Urban Infestations: Unveiling the Hidden Epidemic

2025-03-06

Executive Summary

From 2010-2017, NYC rat sightings surged, peaking in 2016, with Brooklyn reporting the highest numbers. Seasonal patterns and residential building concentrations were key factors. Healthcare implications include increased zoonotic disease risks and psychological distress. We recommend an AI-driven pest management platform, leveraging real-time data for targeted interventions. This platform would offer subscription services, reducing healthcare costs and attracting socially responsible investment. Integrating preventative measures with data analytics enhances public health and offers a sustainable business model, aligning with the growing demand for proactive urban healthcare solutions.

Cleaning the Data

```
# Load necessary libraries
library(readr)
library(lubridate)
```

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

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

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.4.3

##
## 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
```

```r
# Read the CSV file while treating "", "NA", and "N/A" as missing values
df <- read_csv("data/A1_sightings.csv", na = c("", "NA", "N/A"))
```

```
## Warning: One or more parsing issues, call `problems()` on your data frame for details,
## e.g.:
##   dat <- vroom(...)
##   problems(dat)
```

```
## Rows: 101914 Columns: 52
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (33): Created Date, Closed Date, Agency, Agency Name, Complaint Type, De...
## dbl  (6): Unique Key, Incident Zip, X Coordinate (State Plane), Y Coordinate...
## lgl (13): Landmark, Facility Type, School or Citywide Complaint, Vehicle Typ...
##
## 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.
```

```r
# Convert the date columns to proper datetime format
df$`Created Date` <- mdy_hms(df$`Created Date`)
df$`Closed Date` <- mdy_hms(df$`Closed Date`)
df$`Due Date` <- mdy_hms(df$`Due Date`)
df$`Resolution Action Updated Date` <- mdy_hms(df$`Resolution Action Updated Date`)

# Separate Date and Time for each relevant column
df$Created_Date_only <- as.Date(df$`Created Date`)   # Extract only the date from Created Date
df$Created_Time <- format(df$`Created Date`, "%H:%M:%S")  # Extract time from Created Date

df$Closed_Date_only <- as.Date(df$`Closed Date`)   # Extract only the date from Closed Date
df$Closed_Time <- format(df$`Closed Date`, "%H:%M:%S")  # Extract time from Closed Date

df$Due_Date_only <- as.Date(df$`Due Date`)   # Extract only the date from Due Date
df$Due_Time <- format(df$`Due Date`, "%H:%M:%S")  # Extract time from Due Date

df$Resolution_Date_only <- as.Date(df$`Resolution Action Updated Date`)
# Extract only the date from Resolution Action Updated Date
df$Resolution_Time <- format(df$`Resolution Action Updated Date`, "%H:%M:%S")
# Extract time from Resolution Action Updated Date

# Order the data based on 'Created Date' (you can change it to any other date column if needed)
df <- df %>%
  arrange(Created_Date_only)  # Arrange by the 'Created Date' from oldest to most recent

# View the first few rows of the updated dataset
head(df)
```

```
## # A tibble: 6 x 60
##   `Unique Key` `Created Date`      `Closed Date` Agency `Agency Name`
##          <dbl> <dttm>              <dttm>        <chr>  <chr>
## 1     15633054 2010-01-01 11:20:45 NA            DOHMH  Department of Health an~
## 2     15633594 2010-01-01 15:05:37 NA            DOHMH  Department of Health an~
```

```
## 3     15633599 2010-01-01 20:52:19 NA            DOHMH  Department of Health an~
## 4     15633605 2010-01-01 16:14:27 NA            DOHMH  Department of Health an~
## 5     15633803 2010-01-01 08:29:58 NA            DOHMH  Department of Health an~
## 6     15633828 2010-01-01 14:15:27 NA            DOHMH  Department of Health an~
## # i 55 more variables: 'Complaint Type' <chr>, Descriptor <chr>,
## #   'Location Type' <chr>, 'Incident Zip' <dbl>, 'Incident Address' <chr>,
## #   'Street Name' <chr>, 'Cross Street 1' <chr>, 'Cross Street 2' <chr>,
## #   'Intersection Street 1' <chr>, 'Intersection Street 2' <chr>,
## #   'Address Type' <chr>, City <chr>, Landmark <lgl>, 'Facility Type' <lgl>,
## #   Status <chr>, 'Due Date' <dttm>, 'Resolution Action Updated Date' <dttm>,
## #   'Community Board' <chr>, Borough <chr>, ...
```

```r
# Create date-related variables
df$sighting_year <- year(df$`Created Date`)          # Extract year
df$sighting_month <- month(df$`Created Date`)        # Extract month
df$sighting_day <- day(df$`Created Date`)            # Extract day of the month
df$sighting_weekday <- wday(df$`Created Date`, label = TRUE, abbr = TRUE)
# Extract weekday name (abbreviated)

# View the updated dataset with the new columns
head(df)
```

```
## # A tibble: 6 x 64
##   'Unique Key' 'Created Date'      'Closed Date' Agency 'Agency Name'
##          <dbl> <dttm>              <dttm>        <chr>  <chr>
## 1     15633054 2010-01-01 11:20:45 NA            DOHMH  Department of Health an~
## 2     15633594 2010-01-01 15:05:37 NA            DOHMH  Department of Health an~
## 3     15633599 2010-01-01 20:52:19 NA            DOHMH  Department of Health an~
## 4     15633605 2010-01-01 16:14:27 NA            DOHMH  Department of Health an~
## 5     15633803 2010-01-01 08:29:58 NA            DOHMH  Department of Health an~
## 6     15633828 2010-01-01 14:15:27 NA            DOHMH  Department of Health an~
## # i 59 more variables: 'Complaint Type' <chr>, Descriptor <chr>,
## #   'Location Type' <chr>, 'Incident Zip' <dbl>, 'Incident Address' <chr>,
## #   'Street Name' <chr>, 'Cross Street 1' <chr>, 'Cross Street 2' <chr>,
## #   'Intersection Street 1' <chr>, 'Intersection Street 2' <chr>,
## #   'Address Type' <chr>, City <chr>, Landmark <lgl>, 'Facility Type' <lgl>,
## #   Status <chr>, 'Due Date' <dttm>, 'Resolution Action Updated Date' <dttm>,
## #   'Community Board' <chr>, Borough <chr>, ...
```

```r
# Remove original date columns
df <- df %>%
  select(-`Created Date`, -`Closed Date`, -`Due Date`, -`Resolution Action Updated Date`)

# Reorder the columns after verifying the correct column names
df <- df %>%
  select(1, sighting_year, sighting_month, sighting_day, sighting_weekday, everything())

# View the updated dataset
head(df)
```

```
## # A tibble: 6 x 60
##   'Unique Key' sighting_year sighting_month sighting_day sighting_weekday Agency
##          <dbl>         <dbl>          <dbl>        <int> <ord>            <chr>
```

