AE 01 - UN Votes

Max Asmar

2026-02-15

Contents

Introduction

How do various countries vote in the United Nations General Assembly, how have their voting patterns evolved throughout time, and how similarly or differently do they view certain issues? Answering these questions (at a high level) is the focus of this analysis.

In this application exercise, you will:

- Practice the fork-then-clone workflow
- Load and explore data from an R package
- Join multiple datasets together
- Create visualizations with ggplot2
- Use `group_by()` and `summarize()` to calculate percentages
- Interpret trends over time

Estimated time: 30-45 minutes

Getting Started

Fork and Clone Workflow

IMPORTANT: Application exercises use the same fork-then-clone workflow as labs. Here's how to get started:

Step 1: Fork the Repository

1. Go to the course application exercises repository on GitHub: `[CourseOrg]/application-exercises`
2. Click the **Fork** button in the top-right corner
3. This creates YOUR OWN copy: `YourUsername/application-exercises`
4. You now have full permissions to push to your fork

Why fork? - Creates your personal copy under your GitHub account - Gives you permission to push your work - Keeps your work separate from other students - Allows instructors to see your individual progress

Step 2: Clone Your Fork to JupyterHub

1. In JupyterHub, open RStudio
2. Click **File → New Project → Version Control → Git**
3. **Repository URL:** Paste the URL of **YOUR forked repository**
   - Your fork: `https://github.com/YourUsername/application-exercises`
   - NOT the course repo: `https://github.com/CourseOrg/application-exercises`
4. **Project directory name:** Leave as `application-exercises`
5. **Create project as subdirectory of:** Choose where you want it (default is fine)
6. Click **Create Project**

Step 3: Verify Your Setup **Check that you're in YOUR fork:** - Look at the **Git pane** in RStudio - The remote should show: `YourUsername/application-exercises` - If it shows the course organization name, you cloned the wrong repository!

If you cloned the course repository by mistake: - Delete the project folder - Start over with Step 1 (Fork first!)

Step 4: Navigate to This Exercise

1. In RStudio's **Files** pane, navigate to the `ae-01-un-votes` folder
2. Open `ae-01-un-votes.Rmd`
3. Update the YAML with your name
4. Click **Knit** to make sure it works

Step 5: Commit and Push After completing the exercise (or periodically as you work):

1. Click the **Git** pane
2. Check the boxes next to changed files
3. Click **Commit**
4. Write a message (e.g., "Completed UN Votes exercise")
5. Click **Commit** then **Push**
6. Verify your work appears in **YOUR fork** on GitHub

Remember: You're pushing to `YourUsername/application-exercises`, NOT the course repository!

Packages

We will use the **tidyverse**, **lubridate**, and **scales** packages for data wrangling and visualization, the **DT** package for interactive display of tabular output, and the **unvotes** package for the data.

```
library(tidyverse)
library(lubridate)
library(scales)
library(DT)
library(unvotes)
```

What these packages do: - **tidyverse:** Data wrangling (dplyr) and visualization (ggplot2) - **lubridate:** Working with dates and times - **scales:** Formatting axis labels (like percentages) - **DT:** Creating interactive tables - **unvotes:** Contains UN voting data

---

Data

The data we're using originally come from the **unvotes** package. In the chunk below we modify the data by joining the various data frames provided in the package to help you get started with the analysis.

```
unvotes <- un_votes %>%
  inner_join(un_roll_calls, by = "rcid") %>%
  inner_join(un_roll_call_issues, by = "rcid") %>%
  mutate(date = as.Date(date))  # Convert to proper Date format
```

```
## Warning in inner_join(., un_roll_call_issues, by = "rcid"): Detected an unexpected many-to-many rela
## i Row 382 of `x` matches multiple rows in `y`.
## i Row 3009 of `y` matches multiple rows in `x`.
## i If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

What this code does: - `un_votes`: Contains how each country voted on each resolution - `un_roll_calls`: Contains information about each resolution (date, description) - `un_roll_call_issues`: Contains the issue category for each resolution - `inner_join()`: Combines these datasets based on the resolution ID (`rcid`) - `mutate(date = as.Date(date))`: Converts date to proper Date format - Result: One dataset with votes, dates, and issue categories

Explore the data:

