A1_Final

Huy Tran

2024-02-24

```
library(dplyr)
library(readr)
sightings<- read_csv("C:/Users/Erick/R/A1/A1_sightings.csv")
```

#Process for Creating NYC Rat Sightings Visualization and Cleaning the Data This memo delineates the procedural steps that I undertaken in order to produce the visual representation depicting the yearly progression of rodent sightings among the municipalities of New York City. The data, which was obtained from the NYC Open Data portal between 2010 and 2018, details documented rodent sightings by borough. I took into account the intended recipients of the graphic, which comprised policymakers, community leaders, and residents with an interest in rodent sightings. Over time, the primary message was to convey the general tendencies across municipalities. In order to adhere to the CRAP (clarity, relevance, accuracy, presentation) principles, I implemented the following: utilized clear labels, incorporated solely pertinent data, verified accuracy twice, and designed a visually appealing format.

Furthermore, I implemented Kieran Healy's four tenets of effective visualization—expectation, transparency, involvement, and accuracy—by employing eye-catching colors, employing unconventional bar charts to represent time series, and utilizing plain labels, unexpectedness, engagement, and truth. Furthermore, I emulated the attributes of Alberto Cairo—illuminating, eliciting emotion, surprising, sparking dialogue, and representing the data truthfully—by offering fresh perspectives on rat trends, which might elicit disgust or concern, the unexpectedness of the bar chart format, which most likely would stimulate discussion, and which I accomplished by providing accurate information. In general, by implementing these principles, a successful visual representation of significant rodent sighting trends for the target audience was produced.

```
# 2.0 Clean & Transform Data
library(dplyr)
library(readr)
library(lubridate)
library(ggplot2)

# Load the data from the CSV file
sightings <- read_csv("C:/Users/Erick/R/A1/A1_sightings.csv")

# Now, use the 'sightings' dataframe instead of 'rats_raw_df'
# Transforming and renaming existing columns
sightings <- sightings %>%
  mutate(
    `Created Date` = mdy_hms(`Created Date`),
    `Closed Date` = mdy_hms(`Closed Date`),
    `Due Date` = mdy_hms(`Due Date`),
    `Resolution Action Updated Date` = mdy_hms(`Resolution Action Updated
       Date`),

    # Create new variables based on 'Created Date' after conversion
```

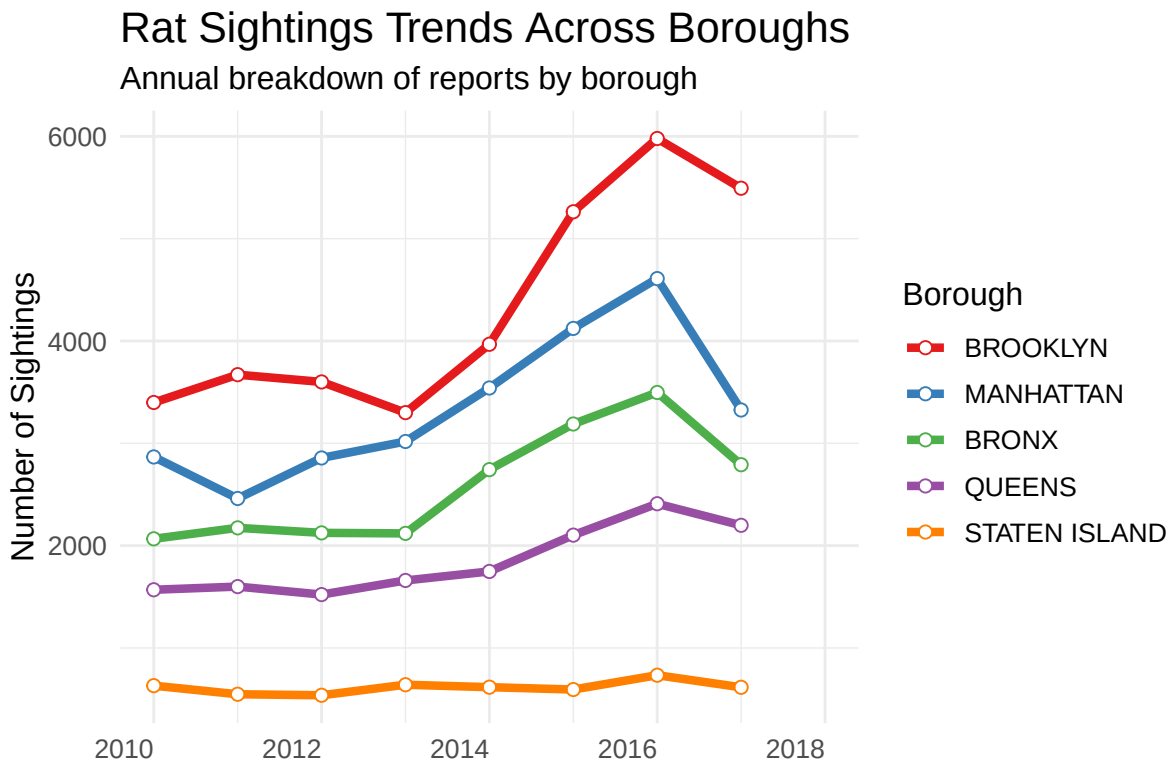
```
    sighting_year = year(`Created Date`),
    sighting_month = month(`Created Date`),
    sighting_month_name = month(`Created Date`, label = TRUE, abbr = FALSE
      ),
    sighting_day = day(`Created Date`),
    sighting_day_name = wday(`Created Date`, label = TRUE, abbr = FALSE),
    sighting_weekday = wday(`Created Date`),
    sighting_name_weekday = weekdays(`Created Date`)
  )

# Check the structure of the transformed and newly created columns
str(select(sightings, `Created Date`, `Closed Date`, `Due Date`, `
   Resolution Action Updated Date`, sighting_year, sighting_month,
   sighting_day))

## tibble [101,914 x 7] (S3: tbl_df/tbl/data.frame)
##  $ Created Date                  : POSIXct[1:101914], format:
   "2015-09-04 00:00:00" "2015-09-04 00:00:00" ...
##  $ Closed Date                   : POSIXct[1:101914], format:
   "2015-09-18 00:00:00" "2015-10-28 00:00:00" ...
##  $ Due Date                      : POSIXct[1:101914], format:
   "2015-10-04 15:01:02" "2015-10-04 10:02:58" ...
##  $ Resolution Action Updated Date: POSIXct[1:101914], format:
   "2015-09-18 00:00:00" "2015-10-28 00:00:00" ...
##  $ sighting_year                 : num [1:101914] 2015 2015 2015 2015
   2015 ...
##  $ sighting_month                : num [1:101914] 9 9 9 9 9 9 9 9 9 9
   ...
##  $ sighting_day                  : int [1:101914] 4 4 4 4 4 4 4 4 4 4
   ...

# List of columns to keep
columns_to_keep <- c("Unique Key",
                     "Created Date",
                     "sighting_year",
                     "sighting_month",
                     "sighting_day",
                     "sighting_weekday",
                     "Closed Date",
                     "Location Type",
                     "Incident Zip",
                     "City",
                     "Status",
                     "Borough",
                     "X Coordinate (State Plane)",
                     "Y Coordinate (State Plane)",
                     "Latitude",
                     "Longitude",
                     "Location")

# Drop variables not in 'columns_to_keep'
sightings <- sightings %>% select(all_of(columns_to_keep))
sightings <- sightings %>% mutate_all(~replace(., is.na(.), 0))
# Check the first few rows of the cleaned and transformed data frame
```

```
head(sightings)
```

```
## # A tibble: 6 x 17
##   `Unique Key` `Created Date`      sighting_year sighting_month
   sighting_day
##          <dbl> <dttm>                      <dbl>          <dbl>
   <dbl>
## 1     31464015 2015-09-04 00:00:00          2015              9
              4
## 2     31464024 2015-09-04 00:00:00          2015              9
              4
## 3     31464025 2015-09-04 00:00:00          2015              9
              4
## 4     31464026 2015-09-04 00:00:00          2015              9
              4
## 5     31464027 2015-09-04 00:00:00          2015              9
              4
## 6     31464188 2015-09-04 00:00:00          2015              9
              4
## # i 12 more variables: sighting_weekday <dbl>, `Closed Date` <dttm>,
## #   `Location Type` <chr>, `Incident Zip` <dbl>, City <chr>, Status <
   chr>,
## #   Borough <chr>, `X Coordinate (State Plane)` <dbl>,
## #   `Y Coordinate (State Plane)` <dbl>, Latitude <dbl>, Longitude <dbl
   >,
## #   Location <chr>
```

