Lab 1: First visualization

Dhyey Mavani

Day 1

Instructions

You have two options for creating your first visualization in this course. The **UN Votes** option involes the voting history of countries in the United Nations General Assembly. The **Coronavirus** option involves the cumulative death toll due to coronavirus across reporting countries.

Use the outline button (top right) to jump down to the section you are interested in. For either option, you can customize the visualization by choosing countries of interest to you for the comparison.

Option 1 UN Votes

Let's take a look at the voting history of countries in the United Nations General Assembly. We will be using data from the **unvotes** package. Additionally, we will make use of the **tidyverse** and **lubridate** packages for the analysis, and the **DT** package for interactive display of tabular output.

The **unvotes** package provides three datasets we can work with: `un_roll_calls`, `un_roll_call_issues`, and `un_votes`. Each of these datasets contains a variable called `rcid`, the roll call id, which can be used as a unique identifier to join the three datasets together.

The `un_votes` dataset provides information on the voting history of the United Nations General Assembly. It contains one row for each country-vote pair.

```
head(un_votes)
```

```
# A tibble: 6 x 4
   rcid country            country_code vote
  <dbl> <chr>              <chr>        <fct>
1     3 United States      US           yes
2     3 Canada             CA           no
3     3 Cuba               CU           yes
4     3 Haiti              HT           yes
5     3 Dominican Republic DO           yes
6     3 Mexico             MX           yes
```

The `un_roll_calls` dataset contains information on each roll call vote of the United Nations General Assembly.

```
head(un_roll_calls)
```

```
# A tibble: 6 x 9
   rcid session importantvote date       unres   amend  para short    descr
  <int>   <dbl>         <int> <date>     <chr>   <int> <int> <chr>    <chr>
1     3       1             0 1946-01-01 R/1/66      1     0 AMENDME~ "TO ADOPT~
2     4       1             0 1946-01-02 R/1/79      0     0 SECURIT~ "TO ADOPT~
3     5       1             0 1946-01-04 R/1/98      0     0 VOTING ~ "TO ADOPT~
4     6       1             0 1946-01-04 R/1/107     0     0 DECLARA~ "TO ADOPT~
5     7       1             0 1946-01-02 R/1/295     1     0 GENERAL~ "TO ADOPT~
6     8       1             0 1946-01-05 R/1/297     1     0 ECOSOC ~ "TO ADOPT~
```

The `un_roll_call_issues` dataset contains (topic) classifications of roll call votes of the United Nations General Assembly. Many votes had no topic, and some have more than one.

```
head(un_roll_call_issues)
```

```
# A tibble: 6 x 3
   rcid short_name issue
  <int> <chr>      <fct>
1    77 me         Palestinian conflict
2  9001 me         Palestinian conflict
3  9002 me         Palestinian conflict
4  9003 me         Palestinian conflict
5  9004 me         Palestinian conflict
6  9005 me         Palestinian conflict
```

1.1 Data prep

In order to do our analysis, we first need to combine our three datasets into one.

```
unvotes <- un_votes %>%
  inner_join(un_roll_calls, by = "rcid") %>%
  inner_join(un_roll_call_issues, by = "rcid")
```

1.2 Visualization

Now we can create a visualization that displays how the voting record changed over time across countries on six broader issues. *Modify the code below with a list of countries of interest to you!* Note that the country name should be spelled and capitalized exactly the same way as it appears in the data. A full list of the countries in the data frame is provided in the UN country list at the end of this section.

```
unvotes %>%
  filter(country %in% c("United States",
                        "Brazil",
                        "France")) %>%
  group_by(country, year = year(date), issue) %>%
  summarize(votes = n(),
            percent_yes = mean(vote == "yes")) %>%
  filter(votes > 5) %>%  # only use records where there are more than 5 votes
  ggplot(mapping = aes(x = year,
                       y = percent_yes,
                       color = country)) +
    geom_point() +
    geom_smooth(method = "loess", se = FALSE) +
    facet_wrap(~ issue) +
    labs(title = "Percentage of 'Yes' votes in the UN General Assembly",
         subtitle = "1946 to 2019",
         y = "% Yes",
         x = "Year",
         color = "Country")
```

1.3 References

1. David Robinson (2017). unvotes: United Nations General Assembly Voting Data. R package version 0.2.0. https://CRAN.R-project.org/package=unvotes.
2. Erik Voeten "Data and Analyses of Voting in the UN General Assembly" Routledge Handbook of International Organization, edited by Bob Reinalda (published May 27, 2013).
3. Much of the analysis has been modeled on the examples presented in the unvotes package vignette.

1.4 UN country list

Below is a list of countries in the dataset:

```
un_votes %>%
  select(country) %>%
  arrange(country) %>%
```

```
  distinct() %>%
  datatable()
```

Option 2 Coronavirus

In this section we explore how the trajectory of the cumulative COVID-19 cases differs across a number of countries.

