

LAcrimedata+Covid

Isha Ganavdiya

2024-11-12

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed      dist
##  Min.   : 4.0    Min.   :  2.00
##  1st Qu.:12.0    1st Qu.: 26.00
##  Median :15.0    Median : 36.00
##  Mean   :15.4    Mean   : 42.98
##  3rd Qu.:19.0    3rd Qu.: 56.00
##  Max.   :25.0    Max.   :120.00
```

Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

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

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##   filter, lag

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

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

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

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

```

# Load the data
crime_data <- read.csv("/Users/ishaganavdiya/Downloads/Crime_Data_from_2020_to_Present.csv")
# Convert 'Date.Rptd' to Date format, extracting only the date part
crime_data$Date.Rptd <- mdy_hms(crime_data$Date.Rptd)
# Extract the year from the 'Date.Rptd' column
crime_data$Year <- year(crime_data$Date.Rptd)

# Calculate annual crime counts
annual_crime_counts <- crime_data %>%
  group_by(Year) %>%
  summarise(total_crimes = n())

# View the results
print(annual_crime_counts)

```

```

## # A tibble: 5 x 2
##   Year total_crimes
##   <dbl>     <int>
## 1  2020     192708
## 2  2021     208284
## 3  2022     235065
## 4  2023     234642
## 5  2024     111939

```

```

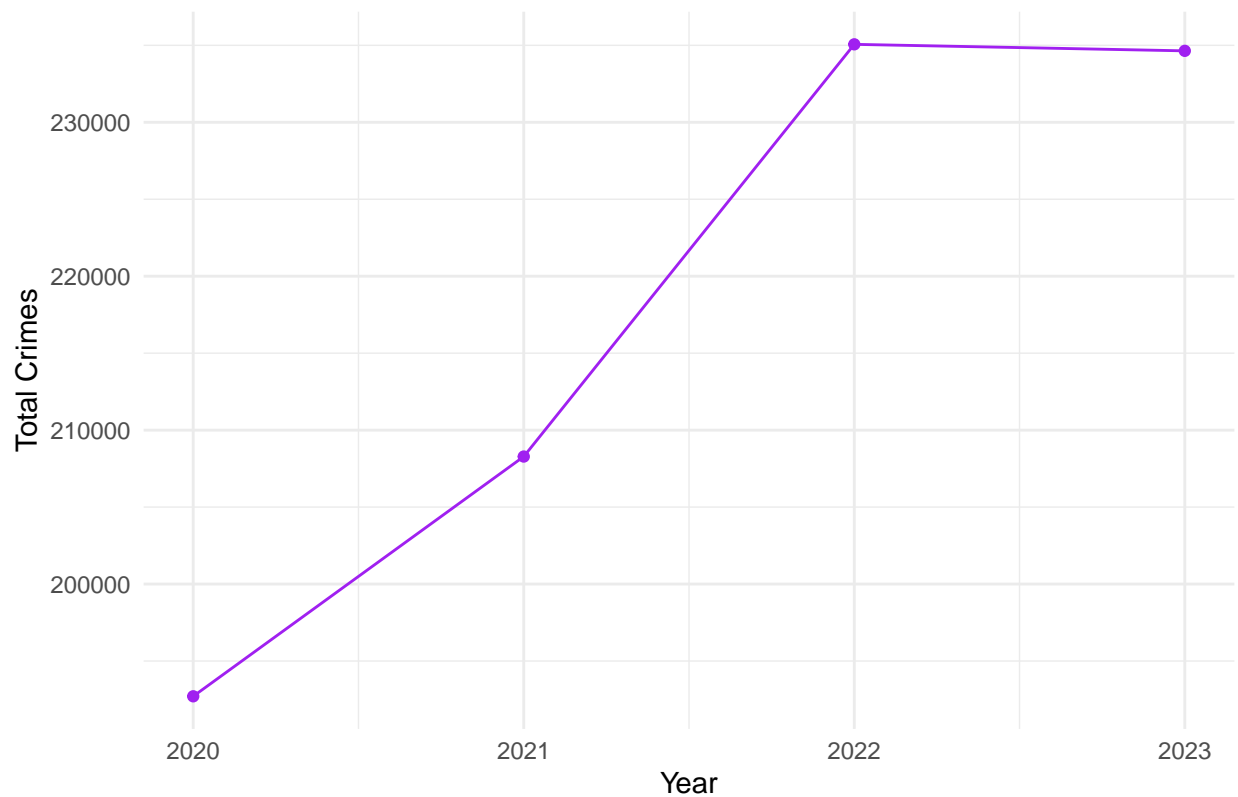
# Load necessary library
library(ggplot2)

