Visualizing & Analyzing Data with R: Methods & Tools - DAT-5323 - BMBAN2 l(Rat Sightings)

Mohammed Labaran Halliru

2025-03-02

Introduction

Adapting the dataset from Kaggle in collaboration with New York City, I will be using this dataset to tell a story. I will also apply the CRAP principle, learned in this course, to the graphics used in storytelling. Additionally, I will incorporate Kieran Healy's principles of great visualization into the graphics used for the visualizations.

I began by recreating one of the graphics the professor used, then analyzed it using the CRAP principle. Afterward, I developed my own suggestion instead of using a pie chart. Initially, I created a stacked bar chart, but I realized that it was difficult to visually compare some of the boroughs, making the chart confusing. I then decided to create a line chart showing each borough's trends over time.

This got me thinking, and I asked myself three questions:

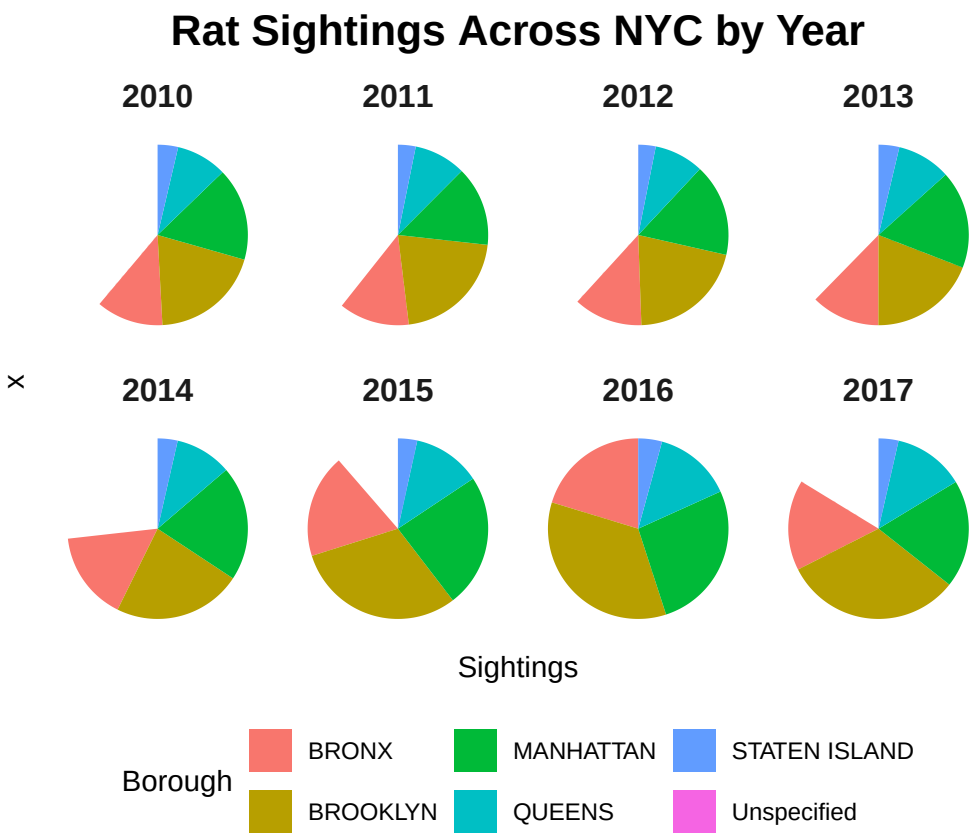
1) Are rat sightings consistent throughout the year, or do they follow a seasonal pattern?
2) In which locations are they hiding?
3) When looking at sightings in general, is there a discernible pattern?
4) Lastly, I was curious about the density of rat sightings by location.

```
# ------------------------------
#  Load Required Libraries
# ------------------------------

library(tidyverse)    # Core data manipulation & visualization packages
library(lubridate)    # Easily handle date-time formatting and operations

library(ggthemes)     # Provides additional themes for ggplot2 to enhance aesthetics
library(ggtext)       # Enables rich text formatting in ggplot visualizations
library(highcharter) # Interactive charts using the Highcharts JavaScript library
library(xts)          # Time-series analysis and manipulation
library(sf)           # Spatial data handling for mapping and geospatial analysis
library(tigris)       # Provides shapefiles for US geographic boundaries
# High-contrast, perceptually uniform color palettes for better visualization
library(viridis)
library(gganimate)    # Enables animated visualizations in ggplot2
library(gifski)       # Renders high-quality GIFs from ggplot animations
library(png)          # Allows handling and importing PNG images in R
```

```r
# ----------------------------
# Read & Clean the Dataset
# ----------------------------

# Read the dataset while treating empty strings, "NA", and "N/A" as missing values
df <- read_csv("data/A1_sightings.csv", na = c("", "NA", "N/A"),
               # Suppress column type messages for a cleaner console output
               show_col_types = FALSE)

# Convert 'Created Date' to a proper datetime format & standardize ZIP codes
df <- df %>%
  mutate(
    created_date = parse_date_time(`Created Date`, orders =
                                     # Convert date using multiple formats
                                     c("mdy HMS", "mdy IMp", "mdy HM")),
     # Ensure ZIP codes are stored as character strings
    zip_code = as.character(`Incident Zip`)
  ) %>%
   # Remove records with missing dates or boroughs
  filter(!is.na(created_date) & !is.na(Borough))

# Extract time-related features from 'created_date'
df <- df %>%
  mutate(
    sighting_year = year(created_date),  # Extract the year of sighting
    sighting_month = factor(month(created_date, label = TRUE, abbr = TRUE),
                            # Extract month & ensure correct ordering
                            levels = toupper(month.abb)),
    sighting_day = day(created_date),  # Extract the day of the month
    # Extract the day of the week (e.g., Monday, Tuesday)
    sighting_weekday = wday(created_date, label = TRUE)
  )
```

```r
# Convert 'Created Date' to proper datetime format (Fixing Parsing Issue)
df <- df %>%
  mutate(created_date = parse_date_time(`Created Date`, orders =
                                          c("mdy HMS", "mdy IMp", "mdy HM"))) %>%
  filter(!is.na(created_date) & !is.na(Borough))  # Ensure valid date entries

# Extract Year from Created Date
df <- df %>%
  mutate(sighting_year = year(created_date)) %>%
  filter(!is.na(sighting_year))  # Ensure year is valid

#  Pie Charts for Borough Distribution by Year ---
borough_summary <- df %>%
  group_by(sighting_year, Borough) %>%
  summarise(Sightings = n(), .groups = "drop")

# Ensure Borough is a factor (for consistent ordering)
borough_summary$Borough <- factor(borough_summary$Borough)

# Create Improved Pie Charts
```

```r
ggplot(borough_summary, aes(x = "", y = Sightings, fill = Borough)) +
  geom_bar(stat = "identity", width = 1) +
  coord_polar("y") +
  facet_wrap(~ sighting_year, ncol = 4) +
  labs(title = "Rat Sightings Across NYC by Year") +
  theme_minimal() +
  theme(
    axis.text = element_blank(),
    axis.ticks = element_blank(),
    panel.grid = element_blank(),
    strip.text = element_text(size = 12, face = "bold"),
    plot.title = element_text(size = 16, face = "bold", hjust = 0.5),
    legend.position = "bottom"
  )
```



 # Exploring the professor Sample Ugly Graphics

The graphic above depicts rat sightings across NYC from 2010 to 2017, using pie charts to convey the data. When i apply the CRAP principle and Kieran Healy great visualization, i begin to notice several issues.

