STATS-506 HW4

Zekai Xu

2025-10-17

Github Repo

Question 1

Part a

```r
# Load Dataset
data(nzge)

tbl <- nzge %>%
  group_by(election_year, voting_type) %>%
  summarize(vote_count = sum(votes, na.rm = TRUE), .groups="drop") %>%
  arrange(desc(vote_count))

print(tbl)
```

```
# A tibble: 10 x 3
   election_year voting_type vote_count
           <dbl> <chr>            <dbl>
 1          2014 Party          2416479
 2          2014 Candidate      2375493
 3          2008 Party          2356536
 4          2008 Candidate      2325598
 5          2005 Party          2286190
 6          2005 Candidate      2260670
 7          2011 Party          2257336
 8          2011 Candidate      2225766
 9          2002 Party          2040248
10          2002 Candidate      2022115
```

Part b

```r
tbl <- nzge %>%
  filter(voting_type == "Candidate" & election_year == "2014") %>%
  group_by(party) %>%
  summarize(vote_count = sum(votes, na.rm=TRUE), .groups = "drop") %>%
  mutate(percent = vote_count / sum(vote_count) * 100) %>%
  arrange(desc(percent))

print(tbl)
```

```
# A tibble: 25 x 3
   party                     vote_count percent
   <chr>                          <dbl>   <dbl>
 1 National Party               1081787  45.5
 2 Labour Party                  801287  33.7
 3 Green Party                   165718   6.98
 4 Conservative Party             81075   3.41
 5 New Zealand First Party        73384   3.09
 6 Maori Party                    42108   1.77
 7 MANA Movement                  32333   1.36
 8 Informal Candidate Votes       27886   1.17
 9 ACT New Zealand                27778   1.17
10 United Future                  14722   0.620
# i 15 more rows
```

Part c

```r
tbl <- nzge %>%
  select(election_year, voting_type, party, votes) %>%
  group_by(election_year, voting_type, party) %>%
  summarize(vote_count = sum(votes, na.rm=TRUE), .groups="drop_last") %>%
  mutate(percent = vote_count / sum(vote_count)) %>%
  slice_max(order_by = percent, n = 1, with_ties = FALSE) %>%
  ungroup() %>%
  select(election_year, voting_type, party, percent) %>%
  pivot_wider(
    names_from = voting_type,
    values_from = c(party, percent),
```

```
    names_glue = "{voting_type}_{.value}"
  ) %>%
  arrange(election_year)

print(tbl)
```

```
# A tibble: 5 x 5
  election_year Candidate_party Party_party    Candidate_percent Party_percent
          <dbl> <chr>           <chr>                      <dbl>         <dbl>
1          2002 Labour Party    Labour Party               0.441         0.411
2          2005 National Party  Labour Party               0.399         0.409
3          2008 National Party  National Party             0.461         0.447
4          2011 National Party  National Party             0.462         0.469
5          2014 National Party  National Party             0.455         0.468
```

Question 2

Part a

```
# Load Dataset
url <- "https://raw.githubusercontent.com/JeffSackmann/tennis_atp/refs/heads/master/atp_matc
tennis <- read_csv(url, show_col_types = FALSE)

num_tour_2019 <- tennis %>%
  filter(tourney_date >= "20190101" & tourney_date <= "20191231") %>%
  summarize(n_tournaments = n_distinct(tourney_id))

paste0("Number of tournaments that took place in 2019: ", num_tour_2019)
```

```
[1] "Number of tournaments that took place in 2019: 125"
```

Part b

```
winners <- tennis %>%
  filter(round == "F") %>%
  group_by(winner_id) %>%
  summarize(n_tournaments = n_distinct(tourney_id), .groups="drop") %>%
```

```r
  arrange(desc(n_tournaments))

multi_winners <- winners %>%
  filter(n_tournaments > 1)

paste0("Number of players who won more than one tournament: ", dim(multi_winners)[1])
```

```
[1] "Number of players who won more than one tournament: 12"
```

```r
paste0("Number of tournaments that the most winning player win: ", multi_winners[1, 2])
```

```
[1] "Number of tournaments that the most winning player win: 5"
```

