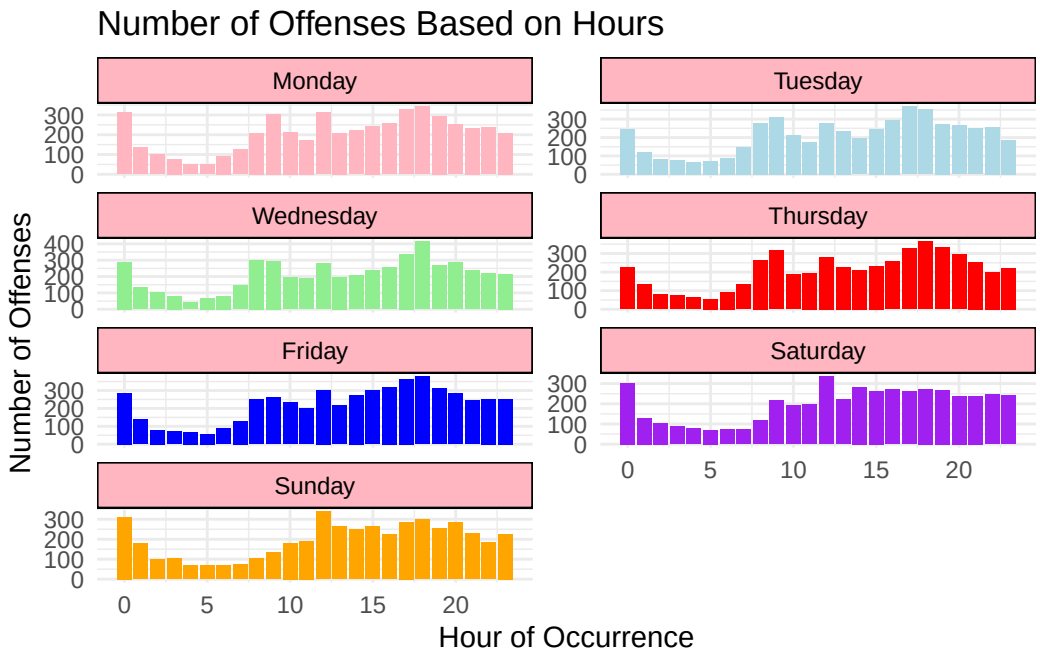
CHANGE TITLE

Ruibo Sun and Nixi Huang

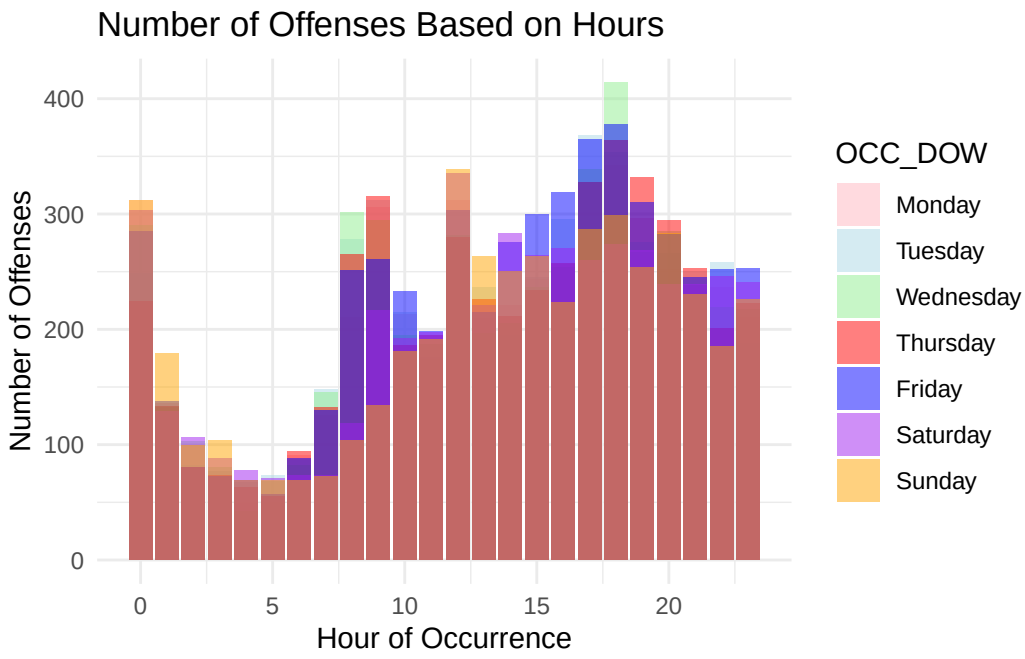
11 March 2024

ADD ABSTRACT

0.1 Plot 1 version 1



0.2 plot 1 version2

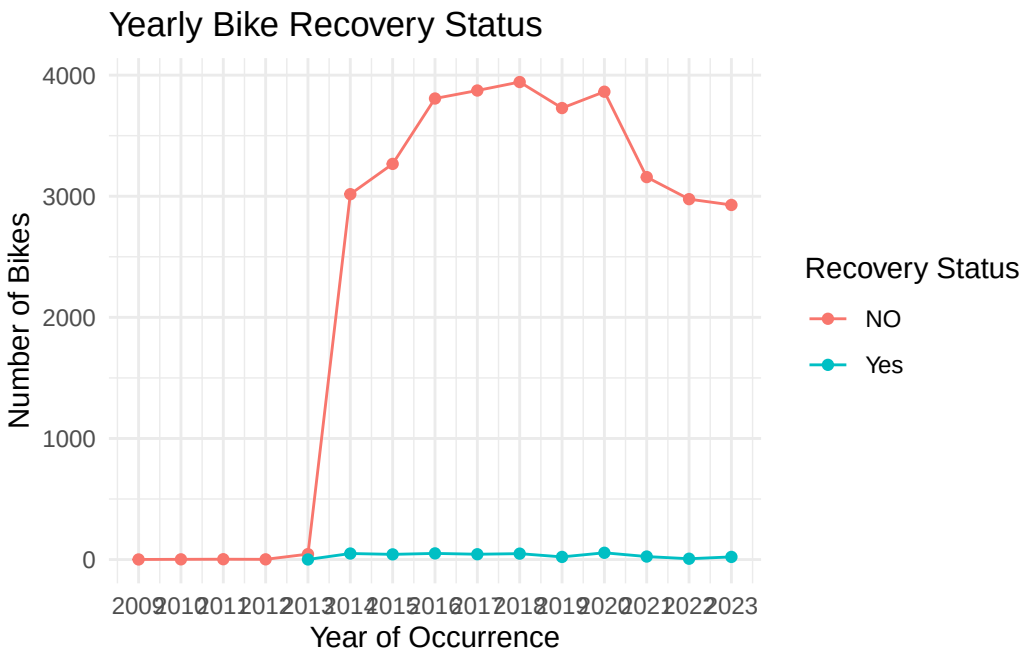


0.3 plot 2

```
# data cleaning for plot 2
#| include: false
#| warning: false
#| message: false

bike_raw$y_n <- ifelse(bike_raw$STATUS =='RECOVERED','Yes','NO')
```

```
#| echo: false
#| warning: false
#| message: false


# Assuming bike_raw has a column 'OCC_YEAR' for the year of occurrence
# and a column 'y_n' indicating if the bike was recovered ('Yes' or 'No')

# Summarize data
```

```r
yearly_counts <- bike_raw %>% filter(OCC_YEAR>2000) %>%
  group_by(OCC_YEAR, y_n) %>%
  summarise(count = n(), .groups = 'drop')

# Create the time series plot
ggplot(yearly_counts, aes(x = OCC_YEAR, y = count, color = y_n)) +
  geom_line() +  # Line plot
  geom_point() + # Add points to the lines
  labs(title = "Yearly Bike Recovery Status",
       x = "Year of Occurrence",
       y = "Number of Bikes",
       color = "Recovery Status") +
  theme_minimal() +
