Pokemon Go

Date, Time and Geo Data

Litta Jose Thottam, 135546

2024-06-17

Contents

Pokemon Go

Pokémon GO is an augmented reality (AR) mobile game developed and published by Niantic in collaboration with Nintendo and The Pokémon Company. It was released in July 2016 and quickly became a global phenomenon, attracting millions of players worldwide.

The game utilizes GPS and the camera of compatible devices to allow players to capture, battle, and train virtual creatures called Pokémon, which appear as if they are in the real world. Players explore their surroundings to find and catch Pokémon, visit PokéStops (real-world locations marked on the in-game map), and participate in Gym battles to compete with other players.

Gameplay Data

The `GameplayLocationHistory.tsv` file contains Pokémon GO gameplay location data recorded during player sessions. This file follows the TSV (Tab-Separated Values) format, which structures data similarly to CSV files but uses tabs to separate fields.

The dataset includes essential information such as the date and time of each reported location, along with latitude and longitude coordinates. This data provides valuable insights into player behavior, allowing analysis of movement patterns and visualization of gameplay activity across different time periods and geographical areas. Researchers and analysts can leverage this dataset to better understand player engagement, popular gaming locations, and trends in Pokémon GO gameplay.

Gameplay Map (4 pts)

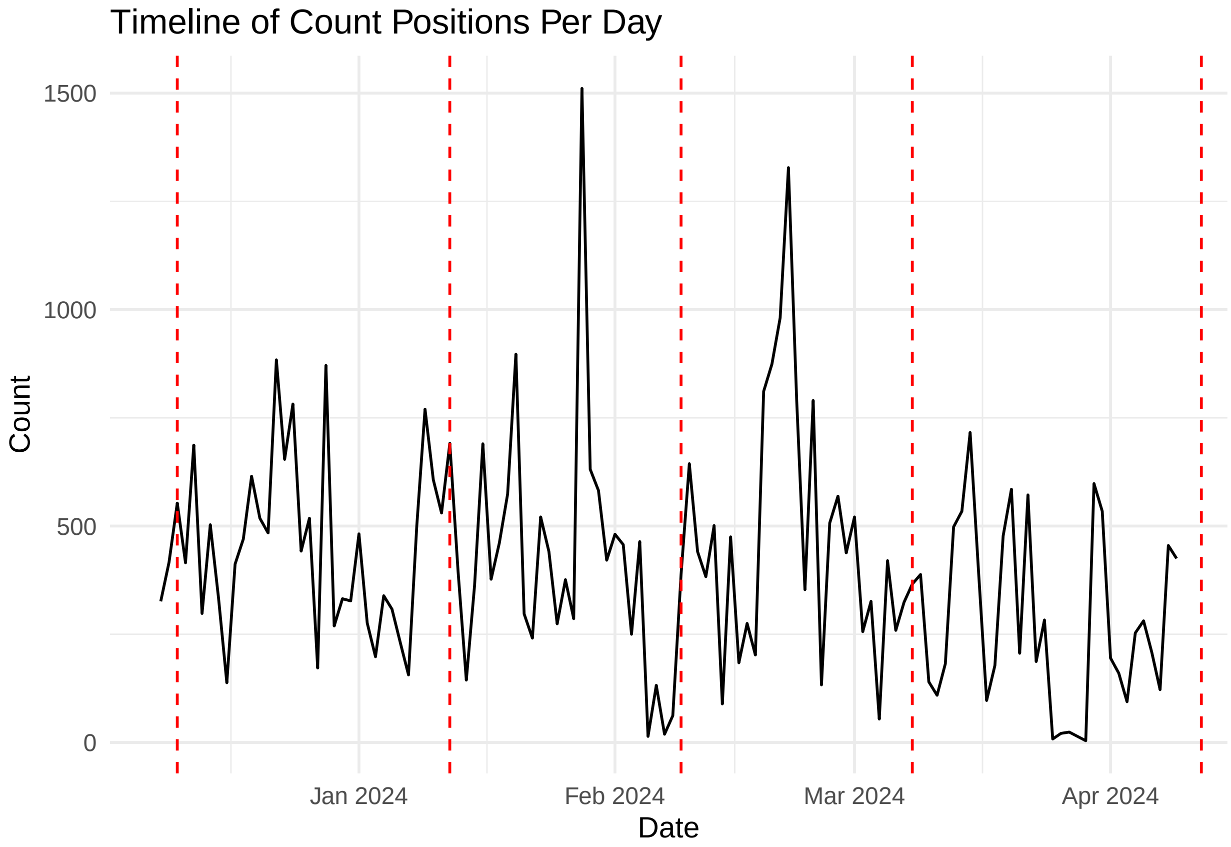
Please create an interactive `leaflet` map that displays all available coordinates within Germany, with the color indicating the time since the start of data collection. You can have a slightly different style for the map. It should be any leaflet format. The timing of data points should be visible by any means.

```r
library(dplyr)
library(leaflet)
library(readr)
library(lubridate)

# Read the data
gameplay_data <- read_tsv("GameplayLocationHistory.tsv")
colnames(gameplay_data) <- c("Date and Time", "latitude", "longitude")

# Convert date/time to POSIXct format
gameplay_data$`Date and Time` <- as.POSIXct(gameplay_data$`Date and Time`)

# Filter the data to include only coordinates within Germany
# Assuming you have 'latitude' and 'longitude' columns
germany_bounds <- list(
  north = 55.058347,
  south = 47.2701114,
  west = 5.8663425,
  east = 15.0419319
)

gameplay_data_germany <- gameplay_data %>%
  filter(
    latitude >= germany_bounds$south & latitude <= germany_bounds$north &
    longitude >= germany_bounds$west & longitude <= germany_bounds$east
  )

# Calculate days since the start date
start_date <- min(gameplay_data_germany$`Date and Time`)
gameplay_data_germany$days_since_start <- as.numeric(difftime(gameplay_data_germany$`Date and Time`, sta

# Create a color palette
pal <- colorNumeric(
  palette = "viridis",
  domain = gameplay_data_germany$days_since_start
)
```