Part c

We use Bootstrap to build a 95% confidence interval for the mean difference without assuming normality, and the hypothesis is:

$$H_0 : ace_{winner} - ace_{loser} > 0, \quad H_1 : ace_{winner} - ace_{loser} \leq 0$$

```r
# Compute per-match difference in aces
ace_diff <- tennis %>%
  transmute(diff = w_ace - l_ace) %>%
  filter(!is.na(diff))

# Observed mean difference
obs_mean <- ace_diff %>%
  specify(response = diff) %>%
  calculate(stat = "mean")

# Bootstrap resampling
set.seed(506)
boot_dist <- ace_diff %>%
  specify(response = diff) %>%
  generate(reps = 5000, type = "bootstrap") %>%
  calculate(stat = "mean")

# 95% percentile confidence interval
lower_bound <- quantile(boot_dist$stat, probs = 0.95)
paste0("One-sided confidence interval is: [", lower_bound, ", Inf).")
```

```
[1] "One-sided confidence interval is: [1.93244246473645, Inf)."
```

The 95% doesn't contain 0, and therefore we fail to reject the null hypothesis.

Part d

```r
win_rate <- tennis %>%
  select(winner_id, winner_name, loser_id, loser_name) %>%
  pivot_longer(
    cols = c(winner_id, loser_id),
    names_to = "result",
    values_to = "player_id"
  ) %>%
  mutate(
    player_name = if_else(result == "winner_id", winner_name, loser_name),
    win = if_else(result == "winner_id", 1, 0)
  ) %>%
  group_by(player_id, player_name) %>%
  summarize(
    matches = n(),
    wins = sum(win),
    win_rate = wins / matches,
    .groups = "drop"
  ) %>%
  filter(matches >= 5) %>%
  arrange(desc(win_rate))

win_rate
```

```
# A tibble: 167 x 5
   player_id player_name        matches  wins win_rate
       <dbl> <chr>                <int> <dbl>    <dbl>
 1    104745 Rafael Nadal            69    60    0.870
 2    104925 Novak Djokovic          69    58    0.841
 3    103819 Roger Federer           66    55    0.833
 4    106421 Daniil Medvedev         80    59    0.738
 5    104731 Kevin Anderson          15    11    0.733
 6    106233 Dominic Thiem           69    50    0.725
 7    105226 Attila Balazs           10     7    0.7
 8    126774 Stefanos Tsitsipas      80    55    0.688
```

```
 9     200282 Alex De Minaur          62     42     0.677
10     105453 Kei Nishikori           43     29     0.674
# i 157 more rows
```

```
paste0("The player with the highest wub-rate is: ", win_rate$player_name[1])
```