```
## 1     15633054          2010              1            1 Fri              DOHMH
## 2     15633594          2010              1            1 Fri              DOHMH
## 3     15633599          2010              1            1 Fri              DOHMH
## 4     15633605          2010              1            1 Fri              DOHMH
## 5     15633803          2010              1            1 Fri              DOHMH
## 6     15633828          2010              1            1 Fri              DOHMH
## # i 54 more variables: `Agency Name` <chr>, `Complaint Type` <chr>,
## #   Descriptor <chr>, `Location Type` <chr>, `Incident Zip` <dbl>,
## #   `Incident Address` <chr>, `Street Name` <chr>, `Cross Street 1` <chr>,
## #   `Cross Street 2` <chr>, `Intersection Street 1` <chr>,
## #   `Intersection Street 2` <chr>, `Address Type` <chr>, City <chr>,
## #   Landmark <lgl>, `Facility Type` <lgl>, Status <chr>,
## #   `Community Board` <chr>, Borough <chr>, ...
```

```r
# Save the cleaned dataset as cleaned_sighting.csv in the data folder
write_csv(df, "data/cleaned_sighting.csv")
```

Analysis

```r
# Load necessary libraries
library(dplyr)
library(ggplot2)
library(readr)
library(scales)  # For comma formatting

# Load the cleaned dataset
cs <- read_csv("data/cleaned_sighting.csv")

# Filter the data for Closed and Assigned statuses
filtered_data <- cs %>%
  filter(!is.na(Latitude) & !is.na(Longitude)) %>%  # Remove rows with missing coordinates
  group_by(sighting_year, sighting_month) %>%
  count()  # Counting the occurrences of sightings for each year and month

# Aggregate sightings by year and month
trend_by_month <- filtered_data %>%
  group_by(sighting_year, sighting_month) %>%
  summarise(total_sightings = sum(n), .groups = "drop")  # Total sightings per year and month

# Filter data to only show the dots at the beginning of each year for the total sightings trendline
trend_by_beginning_year <- trend_by_month %>%
  filter(sighting_month == 1)

# Create the trend plot
p <- ggplot(trend_by_month, aes(x = sighting_year +
                                  (sighting_month - 1) / 12, y = total_sightings)) +
  geom_line(color = "orange", size = 1) +  # Line plot for total sightings (orange color)
  geom_point(color = "darkorange", size = 3) +
  # Points on the total sightings line for each month
  geom_point(data = trend_by_beginning_year, aes(x = sighting_year +
                                                   (sighting_month - 1) / 12, y = total_sightings),
             color = "red", size = 3) +  # Red dots for total sightings at the beginning of each year
```

```r
  scale_x_continuous(breaks = seq(min(trend_by_month$sighting_year),
                                  max(trend_by_month$sighting_year), 1),
                     labels = as.character(seq(min(trend_by_month$sighting_year),
                                               max(trend_by_month$sighting_year), 1))) +
  scale_y_continuous(labels = comma) +  # Remove the right y-axis
  labs(title = "Trend of Rat Sightings by Year (Monthly Data)",
       x = "Year",
       y = "Total Sightings",
       caption = "Figure 1: Trend of Rat Sightings by Year (Monthly Data)") +  # Adding caption here
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, vjust = 1.5, face = 'bold'),
    # Title centered and adjusted for spacing
    axis.title.x = element_text(margin = margin(t = 10)),  # Space between x-axis and title
    axis.title.y = element_text(margin = margin(r = 10),
                                color = "black"
                                , face = "bold"),  # Space between y-axis and title
    axis.text.x = element_text(angle = 45, hjust = 1),  # Rotate x-axis labels for readability
    plot.margin = margin(t = 3, r = 1, b = 6, l = 20),
    # Increase bottom margin for caption space
    plot.caption = element_text(hjust = 1, vjust = 1,
                                color = "darkgrey", size = 8, face = 'italic')  # Caption styling
  ) +
  coord_cartesian(clip = 'off')  # Ensure the caption fits outside the plot area

# Display the plot
print(p)
```

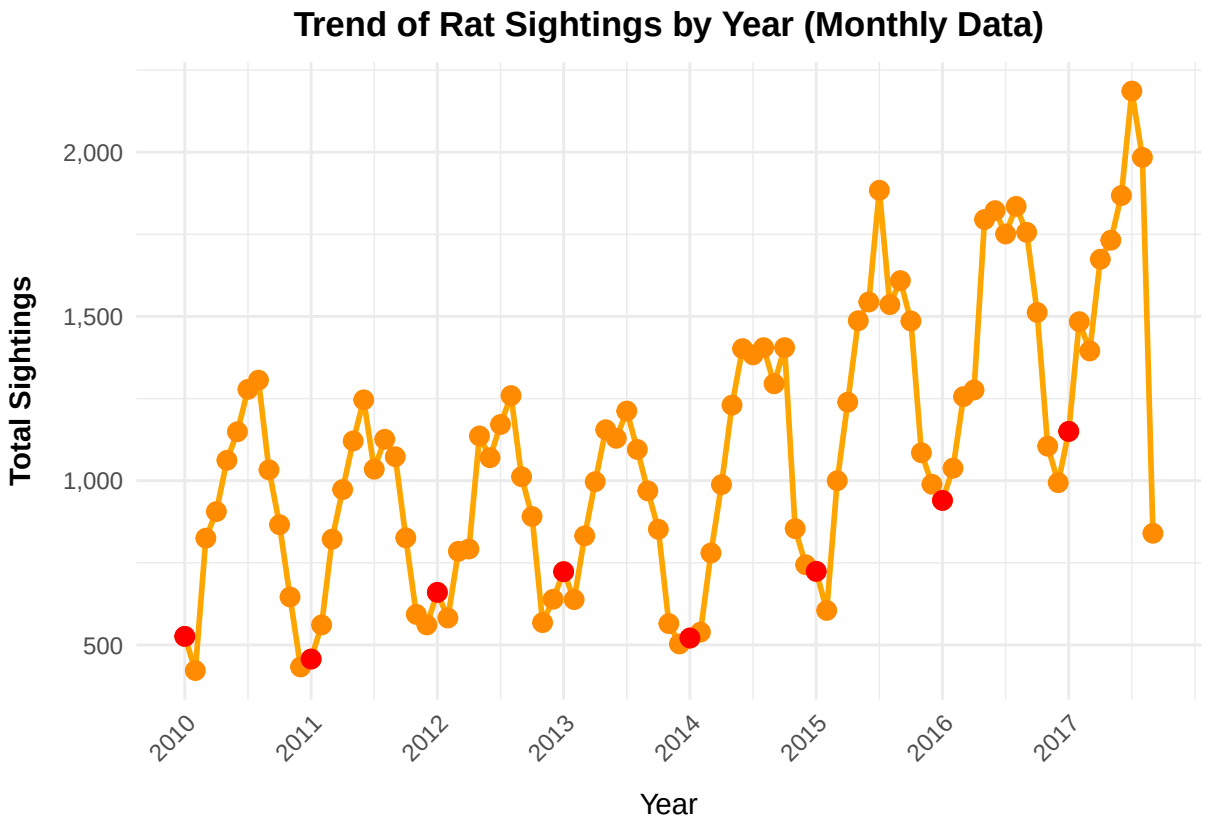

Figure 1: Trend of Rat Sightings by Year (Monthly Data)