  scale_x_continuous(breaks = unique(yearly_counts$OCC_YEAR))  # Ensure all years are incl
```



0.4 plot 3: the map!!

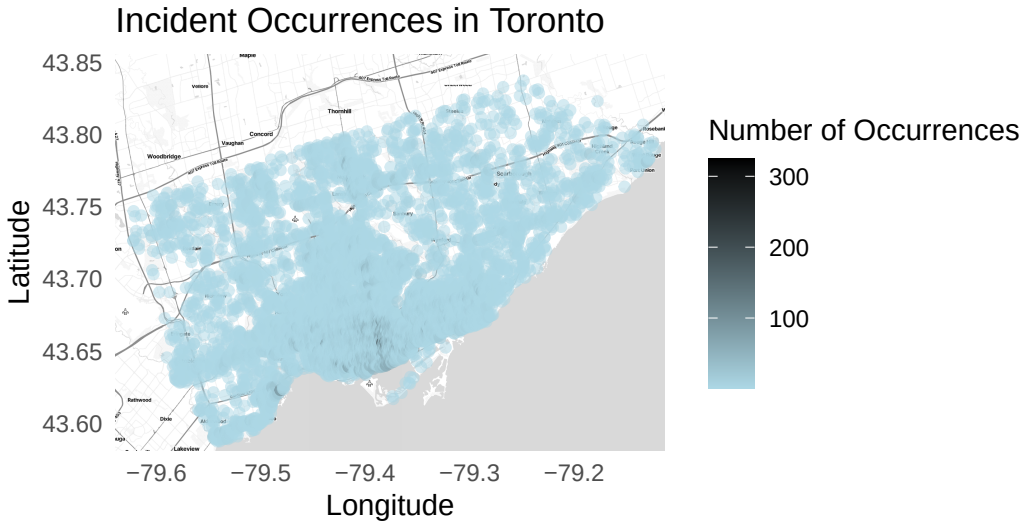
```
register_stadiamaps(key = "ddfcfc93-13e6-497f-8336-c5fdd043f311")

toronto_map <- get_stadiamap(bbox = c(left = -79.639219, bottom = 43.581024, right = -79.1
                             maptype = "stamen_toner_lite",
                             zoom = 12)
```

```
i © Stadia Maps © Stamen Design © OpenMapTiles © OpenStreetMap contributors.
```

```
# data cleaning
bike_aggregated <- bike_raw %>%
  group_by(LONG_WGS84, LAT_WGS84) %>%
  summarise(count = n(), .groups = 'drop')
```

```
ggmap(toronto_map) +
  geom_point(data = bike_aggregated, aes(x = LONG_WGS84, y = LAT_WGS84, color = count), al
  scale_color_gradient(low = "lightblue", high = "black") +  # Adjust color gradient as ne
  labs(title = "Incident Occurrences in Toronto",
       x = "Longitude",
       y = "Latitude",
       color = "Number of Occurrences") +
  theme_minimal()
```

```
Warning: Removed 1 rows containing missing values (`geom_point()`).
```



