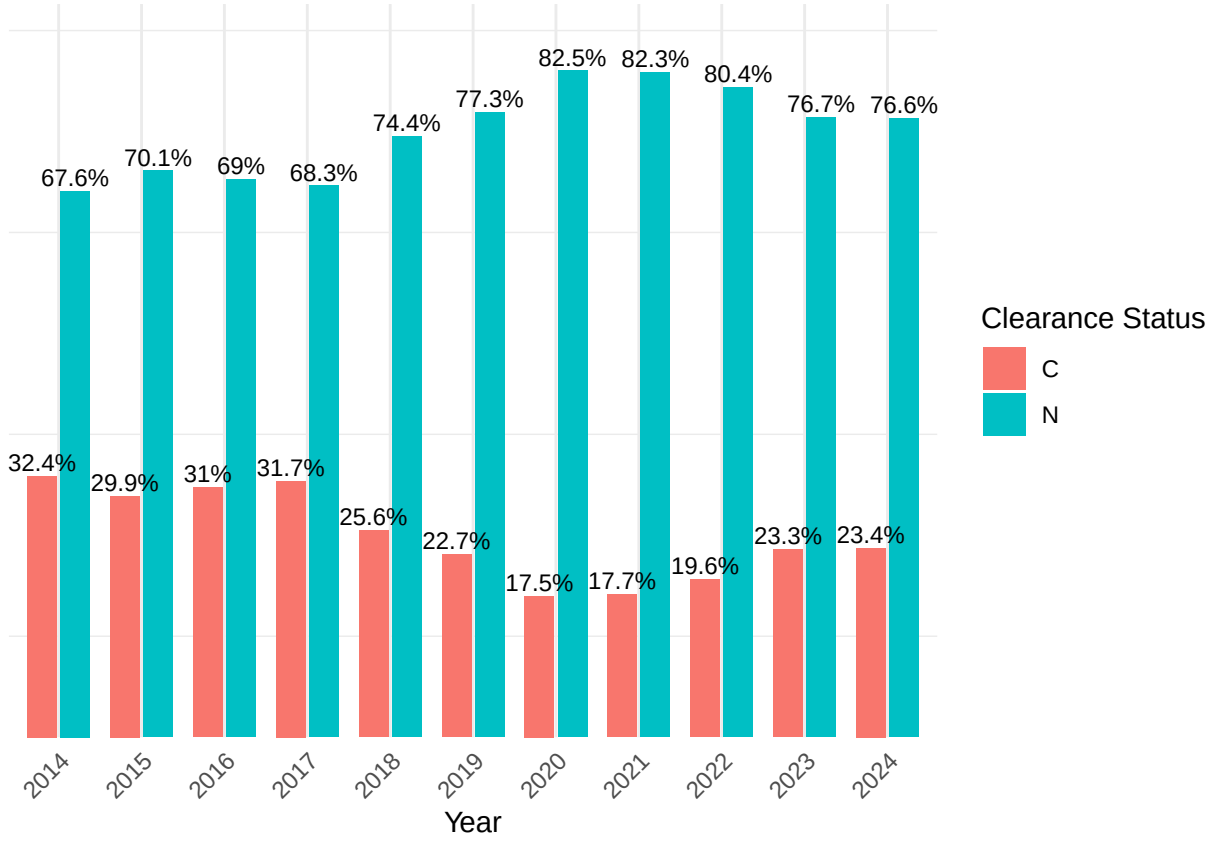
```
# Compute yearly percentages of clearance_status
yearly_pct <- crime %>%
  # count per year and status
  count(occurred_date_time_year, clearance_status) %>%
  group_by(occurred_date_time_year) %>%
  mutate(
    pct = n / sum(n) * 100,
    pct_label = paste0(round(pct, 1), "%")
  ) %>%
  ungroup()

# 1. Years 2003-2013
yearly_pct %>%
  filter(between(occurred_date_time_year, 2003, 2013)) %>%
  ggplot(aes(
    x = factor(occurred_date_time_year),
    y = pct,
    fill = clearance_status
  )) +
  geom_col(position = position_dodge(width = 0.8), width = 0.7) +
  geom_text(aes(label = pct_label),
            position = position_dodge(width = 0.8),
            vjust = -0.3,
            size = 3) +
  scale_y_continuous(expand = expansion(mult = c(0, 0.1))) +
```

```
  labs(
    x        = "Year",
    y        = NULL,
    fill     = "Clearance Status"
  ) +
  theme_minimal() +
  theme(
    axis.ticks.y       = element_blank(),
    axis.text.y        = element_blank(),