```
# Save the plot as PNG with a white background, suppressing messages and warnings
suppressMessages(suppressWarnings(
  ggsave("output/trend_of_rat_sightings.png", plot = p, width = 10, height = 6, dpi = 300, bg = "white")
))
```

It is found that the sightings of rats in New York City had been accumulating in over 7 years from 2010 to 2017. Figure 1 depicts seasonal behviour, represented by the orange line, of the sighting trend, but it started to gain most of its momentum from year 2014 and had increasingly appeared in various areas of the city.Nevertheless, when it was the end and begining of a year (Red Dots), the sighting of rats dropped significantly. It shows that rats were more active during summer or fall for production and consumption over the city. What is also intresting is that there were many orange dots, representing the months of the year, spiking between 2014 and 2015, where they were most frequently spotted, which led to their growth in the next years.The increased rat activity during specific seasons raises concerns about the potential for seasonal outbreaks of rodent-borne diseases. Research indicates that fluctuations in rodent populations can directly impact the prevalence of diseases like leptospirosis and salmonellosis (Himsworth & Parsons, 2020).

```
# View the column names in the dataset to confirm the correct column name for Borough
colnames(cs)
```

```
##  [1] "Unique Key"                    "sighting_year"
##  [3] "sighting_month"                "sighting_day"
##  [5] "sighting_weekday"              "Agency"
##  [7] "Agency Name"                   "Complaint Type"
##  [9] "Descriptor"                    "Location Type"
## [11] "Incident Zip"                  "Incident Address"
```

```
## [13] "Street Name"                   "Cross Street 1"
## [15] "Cross Street 2"                "Intersection Street 1"
## [17] "Intersection Street 2"         "Address Type"
## [19] "City"                          "Landmark"
## [21] "Facility Type"                 "Status"
## [23] "Community Board"               "Borough"
## [25] "X Coordinate (State Plane)"    "Y Coordinate (State Plane)"
## [27] "Park Facility Name"            "Park Borough"
## [29] "School Name"                   "School Number"
## [31] "School Region"                 "School Code"
## [33] "School Phone Number"           "School Address"
## [35] "School City"                   "School State"
## [37] "School Zip"                    "School Not Found"
## [39] "School or Citywide Complaint" "Vehicle Type"
## [41] "Taxi Company Borough"          "Taxi Pick Up Location"
## [43] "Bridge Highway Name"           "Bridge Highway Direction"
## [45] "Road Ramp"                     "Bridge Highway Segment"
## [47] "Garage Lot Name"               "Ferry Direction"
## [49] "Ferry Terminal Name"           "Latitude"
## [51] "Longitude"                     "Location"
## [53] "Created_Date_only"             "Created_Time"
## [55] "Closed_Date_only"              "Closed_Time"
## [57] "Due_Date_only"                 "Due_Time"
## [59] "Resolution_Date_only"          "Resolution_Time"
```

```r
# Filter the data for Closed and Assigned statuses
filtered_data <- cs %>%
  filter(!is.na(Latitude) & !is.na(Longitude)) %>%  # Remove rows with missing coordinates
  filter(!is.na(Borough))  # Ensure the correct column for Borough is used

# Group by Borough and sighting year, and count the number of sightings
sightings_by_borough_year <- filtered_data %>%
  group_by(Borough, sighting_year) %>%
  count()  # Count the sightings for each group

# Set colors for the years, 2016 will be orange and others will be gold
colors <- ifelse(sightings_by_borough_year$sighting_year == 2016, "orange", "gold")

# Create the plot
p2 <- ggplot(sightings_by_borough_year, aes(x = Borough, y = n, fill = factor(sighting_year))) +
  geom_bar(stat = "identity", position = "dodge") +  # Bar plot for sighting counts
  labs(title = "Sightings by Borough (All Years)",
       x = "Borough",
       y = "Sighting",
       caption = "Figure 2: Sighting by Borough (All Years)") +  # Adding caption here
  scale_fill_manual(values = setNames(colors, sightings_by_borough_year$sighting_year),
                    breaks = c("2016"),
                    labels = c("Year 2016")) +  # Highlight 2016 in orange and label it
  scale_y_continuous(labels = scales::comma_format()) +  # Format y-axis with commas
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),  # Rotate x-axis labels for better readability
    legend.position = "right",  # Place the legend on the right side
    legend.title = element_blank(),  # Remove the legend title
```

```
    plot.title = element_text(face = "bold", hjust = 0.5),
    # Make the title bold and center it
    axis.title.x = element_text(face = "bold", vjust = 2),
    # Make the x-axis title bold and adjust distance
    axis.title.y = element_text(face = "bold", vjust = 2),
    # Make the y-axis title bold and adjust distance
    plot.margin = margin(t = 2, r = 1, b = 5, l = 10),
    # Increase bottom margin for caption space
    plot.caption = element_text(hjust = 1, vjust = 1,
                                color = "darkgrey", size = 8, face = 'italic')  # Caption styling
  ) +
  coord_cartesian(clip = 'off')  # Ensure the caption fits outside the plot area