```
# View the structure
glimpse(unvotes)
```

```
## Rows: 857,878
## Columns: 14
## $ rcid          <dbl> 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6,~
## $ country       <chr> "United States", "Canada", "Cuba", "Dominican Republic",~
## $ country_code  <chr> "US", "CA", "CU", "DO", "MX", "GT", "HN", "SV", "NI", "P~
## $ vote          <fct> no, no, yes, abstain, yes, no, yes, abstain, yes, abstai~
## $ session       <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
## $ importantvote <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ date          <date> 1946-01-04, 1946-01-04, 1946-01-04, 1946-01-04, 1946-01~
## $ unres         <chr> "R/1/107", "R/1/107", "R/1/107", "R/1/107", "R/1/107", "~
## $ amend         <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ para          <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ short         <chr> "DECLARATION OF HUMAN RIGHTS", "DECLARATION OF HUMAN RIG~
## $ descr         <chr> "TO ADOPT A CUBAN PROPOSAL (A/3-C) THAT AN ITEM ON A DEC~
```

```
## $ short_name    <chr> "hr", "hr", "hr", "hr", "hr", "hr", "hr", "hr", "hr", "h~
## $ issue         <fct> Human rights, Human rights, Human rights, Human rights, ~
```

```
# See the first few rows
head(unvotes)
```

```
## # A tibble: 6 x 14
##    rcid country  country_code vote  session importantvote date       unres amend
##   <dbl> <chr>    <chr>        <fct>   <dbl>         <int> <date>     <chr> <int>
## 1     6 United ~ US           no          1             0 1946-01-04 R/1/~     0
## 2     6 Canada   CA           no          1             0 1946-01-04 R/1/~     0
## 3     6 Cuba     CU           yes         1             0 1946-01-04 R/1/~     0
## 4     6 Dominic~ DO           abst~       1             0 1946-01-04 R/1/~     0
## 5     6 Mexico   MX           yes         1             0 1946-01-04 R/1/~     0
## 6     6 Guatema~ GT           no          1             0 1946-01-04 R/1/~     0
## # i 5 more variables: para <int>, short <chr>, descr <chr>, short_name <chr>,
## #   issue <fct>
```

Key variables: - `country`: Name of the country - `vote`: How they voted (yes, no, abstain) - `date`: Date of the vote - `issue`: Category of the resolution (e.g., Human rights, Colonialism)

Exercise 1: Explore the Data

Before creating visualizations, let's explore the data to understand what we're working with.

Task 1.1: How many unique countries are in the dataset?

```
# Count unique countries
unvotes %>%
  distinct(country) %>%
  nrow()
```

```
## [1] 200
```

Answer: There are 200 countries in the dataset.

Task 1.2: What are the unique vote options?

```
# See unique values of vote
unvotes %>%
  distinct(vote)
```

```
## # A tibble: 3 x 1
##   vote
##   <fct>
## 1 no
## 2 yes
## 3 abstain
```

Answer: The vote options are: no, yes, and abstain.

Task 1.3: What issue categories are included?