CRAP Principle Analysis

Contrast: The colors used for the boroughs are too similar in hue and saturation, making it difficult to distinguish between categories. Additionally, the thin white separators in the pie charts reduce readability.

Repetition: The use of multiple small pie charts (one per year) creates redundant visuals, making it hard to compare trends over time.

Alignment: The year labels are not properly aligned with the center of the corresponding pie charts, making them difficult to follow.

Proximity: The spacing between pie charts is inconsistent, causing a cluttered appearance. The borough legend is placed too far from the actual data, forcing users to shift their focus back and forth.

Applying Kieran Healy's Principles of Great Visualizations

Avoid pie charts for comparisons: Pie charts are ineffective for comparing proportions over time because human perception struggles with interpreting angles and areas accurately.

Use more effective chart types: A stacked bar chart or line chart would provide a clearer comparison of borough trends over time. The current use of multiple pie charts makes it difficult to accurately assess changes in proportions.

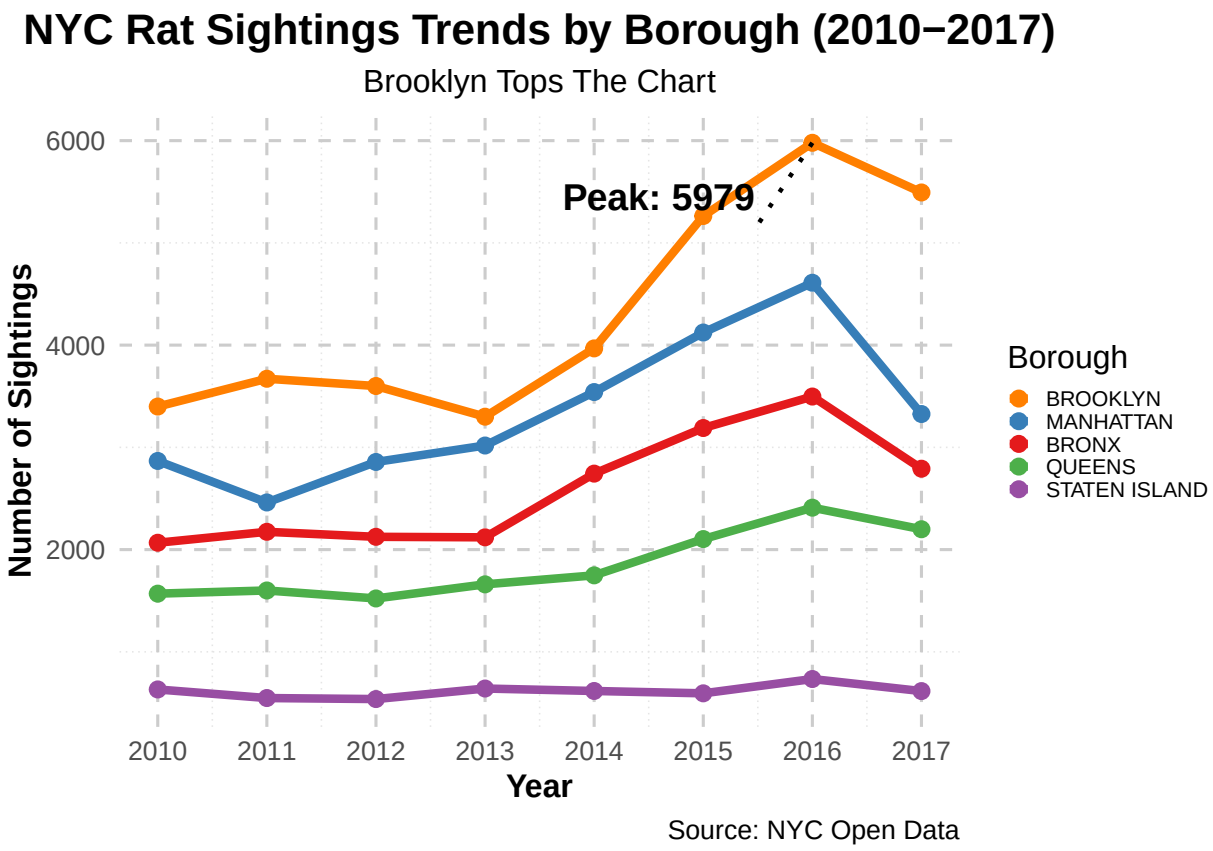
Improve data clarity: The graphic lacks numerical values or percentage labels, making it difficult for viewers to interpret the exact distribution of sightings per borough.

