Cornell Gorges

A Data-Driven Look at the Ithacan Gorges

Ming DeMers

September 10, 2023

Contents

Introduction

The gorges at Cornell University are one of the university's most iconic and scenic features. Nestled in the heart of the campus, Cornell's gorges are a series of deep and narrow ravines carved by the flow of water over thousands of years. These beautiful natural formations offer a tranquil escape from the hustle and bustle of campus life, providing students and visitors with picturesque spots for relaxation and exploration. While the gorges are captivating, they can be deceptively dangerous. Safety measures are enforced and visitors are urged to follow the guidelines.

The Gorge Stewards Program was established in 2014 to improve safety and enjoyment of the gorges, following the death of a student earlier that year. Gorge stewards walk the paths from May to September to provide information about trails, safety rules, natural history, activities, and swimming alternatives. Stewards also track visitor use, including unauthorized or illegal uses.

This project will explore how usage of the Gorges has fluctuated, how violations of gorge rules has decreased, and explain overall the changing landscape of the Gorges.

The Data

The data is sourced from the Gorge Stewards Program, courtesy of the Cornell Botanic Gardens and Cornell Outdoor Education (COE). It comprises of a table for each year recorded, with a number of variables detailing gorge usage, violations, weather and more.

Data Cleaning

```r
clean_gorge <- function(df) {
  df <- df |>
  mutate(Date = as.Date(Date, format('%m/%d/%Y'))) |>
  drop_na(Date) |>
  mutate(High.Temperature = as.numeric(High.Temperature)) |>
  clean_names()

  colnames(df) <- gsub("number\\_of\\_", "", colnames(df))

  return(df)
}

# lapply(gorges, function)

gorge_2022_clean <- gorge_2022_raw |>
  clean_gorge()
```

```
## mutate: converted 'Date' from character to Date (1 new NA)
```

```
## drop_na: removed one row (1%), 108 rows remaining
```

```
## Warning: There was 1 warning in `.fun()`.
## i In argument: `High.Temperature = as.numeric(High.Temperature)`.
## Caused by warning:
## ! NAs introduced by coercion
```

```
## mutate: converted 'High.Temperature' from character to double (2 new NA)
```

```
gorge_2022_clean <- gorge_2022_clean |>
  mutate(date = replace(date, row_number() == 23, '2022-06-09')) |>
  mutate(steward_name = if_else(steward_name == "Philip",
                                "Phillip",
                                steward_name)) |>
  mutate(steward_name = str_replace_all(
    str_to_title(steward_name),
    " ",
    ""))
```

```
## mutate: changed one value (1%) of 'date' (0 new NA)
```

```
## mutate: changed one value (1%) of 'steward_name' (0 new NA)
```

```
## mutate: changed 9 values (8%) of 'steward_name' (0 new NA)
```

```
gorge_2022 <- gorge_2022_clean
```

We first import the sets and clean up the data best we can.

Gorge Statistics

We are interested in understanding the Gorges by the numbers. `Gorges` is a dataset that has all the years joined into one dataset. Due to format of the data changing, some granular detail is lost.

Join the Gorges

```
gorges_raw <- read.csv('data/gorges.csv')

gorges_clean <- gorges_raw |>
  drop_na(Date) |>
  drop_na(Number.of.observed.gorge.users) |>
  drop_na(total.Number.of.people.obseved.violating.a.rule) |>
  mutate(Date = as.Date(Date, format('%m/%d/%Y')),
         observed_gorge_users = as.numeric(Number.of.observed.gorge.users),
         observed_violations = as.numeric(
           total.Number.of.people.obseved.violating.a.rule),
         High.Temperature = as.numeric(High.Temperature)) |>
  clean_names()
```

```
## drop_na: no rows removed
## drop_na: no rows removed
## drop_na: no rows removed
```

```
## Warning: There were 3 warnings in `.fun()`.
## The first warning was:
## i In argument: `observed_gorge_users =
##   as.numeric(Number.of.observed.gorge.users)`.
## Caused by warning:
## ! NAs introduced by coercion
## i Run `dplyr::last_dplyr_warnings()` to see the 2 remaining warnings.
```

```
## mutate: converted 'Date' from character to Date (131 new NA)

##         converted 'High.Temperature' from character to double (84 new NA)

##         new variable 'observed_gorge_users' (double) with 327 unique values and 15% NA

##         new variable 'observed_violations' (double) with 41 unique values and 16% NA
```

```
gorges <- gorges_clean |>
  mutate(observed_gorge_users = ifelse(is.na(observed_gorge_users),
                                       0,
                                       observed_gorge_users),
         observed_violations = ifelse(is.na(observed_violations),
                                      0,
                                      observed_violations),
         high_temperature = ifelse(is.na(high_temperature),
                                   0,
                                   high_temperature)
         ) |>
  filter(year(date) > (2014))
```

```
## mutate: changed 84 values (8%) of 'high_temperature' (84 fewer NA)

##         changed 163 values (15%) of 'observed_gorge_users' (163 fewer NA)

##         changed 173 values (16%) of 'observed_violations' (173 fewer NA)

## filter: removed 132 rows (12%), 975 rows remaining
```

