When and Where: Examining Crisis Service Calls in Toronto*

A Detailed Look at Temporal and Spatial Trends in Crisis Response Calls

Xinxiang Gao

September 23, 2024

First sentence. Second sentence. Third sentence. Fourth sentence.

1 Introduction

R Core Team (2023), Wickham et al. (2019), Wickham (2016), Wilke (2024), Auguie (2017) Gelfand (2022)

```
data <- read.csv("../data/analysis_data/piccfsa.csv")
```

2 Data

```
ggplot(data, aes(x = event_month, fill = event_type)) +
  geom_bar(position = "dodge") +
  labs(title = "Event Type Counts Grouped by Day of the Week",
       x = "Day of the Week",
       y = "Count",
       fill = "Event Type") +
  theme_minimal() +
  coord_flip()
```

*Code and data are available at: https://github.com/xgao28/PICCFSA.

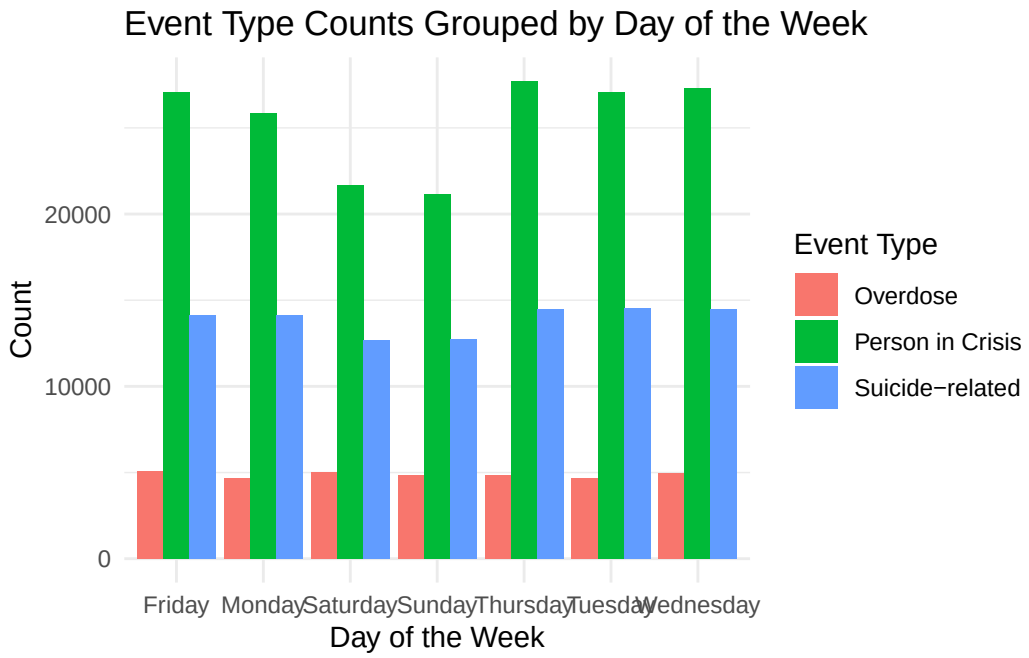
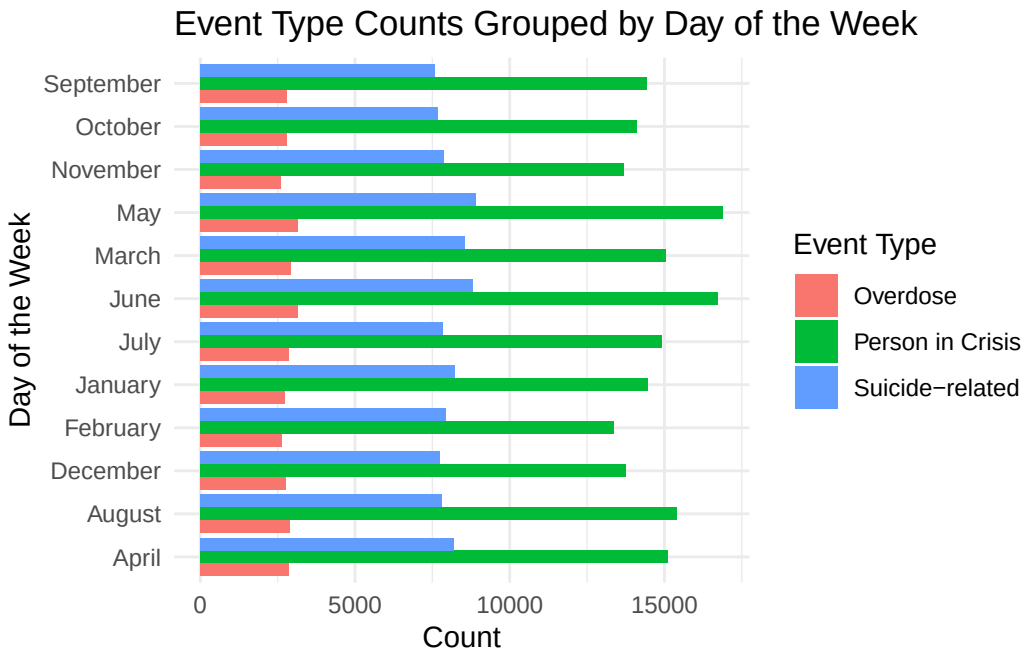