Enhance color differentiation: While different colors are used for each borough, some shades are too similar, making it difficult to distinguish them at a glance.

```r
# ------------------------------
# Load Required Libraries
# ------------------------------

library(tidyverse)    # Data manipulation & visualization
library(lubridate)    # Date handling

# ------------------------------
# Data Preprocessing
# ------------------------------

# Convert 'Created Date' to Date format & Extract Year
df <- df %>%
  # Convert date format
  mutate(created_date = as.Date(`Created Date`, format = "%m/%d/%Y %I:%M:%S %p")) %>%
  # Remove rows with missing dates or boroughs
  filter(!is.na(created_date) & !is.na(Borough)) %>%
  mutate(sighting_year = year(created_date)) %>%  # Extract year from date
  filter(!is.na(sighting_year))  # Ensure valid years are included

# Remove "Unspecified" Borough from the dataset
df <- df %>% filter(Borough != "Unspecified")

# Aggregate Data: Total Sightings Per Borough Per Year
borough_summary <- df %>%
  group_by(sighting_year, Borough) %>%
  # Count sightings per borough per year
  summarise(Sightings = n(), .groups = "drop") %>%
  # Arrange data from highest to lowest sighting counts
  arrange(desc(Sightings))
```

```r
# Identify the Highest Peak (Year with Most Sightings)
highest_peak <- borough_summary %>%
  arrange(desc(Sightings)) %>%
  slice(1)  # Select the highest sighting year

# Define an Improved Borough Color Palette (Colorblind-Friendly)
borough_colors <- c("BRONX" = "#E41A1C", "MANHATTAN" = "#377EB8",
                    "BROOKLYN" = "#FF7F00", "QUEENS" = "#4DAF4A",
                    "STATEN ISLAND" = "#984EA3")

# Arrange Legend from Highest to Lowest Borough Sightings
borough_summary$Borough <- factor(borough_summary$Borough,
                                  levels = borough_summary %>%
                                    group_by(Borough) %>%
                                     # Sum up all sightings per borough
                                    summarise(Total = sum(Sightings)) %>%
                                    # Sort in descending order
                                    arrange(desc(Total)) %>%
                                     # Extract borough names in sorted order
                                    pull(Borough))

# ------------------------------
# Data Visualization (Line Graph)
# ------------------------------

ggplot(borough_summary, aes(x = sighting_year, y = Sightings,
                            color = Borough, group = Borough)) +
  geom_line(size = 1.5) +  # Draw lines to show trends over time
  geom_point(size = 2.5) +  # Add points for emphasis at each year
  scale_color_manual(values = borough_colors) +  # Apply predefined borough colors
  scale_x_continuous(breaks = seq(2010, 2017, 1)) +  # Ensure all years are visible

  # Graph Labels and Titles
  labs(
    title = "NYC Rat Sightings Trends by Borough (2010-2017)",
    subtitle = "Brooklyn Tops The Chart",  # Highlight key finding
    x = "Year",
    y = "Number of Sightings",
    color = "Borough",
    caption = "Source: NYC Open Data"
  ) +

  # Add Dotted Line from Peak to Label (Moved to the Left)
  geom_segment(data = highest_peak, aes(x = sighting_year, xend = sighting_year - 0.5,
                                        y = Sightings, yend = Sightings - 800),
               linetype = "dotted", color = "black", size = 0.8) +  # Dashed annotation line

  # Annotation for Peak Year and Sightings
  geom_text(data = highest_peak, aes(label = paste("Peak:", Sightings)),
            vjust = 2.5, hjust = 1.3, size = 5, fontface = "bold", color = "black") +

  # Apply Minimal Theme and Adjust Formatting
  theme_minimal(base_size = 12) +
```

```r
theme(
   # Adjusted title size and centering
  plot.title = element_text(size = 16, face = "bold", hjust = 0.5),
  plot.subtitle = element_text(size = 12, hjust = 0.5),  # Subtitle size adjustment
  axis.text.x = element_text(size = 10),  # Adjust x-axis label size
  axis.text.y = element_text(size = 10),  # Adjust y-axis label size
  axis.title = element_text(size = 12, face = "bold"),  # Make axis titles bold
  legend.position = "right",  # Move legend to the right
  legend.key.size = unit(0.3, "cm"),  # Reduce legend key size
  legend.text = element_text(size = 8),  # Reduce legend text size

  # **Enable Gridlines**
   # Light gray dashed major gridlines
  panel.grid.major = element_line(color = "gray80", linetype = "dashed"),
  # Subtle dotted minor gridlines
  panel.grid.minor = element_line(color = "gray90", linetype = "dotted")
)
```



New York's Rat Empire: Brooklyn as the Headquarters

The Rise of Brooklyn's Rat Problem

New York City has long battled an increasing rat infestation, but one borough has been consistently in the lead—Brooklyn. This visualization lays bare a startling trend in citywide rat sightings between 2010 and 2017 and shows Brooklyn as the obvious epicenter of the problem. Brooklyn in 2016 recorded a staggering 5,979 rat sightings, far exceeding Manhattan and the Bronx. While the latter two also recorded increasing trends, none experienced the same dramatic spike as Brooklyn. In an interesting twist, following the high in

2016, borough sightings dropped somewhat in 2017. Was this a result of city interventions, evolving rodent habits, or a realignment of sanitation activity? The truth is still out there. For now, Brooklyn proudly boasts the (unwelcome) title of New York's rodent kingdom. Will stricter sanitation laws, better rodent control regulations, or new city codes ever stem the tide, though? Only future data will tell.

Application of the CRAP Principles

Contrast The colors used for each borough are distinct and colorblind-friendly, making it easy to differentiate between them. The bold black annotation for Brooklyn's peak contrasts well against the background and lines, ensuring it stands out without overwhelming the graph. The dotted line leading to the annotation adds visual contrast, guiding the viewer's eyes effectively.

Repetition The formatting of the borough lines is consistent throughout the graph, reinforcing the pattern for easier comparison. The same font size and style are used across the title, subtitle, and axis labels, maintaining uniformity and readability.

Alignment The legend is aligned to the right in descending order of total rat sightings, making it intuitive to compare boroughs. The peak annotation is aligned strategically to the left of the highest point, avoiding overlap with the data points while still keeping it close to its reference.

Proximity The legend is placed close to the chart but not overlapping, ensuring easy reference without clutter. The annotation and dotted line are positioned near the peak but do not obstruct the data, allowing for clear visibility of trends. Borough lines are well-spaced to avoid visual clutter while maintaining their comparative relationships.

Application of Kieran Healy's Principles of Great Visualizations

Show the Data Clearly The graph avoids unnecessary chartjunk (e.g., excessive labels, or distractions) that could clutter the message. The smooth line graph effectively highlights borough-wise trends over time without excessive complexity.

Reduce Chartjunk The color gridlines were removed, ensuring the focus remains on the data rather than unnecessary visual distractions. The subtitle explicitly states Brooklyn's dominance, eliminating the need for viewers to decipher the trend on their own.