Yearly Stats

```
gorges_tbl <- gorges |>
  group_by(year(date)) |>
  summarise(shifts_completed = n(),
            visitors = sum(observed_gorge_users),
            violations = sum(observed_violations),
            stewards = length(unique(steward_name)),
            avg_temp = mean(high_temperature[high_temperature != 0])) |>
  mutate(year = `year(date)`)
```

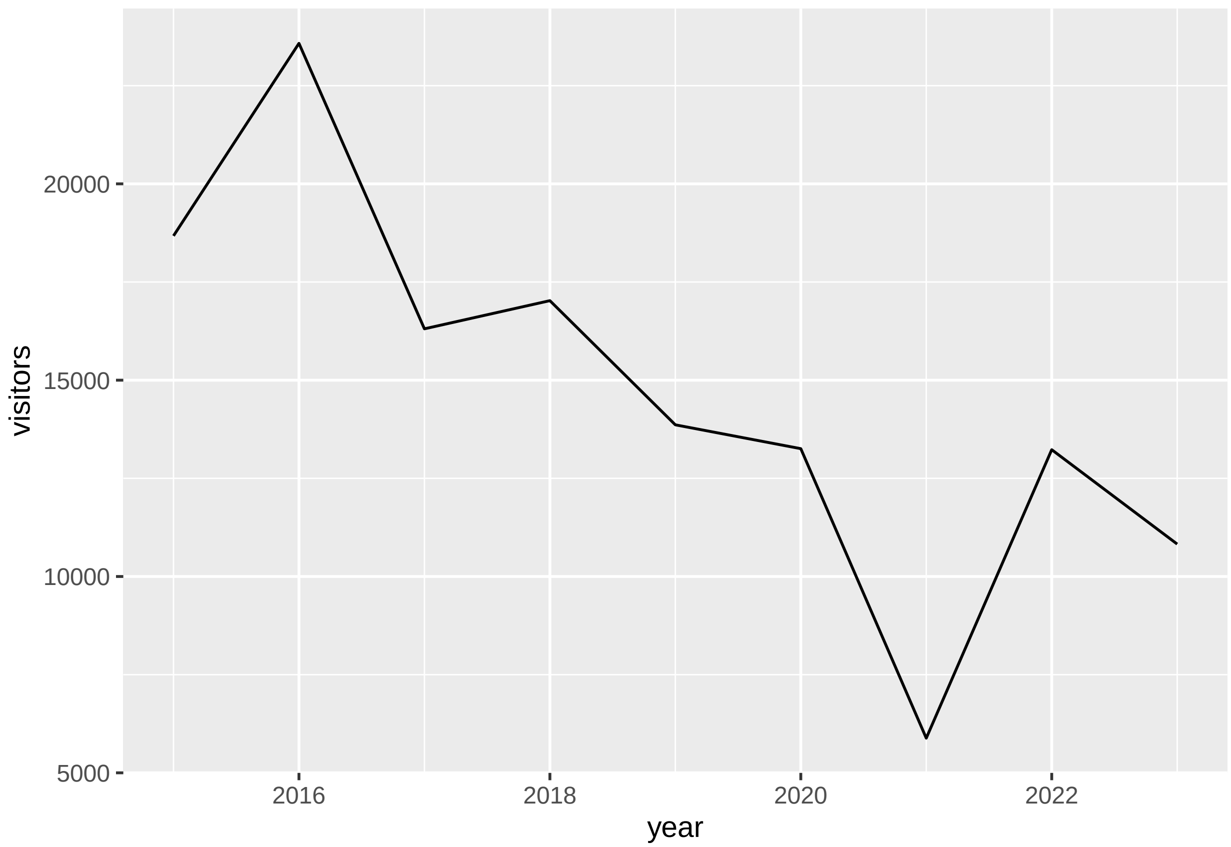
```
## group_by: one grouping variable (year(date))

## summarise: now 9 rows and 6 columns, ungrouped

## mutate: new variable 'year' (double) with 9 unique values and 0% NA
```

```
gorges_tbl
```

```
## # A tibble: 9 x 7
##   `year(date)` shifts_completed visitors violations stewards avg_temp  year
##          <dbl>            <int>    <dbl>      <dbl>    <int>    <dbl> <dbl>
## 1         2015              124    18675        407       10     77.9  2015
## 2         2016               99    23578        807        8     82.3  2016
## 3         2017              107    16308        638       13     74.8  2017
## 4         2018               99    17025        283       13     74.8  2018
## 5         2019               91    13863         69        8     78.0  2019
## 6         2020              154    13256        958       16     81.7  2020
## 7         2021               96     5883        173       27     79.4  2021
## 8         2022              107    13229        562       12     81.2  2022
## 9         2023               98    10826        273       14     77.2  2023
```

