Web scraping

Week 13

AEM 2850 / 5850 : R for Business Analytics
Cornell Dyson
Fall 2025

Announcements

This is our last full week of class! 😢

Homework - Week 13 will be due Monday

Homework - Week 14 will not exist due to time constraints

- But next Tuesday's material is still fair game for Prelim 2!

Prelim 2 is December 4 at 7:30pm

- I will provide more information at the end of class today
- I will post practice problems before Thanksgiving Break

Questions before we get started?

Plan for today

Web scraping basics

Web scraping with rvest

- Cornell sports
- College rankings

robots.txt

Prelim 2

Web scraping basics

What is web scraping?

Getting data or "content" off the web and onto our computers

We get content off the web all the time!

- Copy and paste
- Read and take notes
- Screenshot

The goal of web **scraping** is to write computer code to help us automate this process and store the results in a machine-readable format

Why would we want to scrape data?

When is web scraping useful?

- When the data is publicly available
- When you can't get the data in a more convenient format

When is web scraping not useful?

- When data is publicly available in other formats (e.g., csv)
- When the site owner offers a way to access data directly (e.g., via an API)

Web scraping is time consuming and costly (for both you and "them")

Server-side vs client-side content

1. Server-side

- Host server "builds" site and sends HTML code that our browser renders
- All the information is embedded in the website's HTML

2. Client-side

- Site contains an empty template of HTML and CSS
- When we visit, our browser sends a *request* to the host server
- The server sends a *response* script that our browser uses to populate the HTML template with information we want

We will focus on server-side web scraping due to time constraints

What is HTML?

HTML stands for "HyperText Markup Language" and looks like this:

```html
<html>
<head>
  <title>Page title</title>
</head>
<body>
  <h1 id='first'>A heading</h1>
  <p>Some text &amp; <b>some bold text.</b></p>
  <img src='myimg.png' width='100' height='100'>
</body>
```

What is HTML?

HTML has a hierarchical structure formed by **elements** that consist of:

1. a start tag
    - optional attributes
2. contents
3. an end tag

What is HTML?

HTML has a hierarchical structure formed by **elements** that consist of:

1. a start tag (e.g., `<h1>`)
    - optional attributes (e.g., `id='first'`)
2. contents in between tags (e.g., `A heading`)
3. an end tag (e.g., `</h1>`)

```
<html>
<head>
  <title>Page title</title>
</head>
<body>
  <h1 id='first'>A heading</h1>
  <p>Some text &amp; <b>some bold text.</b></p>
  <img src='myimg.png' width='100' height='100'>
</body>
```

What is HTML?

Elements

- There are over 100 HTML elements
- Look up tags to learn about them as needed

Contents

- Most elements can have content in between start and end tags
- Content can be text or more elements (as **children**)

Attributes

- Attributes like `id` and `class` are used with CSS to control page appearance
- These attributes are useful for scraping data

What is CSS?

CSS stands for **C**ascading **S**tyle **S**heets

- Tool for defining visual appearance of HTML

CSS selectors help identify what we want to scrape

We will learn by example using the extension/bookmarklet SelectorGadget

Web scraping with rvest

The rvest package

rvest (as in "harvest") is part of the tidyverse

```
library(rvest) # installed with tidyverse but needs to be loaded
```