# Filter out the year 2024 from the data
annual_crime_counts_filtered <- annual_crime_counts %>%
  filter(Year < 2024)

# Plot annual crime counts excluding 2024
ggplot(annual_crime_counts_filtered, aes(x = Year, y = total_crimes)) +
  geom_line(color = "purple") +
  geom_point(color = "purple") +
  labs(
    title = "Annual Crime Counts in Los Angeles (2020-2023)",
    x = "Year",
    y = "Total Crimes"
  ) +
  theme_minimal()

```

Annual Crime Counts in Los Angeles (2020–2023)



```
#Load COVID-19 Case Data
```

```
covid_data <- read.csv("/Users/ishaganavdiya/Downloads/LA_County_COVID_Cases.csv")
colnames(covid_data)
```

```
## [1] "county"      "state"       "fips"        "date"
## [5] "Lat"         "Lon"         "cases"       "deaths"
## [9] "people_tested" "state_cases" "state_deaths" "new_cases"
## [13] "new_deaths"  "new_state_cases" "new_state_deaths"
```

```
# Load libraries
```

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

```
# Convert 'date' column to Date format
```

```
covid_data$date <- mdy_hms(covid_data$date)
```

```
# Sort data by date (if it's not sorted)
```

```
covid_data <- covid_data %>% arrange(date)
```

```
# Calculate new cases by subtracting the previous day's cumulative cases from the current day's cases
```

```
covid_data <- covid_data %>%
```

```
  mutate(new_cases = cases - lag(cases, default = first(cases))) %>%
```

```
  mutate(new_deaths = deaths - lag(deaths, default = first(deaths)))
```

```
# Add Year column
```

```
covid_data$Year <- year(covid_data$date)

# Group by Year and calculate total new cases and new deaths for each year
annual_covid_counts <- covid_data %>%
  group_by(Year) %>%
  summarise(
    total_new_cases = sum(new_cases, na.rm = TRUE),
    total_new_deaths = sum(new_deaths, na.rm = TRUE)
  )

# View the results
print(annual_covid_counts)
```

```
## # A tibble: 4 x 3
##   Year total_new_cases total_new_deaths
##   <dbl>         <int>         <int>
## 1  2020         828084         10826
## 2  2021         950605         16819
## 3  2022        1860711          7085
## 4  2023         69721          789
```

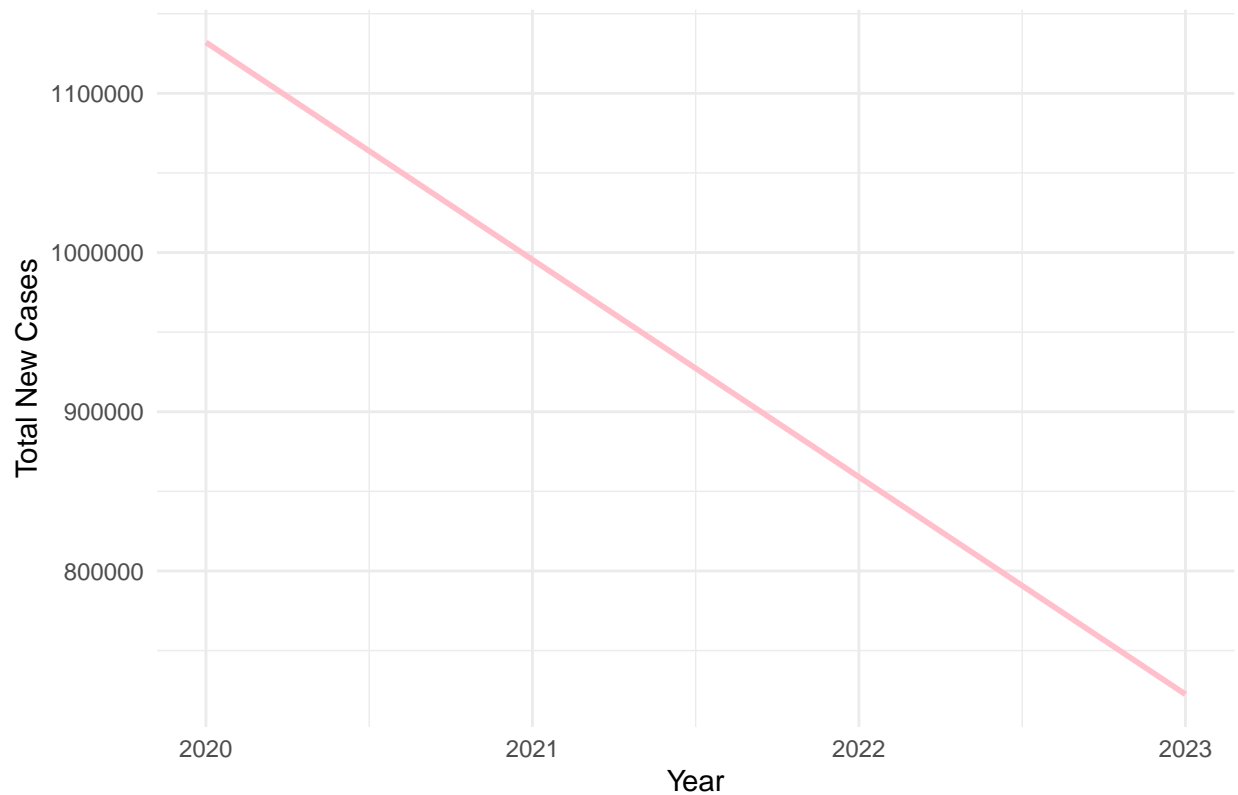
```
# Load necessary libraries
library(ggplot2)