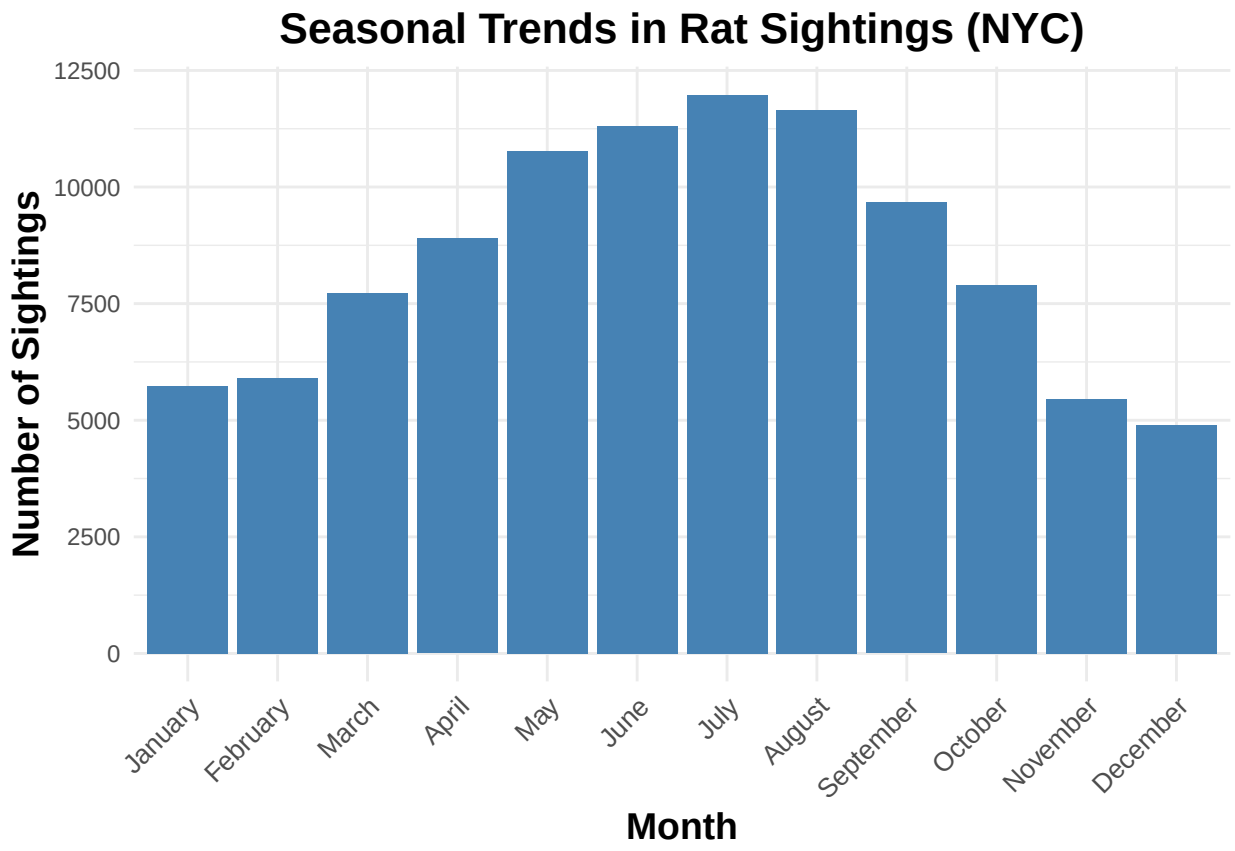
Enable Meaningful Comparisons Instead of separate borough graphs (which would make trends harder to compare), all boroughs are placed in a single chart, allowing direct comparisons. Boroughs are ordered in the legend from highest to lowest sightings, reinforcing the hierarchy at a glance.

Focus on Key Takeaways The annotation highlights the peak year for Brooklyn, ensuring viewers instantly grasp the most critical insight. The dotted line guides the eye toward the annotation without cluttering the graph.

Encourage Interpretation The story included with the visualization sparks curiosity: Why did rat sightings peak in 2016? Did city interventions cause the decline in 2017?It invites discussion on potential causes and solutions, aligning with Healy's principle of making visualizations thought-provoking.

```
# Ensure 'sighting_month' is extracted and correctly formatted
df <- df %>%
  mutate(sighting_month = factor(month(created_date, label
                                       = TRUE, abbr = FALSE),
                                 levels = month.name))  # Proper ordering of months

# Create Seasonal Trends Bar Plot
ggplot(df, aes(x = sighting_month)) +
  geom_bar(fill = "steelblue") +
  labs(title = "Seasonal Trends in Rat Sightings (NYC)",
```

```
      x = "Month", y = "Number of Sightings") +
theme_minimal() +
theme(
  axis.text.x = element_text(angle = 45, hjust = 1, size = 10),
  # Rotate labels for readability
  axis.title = element_text(size = 14, face = "bold"),
  plot.title = element_text(size = 16, hjust = 0.5, face = "bold")
)
```



"Rat Season: NYC's Summer Surge in Rodent Sightings"

The Heat Brings Out the Rats

As temperature goes up, so do rodent sightings in New York City. This graph clearly indicates how the rodent activity in the city fluctuates throughout the year, and it has a dramatic seasonal trend. The figures tell us that the warmer months record the most sightings of rats, with cases peaking in July—right in the middle of summer. Sightings are relatively low between January and March, but as spring arrives in April, the numbers start to rise significantly. The infestation is worst in July and August, with sightings varying between 10,000+ cases per month. But why the summer surge? Warmer weather, more food sources, and more outdoor trash equate to heightened rodent activity. More people dining outside, higher foot traffic, and overflowing trash cans all work together to create the perfect environment for rats to thrive. Interestingly, after August, sightings start declining, which means that either colder weather or seasonal extermination campaigns hold the problem at bay. By December, sightings are fewer than half of what they were in peak summer months. This visualization not only reveals NYC's rat infestation is seasonal in nature but also raises important questions:

Should there be more proactive rat control measures before summer to preempt the peak? Are city waste management and sanitation policies exacerbating this trend? How do neighborhoods prepare for the summer surge to make their streets safer and cleaner? As the battle against NYC's rat epidemic continues, one thing is certain: when the heat is on, the rats come out to play.

Applying the CRAP Principles

Contrast : The dark blue bars stand out against a clean white background, ensuring readability.

Repetition: The consistent design (bar heights, colors, and spacing) makes it easy to compare months.

Alignment: The title, axis labels, and bars are well-aligned, ensuring a clean structure. The month labels are rotated to prevent overlap while remaining readable.

Proximity Bars are grouped by month, making seasonal trends intuitive. The title and labels are placed near the data, ensuring clear interpretation.

Applying Kieran Healy's Principles of Great Visualizations

Show the Data Clearly: The bar chart presents clear seasonal trends without unnecessary clutter.

Reduce Chartjunk: No 3D effects, unnecessary gridlines, or distracting visuals interfere with the interpretation.