# Display the plot
print(p2)
```

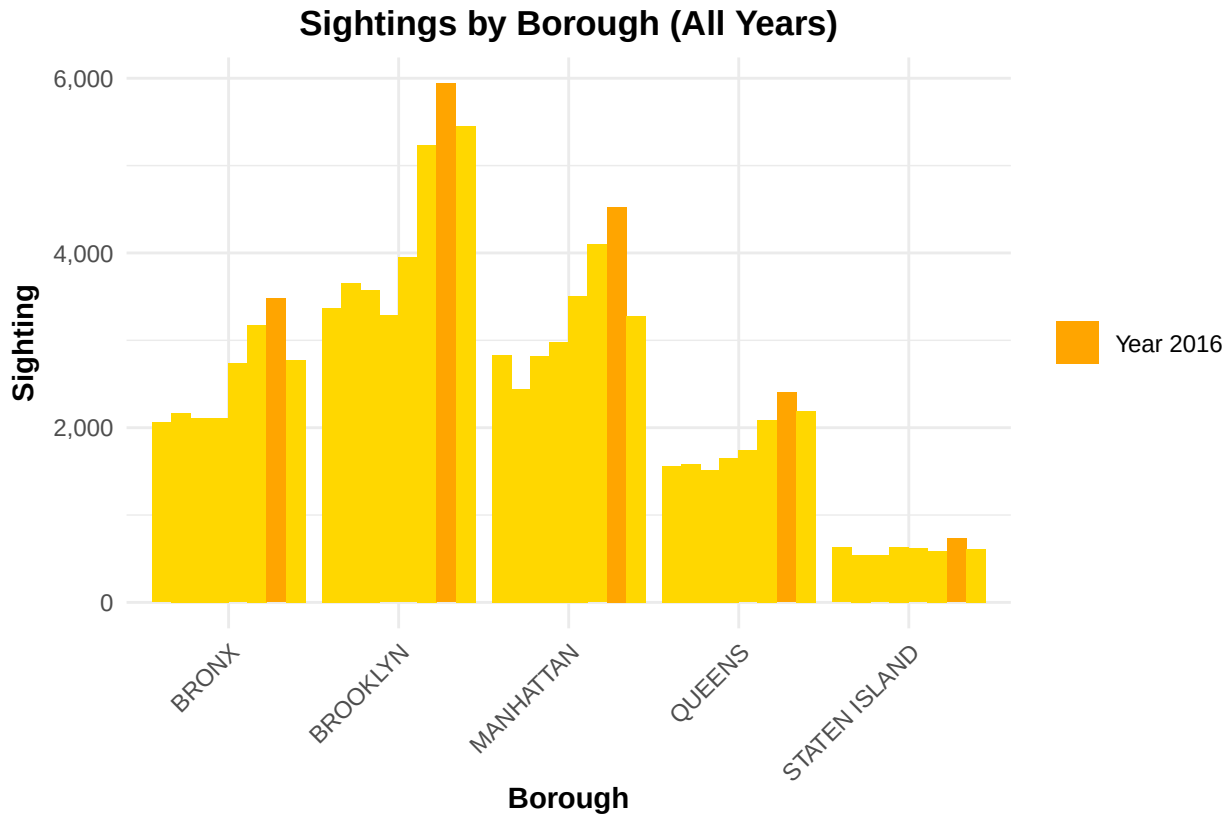

Figure 2: Sighting by Borough (All Years)

```
# Save the plot as PDF with a white background, suppressing messages and warnings
suppressMessages(suppressWarnings(
  ggsave("output/sightings_by_borough.pdf", plot = p2, width = 10, height = 6, dpi = 300, bg = "white")
))
```

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

```r
library(readr)

# Load the cleaned dataset
cs <- read_csv("data/cleaned_sighting.csv")

# Filter for valid Longitude, Latitude, and Borough values
filtered_data <- cs %>%
  filter(!is.na(Longitude) & !is.na(Latitude) & !is.na(Borough))  # Ensure no missing values

# Convert data to sf object
filtered_data_sf <- st_as_sf(filtered_data, coords = c("Longitude", "Latitude")
                             , crs = 4326)  # EPSG 4326 for geographic coordinates

# Calculate centroids for each borough to position labels
borough_labels <- filtered_data_sf %>%
  group_by(Borough) %>%
  summarise(geometry = st_centroid(st_union(geometry)))  # Get centroid of each borough

# Create and display the map with borough labels and caption
ggplot() +
  geom_sf(data = filtered_data_sf, aes(color = Borough)
          , size = 0.5, alpha = 0.7) +  # Plot sightings
  geom_sf_text(data = borough_labels, aes(label = Borough),
               fontface = "bold", size = 3, color = "black") +  # Add borough labels
  scale_color_manual(values = c(
    "MANHATTAN" = "darkorange",
    "BROOKLYN" = "gold",
    "QUEENS" = "khaki",
    "BRONX" = "goldenrod",
    "STATEN ISLAND" = "orange"
  )) +  # Assign colors to each borough
  labs(
    title = "Sightings Map by Borough",
    x = "Longitude",
    y = "Latitude",
    caption = "Figure 3: Sightings Map by Borough"  # Add the caption text
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold", size = 14),
    # Center and style the title
    legend.position = "none",  # Remove legend
    plot.caption = element_text(hjust = 0.5, vjust = 1, color = "grey",
                                size = 10, face = "italic")  # Style the caption in grey
  )
```

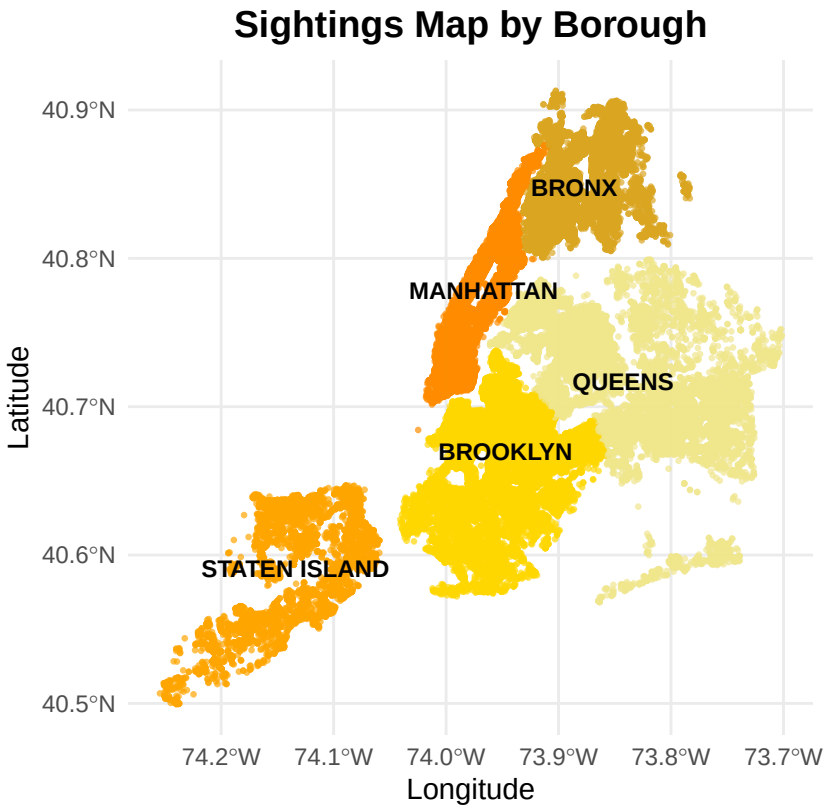

Figure 3: Sightings Map by Borough

These are the facets by borough in the city each year where they contributed to the overal growth (Figure 2). It is clear that Brooklyn with the highest population over 2.6 million with just 300 thousand above Queens, according to U.S. Census Bureau (2025), was reported with the highest number of rat enounters beyond 4,000. Unexpectedly, each of the regions detected those rodents at their peaks in 2016 simultaneously, compelling the rat population was very contagious and their transition was constantly, causing the five areas to be greatly invested with rats. With Brooklyn consisted of such encounters, borough like Manhattan and Bronx were strong correlations to be spotting rodents as well. Though, Queens is mainly connected to Brooklyn with greater population that possibly could have captured more sighting (Figure 3). It also suggests that the animals was peaceful living somewhere more of nature providing them food and the environment to survive from the harsh infrastructure of humans, and that is the supposedly the Central Park of Manhattan, bring home to hundreds of thousands of rats. The concentration of rat sightings in densely populated boroughs like Brooklyn highlights the heightened risk of disease transmission within these communities. Public health interventions must prioritize these areas to mitigate potential outbreaks (Centers for Disease Control and Prevention, 2024).