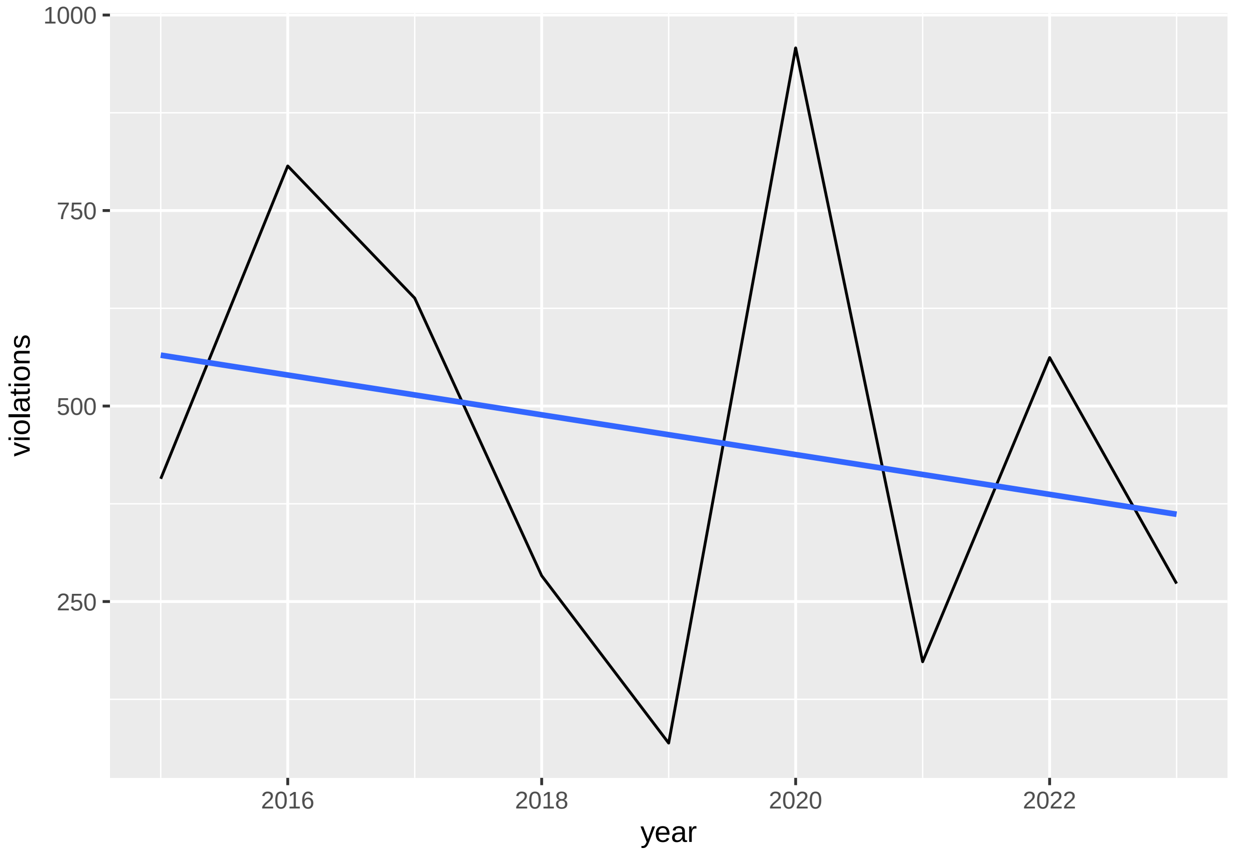
We see some interesting points. Overall, it seems that the around 14,000 visitors experience the Gorges every year, and around 12 stewards patrol every Summer. The average temperature is a high 70s, and around 100 shifts are completed every year. Notably are the years 2018, 2019, 2020, and 2021.

```
gorges_tbl |>
  ggplot(aes(x = year, y = visitors)) +
  geom_line()
```