Game Data TimeLine(2 pts)

Please create a timeline of all observations per day to inspect the total activity. The overall gameplay seems to have some days of increased activity. Let's see if we can connect them with event data. Pokémon GO Community Days are special monthly events organized by Niantic, the developer of Pokémon GO. During Community Days, players have the opportunity to encounter a specific Pokémon more frequently in the wild for a limited time period, typically three hours. Please add community days to the graph to see if peaks are related to the events. Adding the event data is optional!

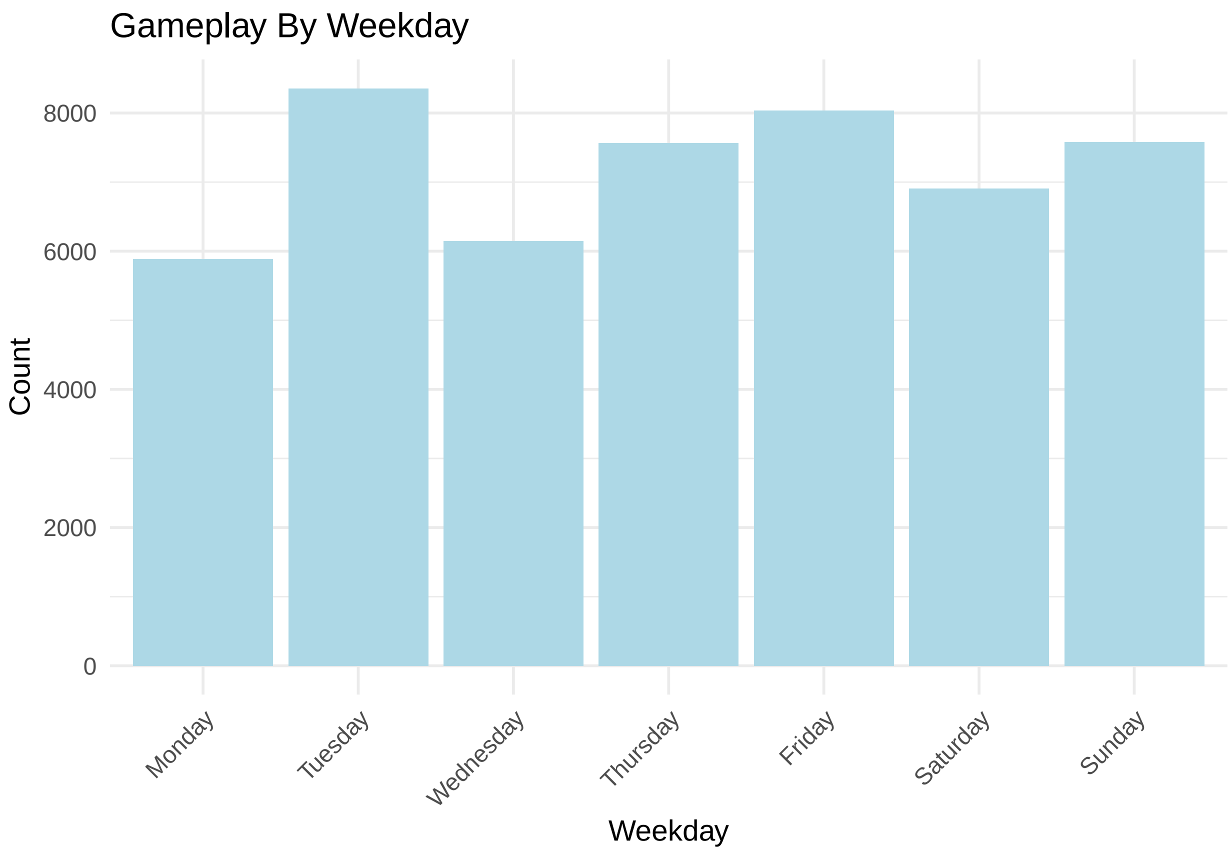
```r
library(dplyr)
library(readr)
library(ggplot2)
# Read the data and adjust column names
gameplay_data <- read_tsv("GameplayLocationHistory.tsv")

colnames(gameplay_data) <- c("time_since_start", "latitude", "longitude")

# Extract date from time_since_start
gameplay_data$date <- as.Date(gameplay_data$time_since_start)

# Aggregate data to count the number of observations per day
daily_activity <- gameplay_data %>%
  group_by(date) %>%
  summarize(count = n())

# Define Pokémon GO Community Days (example data, replace with actual dates)
community_days <- as.Date(c(
  "2023-12-10", "2024-01-12", "2024-02-09", "2024-03-08",
  "2024-04-12", "2024-05-10", "2024-06-14"
))

# Create the plot
activity_plot <- ggplot(daily_activity, aes(x = date, y = count)) +
  geom_line(color = "black") +
  geom_vline(xintercept = as.numeric(community_days), color = "red", linetype = "dashed") +