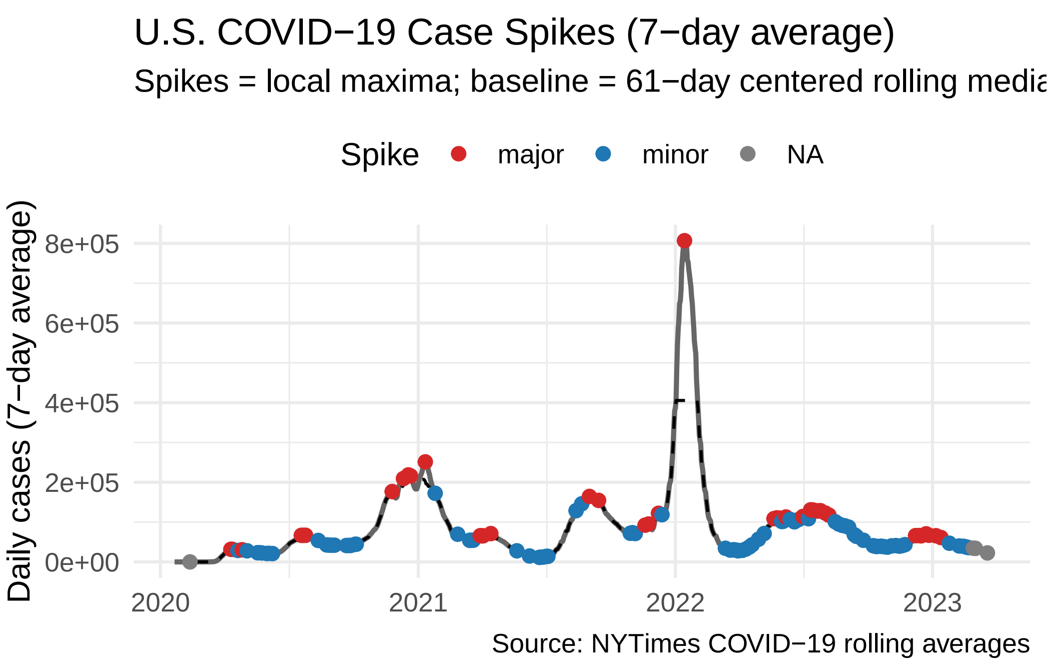
```
[1] "The player with the highest wub-rate is: Rafael Nadal"
```

Question 3

Part a

```
# Load Dataset
covid <- read_csv("https://raw.githubusercontent.com/nytimes/covid-19-data/refs/heads/master/

# Identify Spikes
k <- 61  # centered rolling window length (~±30 days)

covid_peaks <- covid %>%
  arrange(date) %>%
  mutate(
    # local maxima
    is_peak  = cases_avg > dplyr::lag(cases_avg) & cases_avg > dplyr::lead(cases_avg),
    # 61-day centered rolling median as baseline
    base_med = zoo::rollmedian(cases_avg, k = k, fill = NA, align = "center"),
    prominence = pmax(cases_avg - base_med, 0)
  ) %>%
  filter(is_peak) %>%
  mutate(
    thresh = quantile(prominence, 2/3, na.rm = TRUE),
    spike_type = if_else(prominence >= thresh, "major", "minor")
  )

ggplot(covid, aes(date, cases_avg)) +
  geom_line(linewidth = 0.9, color = "gray40") +
  geom_line(aes(y = zoo::rollmedian(cases_avg, k = 61, fill = NA, align = "center")),
            color = "black", linetype = "dashed", na.rm = TRUE) +
  geom_point(data = covid_peaks, aes(y = cases_avg, color = spike_type), size = 2) +
  scale_color_manual(values = c(major = "#d62728", minor = "#1f77b4")) +
```

```
  labs(
    title = "U.S. COVID-19 Case Spikes (7-day average)",
    subtitle = "Spikes = local maxima; baseline = 61-day centered rolling median; prominence
    x = NULL, y = "Daily cases (7-day average)",
    color = "Spike",
    caption = "Source: NYTimes COVID-19 rolling averages"
  ) +
  theme_minimal(base_size = 12) +
  theme(legend.position = "top")
```