```
# See unique issues
unvotes %>%
  distinct(issue)
```

```
## # A tibble: 6 x 1
##   issue
```

```
##   <fct>
## 1 Human rights
## 2 Economic development
## 3 Colonialism
## 4 Palestinian conflict
## 5 Arms control and disarmament
## 6 Nuclear weapons and nuclear material
```

Answer: The issue categories are: Human rights, Economic development, Colonialism, Palestinian Conflict, Arms control and disarmament, Nuclear weapons and nuclear material

---

UN Voting Patterns

Now let's create a data visualization that displays how the voting record of the UK & NI changed over time on a variety of issues, and compares it to two other countries: US and Turkey.

Understanding the Code

Let's break down what the code below does, step by step:

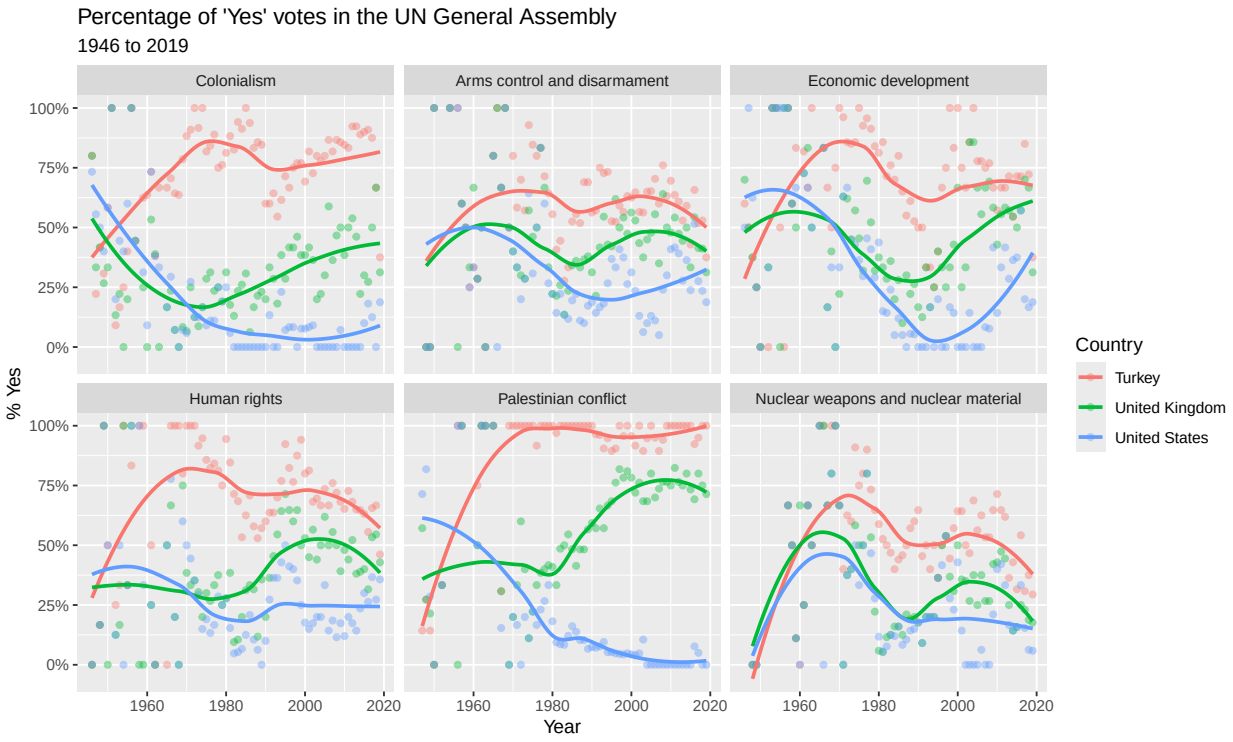
```
unvotes %>%
  # Step 1: Filter for specific countries
  filter(country %in% c("United States", "United Kingdom", "Turkey")) %>%