# Create a data frame for total new cases and deaths
covid_data <- data.frame(
  Year = c(2020, 2021, 2022, 2023),
  total_new_cases = c(828084, 950605, 1860711, 69721),
  total_new_deaths = c(10826, 16819, 7085, 789)
)

# Plot only the trend line for total new cases in pink
ggplot(covid_data, aes(x = Year, y = total_new_cases)) +
  geom_smooth(method = "lm", color = "pink", se = FALSE) + # Trend line in pink
  labs(title = "Trend of Total New COVID Cases (2020-2023)",
    x = "Year",
    y = "Total New Cases") +
  theme_minimal()
```

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

Trend of Total New COVID Cases (2020–2023)



```
# Interpolate missing values for 'total_new_cases'
library(zoo)
```

```
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##   as.Date, as.Date.numeric
```

```
# Merge crime data and covid data on the 'Year' column
merged_data <- merge(annual_crime_counts, annual_covid_counts, by = "Year")

merged_data$total_new_cases <- na.approx(merged_data$total_new_cases, na.rm = FALSE)
head(merged_data %>% filter(Year < 2024))
```

```
##   Year total_crimes total_new_cases total_new_deaths
## 1 2020      192708      828084         10826
## 2 2021      208284      950605         16819
## 3 2022      235065     1860711          7085
## 4 2023      234642       69721          789
```

```
summary(merged_data %>% filter(Year < 2024))
```

| | Year | total_crimes | total_new_cases | total_new_deaths |
|----|--------------|----------------|-----------------|------------------|
| ## | Min. :2020 | Min. :192708 | Min. : 69721 | Min. : 789 |
| ## | 1st Qu.:2021 | 1st Qu.:204390 | 1st Qu.: 638493 | 1st Qu.: 5511 |
| ## | Median :2022 | Median :221463 | Median : 889344 | Median : 8956 |
| ## | Mean :2022 | Mean :217675 | Mean : 927280 | Mean : 8880 |
| ## | 3rd Qu.:2022 | 3rd Qu.:234748 | 3rd Qu.:1178132 | 3rd Qu.:12324 |
| ## | Max. :2023 | Max. :235065 | Max. :1860711 | Max. :16819 |

```

ggplot(merged_data %>% filter(Year < 2024), aes(x = Year)) +
  # Crime data (primary y-axis) with log scale if needed
  geom_line(aes(y = total_crimes, color = "Crime Trend"), size = 1) +
  # COVID data (secondary y-axis)
  geom_line(aes(y = total_new_cases, color = "COVID Trend"), size = 1, linetype = "dashed") +
  scale_color_manual(values = c("Crime Trend" = "purple", "COVID Trend" = "pink")) +
  labs(title = "Crime vs COVID Trends in Los Angeles (2020-2023)",
       x = "Year",
       y = "Crime Count",
       color = "Legend") +
  scale_y_continuous(name = "Crime Count",
                     limits = c(0, max(merged_data$total_crimes, na.rm = TRUE)),
                     trans = "log") + # Optional: use log scale for crime data
  scale_y_continuous(sec.axis = sec_axis(~ ., name = "COVID New Cases")) +
  theme_minimal() +
  theme(legend.position = "bottom")

```