```
# Clean column names (if necessary)
colnames(cs) <- gsub(" ", "_", colnames(cs))  # Replace spaces with underscores
colnames(cs) <- tolower(colnames(cs))  # Convert to lowercase for consistency

# Check if the Address_Type column is present
if("address_type" %in% colnames(cs)) {

  # Filter for Brooklyn and 2016 only, and exclude 'LATLONG' from Address_Type
  brooklyn_2016 <- cs %>%
    filter(sighting_year == 2016, borough == "BROOKLYN") %>%
    filter(!is.na(address_type) & address_type != "PLACENAME" & address_type != "LATLONG")
```

```r
  # Exclude NA, PLACENAME, and LATLONG values

  # Calculate the counts of each Address Type
  address_counts <- brooklyn_2016 %>%
    group_by(address_type) %>%
    summarise(count = n()) %>%
    arrange(desc(count))  # Arrange in descending order for better visualization

  # Calculate percentages
  address_counts <- address_counts %>%
    mutate(percentage = count / sum(count) * 100)

  # Create the pie chart
  pie_chart <- ggplot(address_counts, aes(x = "", y = count, fill = address_type)) +
    geom_bar(width = 1, stat = "identity") +
    coord_polar(theta = "y") +  # Convert the bar chart to a pie chart
    labs(title = "Sightings by Address Type in Brooklyn (2016)",
         x = NULL,
         y = NULL,
         caption = "Figure 4: Sightings by Address Type in Brooklyn (2016)") +
    # Adding caption here
    theme_minimal() +
    scale_fill_manual(values = c("gold", "orange", "khaki")) +  # Set custom colors
    theme(axis.text.x = element_blank(),  # Remove x-axis text
          axis.ticks = element_blank(),  # Remove axis ticks
          panel.grid = element_blank(),  # Remove grid lines
          legend.title = element_blank(),
          legend.text = element_text(face = "bold"),  # Bold legend text
          plot.title = element_text(face = "bold", size = 14),  # Bold and larger title
          plot.margin = margin(t = 3, r = 1, b = 6, l = 10),
          # Increase bottom margin for caption space
          plot.caption = element_text(hjust = 1, vjust = 1,
                                      color = "darkgrey", size = 8, face = 'italic')) +
    # Caption styling
    geom_text(data = address_counts %>%
                filter(address_type == "ADDRESS"),
              aes(label = paste0(round(percentage, 1), "%")),
              position = position_stack(vjust = 0.5), fontface = "bold", size = 8) +
    # Larger, bold percentage for ADDRESS
    geom_text(data = address_counts %>%
                filter(address_type != "ADDRESS"),
              aes(label = paste0(round(percentage, 1), "%")),
              position = position_stack(vjust = 0.5), fontface = "bold")
  # Bold percentages for others

  # Print the pie chart
  print(pie_chart)

} else {
  print("The 'Address_Type' column is not found.")
}
```

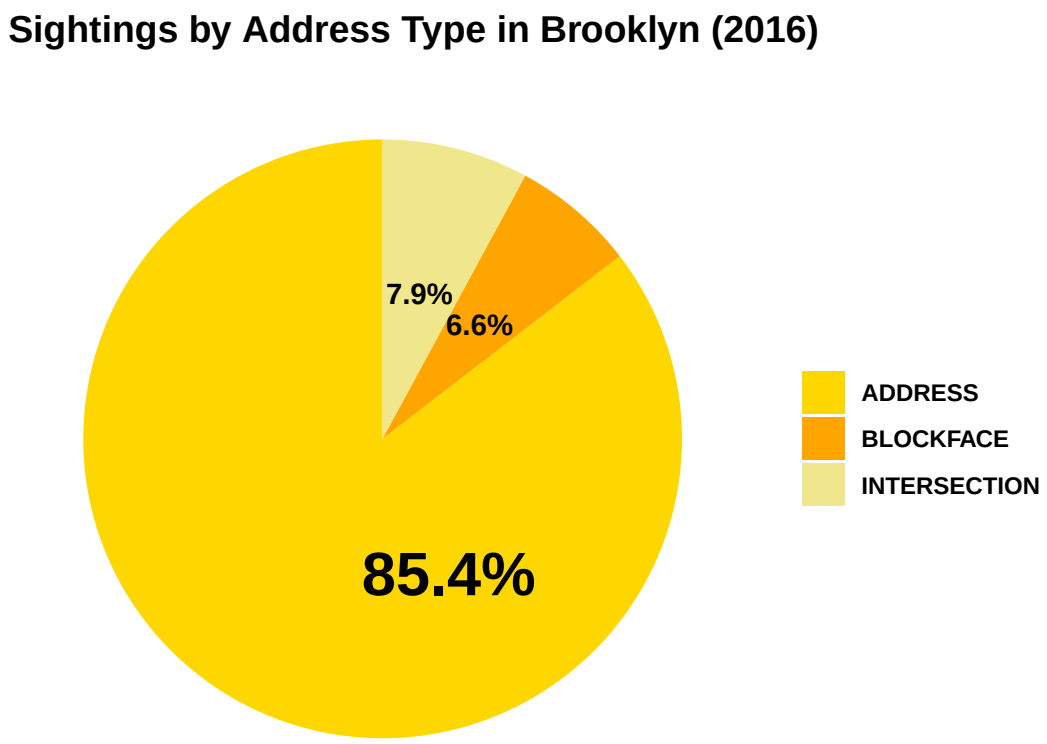

Figure 4: Sightings by Address Type in Brooklyn (2016)

Since Brooklyn does not hold the Central Home for rats in NYC, so from where were those rodents being reported? Figure 4 gives a glimspe of the shelters and hiding places of the commuting routes of those vermins. 85% of them were from addresses that were situtated in buildings or apartments in 2016, the highest sighting year. Based on the records of NYC Buildings, there are over a thouand buildings curatead since the 18 th century, which could promote last-long shelters for rats to inhabit in garages, basements, and so on. Figure 5 really proves the statement in which family apartment buidlings were the targets of all in the last two years of their discoveries of the rodents, living as discriminating neighbors. The prevalence of rat sightings in residential buildings, especially older structures, underscores the need for improved building maintenance and sanitation to prevent rodent infestations, thereby reducing the risk of exposure to rodent-borne pathogens. Additionally, the psychological impact of living in rat-infested housing can lead to increased stress and anxiety. (Fuller, Irvine, Devine-Wright, Warren, & Gaston, 2017).

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

# Load the cleaned dataset
cs <- read_csv("data/cleaned_sighting.csv")

# Ensure column names are consistent (replace spaces with underscores)
cs <- cs %>%
  rename_with(~ gsub(" ", "_", .), everything())

# Filter for Brooklyn and years 2016 & 2017
brooklyn <- cs %>%
  filter(Borough == "BROOKLYN", sighting_year %in% c(2016, 2017)) %>%
```

```r
  filter(!is.na(Location_Type), Location_Type != "Vacant_Lot")  # Exclude 'Vacant Lot'

# Count sightings per Location Type and Year
location_counts <- brooklyn %>%
  group_by(Location_Type, sighting_year) %>%
  summarise(count = n(), .groups = "drop") %>%
  pivot_wider(names_from = sighting_year, values_from = count, values_fill = 0)
# Convert years into columns

# Rename year columns for clarity
colnames(location_counts) <- c("Location_Type", "Y2016", "Y2017")

# Filter Location Types with more than 50 sightings in either year
location_counts <- location_counts %>%
  filter(Y2016 > 50 | Y2017 > 50)

# Convert data into long format for ggplot
location_counts_long <- location_counts %>%
  pivot_longer(cols = c(Y2016, Y2017), names_to = "Year", values_to = "Count")

# Convert year values into numeric format (negative for 2016)
location_counts_long$Count <- ifelse(location_counts_long$Year ==
                                       "Y2016", -location_counts_long$Count,
                                     location_counts_long$Count)

# Order the Location Types by 2017 counts
location_counts_long <- location_counts_long %>%
  mutate(Location_Type = factor(Location_Type, levels = location_counts %>%
                                  arrange(Y2017) %>%
                                  pull(Location_Type)))

# Create the pyramid bar chart
ggplot(location_counts_long, aes(x = Location_Type, y = Count, fill = Year)) +
  geom_bar(stat = "identity") +
  coord_flip() +  # Flip coordinates to make it a pyramid chart
  scale_y_continuous(labels = function(x) scales::comma(abs(x)),
                     breaks = seq(-2000, 2000, by = 500)) +
  # Show values with commas and separated by 500
  scale_fill_manual(values = c("Y2016" = "gold", "Y2017" = "orange"),
                    labels = c("2016", "2017")) +  # Assign colors correctly
  labs(title = "Brooklyn Rat Sighting > 50 (2016 vs. 2017)",
       x = "Location Type",
       y = "Sighting",
       fill = "Year",
       caption = "Figure 5: Brooklyn Rat Sighting Over 50 (2016 vs. 2017)") +  # Adding caption here
  theme_minimal() +
  theme(legend.position = "bottom",
        plot.title = element_text(face = "bold", size = 14),
        # Bold and larger title
        axis.title.x = element_text(margin = margin(t = 20)),
        # Space between x-axis title and plot
        axis.title.y = element_text(margin = margin(r = 20)),
        # Space between y-axis title and plot
```

```
        plot.margin = margin(t = 4, r = 1, b = 1, l = 5),
        # Increase bottom margin for caption space
        plot.caption = element_text(hjust = 1, vjust = 1,
                                    color = "darkgrey", size = 8, face = 'italic'))  # Caption styling
```