    panel.grid.major.y = element_blank(),
    axis.text.x        = element_text(angle = 45, hjust = 1)
  )
```




```
# 2. Years 2014-2024
yearly_pct %>%
  filter(between(occurred_date_time_year, 2014, 2024)) %>%
  ggplot(aes(
    x = factor(occurred_date_time_year),
    y = pct,
    fill = clearance_status
  )) +
  geom_col(position = position_dodge(width = 0.8), width = 0.7) +
  geom_text(aes(label = pct_label),
            position = position_dodge(width = 0.8),
            vjust = -0.3,
```

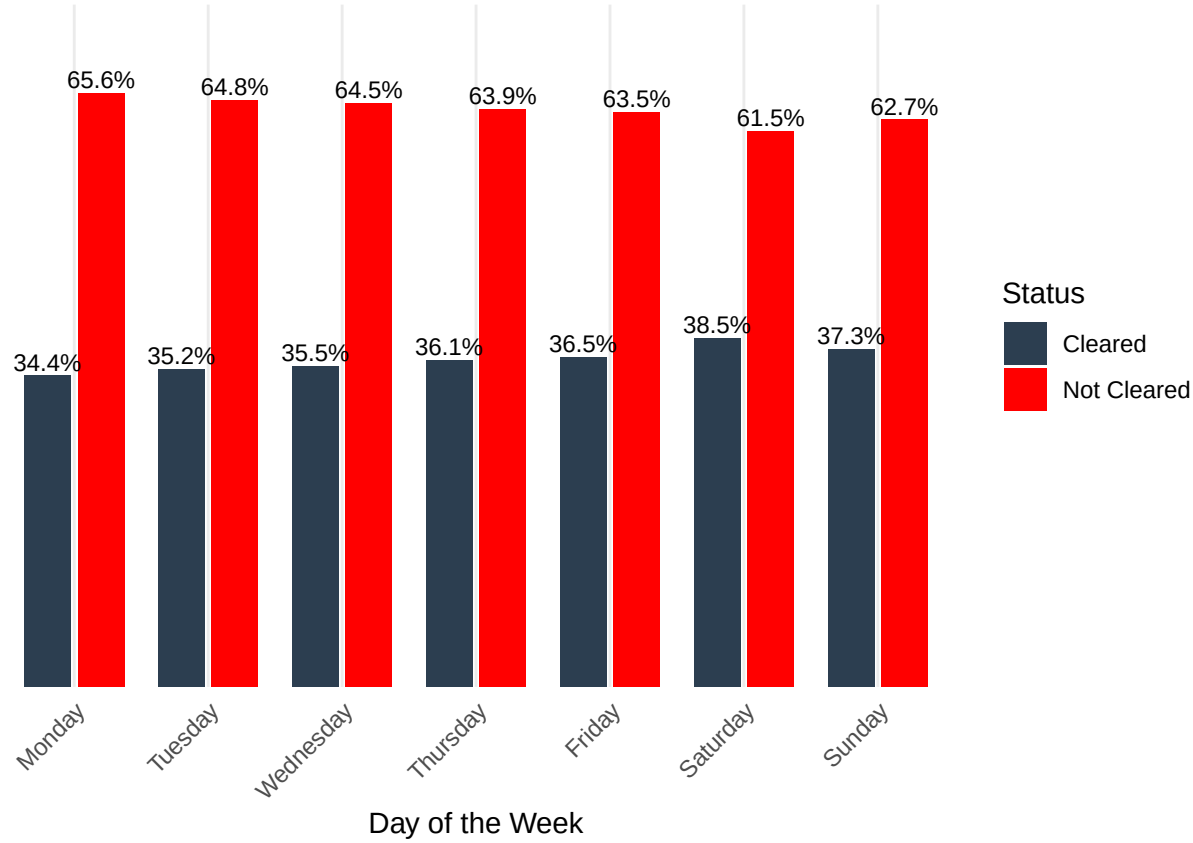
```
            size = 3) +
  scale_y_continuous(expand = expansion(mult = c(0, 0.1))) +
  labs(
    x        = "Year",
    y        = NULL,
    fill     = "Clearance Status"
  ) +
  theme_minimal() +
  theme(
    axis.ticks.y       = element_blank(),
    axis.text.y        = element_blank(),
    panel.grid.major.y = element_blank(),
    axis.text.x        = element_text(angle = 45, hjust = 1)
  )
```



Crime clearance rates according to days in a week.