Figure 1: Bills of penguins



Talk more about it.

And also planes (Figure 2). (You can change the height and width, but don't worry about doing that until you have finished every other aspect of the paper - Quarto will try to make it

look nice and the defaults usually work well once you have enough text.)

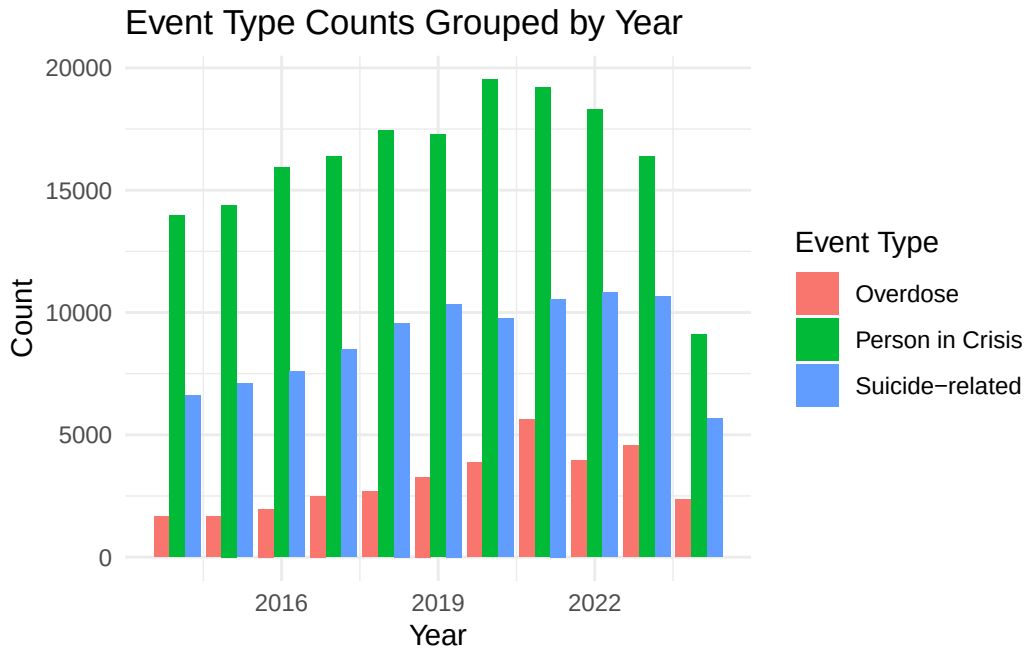


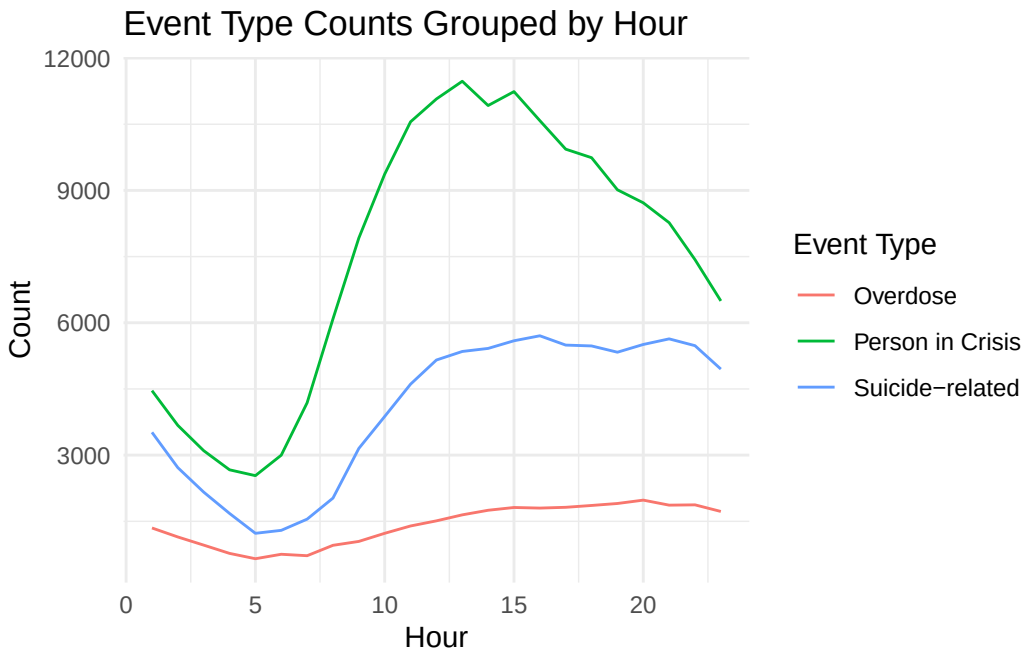
Figure 2: Relationship between wing length and width

```
df_summarized <- data %>%
  group_by(event_hour, event_type) %>%
  summarise(count = n()) %>%
  ungroup()
```

```
`summarise()` has grouped output by 'event_hour'. You can override using the
`.groups` argument.
```

```
# Create the line plot
ggplot(df_summarized, aes(x = event_hour, y = count, color = event_type, group = event_type))
  geom_line() +
  labs(title = "Event Type Counts Grouped by Hour",
       x = "Hour",
       y = "Count",
       color = "Event Type") +
  theme_minimal()
```

```
Warning: Removed 3 rows containing missing values (`geom_line()`).
```




```
library(cowplot)  # For combining plots
```

```
Attaching package: 'cowplot'

The following object is masked from 'package:lubridate':

    stamp
```

```
# Step 1: Summarize the data by counting occurrences of each event_type
df_summarized <- data %>%
  group_by(hood_158, event_type) %>%
  summarise(count = n()) %>%
  ungroup()
```

```
`summarise()` has grouped output by 'hood_158'. You can override using the
`.groups` argument.
```

```
# Step 2: Split the summarized data into separate data frames for each event_type
event_types <- unique(df_summarized$event_type)

# Create a list to hold the top 10 data frames and plots
```

```r
df_list <- list()
plot_list <- list()

for (event in event_types) {
  # Filter the top 10 neighborhoods for the current event type
  df_top10 <- df_summarized %>%
    filter(event_type == event) %>%
    arrange(desc(count)) %>%
    head(10)