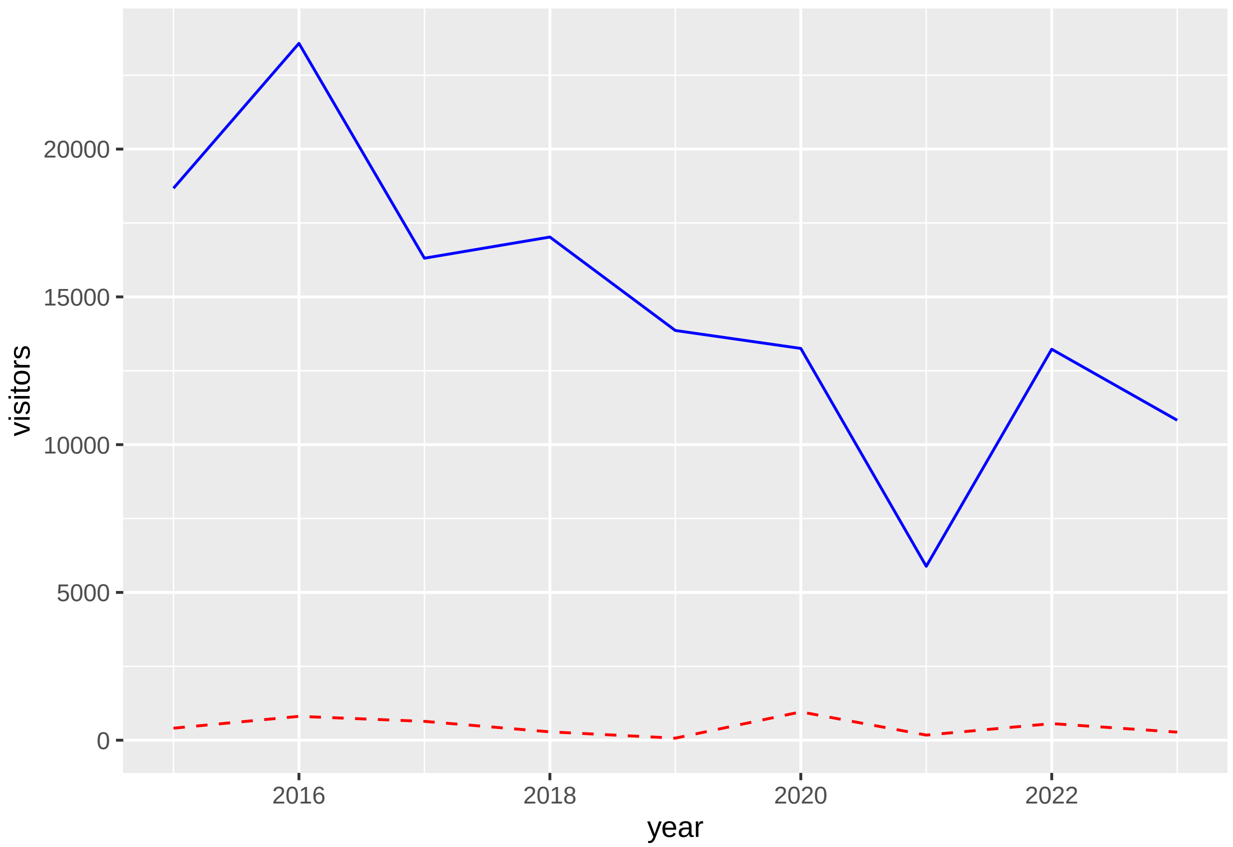


```
gorges_tbl |>
  ggplot(aes(x = year, y = violations)) +
  geom_line() +
  geom_smooth(method = lm, se = F)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



