STATS 506 Problem Set 4

Lingzhi Hao

GitHub Repository:
https://github.com/Lingzhi-Hao/STATS-506-Problem-Set-04

Problem 1 - Tidyverse: New Zealand

(a)

```
library(nzelect)
library(dplyr)
```

```
Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```
library(tidyr)

nzge %>%
  group_by(election_year, voting_type) %>%
  summarise(vote_count = sum(votes, na.rm = TRUE)) %>%
  arrange(desc(vote_count))
```

```
`summarise()` has grouped output by 'election_year'. You can override using the
`.groups` argument.
```

```
# A tibble: 10 x 3
# Groups:   election_year [5]
   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
```

(b)

```r
nzge %>%
  filter(election_year == 2014, voting_type == "Candidate") %>%
  group_by(party) %>%
  summarise(party_vote_count = sum(votes, na.rm = TRUE)) %>%
  mutate(vote_percentage = party_vote_count/sum(party_vote_count, na.rm = TRUE)*100) %>%
  arrange(desc(vote_percentage))
```

```
# A tibble: 25 x 3
   party                     party_vote_count vote_percentage
   <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
```

(c)

```r
nzge %>%
  group_by(election_year, voting_type) %>%
  slice_max(order_by = votes, n = 1, with_ties = FALSE) %>%
  select(election_year, voting_type, party) %>%
  pivot_wider(names_from = voting_type, values_from = party, names_prefix = "Winner_of_")
```

```
# A tibble: 5 x 3
# Groups:   election_year [5]
  election_year Winner_of_Candidate Winner_of_Party
          <dbl> <chr>               <chr>
1          2002 Labour Party        Labour Party
2          2005 Labour Party        Labour Party
3          2008 National Party      National Party
4          2011 National Party      National Party
5          2014 National Party      National Party
```

Problem 2 - Tidyverse: Tennis

```r
ATP_Matches = read.csv("https://raw.githubusercontent.com/JeffSackmann/tennis_atp/refs/heads/
```

(a) *128* tournaments are in the *atp_matches_2019* dataset.

```r
ATP_Matches %>%
  summarise(n_distinct(tourney_id))
```

```
  n_distinct(tourney_id)
1                    128
```

If strictly limiting the tournament date in year 2019, there were *125* tournaments taking place in 2019.

```r
ATP_Matches %>%
  filter(tourney_date >= 20190101 & tourney_date <= 20191231) %>%
  summarise(n_distinct(tourney_id))
```

```
  n_distinct(tourney_id)
1                    125
```

In the following problems (b) (c) (d), all 128 tournaments in the dataset are considered.

(b) *12* players won more than one tournaments. The most winning players Dominic Thiem and Novak Djokovic both won *5* tournaments.

```
ATP_Matches %>%
  filter(round == "F") %>%
  select(tourney_id, winner_name) %>%
  group_by(winner_name) %>%
  count(winner_name, sort = TRUE) %>%
  filter(n >= 2)
```

```
# A tibble: 12 x 2
# Groups:   winner_name [12]
   winner_name            n
   <chr>              <int>
 1 Dominic Thiem          5
 2 Novak Djokovic         5
 3 Daniil Medvedev        4
 4 Rafael Nadal           4
 5 Roger Federer          4
 6 Alex De Minaur         3
 7 Stefanos Tsitsipas     3
 8 Benoit Paire           2
 9 Cristian Garin         2
10 Jo-Wilfried Tsonga     2
11 Matteo Berrettini      2
12 Nick Kyrgios           2
```

(c) There is evidence that winners have more aces than losers. To test this, a one-sample t-test is conducted on the difference of winner's aces and loser's aces for every match.

Null hypothesis: The difference in aces between winners and losers is 0;

Alternative hypothesis: The difference in aces between winners and losers is not 0.