  # Save the data frame to the list
  df_list[[event]] <- df_top10

  # Step 3: Create a bar plot for the top 10 neighborhoods for the current event type
  plot <- ggplot(df_top10, aes(x = reorder(hood_158, count), y = count)) +
    geom_bar(stat = "identity") +
    labs(title = event,
         x = "Neighborhood 158",
         y = "Count") +
    theme_minimal() +
    theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
    coord_flip()

  # Save the plot to the list
  plot_list[[event]] <- plot
}

# Step 4: Get a look-up table
neighborhood_table <- data %>%
  select(hood_158, neighbourhood_158) %>%
  distinct() %>%
  arrange(hood_158)

# Convert the table to a grid object using gridExtra::tableGrob
table_plot <- gridExtra::tableGrob(neighborhood_table)

# Step 5: Combine the individual plots using cowplot
combined_plot <- cowplot::plot_grid(plotlist = plot_list, ncol = 1)

# Step 6: Display the combined plot
print(combined_plot)
```


Overdose
Person in Crisis
Suicide−related
Neighborhood 1
Neighborhood 158
0
1000
2000
3000
4000
6000
Count

References

Auguie, Baptiste. 2017. *gridExtra: Miscellaneous Functions for "Grid" Graphics*. https://CRAN.R-project.org/package=gridExtra.

Gelfand, Sharla. 2022. *Opendatatoronto: Access the City of Toronto Open Data Portal*. https://CRAN.R-project.org/package=opendatatoronto.

R Core Team. 2023. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Wickham, Hadley. 2016. *Ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. https://ggplot2.tidyverse.org.

Wickham, Hadley, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D'Agostino McGowan, Romain François, Garrett Grolemund, et al. 2019. "Welcome to the tidyverse." *Journal of Open Source Software* 4 (43): 1686. https://doi.org/10.21105/joss.01686.

Wilke, Claus O. 2024. *Cowplot: Streamlined Plot Theme and Plot Annotations for 'Ggplot2'*. https://CRAN.R-project.org/package=cowplot.