```
library(tidyverse)

# 1. Compute percentages per day of week
day_pct <- crime %>%
  count(occurred_date_time_day_of_week, clearance_status) %>%
  group_by(occurred_date_time_day_of_week) %>%
  mutate(
    pct       = n / sum(n) * 100,
    pct_label = paste0(round(pct, 1), "%")
```

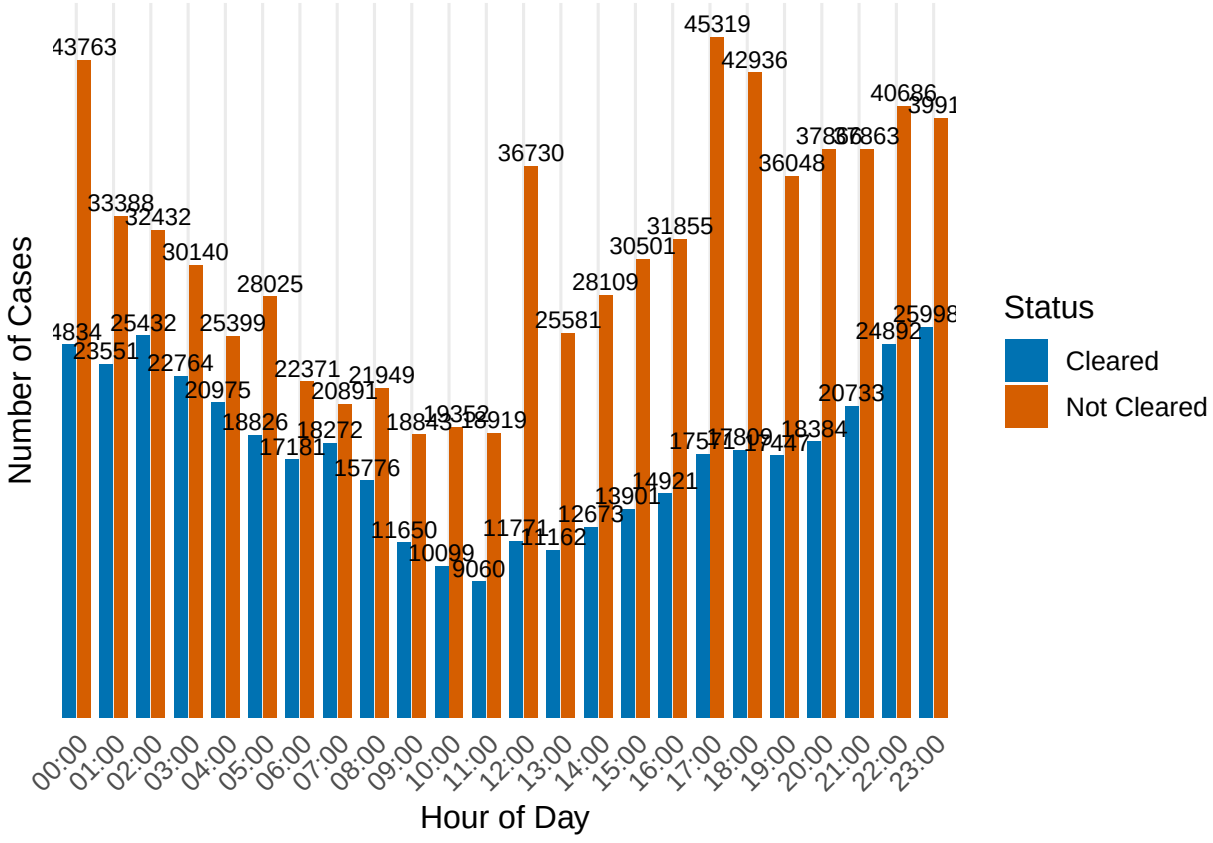
```
  ) %>%
  ungroup()

# 2. Plot
day_pct %>%
  # ensure days go Monday → Sunday
  mutate(day = fct_relevel(occurred_date_time_day_of_week,
                           "Monday","Tuesday","Wednesday",
                           "Thursday","Friday","Saturday","Sunday")) %>%
  ggplot(aes(x = day, y = pct, fill = clearance_status)) +
  geom_col(position = position_dodge(width = 0.8), width = 0.7) +
  geom_text(aes(label = pct_label),
            position = position_dodge(width = 0.8),
            vjust = -0.3, size = 3) +
  scale_fill_manual(
    values = c("C" = "#2C3E50", "N" = "red"),
    labels = c("C" = "Cleared", "N" = "Not Cleared")
  ) +
  scale_y_continuous(expand = expansion(mult = c(0, 0.15))) +
  labs(
    x       = "Day of the Week",
    y       = NULL,
    fill    = "Status"
  ) +
  theme_minimal() +
  theme(
    axis.ticks.y        = element_blank(),
    axis.text.y         = element_blank(),
    panel.grid.major.y  = element_blank(),
    panel.grid.minor.y  = element_blank(),
    axis.text.x         = element_text(angle = 45, hjust = 1)
  )
```



Hour wise Crime Clearances

```
library(dplyr)
library(ggplot2)
library(forcats)

# 1. Compute hour-wise counts
hour_counts <- crime %>%