Since p-value = 3.060403e-34 « 0.0001, t-statistic = 12.37302, estimate = 1.7049, the Null hypothesis is rejected. Winners significantly have more aces than losers. Winners have an average of 1.7049 more aces per match than losers.

```
library(dplyr)
library(tidyr)
library(infer)
```

```r
ATP_Matches %>%
  transmute(ace_diff = w_ace - l_ace) %>%
  drop_na() %>%
  infer::t_test(
    response = ace_diff,
    mu = 0,
    alternative = "two_sided"
  )
```

```
# A tibble: 1 x 7
  statistic  t_df  p_value alternative estimate lower_ci upper_ci
      <dbl> <dbl>    <dbl> <chr>          <dbl>    <dbl>    <dbl>
1      12.4  2693 3.06e-34 two.sided       1.70     1.43     1.98
```

(d) The player (at least 5 matches) with highest win-rate is *Rafael Nadal*.

```r
ATP_Matches %>%
  select(winner_name, loser_name) %>%
  pivot_longer(
    cols = everything(),
    names_to = "outcome",
    values_to = "player_name"
  ) %>%
  group_by(player_name) %>%
  summarise(
    total = n(),
    wins = sum(outcome == "winner_name"),
    .groups = "drop"
    ) %>%
  mutate(win_rate = wins/total) %>%
  filter(total >= 5) %>%
  slice_max(order_by = win_rate, with_ties = TRUE)
```

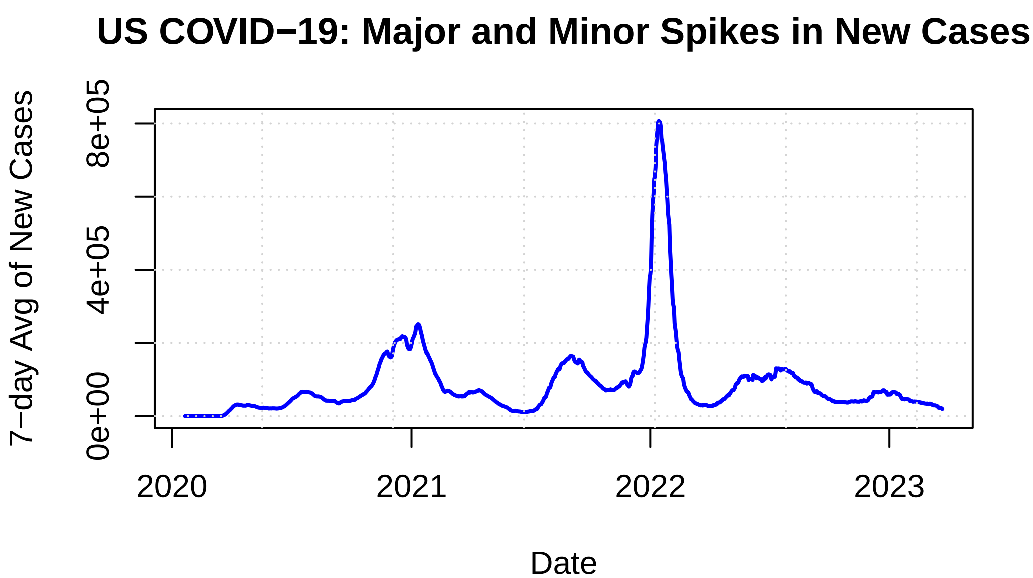
```
# A tibble: 1 x 4
  player_name  total  wins win_rate
  <chr>        <int> <int>    <dbl>
1 Rafael Nadal    69    60    0.870
```

Problem 3 - Visualization

```r
covid <- read.csv("https://raw.githubusercontent.com/nytimes/covid-19-data/refs/heads/master/

covid$date <- as.Date(covid$date)
```

(a) There were *1* major spike and *5* minor spikes.

```r
us <- aggregate(cases_avg ~ date, data = covid, FUN = sum)

plot(us$date, us$cases_avg, type = "l", col = "blue", lwd = 2,
     main = "US COVID-19: Major and Minor Spikes in New Cases",
     xlab = "Date", ylab = "7-day Avg of New Cases")

grid()
```

