AE 10: Scraping multiple pages of articles from the Cornell Review

Suggested answers

APPLICATION EXERCISE ANSWERS

MODIFIED
October 8, 2024

Packages

We will use the following packages in this application exercise.

- **tidyverse**: For data import, wrangling, and visualization.
- **rvest:** For scraping HTML files.
- **robotstxt:** For verifying if we can scrape a website.

```
library(tidyverse)
library(rvest)
library(robotstxt)
```

Part 1 - Data scraping

See the code below stored in `iterate-cornell-review.R`.

```
# load packages
library(tidyverse)
library(rvest)
library(robotstxt)

# check that we can scrape data from the cornell review
paths_allowed("https://www.thecornellreview.org/")

# read the first page
page <- read_html("https://www.thecornellreview.org/")

# extract desired components
titles <- html_elements(x = page, css = "#main .read-title a") |>
  html_text2()

authors <- html_elements(x = page, css = "#main .byline a") |>
  html_text2()
```

```
article_dates <- html_elements(x = page, css = "#main .posts-date") |>
  html_text2()

topics <- html_elements(x = page, css = "#main .cat-links") |>
  html_text2()

abstracts <- html_elements(x = page, css = ".post-description") |>
  html_text2()

post_urls <- html_elements(x = page, css = ".aft-readmore") |>
  html_attr(name = "href")

# create a tibble with this data
review_raw <- tibble(
  title = titles,
  author = authors,
  date = article_dates,
  topic = topics,
  description = abstracts,
  url = post_urls
)

# clean up the data
review <- review_raw |>
  mutate(
    date = mdy(date),
    description = str_remove(string = description, pattern = "\nRead More")
  )

######## write a for loop to scrape the first 10 pages
scrape_results <- vector(mode = "list", length = 10)

for(page_num in 1:length(scrape_results)) {
  # print a message to keep track of where we are in the iteration
  message(str_glue("Scraping page {page_num}"))

  # pause for a couple of seconds to prevent rapid HTTP requests
  Sys.sleep(2)

  # create url
  url <- str_glue("https://www.thecornellreview.org/page/{page_num}/")

  # read the first page
  page <- read_html(url)

  # extract desired components
  titles <- html_elements(x = page, css = "#main .read-title a") |>
    html_text2()
```

```
  authors <- html_elements(x = page, css = "#main .byline a") |>
    html_text2()

  article_dates <- html_elements(x = page, css = "#main .posts-date") |>
    html_text2()

  topics <- html_elements(x = page, css = "#main .cat-links") |>
    html_text2()

  abstracts <- html_elements(x = page, css = ".post-description") |>
    html_text2()

  post_urls <- html_elements(x = page, css = ".aft-readmore") |>
    html_attr(name = "href")

  # create a tibble with this data
  review_raw <- tibble(
    title = titles,
    author = authors,
    date = article_dates,
    topic = topics,
    description = abstracts,
    url = post_urls
  )

  # clean up the data
  review <- review_raw |>
    mutate(
      date = mdy(date),
      description = str_remove(string = description, pattern = "\nRead More")
    )

  # store in list output
  scrape_results[[page_num]] <- review
}

# collapse list of data frames to a single data frame
scrape_df <- list_rbind(x = scrape_results)

######## write a function to scrape a single page and use a map() function
######## to iterate over the first ten pages
# convert to a function
scrape_review <- function(url){
  # pause for a couple of seconds to prevent rapid HTTP requests
  Sys.sleep(2)

  # read the first page
  page <- read_html(url)

  # extract desired components
```

```
  titles <- html_elements(x = page, css = "#main .read-title a") |>
    html_text2()

  authors <- html_elements(x = page, css = "#main .byline a") |>
    html_text2()

  article_dates <- html_elements(x = page, css = "#main .posts-date") |>
    html_text2()

  topics <- html_elements(x = page, css = "#main .cat-links") |>
    html_text2()

  abstracts <- html_elements(x = page, css = ".post-description") |>
    html_text2()

  post_urls <- html_elements(x = page, css = ".aft-readmore") |>
    html_attr(name = "href")

  # create a tibble with this data
  review_raw <- tibble(
    title = titles,
    author = authors,
    date = article_dates,
    topic = topics,
    description = abstracts,
    url = post_urls
  )

  # clean up the data
  review <- review_raw |>
    mutate(
      date = mdy(date),
      description = str_remove(string = description, pattern = "\nRead More")
    )

  # export the resulting data frame
  return(review)
}

# test function
## page 1
scrape_review(url = "https://www.thecornellreview.org/page/1/")

## page 2
scrape_review(url = "https://www.thecornellreview.org/page/2/")

## page 3
scrape_review(url = "https://www.thecornellreview.org/page/3/")

# create a vector of URLs
```

```
page_nums <- 1:10
cr_urls <- str_glue("https://www.thecornellreview.org/page/{page_nums}/")
cr_urls

# map function over URLs
cr_reviews <- map(.x = cr_urls, .f = scrape_review, .progress = TRUE) |>
  list_rbind()

# write data
write_csv(x = cr_reviews, file = "data/cornell-review-all.csv")
```