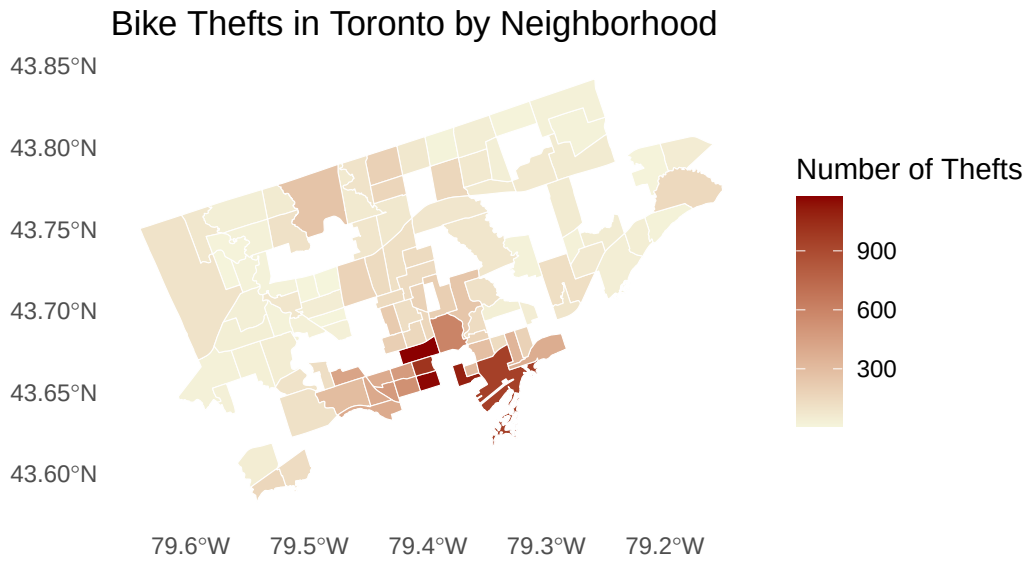

```
# Load necessary libraries
library(ggplot2)
library(sf)
```

```
Linking to GEOS 3.10.2, GDAL 3.4.2, PROJ 8.2.1; sf_use_s2() is TRUE
```

```
library(dplyr)

# Step 1: Load the geographic data
toronto_neighborhoods <- st_read(file.path("input/toneighshape/Neighbourhoods v2_region.sh
```

```
Reading layer `Neighbourhoods v2_region' from data source
  `/Users/xiuzh/Desktop/313/input/toneighshape/Neighbourhoods v2_region.shp'
  using driver `ESRI Shapefile'
Simple feature collection with 140 features and 2 fields
Geometry type: MULTIPOLYGON
Dimension:     XY
Bounding box:  xmin: 609589.4 ymin: 4826145 xmax: 651617.9 ymax: 4857224
Projected CRS: unnamed
```

```r
# Step 2: Aggregate bike theft data
# Replace `neighborhood_name` with the actual column name from your bike data
bike_thefts_agg <- bike_raw %>%
  group_by(NEIGHBOURHOOD_158) %>%
  summarise(count = n(), .groups = 'drop')

# Step 3: Merge the geographic and theft data
# Replace `NAME` with the neighborhood name column from the shapefile
toronto_thefts_geo <- merge(toronto_neighborhoods, bike_thefts_agg,
                            by.x = "NAME", by.y = "NEIGHBOURHOOD_158")

# Step 4: Create the map
ggplot(toronto_thefts_geo) +
  geom_sf(aes(fill = count), color = "white", size = 0.2) +
  scale_fill_gradient(low = "beige", high = "darkred") +  # Adjust color scale as needed
  labs(title = "Bike Thefts in Toronto by Neighborhood",
       fill = "Number of Thefts") +
  theme_minimal() +
  theme(legend.position = "right", panel.grid.major = element_blank(), panel.grid.minor =
```



0.5 plot 4: word cloud

```
library(wordcloud2)

bike_info<-bike_raw%>%select(BIKE_COLOUR,BIKE_COST,BIKE_MODEL)

x<-letterCloud(bike_info, word = "BIKE", color='random-light' , backgroundColor="black")
x
```