  # Step 2: Extract the year from the date
  mutate(year = year(date)) %>%

  # Step 3: Group by country, year, and issue
  group_by(country, year, issue) %>%

  # Step 4: Calculate percentage of "yes" votes
  summarize(percent_yes = mean(vote == "yes")) %>%

  # Step 5: Create the plot
  ggplot(mapping = aes(x = year, y = percent_yes, color = country)) +
  geom_point(alpha = 0.4) +
  geom_smooth(method = "loess", se = FALSE) +
  facet_wrap(~issue) +
  scale_y_continuous(labels = percent) +
  labs(
    title = "Percentage of 'Yes' votes in the UN General Assembly",
    subtitle = "1946 to 2019",
    y = "% Yes",
    x = "Year",
    color = "Country"
  )
```



Code explanation:

1. **filter():** Keeps only rows for UK, US, and Turkey
2. **mutate():** Creates a new `year` variable by extracting the year from the `date`
3. **group_by():** Groups data by country, year, and issue
4. **summarize():** For each group, calculates the percentage of "yes" votes
   - `mean(vote == "yes")` works because TRUE = 1 and FALSE = 0
5. **ggplot():** Creates a scatterplot with smoothed trend lines
   - `geom_point()`: Adds points
   - `geom_smooth()`: Adds trend lines
   - `facet_wrap()`: Creates separate panels for each issue
   - `scale_y_continuous()`: Formats y-axis as percentages

Exercise 2: Interpret the Visualization

Task 2.1: Describe the overall trend for the United Kingdom on Human Rights issues.

Your answer:

From the 1940s to the mid 1980s, the United Kingdom (U.K.) remained relatively consistent in the percentage of "yes" votes that it made towards Human Rights issues. During this time period, the U.K. voted yes around 30% of the time. From the mid 1980s to the mid 2000s, the percentage of "yes" votes increased and hit a maximum of around 50%. After the mid 2000s, there were less "yes" votes and it leveled off to around 37% by 2020.

Task 2.2: Which country appears most consistent in its voting patterns across all issues? Which appears least consistent?

Your answer:

I think the United Kingdom appears most consistent in its voting patterns out of the three countries. In

comparison, I think Turkey is the least consistent in its voting patterns out of the three countries. I am making these observations based on final percentage - initial percentage of yes votes.

Task 2.3: On which issue do the three countries show the most similar voting patterns? On which issue do they differ the most?

Your answer:

Most similar: Arms Control and Disarmament

Most different: Palestinian Conflict

Exercise 3: Modify the Countries

We can easily change which countries are being plotted by changing which countries the code `filter`s for.

Note: The country name should be spelled and capitalized exactly the same way as it appears in the data. See the Appendix for a list of the countries in the data.

Task 3.1: Modify the code to compare three different countries of your choice.

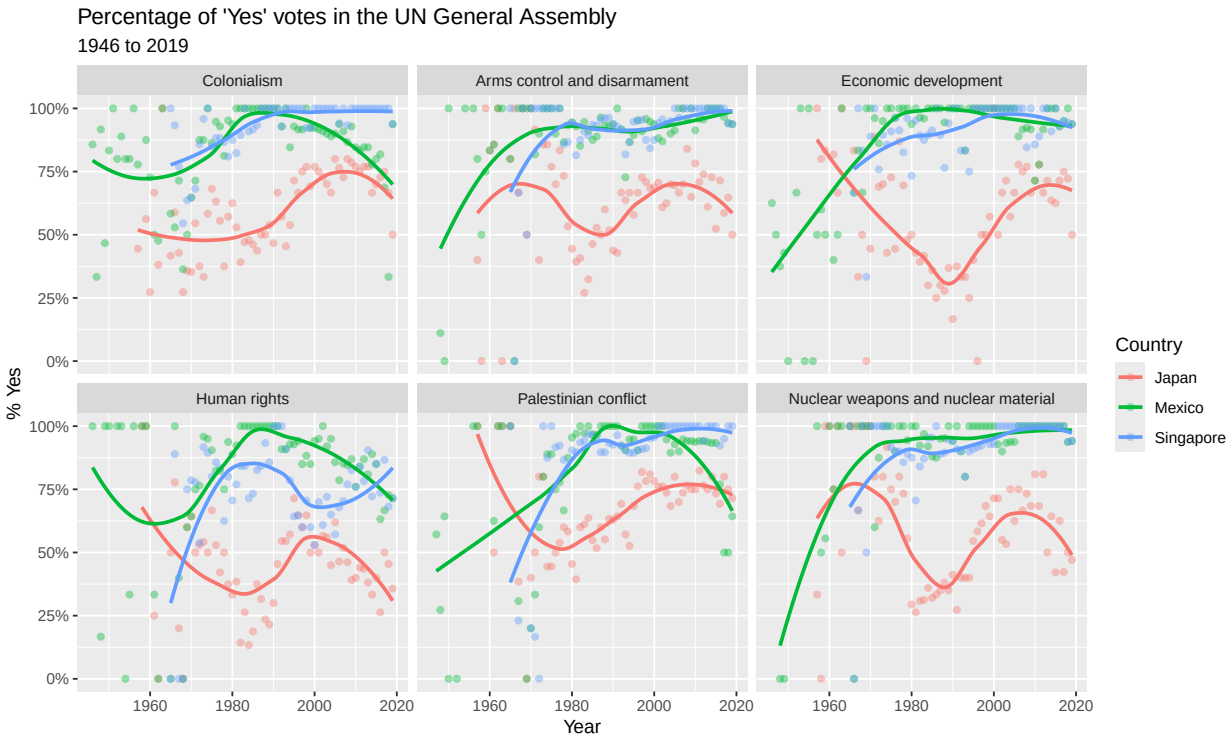
```
unvotes %>%
  # Step 1: Filter for specific countries
  filter(country %in% c("Japan", "Singapore", "Mexico")) %>%