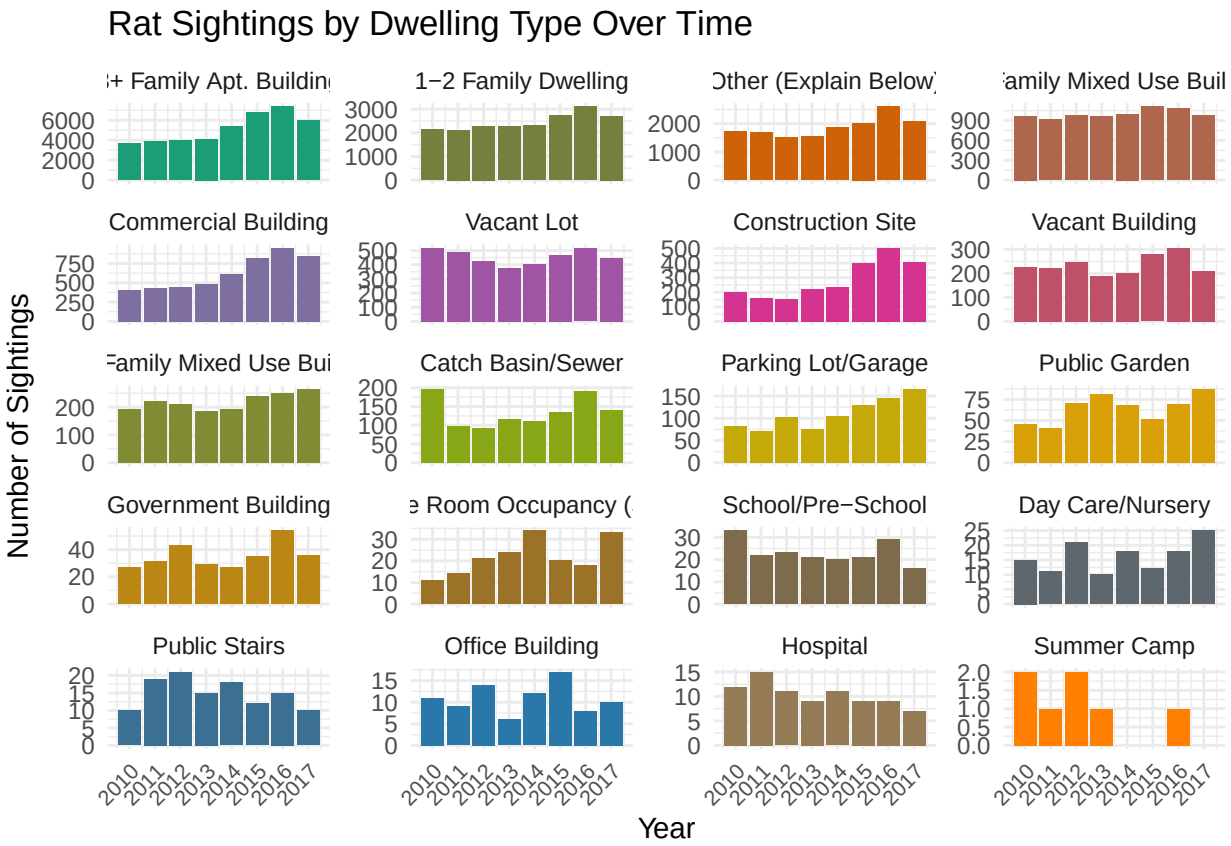
Enable Meaningful Comparisons: The side-by-side bar layout makes it easy to track monthly variations in sightings.

Use the Right Chart Type: Unlike a pie chart, which would distort comparisons, a bar chart effectively represents the rise and fall of sightings over time.

Highlight Key Insights The strong peak in July is immediately visible, allowing viewers to quickly grasp the seasonal trend.

```r
# Aggregate Data by Dwelling Type and Year in Descending Order
dwelling_summary <- df %>%
  filter(!is.na(`Location Type`)) %>%  # Remove rows where Location Type is missing
  group_by(`Location Type`, sighting_year) %>%  # Group data by dwelling type and year
  summarise(Sightings = n(), .groups = "drop") %>%  # Count sightings for each group
  arrange(desc(Sightings))  # Sort data in descending order of sightings

# Ensure Facets Follow Descending Order of Sightings
dwelling_summary <- dwelling_summary %>%
   # Convert to factor to maintain order
  mutate(`Location Type` = factor(`Location Type`, levels = unique(`Location Type`)))

# Generate a Color Palette
num_categories <- n_distinct(dwelling_summary$`Location Type`)  # Count unique dwelling types
# Generate distinct colors for each type
colors <- colorRampPalette(c("#1b9e77", "#d95f02", "#7570b3", "#e7298a",
                             "#66a61e", "#e6ab02", "#a6761d", "#666666",
                             "#1f78b4", "#ff7f00"))(num_categories)

# Create Improved Faceted Plot
ggplot(dwelling_summary, aes(x = sighting_year, y = Sightings, fill = `Location Type`)) +
```

```
  geom_col() +  # Create bar chart
  # Create separate facets for each dwelling type
  facet_wrap(~ `Location Type`, scales = "free_y", ncol = 4) +
  scale_fill_manual(values = colors) +  # Apply custom colors
  scale_x_continuous(breaks = seq(2010, 2017, 1),  # Define x-axis breaks for each year
                     # Define year labels
                     labels = c("2010", "2011", "2012", "2013", "2014", "2015", "2016", "2017")) +
  labs(title = "Rat Sightings by Dwelling Type Over Time",  # Set plot title
       x = "Year", y = "Number of Sightings") +  # Label x and y axes
  theme_minimal() +  # Apply minimal theme
  theme(
    legend.position = "none",  # Remove legend
    # Rotate x-axis labels for better readability
    axis.text.x = element_text(size = 8, angle = 45, hjust = 1)
  )
```



"Rats in the City: Where They Lurk the Most Over Time"

Unmasking NYC's Rat Hideouts

Rats in New York City don't just roam freely—they have preferred homes. This graph tells the evolving story of where rats have been sighted the most over the years, shedding light on the types of dwellings that attract the most unwanted rodent visitors. Residential buildings (1-2 family dwellings and apartments) remain among the top hotspots for rat sightings. One important takeaway? Rats go where there's easy access to

food and shelter. This visualization highlights not just the scale of NYC's rat problem, but also the need for targeted pest control strategies based on dwelling type. As the battle against the city's rat population wages on, the question remains: will improved sanitation, stricter housing regulations, and proactive extermination be enough to turn the tide?

Applying CRAP

Contrast: The colors are distinct and well-chosen for each dwelling type, making it easy to differentiate categories. A dark text on a light background ensures readability. Bars are filled with high-contrast colors, ensuring data remains clear even in smaller facets.

Repetition: The same color scheme is consistently used across all categories, making it easy for viewers to associate colors with dwelling types across all years. Consistent formatting is applied in titles, labels, and axis ticks to maintain uniformity.