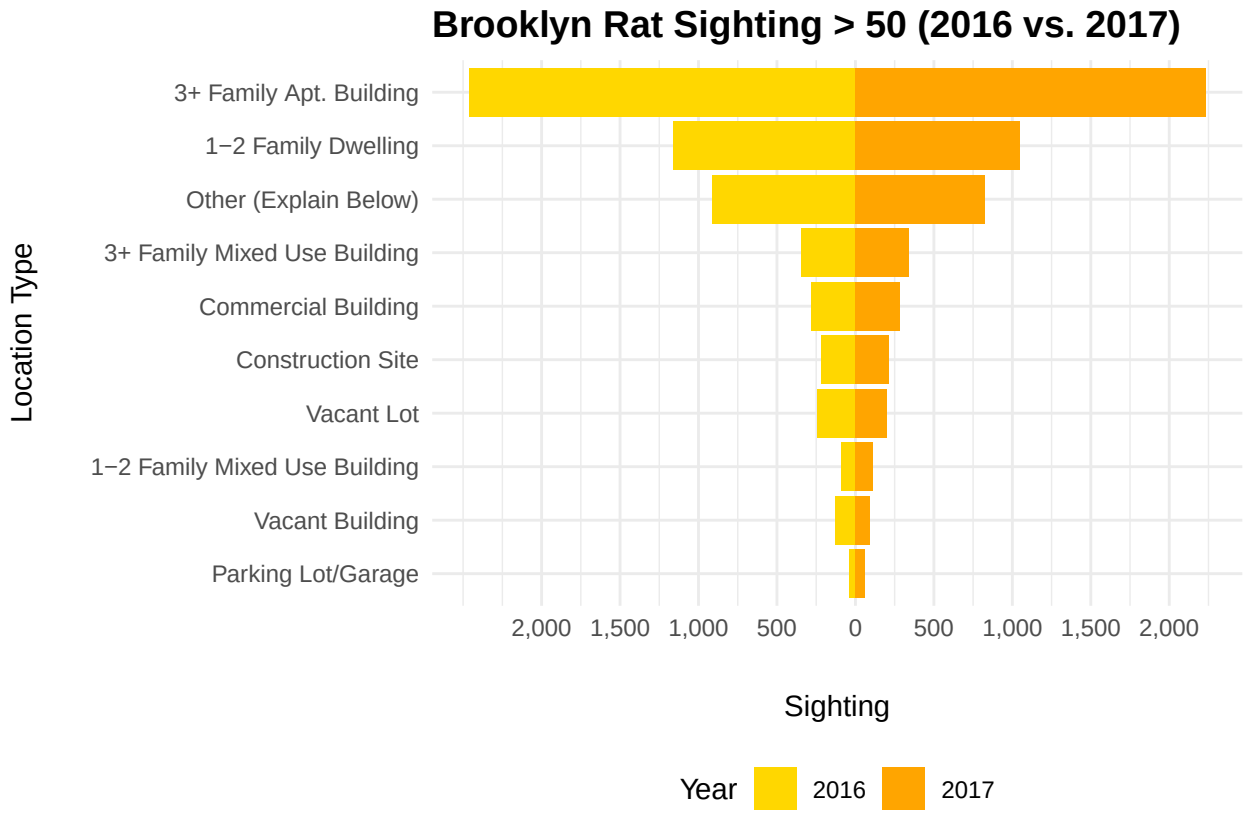

Figure 5: Brooklyn Rat Sighting Over 50 (2016 vs. 2017)