  count(occurred_date_time_hour, clearance_status)

# 2. Plot Counts Cleared vs Not Cleared by hour
hour_counts %>%
  # ensure 0-23 ordering
  mutate(
    hour = factor(
      occurred_date_time_hour,
      levels = 0:23,
      labels = sprintf("%02d:00", 0:23)
    )
  ) %>%
  ggplot(aes(x = hour, y = n, fill = clearance_status)) +
  geom_col(position = position_dodge(width = 0.8), width = 0.7) +
  geom_text(aes(label = n),
            position = position_dodge(width = 0.8),
            vjust = -0.3,
            size = 3) +
  scale_fill_manual(
```

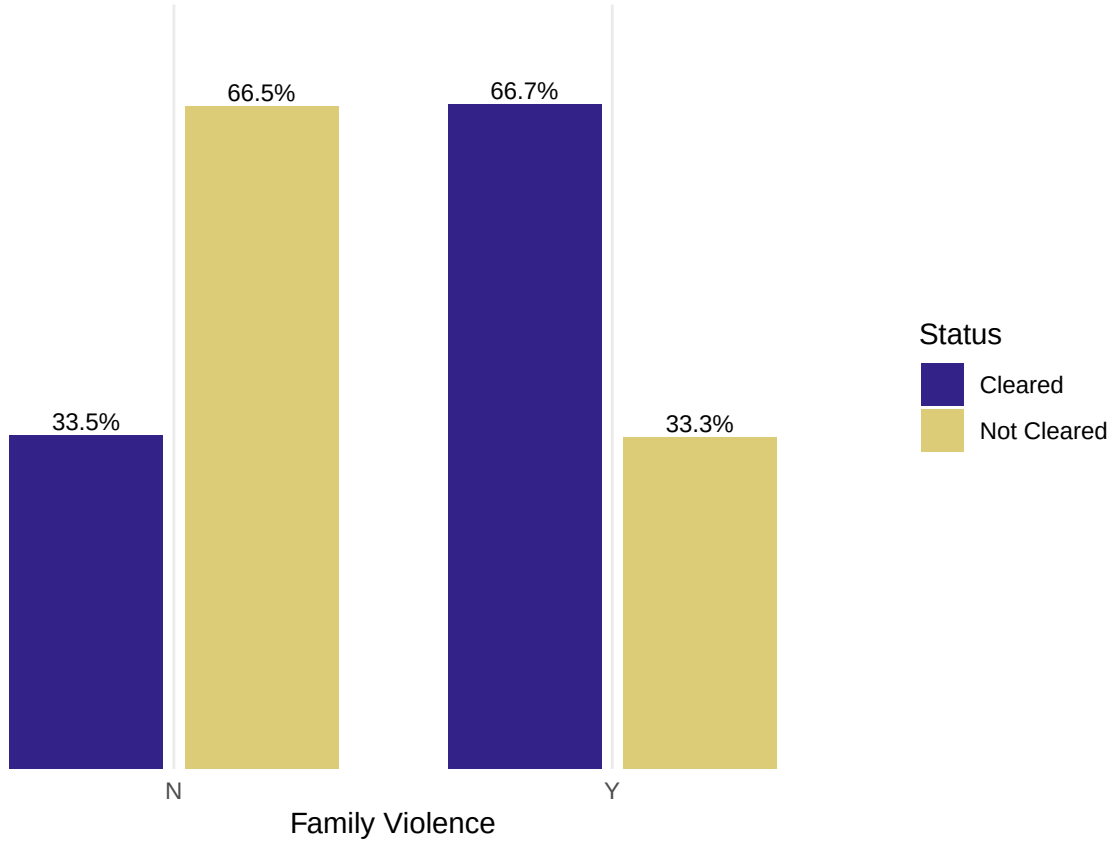
```
    name   = "Status",
    values = c("C" = "#0072B2",  # blue
               "N" = "#D55E00"), # orange
    labels = c("C" = "Cleared", "N" = "Not Cleared")
  ) +
  scale_y_continuous(expand = expansion(mult = c(0, 0.05))) +
  labs(
    x = "Hour of Day",
    y = "Number of Cases"
  ) +
  theme_minimal(base_size = 12) +
  theme(
    axis.text.x         = element_text(angle = 45, hjust = 1),
    axis.ticks.y        = element_blank(),
    axis.text.y         = element_blank(),
    panel.grid.major.y  = element_blank(),
    panel.grid.minor.y  = element_blank()
  )
```



Family Violence's influence in crime clearance rates

```
library(tidyverse)

# 1. Compute percentages by family_violence & clearance_status
family_pct <- crime %>%
  count(family_violence, clearance_status) %>%
```

```r
  group_by(family_violence) %>%
  mutate(
    pct       = n / sum(n) * 100,
    pct_label = paste0(round(pct, 1), "%")
  ) %>%
  ungroup()