```
gorges_tbl |>
  ggplot(aes(x = year)) +
  geom_line(aes(y = visitors), color = "blue", linetype = "solid") +
  geom_line(aes(y = violations), color = "red", linetype = "dashed")
```



Interestingly, we see the amount of visitors spikes in 2016, and falls to it lowest in 2021. 2020 shows only slightly lesser numbers than previous years, and 2022 shows membership returned near fully. It's hard to determine why this drop occured - the COVID-19 Pandemic would be a likely explanation, but that doesn't seem to account for 2020's regular number.

Over the 8 years of the program, violations overall have a downward trend, which is a good outcome of the Stewards Program. However, the decline is not steady, and there's a noticeable spike in 2020 and 2022. Again, this could be explained by the COVID pandemic: one of the few activities during the pandemic was going outside, and perhaps individuals did not follow rules as they were unaware of them or more careless during that year.

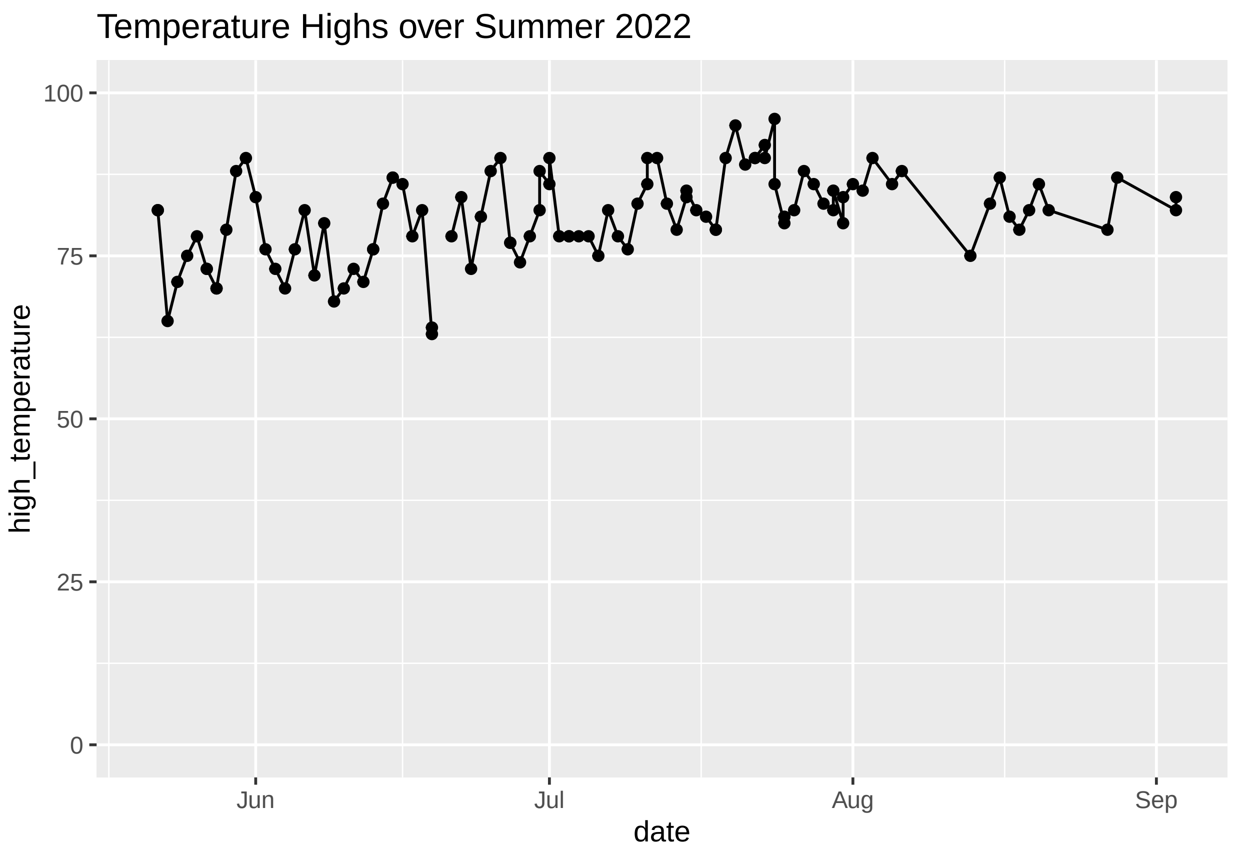
A Look at 2022

We now choose to look at a specific year. We select 2022 as it has high visitor numbers and is more recent.

```
gorge_2022 |>
  ggplot(aes(x = date, y = high_temperature)) +
  geom_point() + ylim(0, 100) + geom_line() +
  labs(
    title = 'Temperature Highs over Summer 2022'
  )
```

```
## Warning: Removed 2 rows containing missing values (`geom_point()`).
```

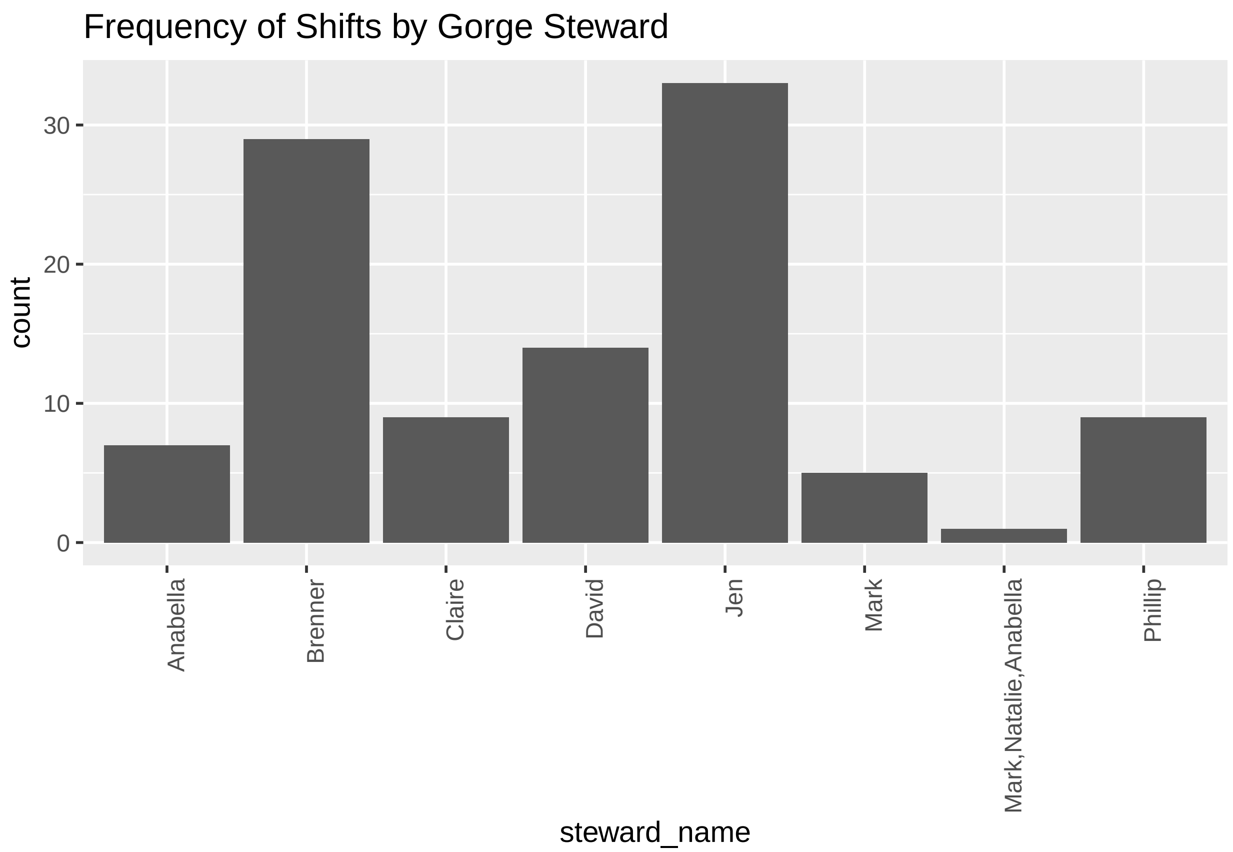
```
## Warning: Removed 1 row containing missing values (`geom_line()`).
```