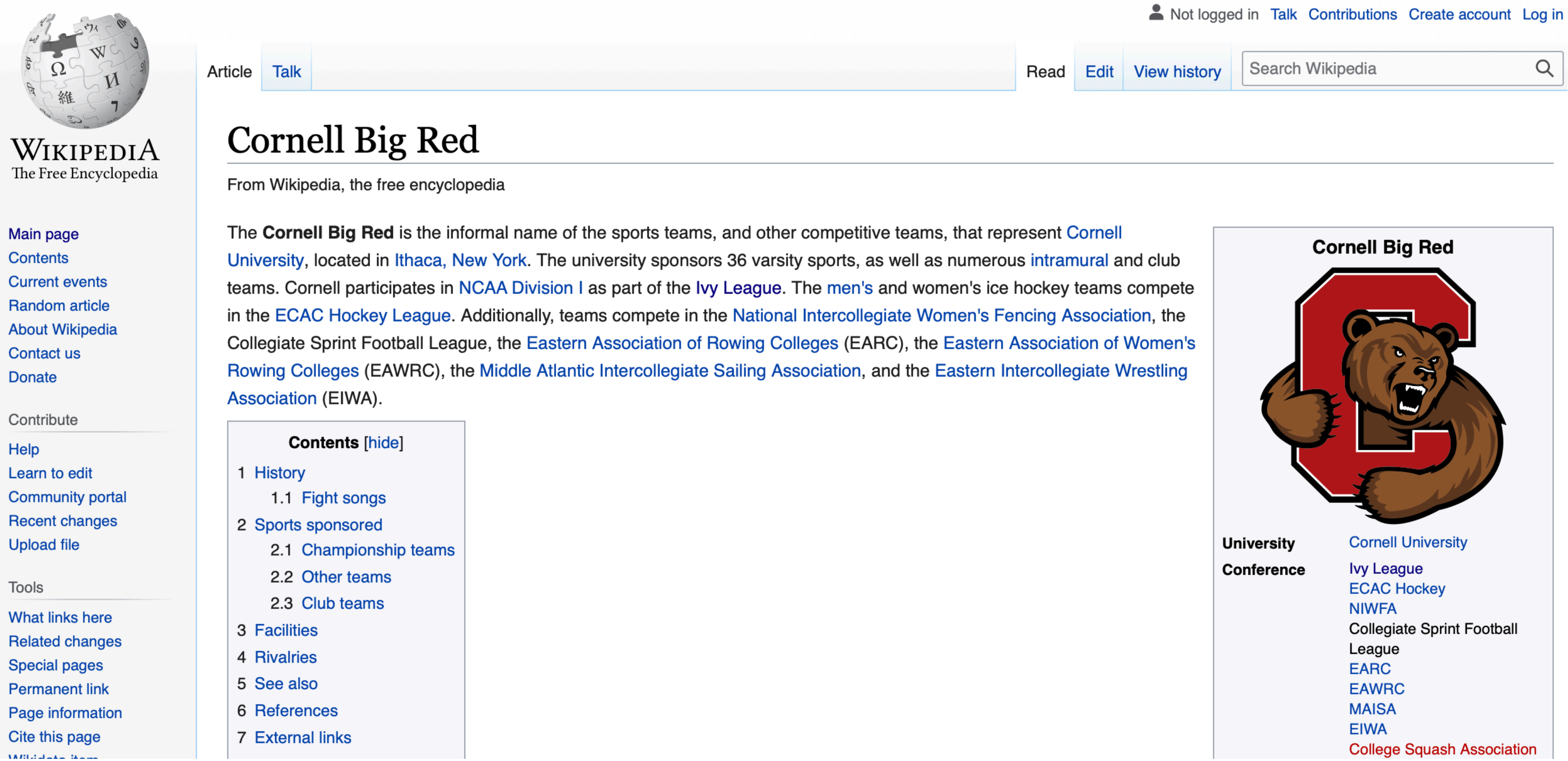
We will cover several functions that make it easy to scrape data from web pages:

- `read_html` reads HTML, much like `read_csv` reads .csv files
- `html_element(s)` find HTML elements using CSS selectors or XPath expressions
- `html_text2` retrieves text from HTML elements
- `html_table` parses HTML tables into data frames

Let's learn these commands by working through two examples

Example 1: Cornell Big Red on Wikipedia

How could we scrape a list of varsity sports?