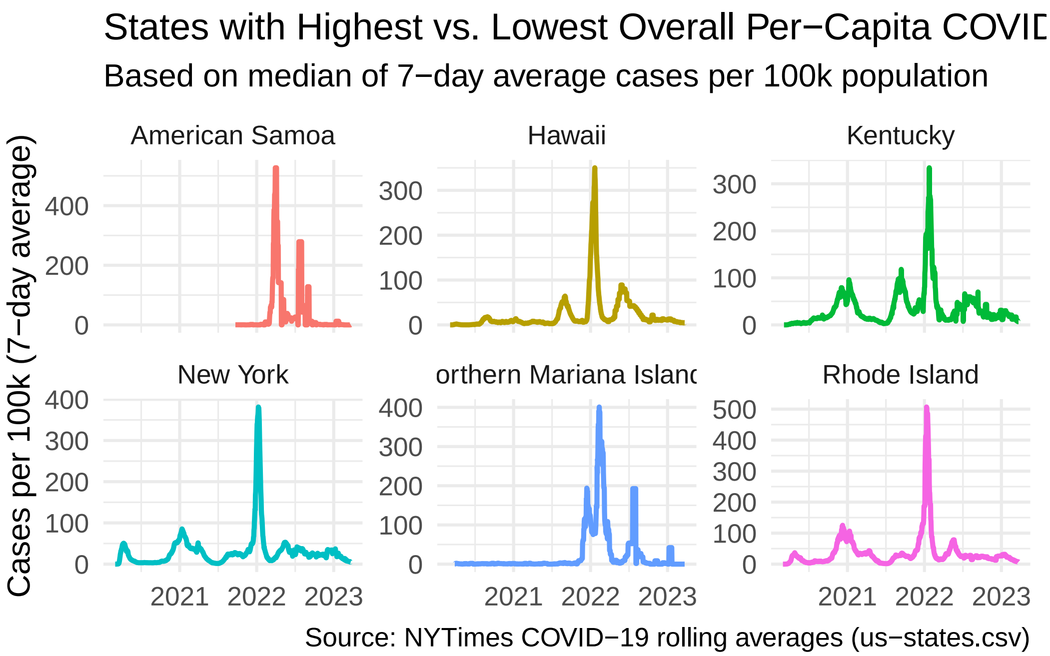
There're roughly 5 major spikes:

1. Spring 2020
2. Winter 2020 - Spring 2021
3. Summer 2021
4. Winter 2021 - Spring 2022
5. Summer 2022

Part b

```
# Load Dataset
covid_states <- read_csv("https://raw.githubusercontent.com/nytimes/covid-19-data/refs/heads/
```

```
                          show_col_types = FALSE)

# Compute overall (median) per-capita rate per state
state_rate <- covid_states %>%
  group_by(state) %>%
  summarise(
    overall_rate = median(cases_avg_per_100k, na.rm = TRUE),
    .groups = "drop"
  ) %>%
  arrange(desc(overall_rate))

# Pick top and bottom 3 states
top_states <- state_rate %>% slice_head(n = 3) %>% pull(state)
bottom_states <- state_rate %>% slice_tail(n = 3) %>% pull(state)

compare_states <- covid_states %>%
  filter(state %in% c(top_states, bottom_states))

ggplot(compare_states, aes(date, cases_avg_per_100k, color = state)) +
  geom_line(linewidth = 0.9) +
  facet_wrap(~ state, ncol = 3, scales = "free_y") +
  labs(
    title = "States with Highest vs. Lowest Overall Per-Capita COVID-19 Rates",
    subtitle = "Based on median of 7-day average cases per 100k population",
    x = NULL, y = "Cases per 100k (7-day average)",
    caption = "Source: NYTimes COVID-19 rolling averages (us-states.csv)"
  ) +
  theme_minimal(base_size = 12) +
  theme(legend.position = "none")
```