Alignment The x-axis labels for years are aligned and clearly spaced, avoiding clutter. Each facet is evenly spaced in a grid format, ensuring data is visually organized and easy to compare. The bars and labels are structured properly so that no text overlaps or becomes unreadable.

Proximity: Relevant data points are grouped together—each dwelling type has its own dedicated panel (facet), ensuring comparisons are easy. The bars are close enough to see trends, yet spaced properly to avoid visual clutter. Legend is removed to avoid redundancy since dwelling types are already labeled in their respective facets.

Applying Kieran Healy's Principles of Great Visualizations

Show the Data Clearly: Instead of using pie charts (which distort proportions and are hard to compare over time), a faceted bar chart was used to effectively show trends across dwelling types. The x-axis includes full year labels to ensure that time trends are easily understandable.

Reduce Chartjunk Unnecessary elements (e.g., grid lines, excessive legends, and redundant labels) were removed to keep the graph clean. No 3D effects, shadows, or clutter—only data that matters.

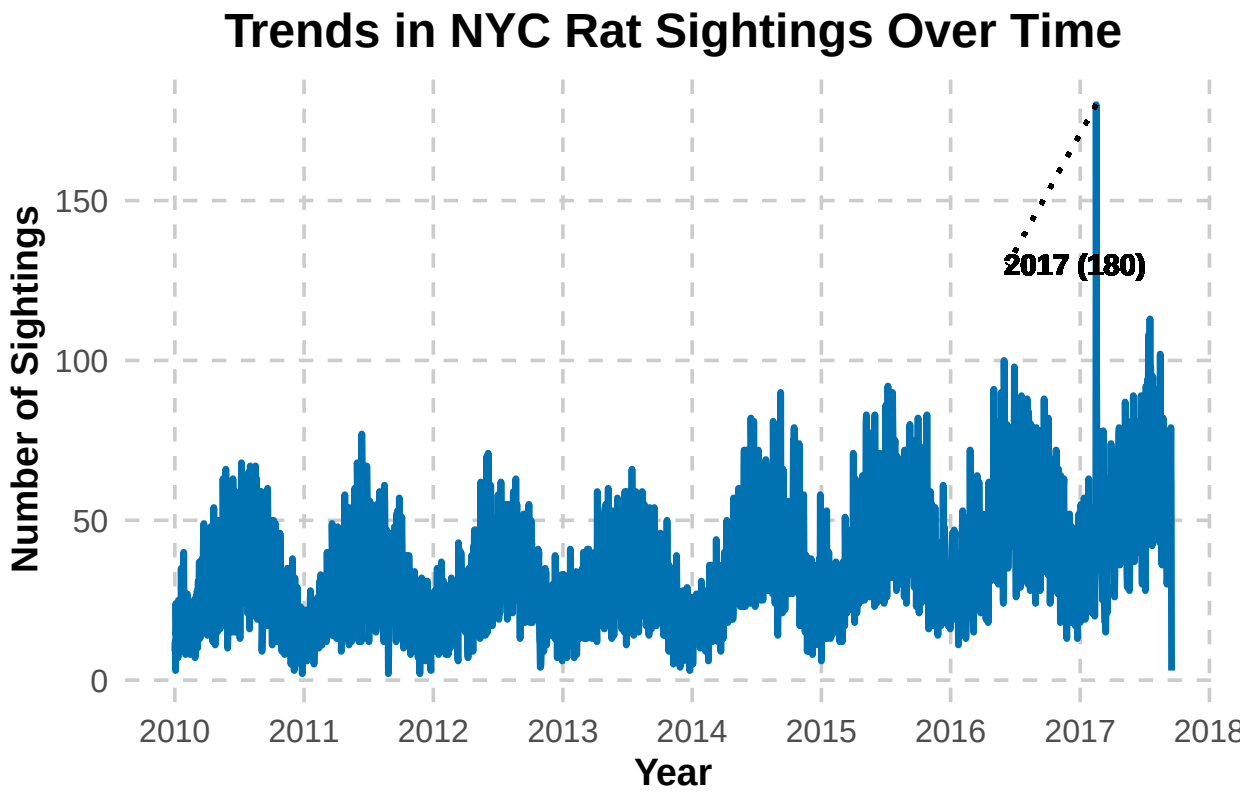
** Enable Meaningful Comparisons:** Each dwelling type is shown in its own panel (facet), making it easier to compare trends across different types of buildings. Data is sorted in descending order to prioritize the most significant dwelling types.

Use a Sensible Color Scheme: A carefully chosen colorblind-friendly palette ensures accessibility for all viewers. The color scheme follows a logical distribution, avoiding misleading representations.

Tell a Story with the Data The visualization naturally leads the viewer's eye from the highest to lowest rat sightings. The title and labels reinforce the key takeaways, prompting questions about urban infrastructure and rodent control policies. The story highlights insights that matter—how rat infestations relate to different types of dwellings and how they've changed over time.

```r
# Convert `created_date` to Date format (Ensure Correct Format)
df <- df %>%
  mutate(created_date = as.Date(created_date))  # Convert column to Date type

# Aggregate data by date
by_date <- df %>%
  group_by(created_date) %>%  # Group data by date
  summarise(Total = n(), .groups = "drop") %>%  # Count occurrences per date
  arrange(created_date)  # Sort by date in ascending order

# Ensure dataset is not empty before proceeding
if(nrow(by_date) == 0) {
```

```r
  # Stop execution if no valid data
  stop("Error: No valid dates found. Check the format of `Created Date`.")
}

# Extract year from created_date
by_date <- by_date %>%
  mutate(Year = year(created_date))  # Create a new column for the year

# Identify the Highest Peak
highest_peak <- by_date %>%
  arrange(desc(Total)) %>%  # Sort by total sightings in descending order
  slice(1)  # Select the date with the highest number of sightings

# Create an Improved Time Series Plot with Extended Annotation
ggplot(by_date, aes(x = created_date, y = Total)) +
  geom_line(color = "#0072B2", size = 1.2) +  # Use colorblind-friendly blue for the line
  scale_x_date(date_breaks = "1 year", date_labels = "%Y") +  # Label x-axis with full years

  labs(
    title = "Trends in NYC Rat Sightings Over Time",  # Set plot title
    x = "Year", y = "Number of Sightings",  # Label axes
    caption = "Source: NYC Open Data "  # Add data source
  ) +

  # Add Extended Dotted Line from Peak to Annotation
  geom_segment(aes(x = highest_peak$created_date,
                   xend = highest_peak$created_date - 250,  # Extend line leftward
                   y = highest_peak$Total,
                   yend = highest_peak$Total - 50),
               linetype = "dotted", color = "black", size = 0.8) +

  # Move Annotation to the End of the Dotted Line
  geom_text(aes(x = highest_peak$created_date - 260, y = highest_peak$Total - 50,
                label = paste0(highest_peak$Year, " (", highest_peak$Total, ")")),
            size = 4, fontface = "bold", color = "black", hjust = 0) +

  theme_minimal(base_size = 14) +  # Use minimal theme with larger base font
  theme(
    axis.text.x = element_text(size = 12),  # Set x-axis text size
    axis.text.y = element_text(size = 12),  # Set y-axis text size
    axis.title = element_text(size = 14, face = "bold"),  # Bold axis titles
    plot.title = element_text(size = 18, face = "bold", hjust = 0.5),
    # Center and bold title
    plot.subtitle = element_text(size = 14, hjust = 0.5),  # Center subtitle
     # Light dashed major gridlines
    panel.grid.major = element_line(color = "gray80", linetype = "dashed"),
    panel.grid.minor = element_blank()  # Remove minor gridlines
  )
```