# 2. Plot % Cleared vs Not Cleared by Family Violence
ggplot(family_pct, aes(
    x    = family_violence,
    y    = pct,
    fill = clearance_status
  )) +
  geom_col(position = position_dodge(width = 0.8), width = 0.7) +
  geom_text(aes(label = pct_label),
            position = position_dodge(width = 0.8),
            vjust = -0.3,
            size = 3) +
  scale_fill_manual(
    values = c("C" = "#332288", "N" = "#DDCC77"),  # color-blind friendly
    labels = c("C" = "Cleared",    "N" = "Not Cleared")
  ) +
  scale_y_continuous(expand = expansion(mult = c(0, 0.15))) +
  labs(
    x       = "Family Violence",
    y       = NULL,
    fill    = "Status"
  ) +
  theme_minimal() +
  theme(
    axis.ticks.y        = element_blank(),
    axis.text.y         = element_blank(),
    panel.grid.major.y  = element_blank(),
    panel.grid.minor.y  = element_blank()
  )
```



Influence of Spatial locations in crime clearing.

```
# 1. Total crime count by APD Sector
sector_count <- crime %>%
  count(apd_sector, sort = TRUE)

ggplot(sector_count, aes(
    x = fct_reorder(apd_sector, n),
    y = n
  )) +
  geom_col(fill = "#009E73") +                                    # color-blind friendly green
  geom_text(aes(label = n),
            vjust = -0.5,
            size = 3) +
  scale_y_continuous(expand = expansion(mult = c(0, 0.1))) +
  labs(
    x       = "APD Sector",
    y       = NULL,
    caption = "Figure 8: Total crime count by APD Sector"
  ) +
  theme_minimal() +
  theme(
    axis.ticks.y        = element_blank(),