  scale_x_date(date_breaks = "1 month", date_labels = "%b %Y") +
  labs(title = "Timeline of Count Positions Per Day",
       x = "Date",
       y = "Count",
       ) +
  theme_minimal()

# Print the plot
#(activity_plot)
##not printing here as it would actually visualise chart again
```



Game Data Weekdays(2 pts)

Please illustrate in a bar plot the count of game plays for each day of the week. Please sort days according to their logical order Monday-Sunday.

```r
# Read the data and adjust column names
gameplay_data <- read_tsv("GameplayLocationHistory.tsv")

colnames(gameplay_data) <- c("time_since_start", "latitude", "longitude")

# Extract the day of the week from time_since_start
gameplay_data$day_of_week <- weekdays(as.Date(gameplay_data$time_since_start))

# Define the order of days from Monday to Sunday
day_order <- c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday")

# Summarize data to count the number of observations per day of the week
weekly_activity <- gameplay_data %>%
  group_by(day_of_week) %>%
  summarise(count = n()) %>%
  mutate(day_of_week = factor(day_of_week, levels = day_order)) %>%
  arrange(day_of_week)

# Create the bar plot
day_plot <- ggplot(weekly_activity, aes(x = day_of_week, y = count)) +
   geom_bar(stat = "identity", fill = "lightblue") +
  scale_fill_brewer(palette = "Set3") +
  labs(title = "Gameplay By Weekday",
       x = "Weekday",
       y = "Count",
      ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

# Print the plot
##day_plot

##not printing here as it would actually visualise chart again
```