```

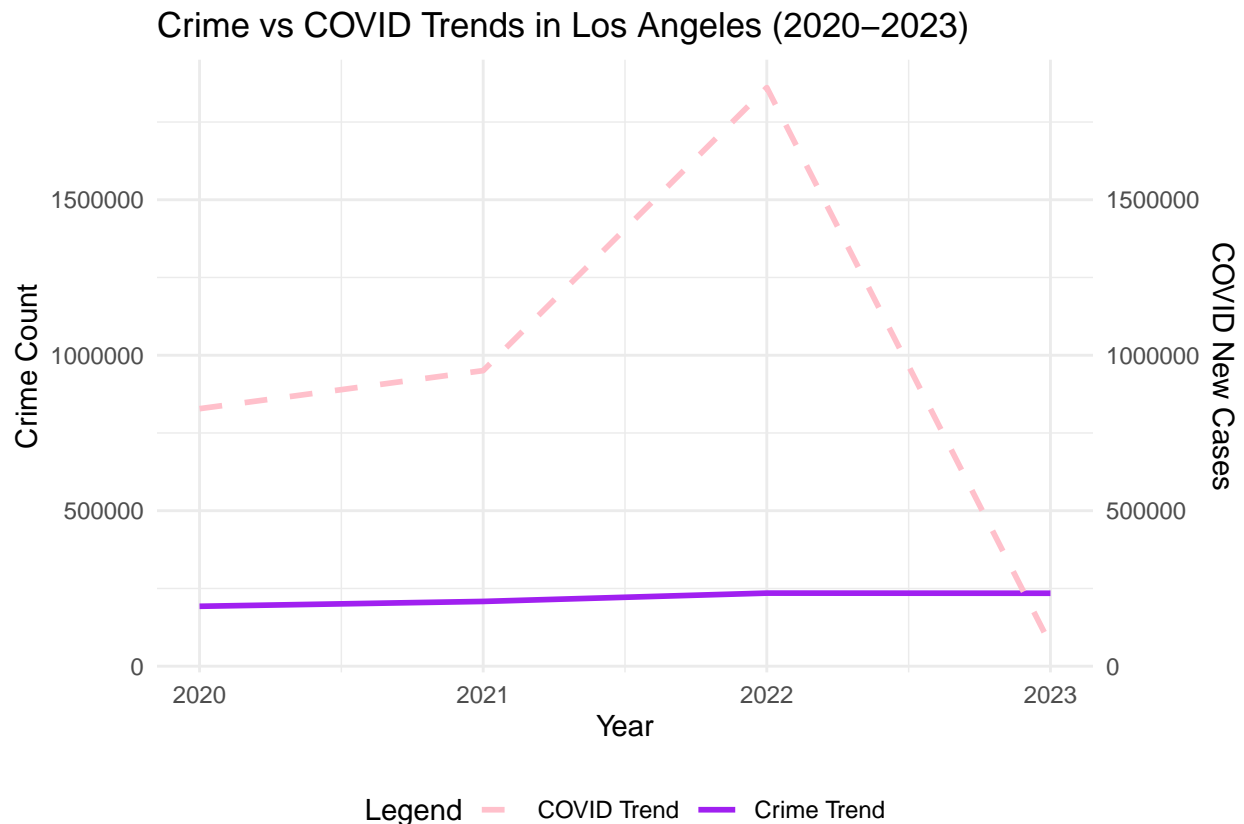
## Warning: Using 'size' aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use 'linewidth' instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.

```

```

## Scale for y is already present.
## Adding another scale for y, which will replace the existing scale.

```



```
merged_data %>%
  filter(Year %in% c(2020, 2021)) %>%
  select(Year, total_crimes, total_new_cases)
```

```
##   Year total_crimes total_new_cases
## 1 2020      192708      828084
## 2 2021      208284      950605
```

```
# Calculate correlation between total crimes and total new cases for 2020 and 2021
correlation_result <- cor(merged_data$total_crimes, merged_data$total_new_cases, use = "complete.obs")
correlation_result
```

```
## [1] 0.08586651
```

I was expecting this show a linear positive correlation, but the correlation coefficient obtained, shows a weak positive linear relationship between the two datasets.