  # Step 2: Extract the year from the date
  mutate(year = year(date)) %>%

  # Step 3: Group by country, year, and issue
  group_by(country, year, issue) %>%

  # Step 4: Calculate percentage of "yes" votes
  summarize(percent_yes = mean(vote == "yes")) %>%

  # Step 5: Create the plot
  ggplot(mapping = aes(x = year, y = percent_yes, color = country)) +
  geom_point(alpha = 0.4) +
  geom_smooth(method = "loess", se = FALSE) +
  facet_wrap(~issue) +
  scale_y_continuous(labels = percent) +
  labs(
    title = "Percentage of 'Yes' votes in the UN General Assembly",
    subtitle = "1946 to 2019",
    y = "% Yes",
    x = "Year",
    color = "Country"
  )
```



Which countries did you choose? Japan, Singapore, and Mexico

Task 3.2: Describe one interesting pattern you notice in your new plot.

Your answer:

One interesting pattern I notice in my new plot is that there was substantial overlap in the percentage of "yes" votes that Singapore and Mexico had across all issues, except for human rights.

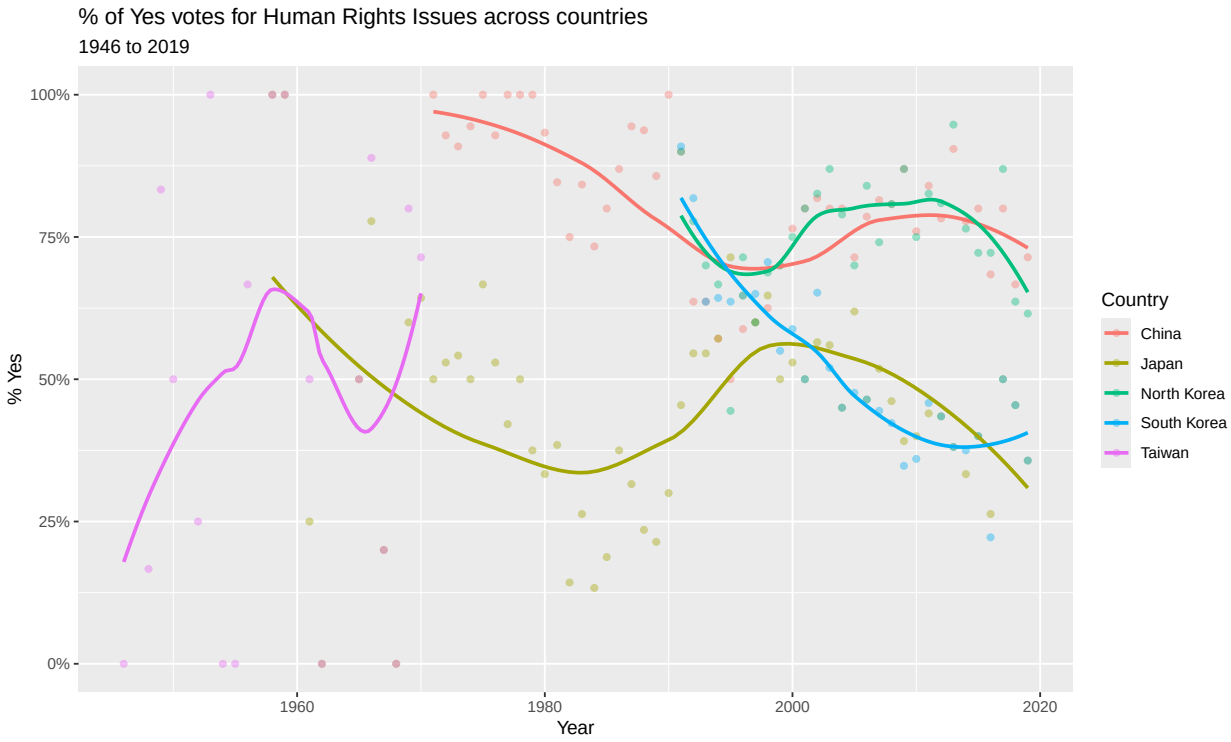
Exercise 4: Focus on One Issue

Task 4.1: Create a plot that shows voting patterns for ONLY the "Human rights" issue, but includes more countries.

Hint: You'll need to add a `filter()` step for the issue, and modify the countries in the first `filter()`.