```
gorge_2022 |>
  filter(steward_name != '') |>
  ggplot(aes(x = steward_name)) +
  geom_histogram(stat = 'Count') +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  labs(
    title = 'Frequency of Shifts by Gorge Steward'
  )
```

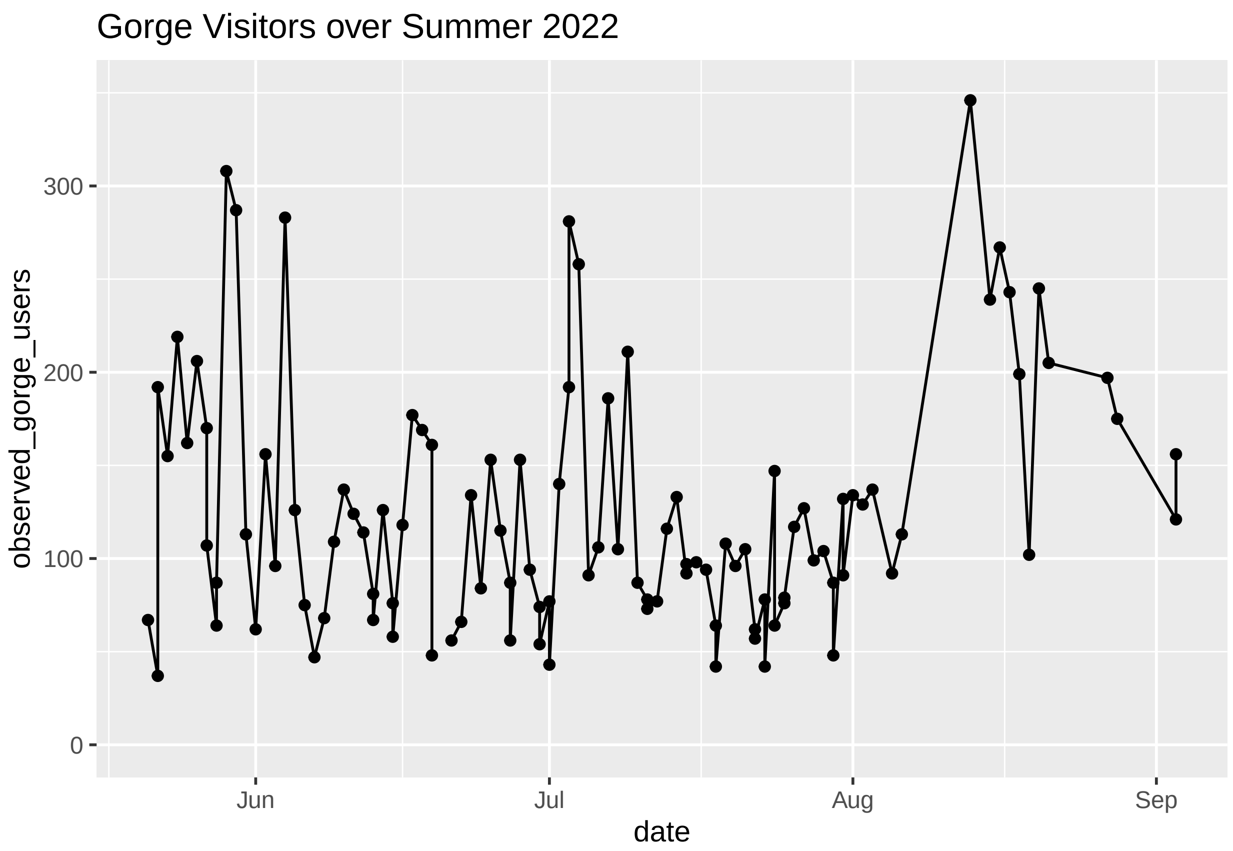
```
## filter: removed one row (1%), 107 rows remaining
```