Cornell Big Red

From Wikipedia, the free encyclopedia

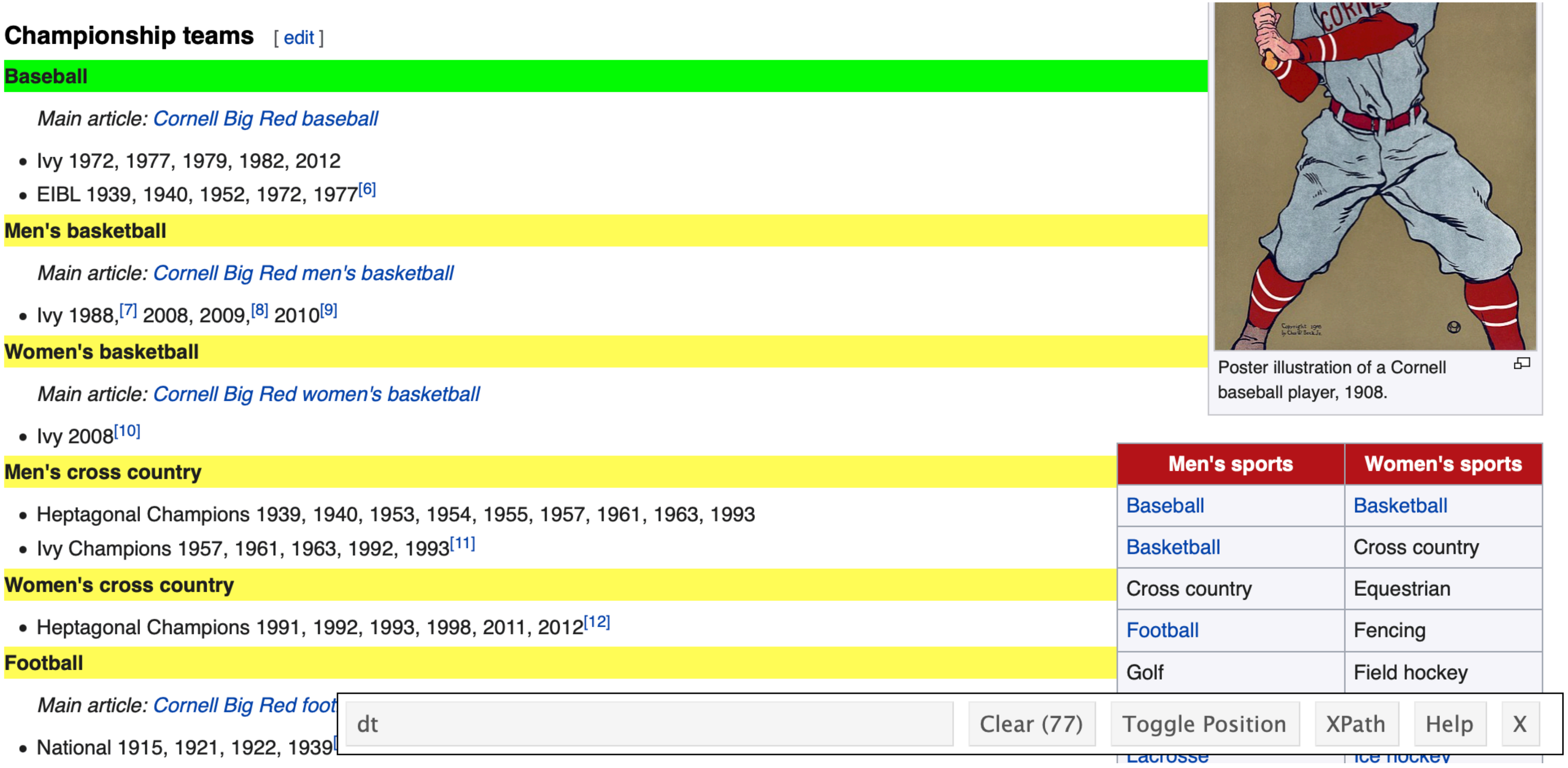
The **Cornell Big Red** is the informal name of the sports teams, and other competitive teams, that represent Cornell University, located in Ithaca, New York. The university sponsors 36 varsity sports, as well as numerous intramural and club teams. Cornell participates in NCAA Division I as part of the Ivy League. The men's and women's ice hockey teams compete in the ECAC Hockey League. Additionally, teams compete in the National Intercollegiate Women's Fencing Association, the Collegiate Sprint Football League, the Eastern Association of Rowing Colleges (EARC), the Eastern Association of Women's Rowing Colleges (EAWRC), the Middle Atlantic Intercollegiate Sailing Association, and the Eastern Intercollegiate Wrestling Association (EIWA).

Contents [hide]

Cornell Big Red

University	Cornell University
Conference	Ivy League ECAC Hockey NIWFA Collegiate Sprint Football League EARC EAWRC MAISA EIWA College Squash Association

Option 1: use `dt` tag to get headings



Championship teams [edit]

Baseball

Main article: Cornell Big Red baseball

- Ivy 1972, 1977, 1979, 1982, 2012
- EIBL 1939, 1940, 1952, 1972, 1977[6]

Men's basketball

Main article: Cornell Big Red men's basketball

- Ivy 1988,[7] 2008, 2009,[8] 2010[9]

Women's basketball

Main article: Cornell Big Red women's basketball

- Ivy 2008[10]

Men's cross country

- Heptagonal Champions 1939, 1940, 1953, 1954, 1955, 1957, 1961, 1963, 1993
- Ivy Champions 1957, 1961, 1963, 1992, 1993[11]

Women's cross country

- Heptagonal Champions 1991, 1992, 1993, 1998, 2011, 2012[12]

Football

Main article: Cornell Big Red foot

- National 1915, 1921, 1922, 1939

Poster illustration of a Cornell baseball player, 1908.