Part 2 - Data analysis

Demo: Import the scraped data set.

```
cr_reviews <- read_csv(file = "data/cornell-review-all.csv")
```

```
Rows: 100 Columns: 6
── Column specification ────────────────────────────────────────────────────────
Delimiter: ","
chr  (5): title, author, topic, description, url
date (1): date

ℹ Use `spec()` to retrieve the full column specification for this data.
ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

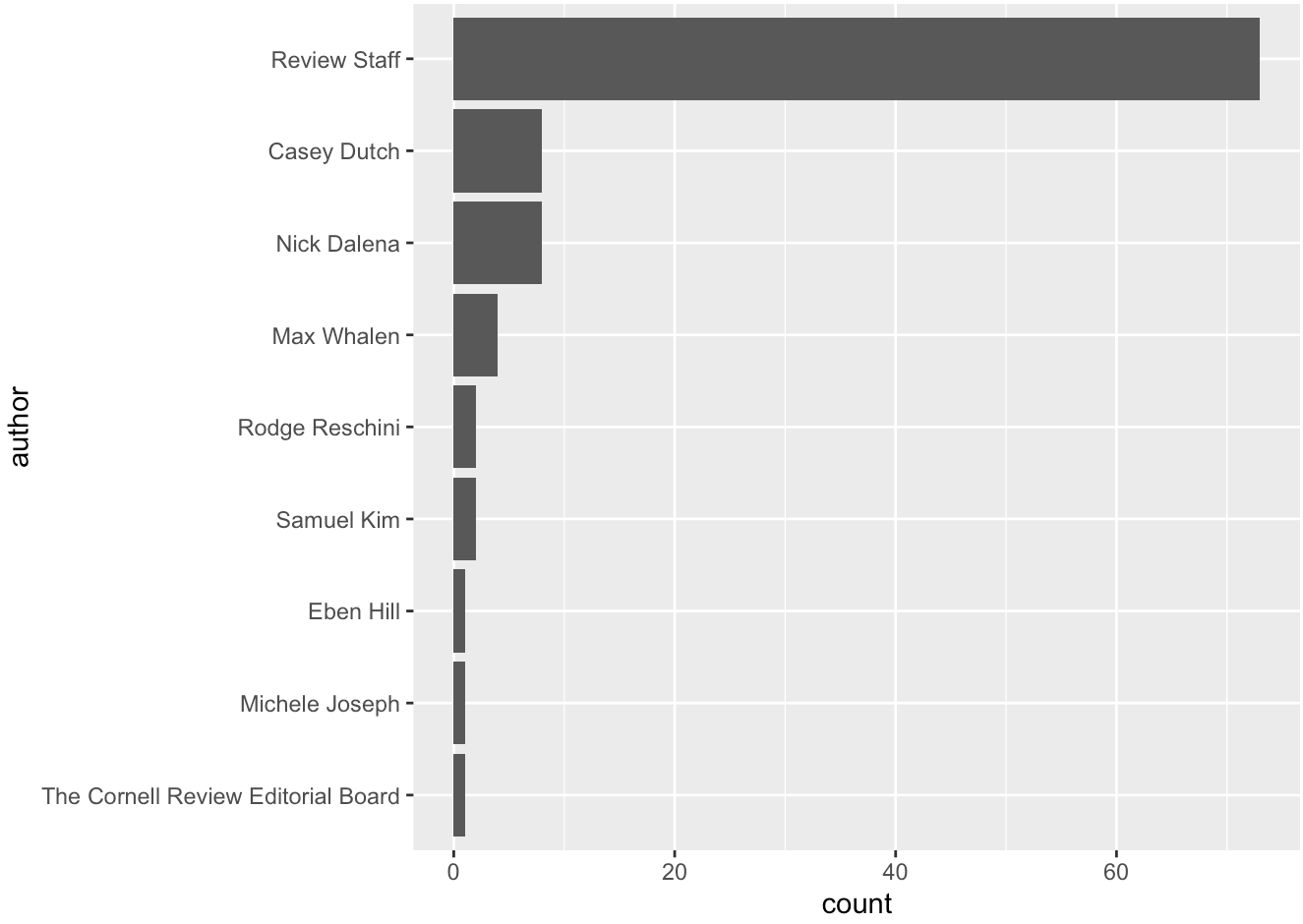
```
cr_reviews
```