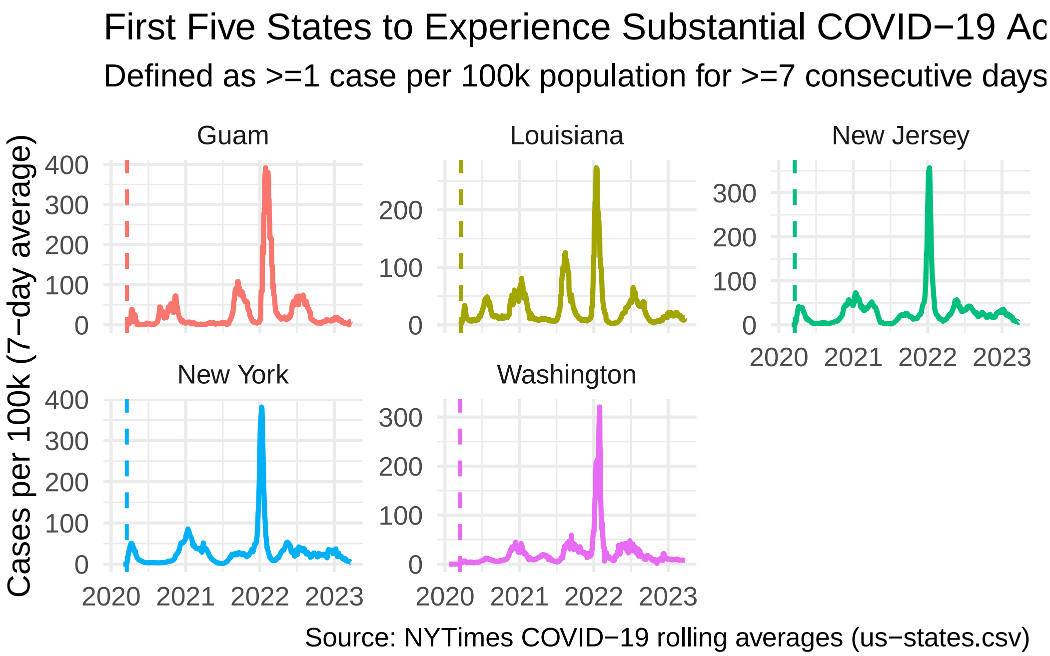
High-rate areas such as `American Samoa`, `Hawaii`, and `Kentucky` show sharp, concentrated peaks, indicating intense but relatively short-lived outbreaks. In contrast, low-rate areas like `New York`, `Northern Mariana Islands`, and `Rhode Island` exhibit lower, broader, or more irregular curves, suggesting more prolonged but less severe transmission.

Overall, the trajectories demonstrate that per-capita intensity and outbreak duration varied greatly across regions, reflecting differences in timing, containment policies, and population density.

Part c

```
# ----- Define "substantial" period -----
threshold <- 1.0    # cases per 100k
min_days  <- 7      # consecutive days

# For each state, find first sustained threshold run
first_substantial <- covid_states %>%
  arrange(state, date) %>%
  group_by(state) %>%
  mutate(
    above = cases_avg_per_100k >= threshold,
    run   = data.table::rleid(above)
  ) %>%
```

```r
  group_by(state, run, .add = TRUE) %>%
  summarise(
    start_date = first(date),
    end_date   = last(date),
    days       = n(),
    above      = first(above),
    .groups = "drop_last"
  ) %>%
  ungroup() %>%
  filter(above, days >= min_days) %>%
  group_by(state) %>%
  summarise(first_substantial_date = min(start_date), .groups = "drop") %>%
  arrange(first_substantial_date)

# First 5 states
first5 <- first_substantial %>% slice_head(n = 5)

# Plot
early_states <- covid_states %>%
  filter(state %in% first5$state)

ggplot(early_states, aes(date, cases_avg_per_100k, color = state)) +
  geom_line(linewidth = 0.9) +
  geom_vline(
    data = first5,
    aes(xintercept = first_substantial_date, color = state),
    linetype = "dashed", linewidth = 0.7
  ) +
  facet_wrap(~ state, ncol = 3, scales = "free_y") +
  labs(
    title = "First Five States to Experience Substantial COVID-19 Activity",
    subtitle = "Defined as  1 case per 100k population for  7 consecutive days",
    x = NULL, y = "Cases per 100k (7-day average)",
    caption = "Source: NYTimes COVID-19 rolling averages (us-states.csv)"
  ) +
  theme_minimal(base_size = 12) +
  theme(legend.position = "none")
```



The first five states (or territories) to experience substantial COVID-19 activity were `Guam`, `Louisiana`, `New Jersey`, `New York`, and `Washington`, with outbreaks emerging around March 2020, marking the start of widespread community transmission in the United States.