Men's sports	Women's sports
Baseball	Basketball
Basketball	Cross country
Cross country	Equestrian
Football	Fencing
Golf	Field hockey
Lacrosse	Ice hockey

dt

Scraping text using `dt` tag

Step 1: use `read_html()` to read in html from the url of interest

```r
big_red <- read_html("https://en.wikipedia.org/wiki/Cornell_Big_Red")

big_red
```

```
## {html_document}
## <html class="client-nojs vector-feature-language-in-header-enabled vector-feature-language-in-main-page
## [1] <head>\n<meta http-equiv="Content-Type" content="text/html; charset=UTF-8 ...
## [2] <body class="skin--responsive skin-vector skin-vector-search-vue mediawik ...
```

Scraping text using `dt` tag

Step 2: use `html_elements()` to extract every instance of a `dt` tag

```
big_red <- read_html("https://en.wikipedia.org/wiki/Cornell_Big_Red")

big_red |>
  html_elements("dt") |> # dt tag is for terms in a description list
  head(8)
```

```
## {xml_nodeset (8)}
## [1] <dt>Baseball</dt>
## [2] <dt>Men's basketball</dt>
## [3] <dt>Women's basketball</dt>
## [4] <dt>Men's cross country</dt>
## [5] <dt>Women's cross country</dt>
## [6] <dt>Women's fencing</dt>
## [7] <dt>Football</dt>
## [8] <dt>Sprint football</dt>
```

Scraping text using `dt` tag

Step 3: use `html_text2()` to convert the sports to a character vector

```
big_red <- read_html("https://en.wikipedia.org/wiki/Cornell_Big_Red")

big_red_text <- big_red |>
  html_elements("dt") |> # dt tag is for terms in a description list
  html_text2()           # convert html to text

head(big_red_text)       # looks good!
```

```
## [1] "Baseball"            "Men's basketball"      "Women's basketball"
## [4] "Men's cross country" "Women's cross country" "Women's fencing"
```

```
length(big_red_text) # hmm...
```

```
## [1] 78
```

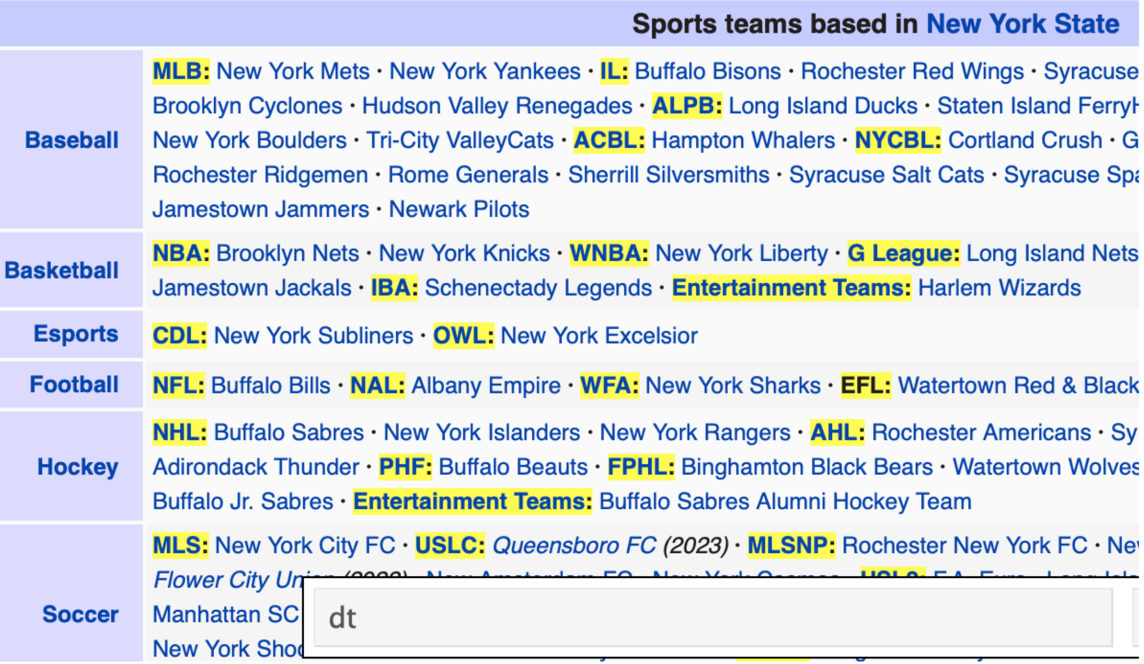
```
tail(big_red_text) # uh-oh...
```

```
## [1] "WFTDA" "MRDA"  "USARL" "NARL"  "USAR"  "WTT"
```

That doesn't seem right...