```
# A tibble: 100 × 6
   title                                author date       topic description url
   <chr>                                <chr>  <date>     <chr> <chr>       <chr>
 1 Playing the Race Card                Revie… 2024-10-07 "Cam… CML and BS… http…
 2 Should Joel Malina Be Fired?         Revie… 2024-10-07 "Bey… Cornell’s … http…
 3 Cornell Drops in 2025 FIRE Free Sp… Revie… 2024-10-03 "Cam… Each year,… http…
 4 Interim Expressive Activity Policy… Revie… 2024-10-02 "Cor… On October… http…
 5 Daryl Davis To Speak on Race Relat… Revie… 2024-10-01 "Cam… Daryl Davi… http…
 6 Happy 100th Birthday, President Ca… Revie… 2024-10-01 "Bey… President … http…
 7 Kavita Bala Named Cornell Provost   Revie… 2024-09-25 "Cam… On Septemb… http…
 8 Ithaca Labor News                   Revie… 2024-09-25 "Ith… Here are t… http…
 9 CML Realizes It Overstepped Social… Revie… 2024-09-25 "Cam… On Wednesd… http…
10 Cornell Republicans to Host Ben Sh… Revie… 2024-09-24 "Ith… On Monday,… http…
# ℹ 90 more rows
```

Demo: Who are the most prolific authors?

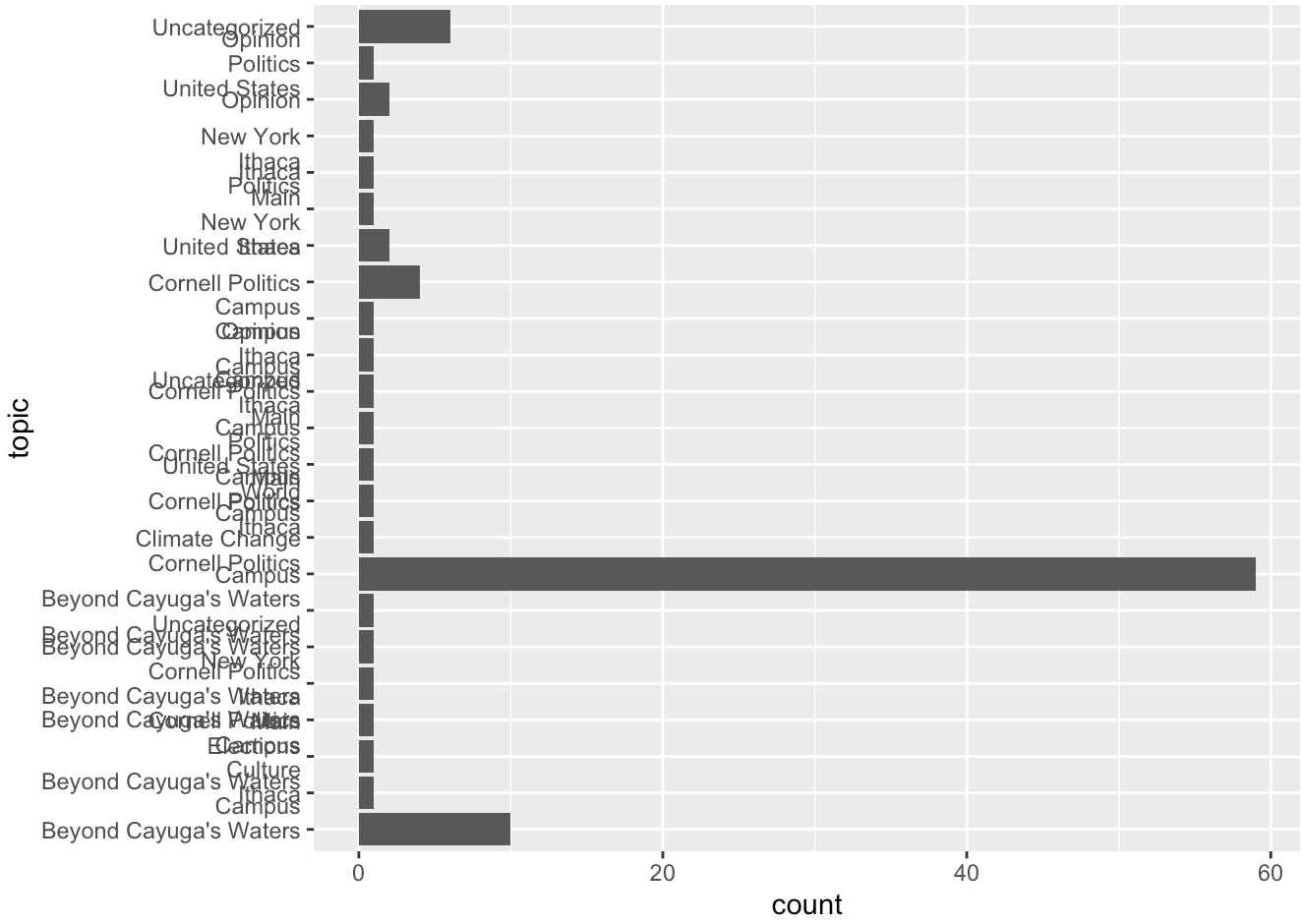
```
cr_reviews |>
  # adjust order of authors so they appear from most to least frequent
```