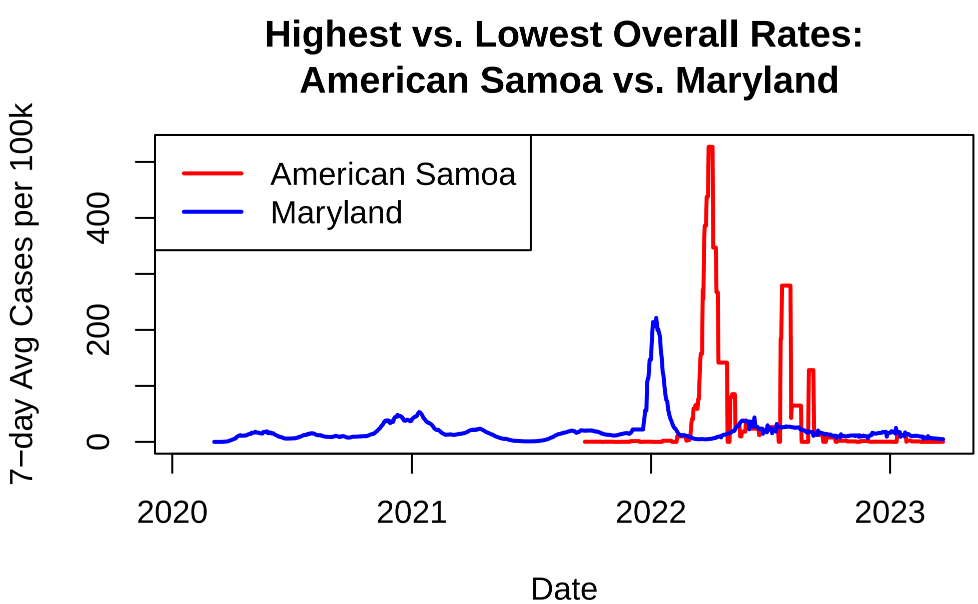


(b) The main differences in the trajectories over time of the highest-rate state (American Samoa) and lowest-rate state (Maryland) are the time span and intensity.

The highest-rate state (American Samoa): It had a delayed and explosive trajectory, with very few but massive and sharp spikes.

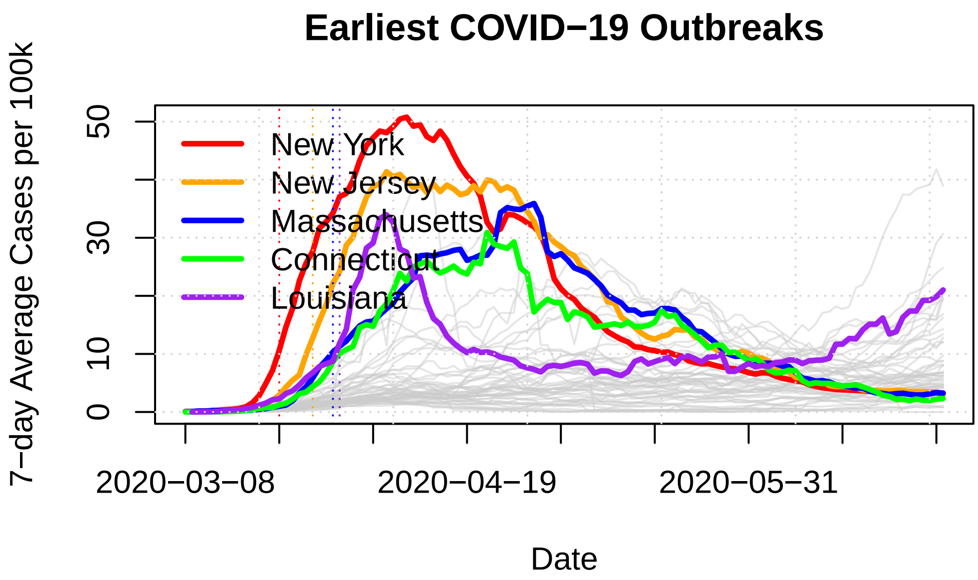
The lowest-rate state (Maryland): It had an immediate and very long-term trajectory (from the start of US COVID to the end), with numerous but much lower waves.

```r
mean_rate <- tapply(covid$cases_avg_per_100k, covid$state, mean, na.rm = TRUE)
high_state <- names(which.max(mean_rate))
low_state  <- names(which.min(mean_rate))

min_date <- min(covid$date, na.rm = TRUE)
max_date <- max(covid$date, na.rm = TRUE)

sel <- covid$state %in% c(high_state, low_state)

plot(covid$date[sel & covid$state == high_state],
     covid$cases_avg_per_100k[sel & covid$state == high_state],
     type = "l", col = "red", lwd = 2,
     xlab = "Date", ylab = "7-day Avg Cases per 100k",
     main = paste("Highest vs. Lowest Overall Rates:\n", high_state, "vs.", low_state),
     xlim = c(min_date, max_date))

lines(covid$date[sel & covid$state == low_state],
      covid$cases_avg_per_100k[sel & covid$state == low_state],
      col = "blue", lwd = 2)
legend("topleft", legend = c(high_state, low_state),
       col = c("red", "blue"), lwd = 2)
```