```
unvotes %>%
  filter(country %in% c("China", "South Korea", "Taiwan", "North Korea", "Japan")) %>%  # Choose 5 coun
  filter(issue == "Human rights") %>%  # Filter for Human rights
  mutate(year = year(date)) %>%
  group_by(country, year) %>%
  summarize(percent_yes = mean(vote == "yes")) %>%
  ggplot(mapping = aes(x = year, y = percent_yes, color = country)) +
  geom_point(alpha = 0.4) +
  geom_smooth(method = "loess", se = FALSE) +
  scale_y_continuous(labels = percent) +
  labs(
    title = "% of Yes votes for Human Rights Issues across countries",  # Update title
    subtitle = "1946 to 2019",
    y = "% Yes",
    x = "Year",
```

```
    color = "Country"
  )
```



Describe what you see:

I see that after 1970, there was no data collected/reported in the unvotes package for Taiwan. I also see that for South Korea, North Korea, Japan, and China, that there was a net decrease in the percentage of "yes" votes for human rights issues from when the data was first collected.

Exercise 5: Calculate Summary Statistics

Task 5.1: Calculate the overall percentage of "yes" votes for each country across all issues and all years.

```
country_yes_percent <- unvotes %>%
  group_by(country) %>%
  summarize(
    total_votes = n(),
    percent_yes = mean(vote == "yes") * 100
  ) %>%
  arrange(desc(percent_yes))

# View the top 10
head(country_yes_percent, 10)
```

```
## # A tibble: 10 x 3
##    country              total_votes percent_yes
##    <chr>                      <int>       <dbl>
##  1 Seychelles                  2304        98.0
##  2 São Tomé & Príncipe         2798        97.3
##  3 Cape Verde                  4242        97.0
```

```
##  4 Guinea-Bissau               3806        96.3
##  5 Timor-Leste                 1516        96.0
##  6 Djibouti                    4323        96.0
##  7 Gambia                      3505        95.3
##  8 United Arab Emirates        4963        95.3
##  9 Kuwait                      5162        95.3
## 10 Maldives                    4781        95.2
```

Which country has the highest overall percentage of "yes" votes? Seychelles has the highest overall percentage of "yes" votes at 98% of 2304 total votes.

Which has the lowest? Zanzibar has the lowest percentage of "yes" votes at 0%, however it only has 1 total vote. The United States has the second lowest percentage of "yes" votes at 20.7% of 5718 total votes.

Task 5.2: Calculate the percentage of "yes" votes by issue, averaged across all countries and all years.