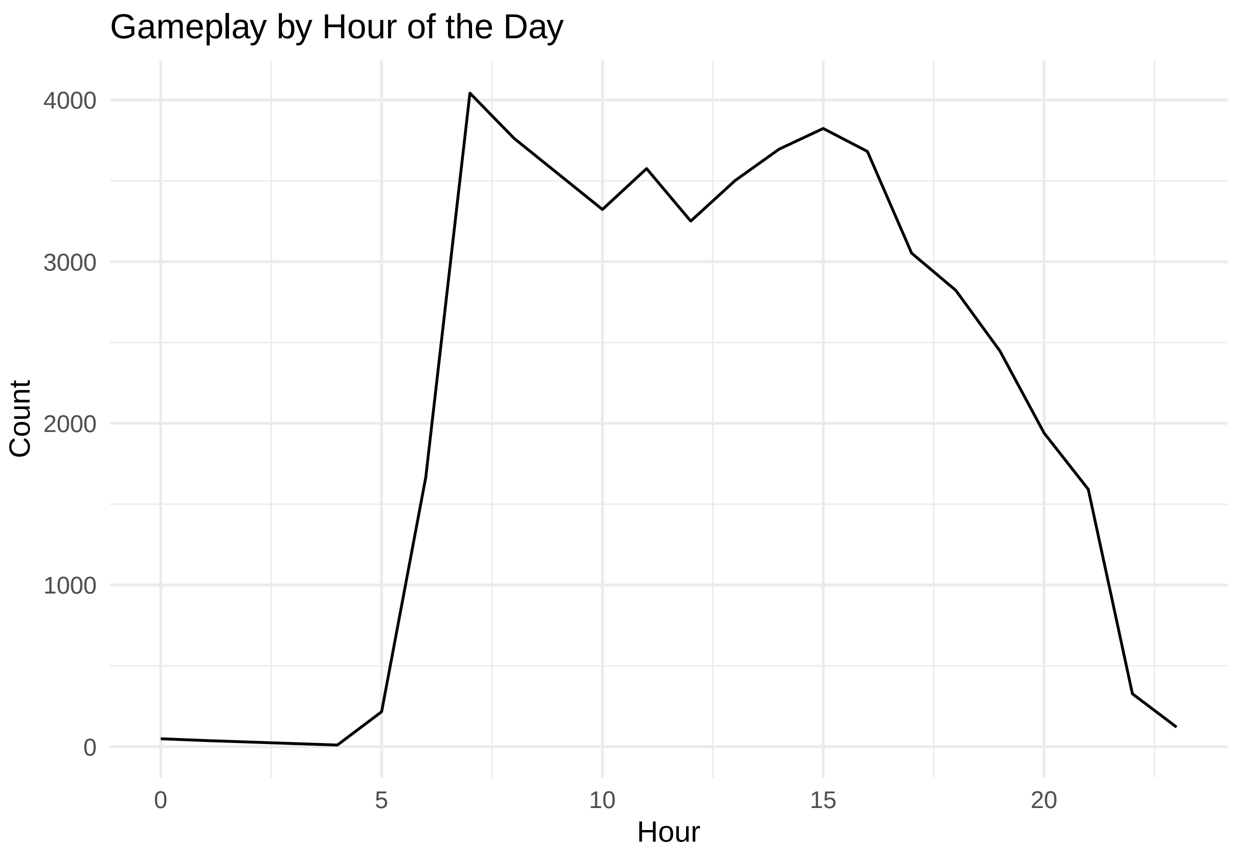


Game Data Time Of Day(2 pts)

When do players sleep? Please summarize the gameplay by the hour of the data in a line plot.

```
library(ggplot2)
library(readr)
library(dplyr)
# Read the data and adjust column names
gameplay_data <- read_tsv("GameplayLocationHistory.tsv")
```

```
colnames(gameplay_data) <- c("time_since_start", "latitude", "longitude")

# Extract the hour from time_since_start
gameplay_data$hour <- format(as.POSIXct(gameplay_data$time_since_start), "%H")

# Summarize data to count the number of observations per hour
hourly_activity <- gameplay_data %>%
  group_by(hour) %>%
  summarise(count = n()) %>%
  mutate(hour = as.numeric(hour)) %>%
  arrange(hour)

# Create the line plot
hourly_plot <- ggplot(hourly_activity, aes(x = hour, y = count)) +
  geom_line(color = "black")  +
 scale_x_continuous(breaks = c(0, 5, 10, 15, 20)) +
  labs(title = "Gameplay by Hour of the Day",
       x = "Hour",
       y = "Count") +
  theme_minimal()

# Print the plot
##hourly_plot

##not printing here as it would actually visualise chart again
```


Gameplay by Hour of the Day
Count
Hour