```
  mutate(author = fct_infreq(f = author) |>
    fct_rev()) |>
  # horizontal bar chart
  ggplot(mapping = aes(y = author)) +
  geom_bar()
```



Demo: What topics does The Cornell Review write about?

```
# basic bar plot
ggplot(data = cr_reviews, mapping = aes(y = topic)) +
  geom_bar()
```



Not super helpful. Each article can have multiple topics. What is the syntax for this column?

```
cr_reviews |>
  select(topic)
```

```
# A tibble: 100 × 1
   topic
   <chr>
 1 "Campus"
 2 "Beyond Cayuga's Waters"
 3 "Campus"
 4 "Cornell Politics"
 5 "Campus"
 6 "Beyond Cayuga's Waters\nUncategorized"
 7 "Campus"
 8 "Ithaca"
 9 "Campus"
10 "Ithaca\nPolitics"
# ℹ 90 more rows
```

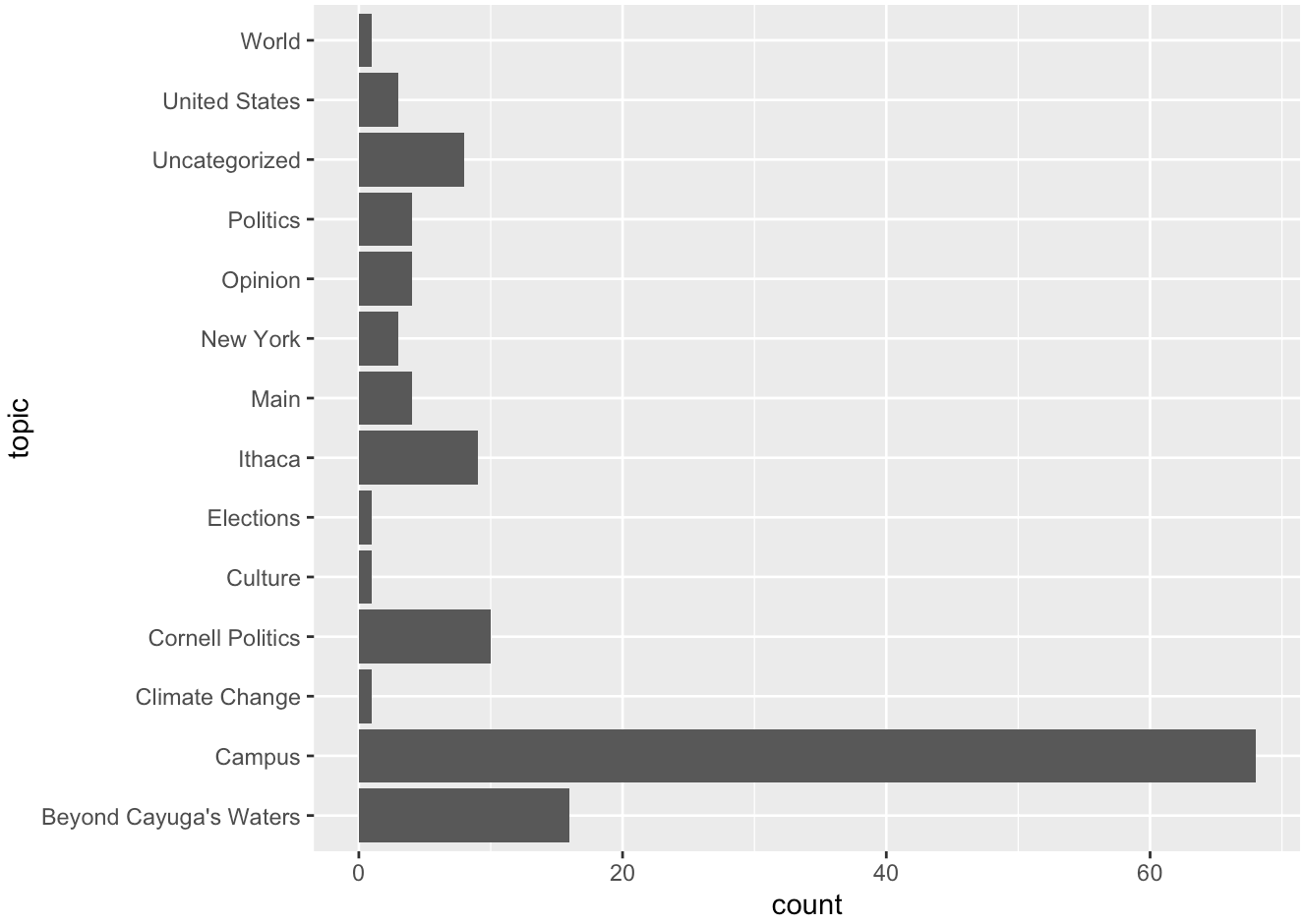
Each topic is separated by a `"\n"`. Since the number of topics varies for each article, we should `separate_longer_delim()` this column. Instead we can use a **stringr** function to split them into distinct character strings.

```
cr_reviews |>
  separate_longer_delim(
    cols = topic,
    delim = "\n"
  )
```