```
## Warning in geom_histogram(stat = "Count"): Ignoring unknown parameters:
## `binwidth`, `bins`, and `pad`
```

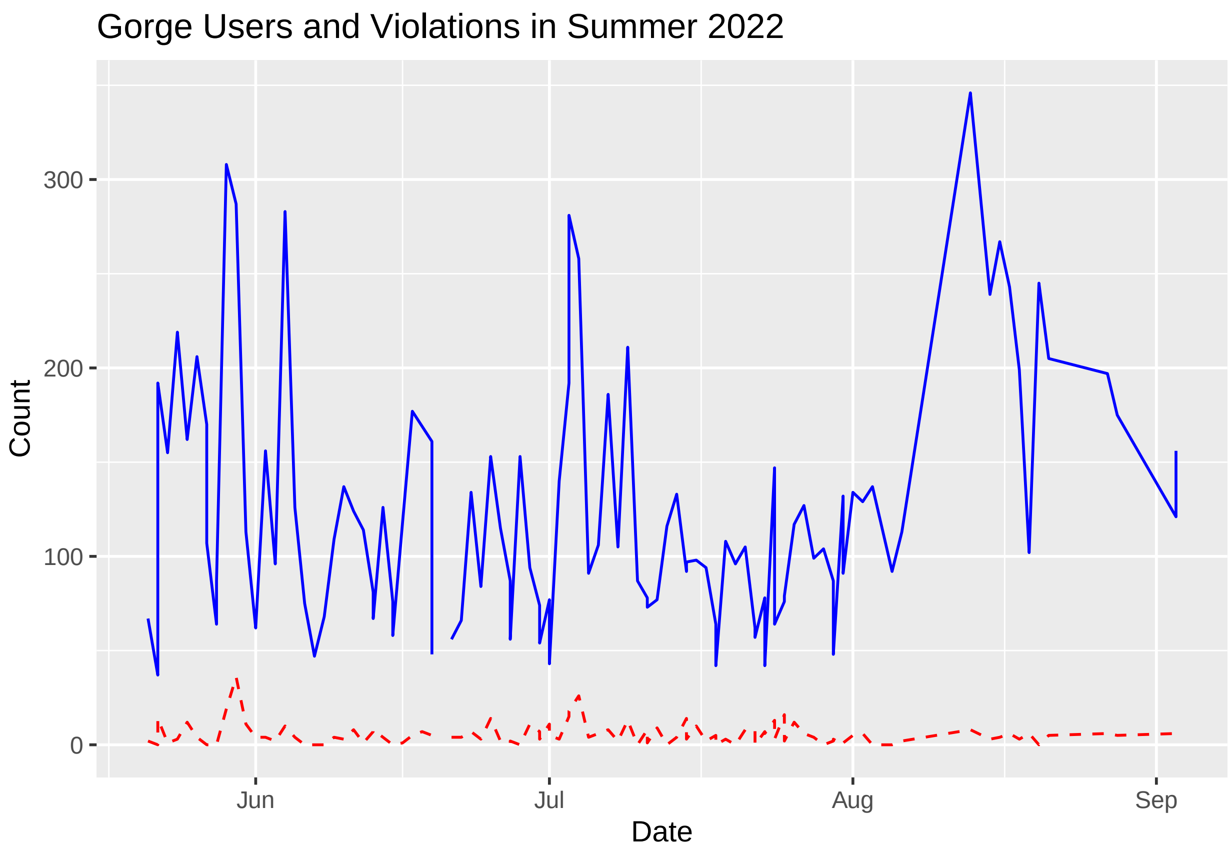