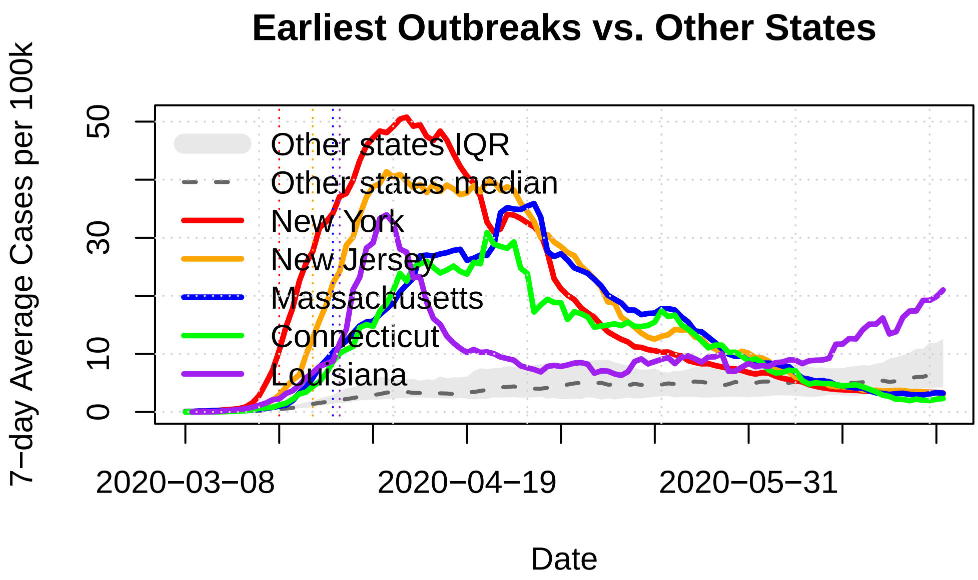
(c) I set the threshold = 10, which means first five states reaching10 new cases per 100k population are the first five states to experience COVID in a substantial way. The first five states are New York, New Jersey, Massachusetts, Connecticut, and Louisiana.

```r
threshold <- 10
first_date <- tapply(covid$date[covid$cases_avg_per_100k >= threshold],
                     covid$state[covid$cases_avg_per_100k >= threshold],
                     min)
first_date <- sort(first_date)
first5_states <- names(head(first_date, 5))
first5_dates  <- as.Date(first_date[first5_states])


start_date <- min(first5_dates) - 14
end_date   <- max(first5_dates) + 90
in_win <- covid$date >= start_date & covid$date <= end_date

plot(NA,
     xlim = c(start_date, end_date),
     ylim = c(0, max(covid$cases_avg_per_100k[in_win], na.rm = TRUE)),
     xlab = "Date", ylab = "7-day Average Cases per 100k",
     main = "Earliest COVID-19 Outbreaks",
     xaxt = "n")

for (st in unique(covid$state)) {
  idx <- in_win & covid$state == st
  lines(covid$date[idx], covid$cases_avg_per_100k[idx],
        col = rgb(0.8, 0.8, 0.8, 0.5))
}

cols <- c("red","orange","blue","green","purple")
for (i in seq_along(first5_states)) {
  st <- first5_states[i]
  idx <- in_win & covid$state == st
  lines(covid$date[idx], covid$cases_avg_per_100k[idx], col = cols[i], lwd = 2.8)

  abline(v = first5_dates[i], col = cols[i], lty = 3)
}
axis.Date(1, at = seq(start_date, end_date, by = "2 weeks"), format = "%Y-%m-%d")
legend("topleft", legend = first5_states, col = cols, lwd = 2.8, bty = "n")
grid()
```




```
threshold <- 10
first_date <- tapply(covid$date[covid$cases_avg_per_100k >= threshold],
                     covid$state[covid$cases_avg_per_100k >= threshold],
                     min)
first_date <- sort(first_date)
first5_states <- names(head(first_date, 5))
first5_dates  <- as.Date(first_date[first5_states])


start_date <- min(first5_dates) - 14