```
# A tibble: 133 × 6
   title                              author date       topic description url
   <chr>                              <chr>  <date>     <chr> <chr>       <chr>
 1 Playing the Race Card              Revie… 2024-10-07 Camp… CML and BS… http…
 2 Should Joel Malina Be Fired?       Revie… 2024-10-07 Beyo… Cornell’s … http…
 3 Cornell Drops in 2025 FIRE Free Sp… Revie… 2024-10-03 Camp… Each year,… http…
 4 Interim Expressive Activity Policy… Revie… 2024-10-02 Corn… On October… http…
 5 Daryl Davis To Speak on Race Relat… Revie… 2024-10-01 Camp… Daryl Davi… http…
 6 Happy 100th Birthday, President Ca… Revie… 2024-10-01 Beyo… President … http…
 7 Happy 100th Birthday, President Ca… Revie… 2024-10-01 Unca… President … http…
 8 Kavita Bala Named Cornell Provost   Revie… 2024-09-25 Camp… On Septemb… http…
 9 Ithaca Labor News                   Revie… 2024-09-25 Itha… Here are t… http…
10 CML Realizes It Overstepped Social… Revie… 2024-09-25 Camp… On Wednesd… http…
# ℹ 123 more rows
```

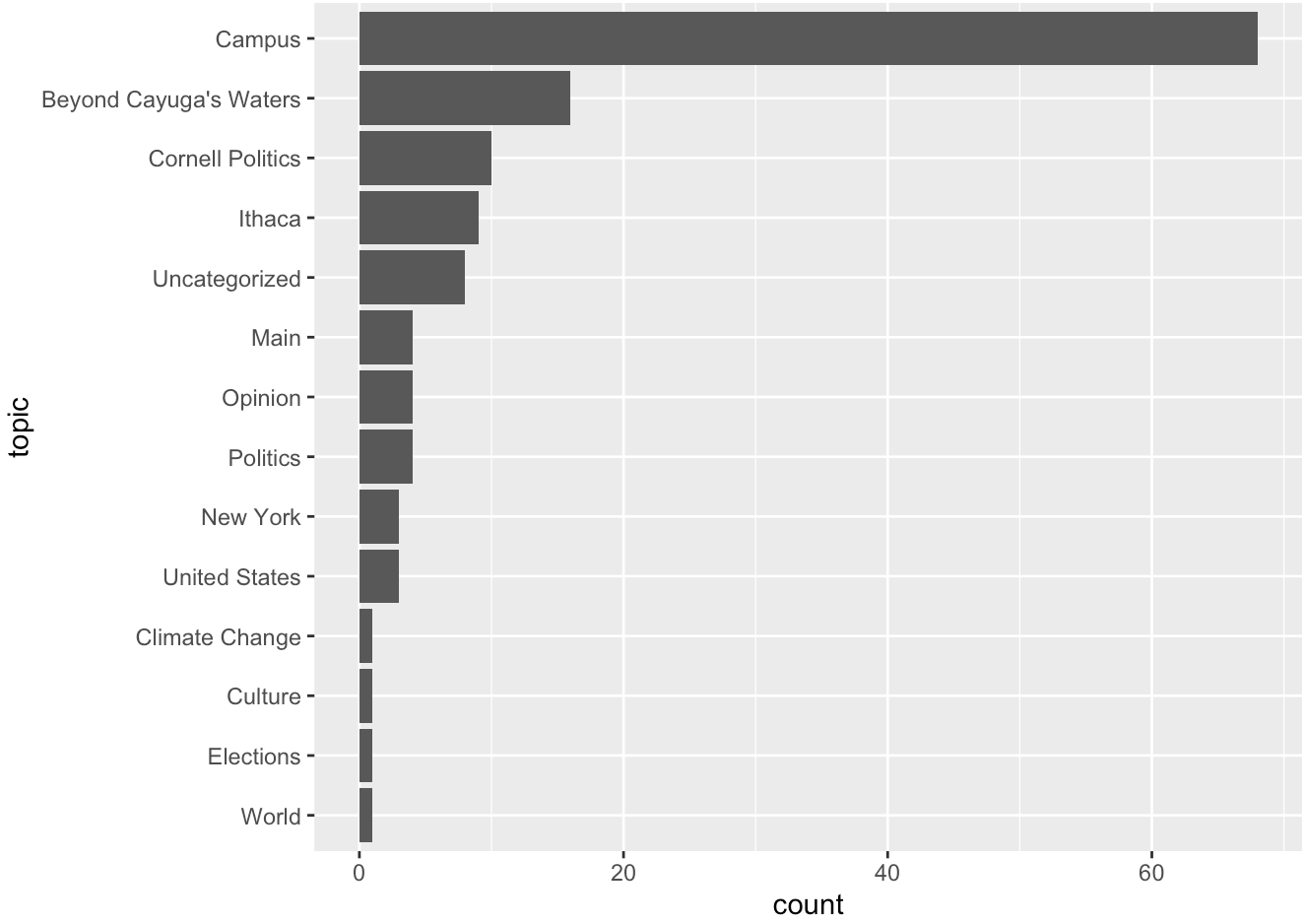
Notice the data frame now has additional rows. The unit of analysis is now an article-topic combination, rather than one-row-per-article. Not entirely a tidy structure, but necessary to construct a chart to visualize topic frequency.

```
cr_reviews |>
  separate_longer_delim(
    cols = topic,
    delim = "\n"
  ) |>
  ggplot(mapping = aes(y = topic)) +
  geom_bar()
```



Let's clean this up like the previous chart.

```
cr_reviews |>
  separate_longer_delim(
    cols = topic,
    delim = "\n"
  ) |>
  mutate(topic = fct_infreq(f = topic) |>
    fct_rev()) |>
  ggplot(mapping = aes(y = topic)) +
  geom_bar()
```



Session information