    axis.text.y         = element_blank(),
    panel.grid.major.y  = element_blank(),
    axis.text.x         = element_text(angle = 45, hjust = 1)
  )
```

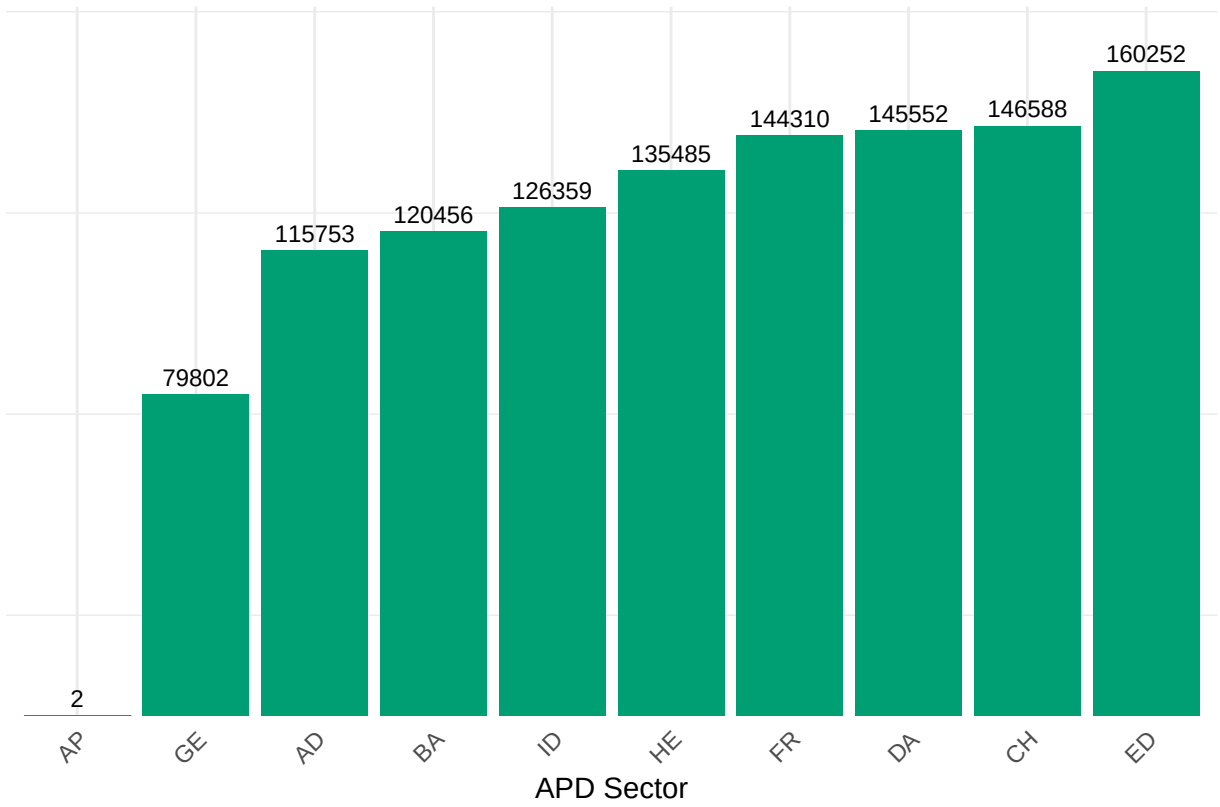


Figure 8: Total crime count by APD Sector

```
# 2. Percentage Cleared vs Not Cleared by APD Sector
sector_pct <- crime %>%
  count(apd_sector, clearance_status) %>%
  left_join(sector_count, by = "apd_sector", suffix = c("", ".total")) %>%
  group_by(apd_sector) %>%
  mutate(
    pct       = n / n.total * 100,
    pct_label = paste0(round(pct, 1), "%")
  ) %>%
  ungroup() %>%
  # keep same sector ordering as the count plot
  mutate(apd_sector = factor(apd_sector, levels = sector_count$apd_sector))

ggplot(sector_pct, aes(
    x = apd_sector,
    y = pct,
    fill = clearance_status
  )) +
  geom_col(position = position_dodge(width = 0.8), width = 0.7) +
  geom_text(aes(label = pct_label),
            position = position_dodge(width = 0.8),
            vjust = -0.3,
            size = 3) +
  scale_fill_manual(
    values = c("C" = "#117733", "N" = "#CC6677"),                # color-blind friendly orange & blue
    labels = c("C" = "Cleared",  "N" = "Not Cleared")
```

```
) +
scale_y_continuous(expand = expansion(mult = c(0, 0.15))) +
labs(
  x        = "APD Sector",
  y        = NULL,
  fill     = "Status"
) +
theme_minimal() +
theme(
  axis.ticks.y        = element_blank(),
  axis.text.y         = element_blank(),
  panel.grid.major.y  = element_blank(),
  axis.text.x         = element_text(angle = 45, hjust = 1)
)
```