end_date   <- max(first5_dates) + 90
in_win <- covid$date >= start_date & covid$date <= end_date
other <- covid[in_win & !(covid$state %in% first5_states),
              c("date","cases_avg_per_100k")]

med_by_day <- aggregate(cases_avg_per_100k ~ date, data = other, median, na.rm = TRUE)
q1_by_day  <- aggregate(cases_avg_per_100k ~ date, data = other,
                        function(x) quantile(x, 0.25, na.rm = TRUE))
q3_by_day  <- aggregate(cases_avg_per_100k ~ date, data = other,
                        function(x) quantile(x, 0.75, na.rm = TRUE))

plot(NA,
     xlim = c(start_date, end_date),
```

```r
     ylim = c(0, max(covid$cases_avg_per_100k[in_win], na.rm = TRUE)),
     xlab = "Date", ylab = "7-day Average Cases per 100k",
     main = "Earliest Outbreaks vs. Other States",
     xaxt = "n")

polygon(c(q1_by_day$date, rev(q3_by_day$date)),
        c(q1_by_day$cases_avg_per_100k, rev(q3_by_day$cases_avg_per_100k)),
        col = rgb(0.7, 0.7, 0.7, 0.3), border = NA)
lines(med_by_day$date, med_by_day$cases_avg_per_100k, col = "gray40", lwd = 2, lty = 2)

cols <- c("red","orange","blue","green","purple")
for (i in seq_along(first5_states)) {
  st <- first5_states[i]
  idx <- in_win & covid$state == st
  lines(covid$date[idx], covid$cases_avg_per_100k[idx], col = cols[i], lwd = 2.8)
  abline(v = first5_dates[i], col = cols[i], lty = 3)
}

axis.Date(1, at = seq(start_date, end_date, by = "2 weeks"), format = "%Y-%m-%d")
legend("topleft",
       legend = c("Other states IQR", "Other states median", first5_states),
       col    = c(rgb(0.7,0.7,0.7,0.3), "gray40", cols),
       lwd    = c(10, 2, rep(2.8, length(first5_states))),
       lty    = c(1, 2, rep(1, length(first5_states))),
       bty = "n")
grid()
```


Earliest Outbreaks vs. Other States
7−day Average Cases per 100k
0
10
30
50
Other states IQR
Other states median
New York
New Jersey
Massachusetts
Connecticut
Louisiana
2020−03−08
2020−04−19
2020−05−31
Date