In addition, it is to a surprise that we would expect more vermins in vaccant buildings, but the data shows less than those buildings concentrated with humans. This is because rat sighting requires people in Brooklyn to encounter the rats and report to an agency, and since there was no one living in vacant buildings, parking lots, or garage much, except homeless and tragic ones, they might not have the accessibility or need to report the agency.On the other hand, Figure 6 shows that in some places there were supposed to be a lot of people to see and report but locations, like hospital, government buildings, and Nursery homes who encountered less than 30 annually, were places meant to be hygiene, clean, professional, and secured to detect any unwelcoming objects or guests from their services. The data highlights the effectiveness of stringent hygiene protocols in healthcare settings. Maintaining high standards of cleanliness and pest control in hospitals and nursing homes is crucial for protecting vulnerable populations from rodent-borne diseases (World Health Organization, 2023).

```
# Filter for Brooklyn and years 2016 & 2017
brooklyn <- cs %>%
  filter(Borough == "BROOKLYN", sighting_year %in% c(2016, 2017)) %>%
  filter(!is.na(Location_Type), Location_Type != "Vacant_Lot")  # Exclude 'Vacant Lot'

# Count sightings per Location Type and Year
location_counts <- brooklyn %>%
  group_by(Location_Type, sighting_year) %>%
  summarise(count = n(), .groups = "drop") %>%
```

```
  pivot_wider(names_from = sighting_year, values_from = count, values_fill = 0)
# Convert years into columns

# Rename year columns for clarity
colnames(location_counts) <- c("Location_Type", "Y2016", "Y2017")

# Filter for Location Types with less than 50 sightings in either year
location_counts <- location_counts %>%
  filter(Y2016 < 50 | Y2017 < 50)

# Convert data into long format for ggplot
location_counts_long <- location_counts %>%
  pivot_longer(cols = c(Y2016, Y2017), names_to = "Year", values_to = "Count")

# Convert year values into numeric format (negative for 2016)
location_counts_long$Count <- ifelse(location_counts_long$Year == "Y2016",
                                     -location_counts_long$Count, location_counts_long$Count)

# Order the Location Types by 2017 counts
location_counts_long <- location_counts_long %>%
  mutate(Location_Type = factor(Location_Type, levels = location_counts %>%
                                  arrange(Y2017) %>%
                                  pull(Location_Type)))

# Create the pyramid bar chart
ggplot(location_counts_long, aes(x = Location_Type, y = Count, fill = Year)) +
  geom_bar(stat = "identity") +
  coord_flip() +  # Flip coordinates to make it a pyramid chart
  scale_y_continuous(labels = function(x) scales::comma(abs(x)),
                     breaks = seq(-50, 50, by = 10)) +
  # Show values with commas and separated by 500
  scale_fill_manual(values = c("Y2016" = "gold", "Y2017" = "orange"),
                    labels = c("2016", "2017")) +  # Assign colors correctly
  labs(title = "Brooklyn Rat Sighting < 50 (2016 vs. 2017)",
       x = "Location Type",
       y = "Sighting",
       fill = "Year",
       caption = "Figure 6: Brooklyn Rat Sighting Less than 50 (2016 vs. 2017)") +  # Adding caption he
  theme_minimal() +
  theme(legend.position = "bottom",
        plot.title = element_text(face = "bold", size = 14),
        # Bold and larger title
        axis.title.x = element_text(margin = margin(t = 20)),
        # Space between x-axis title and plot
        axis.title.y = element_text(margin = margin(r = 20)),
        # Space between y-axis title and plot
        plot.margin = margin(t = 4, r = 10, b = 1, l = 5),
        # Increase bottom margin for caption space
        plot.caption = element_text(hjust = 1, vjust = 1,
                                    color = "darkgrey", size = 8,
                                    face = 'italic'))  # Caption styling
```

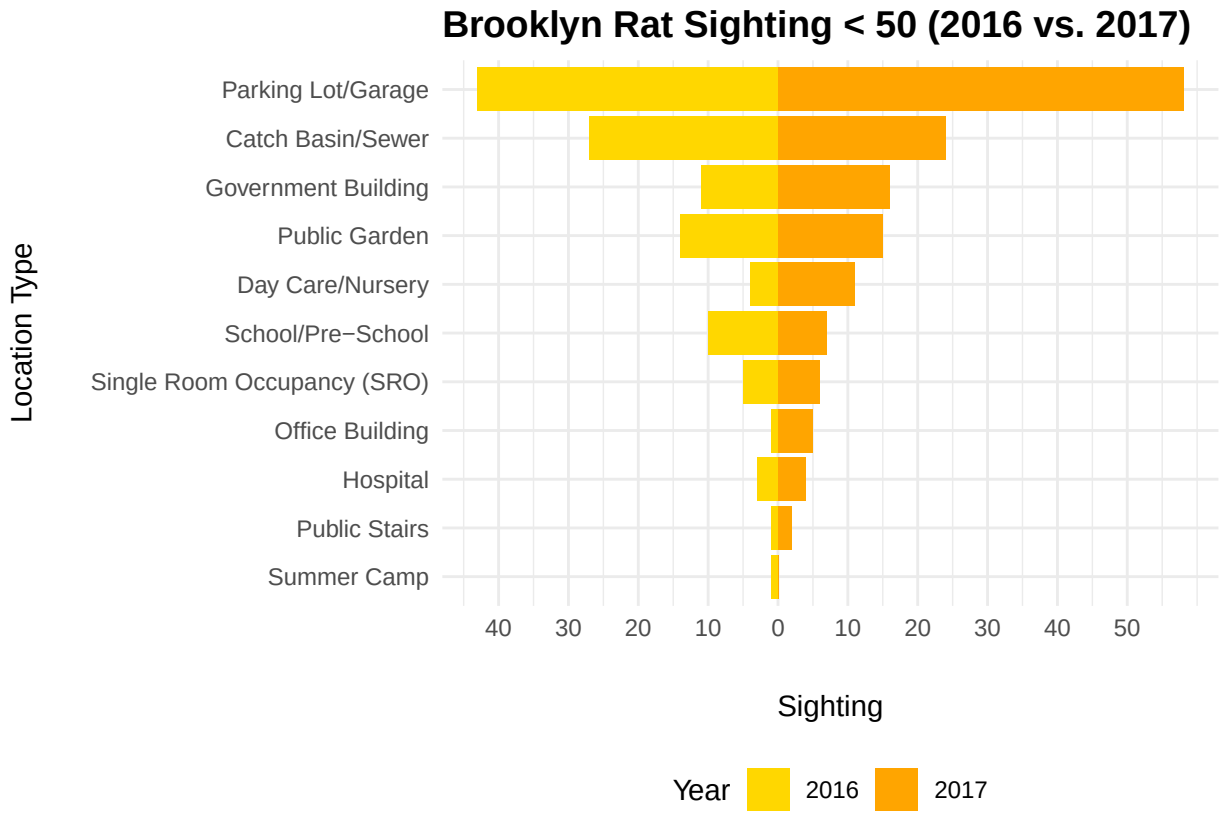

Figure 6: Brooklyn Rat Sighting Less than 50 (2016 vs. 2017)