```
issue_yes_percent <- unvotes %>%
  group_by(issue) %>%
  summarize(
    percent_yes = mean(vote == "yes") * 100
  ) %>%
  arrange(percent_yes)

issue_yes_percent
```

```
## # A tibble: 6 x 2
##   issue                                percent_yes
##   <fct>                                      <dbl>
## 1 Human rights                                73.4
## 2 Colonialism                                 80.6
## 3 Nuclear weapons and nuclear material        81.7
## 4 Economic development                        83.0
## 5 Arms control and disarmament                83.3
## 6 Palestinian conflict                        83.9
```

Which issue has the most "yes" votes on average? The Palestinian conflict has the most "yes" votes on average, with about 83.9% "yes" votes.

Which has the least? Human rights issues have the least "yes" votes on average, with about 73.4% "yes" votes.

Exercise 6: Challenge (Optional)

Task 6.1: Create a visualization that shows how voting similarity between the US and other countries has changed over time for one specific issue.

Hints: - You'll need to calculate agreement (both voted yes, or both voted no) - You might want to focus on a few specific countries - Consider using a line plot with `geom_line()`

```
# Your code here (this is challenging - try your best!)
```

Knit and Submit

Before you finish:

1. Make sure all code chunks run without errors

2. Fill in all answer spaces
3. Knit to HTML first (click **Knit** button)
4. Review the HTML output

Git workflow:

1. Click the **Git** pane
2. Check boxes next to all changed files
3. Click **Commit**
4. Write a commit message: "Completed AE 01 - UN Votes"
5. Click **Commit**
6. Click **Push**
7. Verify your work is on GitHub in **YOUR fork**

Check GitHub: - Go to `github.com/YourUsername/application-exercises` - Navigate to `ae-01-un-votes` folder - Verify you see your updated `.Rmd` file - You should also see the rendered `.html` file

---

Generate PDF for Canvas Submission

Once you've verified everything works and committed your changes to GitHub:

Step 1: Knit to PDF

1. Click the **Knit** dropdown arrow (next to the Knit button)
2. Select **Knit to PDF**
3. Wait for the PDF to generate (this may take a minute)

Troubleshooting PDF generation:

- **Error: "LaTeX not found"**
  - Install tinytex in R Console: `tinytex::install_tinytex()`
  - Restart RStudio after installation
  - Try knitting to PDF again
- **Plot doesn't fit on page**
  - Adjust `fig.width` and `fig.height` in chunk options
  - Or add to YAML: `geometry: margin=1in`
- **Country table (Appendix) causing issues**
  - The interactive `datatable()` might not render well in PDF
  - This is okay - the PDF is mainly for your answers

Step 2: Locate Your PDF

After successful PDF generation:

1. Look in the **Files** pane (bottom-right)
2. Find `ae-01-un-votes.pdf` in your `ae-01-un-votes` folder
3. Click the checkbox next to the PDF file
4. Click **More → Export...**
5. Save the PDF to your computer

Step 3: Submit to Canvas

1. Go to Canvas
2. Navigate to the AE 01 assignment
3. Upload `ae-01-un-votes.pdf`

4. Submit

Note: You submit the PDF to Canvas AND push your work to GitHub. Both are required!

References

1. David Robinson (2017). unvotes: United Nations General Assembly Voting Data. R package version 0.2.0.
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.