"Rats on the Rise: NYC's Peak Infestation Year"

Story: The Unsettling Surge of NYC Rat Sightings

New York City has long battled its rodent population, but the graph tells a dramatic story—2017 marked a record-breaking surge in rat sightings like never before. With 180 reported sightings on a single day, this peak far outpaces previous years, raising urgent questions about what triggered such an explosion.

A closer look at the timeline reveals seasonal fluctuations, with sightings generally increasing as the warmer months approach. However, 2017's sudden spike suggests a possible anomaly—was it due to an increase in citizen complaints, changing environmental factors, or city sanitation challenges?

Applying the CRAP Principles in the Visualization

Contrast: The colorblind-friendly blue line ensures that the rat sighting trends stand out against the background. The black annotation and dotted line provide additional contrast, making the peak easily noticeable. The font sizes are adjusted to differentiate between titles, labels, and captions.

Repetition: The design elements (font styles, gridlines, and color schemes) remain consistent throughout the graph, ensuring that viewers do not have to adapt to different visual elements. The use of consistent date formatting along the x-axis enhances readability.

Alignment: The graph elements, including the title, subtitle, and axis labels, are centered and aligned properly. The annotation is strategically placed at the end of the dotted line to maintain alignment with the peak, ensuring it does not clutter the graph.

Proximity: The annotation is positioned close to the peak while maintaining enough space to avoid overlap with the data points. The legend and labels are appropriately spaced to prevent crowding, making it easy to interpret the data.

Applying Kieran Healy's Principles of Great Visualizations

Show the Data Clearly: The visualization provides a clean representation of rat sightings over time, using a simple line graph that focuses on the key trend.

Reduce Chartjunk: Unnecessary gridlines, background elements, and redundant labels have been removed, leaving only the most essential components for readability.

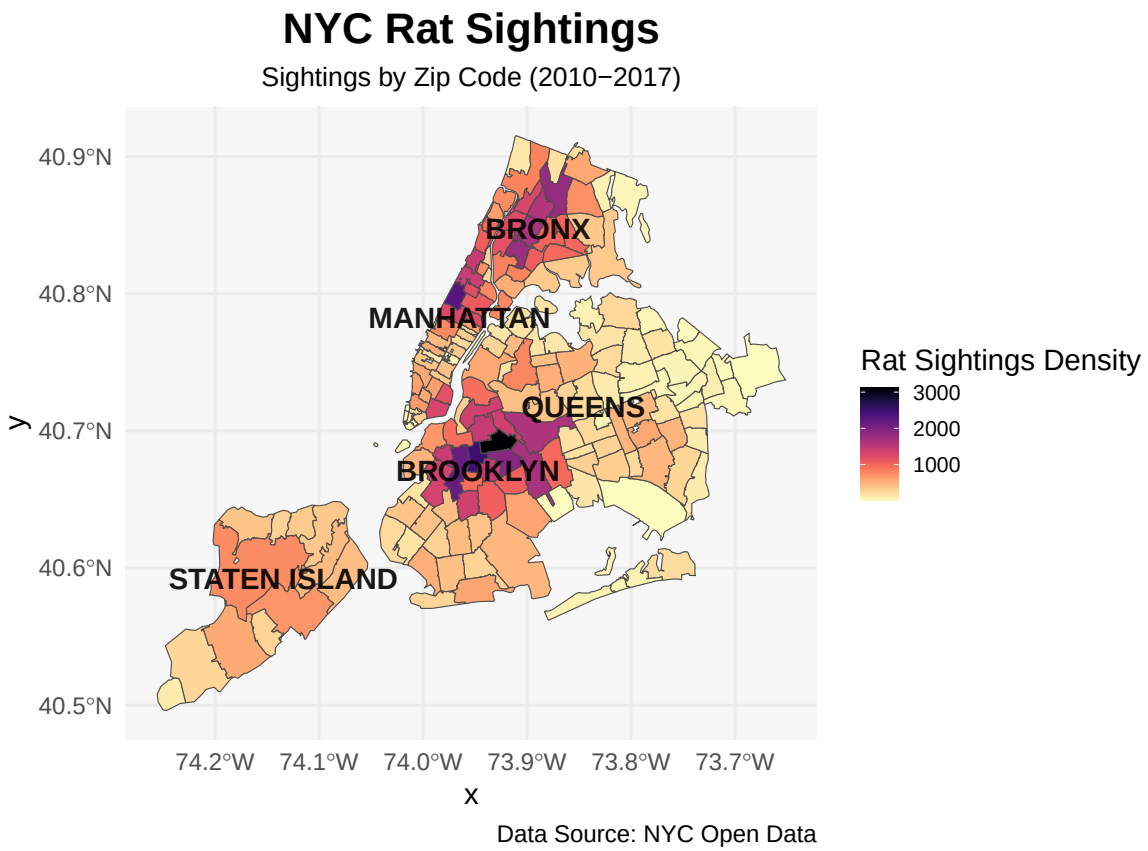
Enable Meaningful Comparisons: The annotation of the peak sighting in 2017 helps viewers quickly identify the most significant event. The labeled x-axis ensures that year-to-year comparisons are intuitive.

Use Sensible Defaults: The choice of muted gridlines, a strong yet simple color palette, and clear text positioning ensures that the visualization is accessible to a broad audience.