```
gorge_2022 |>
  ggplot(aes(x = date, y = observed_gorge_users)) +
  geom_point() + geom_line() + ylim(0, 350) +
  labs(
    title = 'Gorge Visitors over Summer 2022'
  )
```

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

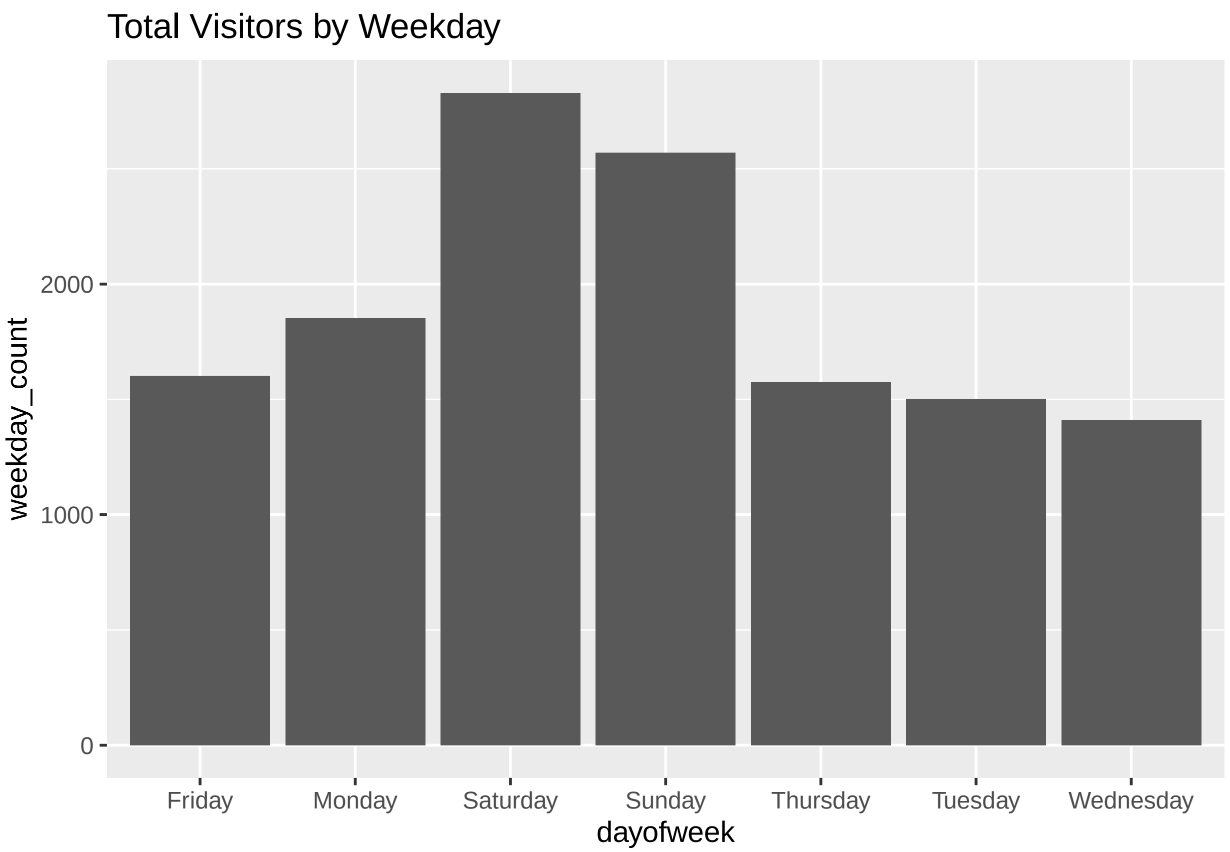


```r
gorge_2022 |>
  ggplot(aes(x = date)) +
  geom_line(aes(y = observed_gorge_users), color = "blue", linetype = "solid") +
  geom_line(aes(y = observed_violations), color = "red", linetype = "dashed") +
  labs(title = "Gorge Users and Violations in Summer 2022",
       x = "Date",
       y = "Count")
```



```r
gorge_2022 |>
  drop_na(observed_gorge_users) |>
  mutate(dayofweek = weekdays(date)) |>
  group_by(dayofweek) |>
  summarize(weekday_count = sum(observed_gorge_users)) |>
  ggplot(aes(dayofweek, weekday_count)) +
  geom_col() +
  labs(
    title = "Total Visitors by Weekday"
  )
```

```
## drop_na: removed one row (1%), 107 rows remaining

## mutate: new variable 'dayofweek' (character) with 7 unique values and 0% NA

## group_by: one grouping variable (dayofweek)

## summarize: now 7 rows and 2 columns, ungrouped
```



# Word Usage

```
library(stringr)

common_prepositions <- c("this", "is", "a", "an", "the", "and", "with", "than", "of", "in", "on", "at",

most_freq_words <- function(df, column, top_n, ignore_words) {
  df %>%
    pull(!!column) %>%
    str_split("\\s+") %>%
    unlist() %>%
    tolower() %>%
    str_replace_all("[[:punct:]]$", "") %>%  # Remove trailing punctuation marks
    str_remove("\\d+") %>%
    setdiff(common_prepositions) %>%
    table() %>%
    as.data.frame() %>%
    arrange(desc(Freq))
}



interact_words <- most_freq_words(gorge_2022, 'description_of_interactions', 50)
```

Questions to answer:

Does temp affect visitors? What day of the week or am/pm is busiest? How have violations gone down? What words are most frequent? How did covid affect gorge usage/how has it rebounded?

Future: Would also like to introduce photos

Public facing installation to highlight safety? Where is are the most violations occuring?

Safety might be a great approach Safety report like from perfect match

What do the national parks do that might be similar?

Photographing the beauty of the gorges