The data come from the **coronavirus** package, which pulls data from the Johns Hopkins University Center for Systems Science and Engineering (JHU CCSE) Coronavirus repository. The coronavirus package provides a tidy format dataset of the 2019 Novel Coronavirus COVID-19 (2019-nCoV) epidemic. The package is available on GitHub here and is updated daily.

For our analysis, in addition to the coronavirus package, we will use the following packages for data wrangling and visualization.

- **tidyverse** for data wrangling and visualization
- **lubridate** package for handling dates
- **glue** package for constructing text strings
- **scales** package for formatting axis labels
- **ggrepel** package for pretty printing of country labels

We will make use of the **DT** package for interactive display of tabular output in the Appendix.

2.1 Data prep

The data frame called `coronavirus` in the coronavirus package provides a daily summary of the Coronavirus (COVID-19) cases by country. Each row in the data frame represents a country (or, where relevant, state/province).

```
head(coronavirus)
```

```
        date province             country       lat      long      type cases
1 2020-01-22                 Afghanistan  33.93911  67.70995 confirmed     0
2 2020-01-22                     Albania  41.15330  20.16830 confirmed     0
3 2020-01-22                     Algeria  28.03390   1.65960 confirmed     0
4 2020-01-22                     Andorra  42.50630   1.52180 confirmed     0
5 2020-01-22                      Angola -11.20270  17.87390 confirmed     0
6 2020-01-22          Antigua and Barbuda  17.06080 -61.79640 confirmed     0
```

Note that the data provided in this package provides daily number of deaths, confirmed cases, and recovered cases. For this report, we will focus on the confirmed cases.

We will start by making our selection for the countries we want to explore. *Modify the code below with a list of countries of interest to you!* Note that the country name should be spelled and capitalized exactly the same way as it appears in the data. A full list of the countries in the data frame is provided in the country list at the end of this section.

```
countries <- c("China",
               "India",
               "United Kingdom",
               "US",
               "South Africa")
```

In the following code chunk we filter the data frame for confirmed cases in the countries we specified above and calculate cumulative number of confirmed cases. We will only visualize data since 10th confirmed case.

```
country_data <- coronavirus %>%
  # filter for confirmed cases in countries of interest
  filter(type == "confirmed",
         country %in% countries) %>%
  # fix country labels for pretty plotting
  mutate(country = case_when(country == "United Kingdom" ~ "UK",
                             TRUE ~ country)) %>%
  # calculate number of total cases for each country and date
  group_by(country, date) %>%
  summarize(tot_cases = sum(cases)) %>%
  # arrange by date in ascending order
  arrange(date) %>%
  # record daily cumulative cases as cumulative_cases
  mutate(cumulative_cases = cumsum(tot_cases)) %>%
  # only use days since the 10th confirmed case
  filter(cumulative_cases > 9) %>%
  # record days elapsed, end date, and end label
  mutate(days_elapsed = as.numeric(date - min(date)),
         end_date = if_else(date == max(date), TRUE, FALSE),
         end_label = if_else(end_date, country, NULL)) %>%
  # ungroup
  ungroup()
```

We also need to take a note of the "as of date" for the data so that we can properly label our visualization.

```
as_of_date <- country_data %>%
  summarize(max(date)) %>%
  pull()
as_of_date_formatted <- glue("{wday(as_of_date, label = TRUE)}, \\
                             {month(as_of_date, label = TRUE)} \\
                             {day(as_of_date)}, {year(as_of_date)}")
```

These data are as of Thu, May 27, 2021.

2.2 Visualization

The following visualization shows the number of cumulative cases vs. days elapsed since the 10th confirmed case in each country. The time span plotted for each country varies since some countries started seeing (and reporting) cases from COVID-19 much later than others.

```
ggplot(data = country_data,
       mapping = aes(x = days_elapsed,
                     y = cumulative_cases,
                     color = country,
                     label = end_label)) +
  # represent cumulative cases with lines
  geom_line(size = 0.7, alpha = 0.8) +
  # add points to line endings
  geom_point(data = country_data %>%
               filter(end_date)) +
  # add country labels, nudged above the lines
  geom_label_repel(nudge_y = 1, direction = "y", hjust = 1) +
```

```
  # turn off legend
  guides(color = FALSE) +
  # use pretty colors
  scale_color_viridis_d() +
  # better formatting for y-axis
  scale_y_continuous(labels = label_comma()) +
  # use minimal theme
  theme_minimal() +
  # customize labels
  labs(x = "Days since 10th confirmed case",
       y = "Cumulative number of cases",
       title = "Cumulative cases from COVID-19, selected countries",
       subtitle = glue("Data as of", as_of_date_formatted, .sep = " "),
       caption = "Source: github.com/RamiKrispin/coronavirus")
```

2.3 Country list

A list of countries in the `coronavirus` data frame is provided below.

```
coronavirus %>%
  select(country) %>%
  arrange(country) %>%
  distinct() %>%
  datatable()
```