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

# Assuming the 'sightings' dataframe is properly set up and aggregated

# Aggregate data by year and borough
sightings_summary <- sightings %>%
  group_by(sighting_year, Borough) %>%
  summarise(count = n(), .groups = 'drop') %>%
  filter(Borough != "Unspecified") # This line removes the "Unspecified"
     data

# Define a color palette with enough colors for each remaining borough
color_palette <- c("#E41A1C", "#377EB8", "#4DAF4A", "#984EA3", "#FF7F00")

# The order of boroughs corresponding to the color palette, without "
   Unspecified"
ordered_boroughs <- c("BROOKLYN", "MANHATTAN", "BRONX", "QUEENS", "STATEN
   ISLAND")

# Ensure that the boroughs are factored in the correct order before
   plotting.
sightings_summary$Borough <- factor(sightings_summary$Borough, levels =
   ordered_boroughs)
```

```r
# Adjusting the headline size and the number of sightings label size
ggplot(sightings_summary, aes(x = sighting_year, y = count, color =
    Borough, group = Borough)) +
  geom_line(size = 1.5) +
  geom_point(size = 2, shape = 21, fill = "white") +
  scale_color_manual(values = color_palette) + # This applies your color
     palette
  expand_limits(x = max(sightings_summary$sighting_year) + 1) +
  theme_minimal() +
  labs(
    title = "Rat Sightings Trends Across Boroughs", # Removed "Different"
       for brevity
    subtitle = "Annual breakdown of reports by borough",
    x = "", # Removed "Year"
    y = "Number of Sightings",
    color = "Borough"
  ) +
  theme(
    plot.title = element_text(hjust = 0, size = 16), # Title aligned to
       the left
    plot.subtitle = element_text(hjust = 0, size = 12), # Subtitle aligned
        to the left
    legend.title = element_text(size = 12),
    legend.text = element_text(size = 10),
    axis.text.x = element_text(angle = 0, hjust = 1, size = 10, vjust =
       0.5),
    axis.text.y = element_text(size = 10),
    axis.title.x = element_text(size = 12),
    axis.title.y = element_text(size = 12),
    plot.margin = unit(c(1, 1, 1, 1), "lines") # Adjusted plot margins
  )
```




```
# Saving the plot
ggsave("rat_sightings_highres.png", plot = last_plot(), dpi = 300, bg = "
    white")
ggsave("rat_sightings_highres.pdf", plot = last_plot(), dpi = 300, bg = "
    white")
```