Appendix

Appendix

Below is a list of countries in the dataset:

Table 1: Countries in the UN Votes Dataset

country
Afghanistan
Albania
Algeria
Andorra
Angola
Antigua & Barbuda
Argentina
Armenia
Australia
Austria
Azerbaijan
Bahamas
Bahrain
Bangladesh
Barbados
Belarus
Belgium
Belize
Benin
Bhutan
Bolivia
Bosnia & Herzegovina
Botswana
Brazil
Brunei
Bulgaria
Burkina Faso
Burundi
Cambodia
Cameroon
Canada

country
Cape Verde
Central African Republic
Chad
Chile
China
Colombia
Comoros
Congo - Brazzaville
Congo - Kinshasa
Costa Rica
Croatia
Cuba
Cyprus
Czechia
Czechoslovakia
Côte d'Ivoire
Denmark
Djibouti
Dominica
Dominican Republic
Ecuador
Egypt
El Salvador
Equatorial Guinea
Eritrea
Estonia
Eswatini
Ethiopia
Federal Republic of Germany
Fiji
Finland
France
Gabon
Gambia
Georgia
German Democratic Republic
Germany
Ghana
Greece
Grenada
Guatemala
Guinea
Guinea-Bissau
Guyana
Haiti
Honduras
Hungary
Iceland
India
Indonesia
Iran
Iraq

country
Ireland
Israel
Italy
Jamaica
Japan
Jordan
Kazakhstan
Kenya
Kiribati
Kuwait
Kyrgyzstan
Laos
Latvia
Lebanon
Lesotho
Liberia
Libya
Liechtenstein
Lithuania
Luxembourg
Madagascar
Malawi
Malaysia
Maldives
Mali
Malta
Marshall Islands
Mauritania
Mauritius
Mexico
Micronesia (Federated States of)
Moldova
Monaco
Mongolia
Montenegro
Morocco
Mozambique
Myanmar (Burma)
Namibia
Nauru
Nepal
Netherlands
New Zealand
Nicaragua
Niger
Nigeria
North Korea
North Macedonia
Norway
Oman
Pakistan
Palau

country
Panama
Papua New Guinea
Paraguay
Peru
Philippines
Poland
Portugal
Qatar
Romania
Russia
Rwanda
Samoa
San Marino
Saudi Arabia
Senegal
Seychelles
Sierra Leone
Singapore
Slovakia
Slovenia
Solomon Islands
Somalia
South Africa
South Korea
South Sudan
Spain
Sri Lanka
St. Kitts & Nevis
St. Lucia
St. Vincent & Grenadines
Sudan
Suriname
Sweden
Switzerland
Syria
São Tomé & Príncipe
Taiwan
Tajikistan
Tanzania
Thailand
Timor-Leste
Togo
Tonga
Trinidad & Tobago
Tunisia
Turkey
Turkmenistan
Tuvalu
Uganda
Ukraine
United Arab Emirates
United Kingdom

country
United States
Uruguay
Uzbekistan
Vanuatu
Venezuela
Vietnam
Yemen
Yemen Arab Republic
Yemen People's Republic
Yugoslavia
Zambia
Zanzibar
Zimbabwe

Troubleshooting

Common issues:

- **"Permission denied" when pushing:** You cloned the course repo instead of your fork. Delete the project and start over with forking first!
- **Error: "could not find function":** Make sure you ran the package loading chunk
- **Plot doesn't show up:** Make sure you ran all the code chunks in order
- **Country name not found:** Check the Appendix for exact spelling and capitalization

PDF-specific issues:

- **"LaTeX not found" or "pdflatex not found"**
  - Install tinytex: Run `tinytex::install_tinytex()` in R Console
  - Restart RStudio
  - Try knitting to PDF again
- **PDF generation fails**
  - Try knitting to HTML first to make sure code works
  - Check that all plots have reasonable `fig.width` and `fig.height`
  - The interactive table in the Appendix may not render in PDF (that's okay)
- **PDF looks different from HTML**
  - This is normal - PDFs have different formatting
  - Focus on making sure your answers and plots are clear
- **Can't find the PDF file**
  - Look in the same folder as your .Rmd file
  - It should be named `ae-01-un-votes.pdf`
  - Use the Files pane in RStudio to locate it

If you're stuck:

- Review the walkthrough video
- Check the course discussion board
- Ask during office hours