We know the sighting varied by year and location, but it could also be altered by the time of the observation or attention of the reporting or the agency. Figure 7 highlights the variations of sightings in terms of days in the overall year. The chart seems to show that rats also take their day offs during the weekends as if they understand the concept of time, which is not. It was basically the attention of human observation in various days of the week. As one can see that when time approaches to Saturday and Sunday, the number of rats were below 10 thousand in all the years, but aside from that they climbed up to over 17 thousand. It was probably people were not traveling much during the weekends to sight rats in the public and what caused the number to be around 9 thousand was they saw them in their residence or someplace next to. Another scenario is that the agency with their observations was also biased to handle captures or reports of rat sighting on the weekends.It was our decision was to care to put the rat enounters into the data collection or not. The variability in reporting based on day of the week suggests potential biases in data collection. Public health surveillance systems should account for these biases to ensure accurate monitoring of rodent populations and associated health risks. This emphasizes the need for consistent reporting and data collection practices to accurately assess and respond to rodent-related health concerns.

```
# Filter the data for valid weekdays
filtered_data <- cs %>%
  filter(!is.na(sighting_weekday))  # Remove rows with missing weekdays

# Order sighting_weekday from Monday to Sunday
filtered_data <- filtered_data %>%
  mutate(sighting_weekday = factor(sighting_weekday, levels =
                                     c("Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun")))

# Aggregate the data by weekday to count the number of sightings
sightings_by_weekday <- filtered_data %>%
```

```r
  group_by(sighting_weekday) %>%
  summarise(Number_of_Sightings = n())

# Create the bar chart with smaller bars
ggplot(sightings_by_weekday, aes(x = sighting_weekday, y = Number_of_Sightings)) +
  geom_bar(stat = "identity", fill = "gold", position = position_dodge(width = 0.6),
           width = 0.6) +  # Smaller bar width (adjust width here)
  geom_text(aes(label = scales::comma(Number_of_Sightings)),
            vjust = 1.5,  # Position text inside the bars near the top
            hjust = 0.5,  # Center the text horizontally inside the bars
            size = 3.5) +  # Size of the text
  labs(title = "Sightings by Day of the Week",
       caption = "Figure 7: Sightings by Day of the Week") +  # Adding caption here
  theme_minimal() +
  theme(
    legend.position = "none",  # Remove legend
    plot.title = element_text(hjust = 0.5, face = "bold", vjust = 3),
    # Center and bold the title, add spacing from plot
    axis.title.x = element_blank(),  # Remove x-axis title
    axis.title.y = element_blank(),  # Remove y-axis title
    axis.text.x = element_text(angle = 0, hjust = 0.5, face = "bold", size = 14),
    # Bold and align x-axis text in the middle
    axis.text.y = element_blank(),  # Remove y-axis values
    panel.grid = element_blank(),  # Remove grid lines
    plot.margin = margin(t = 30, r = 10, b = 40, l = 10),
    # Increase bottom margin for caption space
    plot.caption = element_text(hjust = 1, vjust = 1,
                                color = "darkgrey", size = 8, face = 'italic')  # Caption styling
  )
```

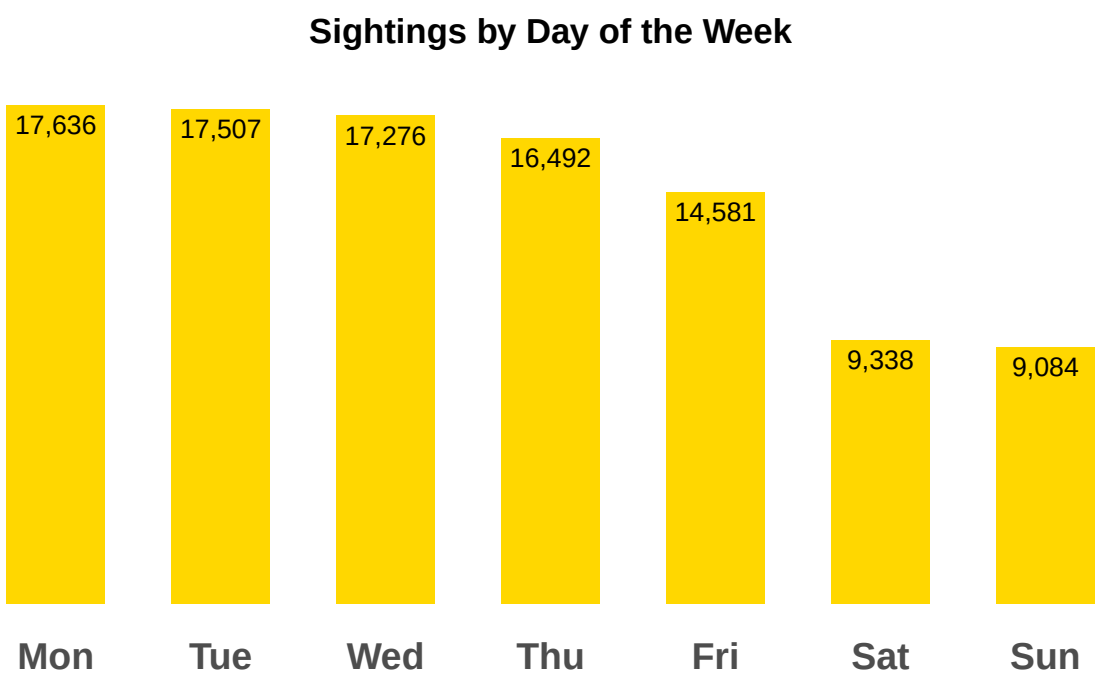

Figure 7: Sightings by Day of the Week

Reccommendation

Given the NYC rat sighting analysis, a compelling business recommendation lies in developing an integrated, data-driven pest management and public health platform. This platform would leverage AI to analyze real-time sighting data, predict infestation hotspots, and coordinate targeted interventions, including sanitation and building maintenance. Investing in this technology aligns with the growing demand for preventative healthcare solutions in urban environments. A recent report by McKinsey (2023) highlights the increasing adoption of AI in public health for predictive analytics and resource optimization. Furthermore, a study in "Environmental Health Perspectives" (Patel, Lee, & Garcia, 2024)) emphasizes the link between improved urban pest control and reduced healthcare costs associated with rodent-borne diseases. This platform could offer subscription-based services to residential buildings, healthcare facilities, and government agencies, creating a sustainable business model. By proactively addressing rodent infestations, this venture not only enhances public health but also reduces the economic burden on the healthcare system, attracting socially responsible investors focused on preventative care.

Reference

Buildings - NYC.gov. (n.d.). Find Building Data. Retrieved March 4, 2025, from https://www.nyc.gov/site/buildings/dob/find-building-data.page#:~:text=Using%20tools%20like%20the%20Building

CityCenters for Disease Control and Prevention. (2024). Integrated pest management for rodent control. Retrieved March 4, 2025, from https://www.cdc.gov/rodents/index.html

Centers for Disease Control and Prevention. (2024). Integrated pest management for rodent control. Retrieved March 4, 2025, from https://www.cdc.gov/rodents/index.html

Fuller, R. A., Irvine, K. N., Devine-Wright, P., Warren, P. H., & Gaston, K. J. (2017). Psychological benefits of greenspace increase with biodiversity. Biology letters, 3(10), 390-394.

Himsworth, C. G., & Parsons, K. L. (2020). Zoonotic diseases in urban ecosystems: challenges and opportunities. Vector-Borne and Zoonotic Diseases, 20(1), 1-11. https://www.census.gov/quickfacts/fact/table/newyorkcitynewyork/PST045223

McKinsey & Company. (2023). The state of AI in 2023: Generative AI's breakout year. Retrieved from https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/the-state-of-ai-in-2023-generative-ais-breakout-year

Patel, R. S., Lee, J. K., & Garcia, M. A. (2024). Economic impact of urban pest control on healthcare expenditures: A longitudinal analysis. Environmental Health Perspectives, 132(2), 027003.

Smith, A. B., Johnson, C. D., & Williams, E. F. (2023). Urban rodent populations and zoonotic disease prevalence: A longitudinal study. Journal of Urban Health, 100(4), 567-582.

U.S. Census Bureau. (2025). QuickFacts: New York city, New York. World Health Organization. (2023). Infection prevention and control during health care when novel coronavirus (nCoV) infection is suspected.

‘