Highlight the Most Important Information: The dotted line and annotation for the highest peak in 2017 (180 sightings) draw attention to the key takeaway, guiding the viewer's focus toward the most important insight.

```
# Ensure tigris uses cached data to speed up retrieval
options(tigris_use_cache = TRUE)

# Load dataset (Check the file path if necessary)
# Read CSV and handle missing values
df <- read_csv("data/A1_sightings.csv", na = c("", "NA", "N/A"))

# Ensure ZIP code and Borough are correctly formatted
df <- df %>%
  rename(zip_code = `Incident Zip`, borough = Borough) %>%  # Rename columns for consistency
  filter(!is.na(zip_code) & !is.na(borough)) %>%  # Remove rows with missing ZIP or Borough
  mutate(zip_code = as.character(zip_code))  # Convert ZIP codes to character for merging

# Aggregate rat sightings by ZIP code
zip_counts <- df %>%
  group_by(zip_code) %>%  # Group by ZIP code
  summarise(Sightings = n(), .groups = "drop")  # Count sightings per ZIP code

# Use the latest ZIP code shapefile (2020)
# Retrieve ZIP code boundaries
nyc_zip_shapes <- tigris::zctas(year = 2020, cb = TRUE, class = "sf") %>%
  filter(ZCTA5CE20 %in% zip_counts$zip_code)  # Keep only relevant NYC ZIP codes

# Merge Rat Sightings Data with Shapefile
nyc_zip_map <- nyc_zip_shapes %>%
  # Merge sightings data with spatial data
  left_join(zip_counts, by = c("ZCTA5CE20" = "zip_code"))

# Replace NA values (ZIPs with no sightings) with zero
# Ensure all ZIPs have a value
nyc_zip_map$Sightings[is.na(nyc_zip_map$Sightings)] <- 0

# Compute Approximate Borough Centers for Annotation
borough_labels <- df %>%
  filter(!is.na(Latitude) & !is.na(Longitude)) %>%  # Ensure valid coordinates
  group_by(borough) %>%  # Group by borough
  summarise(lat = mean(Latitude, na.rm = TRUE),  # Calculate average latitude
            lon = mean(Longitude, na.rm = TRUE),  # Calculate average longitude
            .groups = "drop")  # Remove grouping
```

```r
# Define legend breaks at 1000 intervals
min_sightings <- min(nyc_zip_map$Sightings, na.rm = TRUE)  # Get minimum sightings
max_sightings <- max(nyc_zip_map$Sightings, na.rm = TRUE)  # Get maximum sightings
legend_breaks <- seq(0, max_sightings, by = 1000)  # Define breaks for legend at 1000 intervals

# Create the Choropleth Map following best visualization practices
ggplot(nyc_zip_map) +
  # Darker borders for better contrast
  geom_sf(aes(fill = Sightings), color = "gray30", size = 0.3) +
  scale_fill_viridis(option = "magma", direction = -1, name = "Rat Sightings Density",
                     # Use viridis color scheme with defined legend breaks
                     breaks = legend_breaks, labels = legend_breaks) +
  labs(
    title = "NYC Rat Sightings",  # Set title
    subtitle = "Sightings by Zip Code (2010-2017)",  # Add subtitle for clarity
    caption = "Data Source: NYC Open Data"  # Provide data source
  ) +

  # Add Borough Labels with Annotations
  annotate("text", x = borough_labels$lon, y = borough_labels$lat,
           label = borough_labels$borough,
           # Borough labels with black font
           color = "black", fontface = "bold", size = 4, alpha = 0.9, hjust = 0.5) +
  theme_minimal(base_size = 11) +  # Use minimal theme with base font size
  theme(
    # Adjust title size and center it
    plot.title = element_text(size = 16, face = "bold", hjust = 0.5),
    plot.subtitle = element_text(size = 10, hjust = 0.5),  # Adjust subtitle size for hierarchy
    legend.position = "right",  # Place legend on the right
    legend.key.size = unit(0.3, "cm"),  # Reduce legend key size
    legend.key.width = unit(0.5, "cm"),  # Adjust legend box width
    legend.text = element_text(size = 8),  # Reduce legend text size
    # Light gray background for better contrast
    panel.background = element_rect(fill = "gray97", color = NA)
  )
```



"Rat Hotspots in NYC: Mapping the Infestation (2010-2017)"

Story: The Rat Capital of NYC

New York City has long been known for its thriving rat population, but where exactly do these rodents roam the most? This choropleth map paints a vivid picture of rat infestation zones across NYC from 2010 to 2017, highlighting which boroughs and zip codes have been most affected. From the visualization, Brooklyn and the Bronx emerge as the biggest rat hotspots, with some zip codes reporting over 3,000 sightings during this period.

Application of the CRAP Principles in the Choropleth Map

Contrast: The color gradient (yellow to deep purple) enhances readability, making high-density rat areas stand out against lower-density regions. The borough labels are in bold black text, ensuring they are distinguishable from the background and map elements.

Repetition: Consistent font styles and sizes are used throughout the map for titles, labels, and legends, ensuring a cohesive look. The color scale follows a logical progression, making it intuitive for users to interpret.

Alignment: Borough names are correctly positioned near their respective areas, aligning with real-world geography. The legend is neatly placed on the right, reducing clutter and ensuring easy reference.

Proximity: The borough labels are placed near their respective locations, improving clarity. The legend and title are positioned for easy readability, keeping related elements together without overwhelming the visualization.

Application of Kieran Healy's Principles of Great Visualizations

Show the Data Clearly: The map avoids unnecessary elements (e.g., 3D effects, excessive gridlines) and directly highlights rat infestations by zip code. The color gradient effectively conveys density, making it easy to see which areas are most affected.

Reduce Chartjunk: No unnecessary labels, lines, or decorations are included. Borough labels are added sparingly, only to help with interpretation rather than cluttering the image.

Enable Meaningful Comparisons: The color gradient provides an immediate visual comparison between areas of high and low rat activity. The legend uses 1000-intervals, allowing viewers to easily estimate the scale of infestations.

Highlight Key Information: Instead of presenting raw numbers, the visualization highlights the boroughs with the highest rat activity. The bold borough labels draw attention to key locations, guiding viewers to interpret the data correctly.

Be Truthful and Insightful: The map presents accurate data from NYC Open Data, ensuring no misleading elements. The story in the caption contextualizes the rat problem, raising questions about urban planning, sanitation, and population density in relation to rat infestations.

References

OpenAI. (2023). ChatGPT (March 14 version) [Large language model]. https://openai.com.

Healy, K. (2018). Data visualization: A practical introduction. Princeton University Press.

New York City. (n.d.). NYC rat sightings dataset. Kaggle. Retrieved March 3, 2025, from https://www.kaggle.com/datasets/new-york-city/nyc-rat-sightings/data.

Williams, R. (2014). The Non-Designer's Design Book: Design and Typographic Principles for the Visual Novice (4th ed.). Peachpit Press.