What went wrong?

1. Got irrelevant data



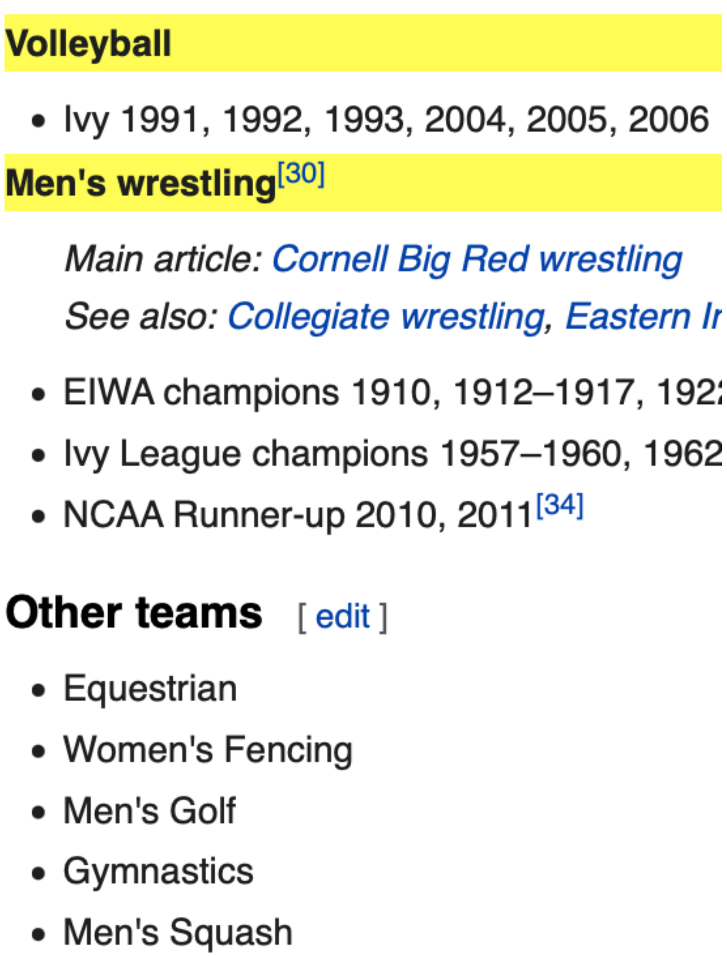
Sports teams based in New York State	
Baseball	**MLB:** New York Mets · New York Yankees · **IL:** Buffalo Bisons · Rochester Red Wings · Syracuse Brooklyn Cyclones · Hudson Valley Renegades · **ALPB:** Long Island Ducks · Staten Island Ferryh New York Boulders · Tri-City ValleyCats · **ACBL:** Hampton Whalers · **NYCBL:** Cortland Crush · G Rochester Ridgemen · Rome Generals · Sherrill Silversmiths · Syracuse Salt Cats · Syracuse Spa Jamestown Jammers · Newark Pilots
Basketball	**NBA:** Brooklyn Nets · New York Knicks · **WNBA:** New York Liberty · **G League:** Long Island Nets Jamestown Jackals · **IBA:** Schenectady Legends · **Entertainment Teams:** Harlem Wizards
Esports	**CDL:** New York Subliners · **OWL:** New York Excelsior
Football	**NFL:** Buffalo Bills · **NAL:** Albany Empire · **WFA:** New York Sharks · **EFL:** Watertown Red & Black
Hockey	**NHL:** Buffalo Sabres · New York Islanders · New York Rangers · **AHL:** Rochester Americans · Sy Adirondack Thunder · **PHF:** Buffalo Beauts · **FPHL:** Binghamton Black Bears · Watertown Wolves Buffalo Jr. Sabres · **Entertainment Teams:** Buffalo Sabres Alumni Hockey Team
Soccer	**MLS:** New York City FC · **USLC:** *Queensboro FC* *(2023)* · **MLSNP:** Rochester New York FC · Ne *Flower City Un* Manhattan SC New York Sho

dt

What went wrong?

1. Got irrelevant data

2. Didn't get relevant data



Volleyball

- Ivy 1991, 1992, 1993, 2004, 2005, 2006

Men's wrestling[30]

Main article: Cornell Big Red wrestling

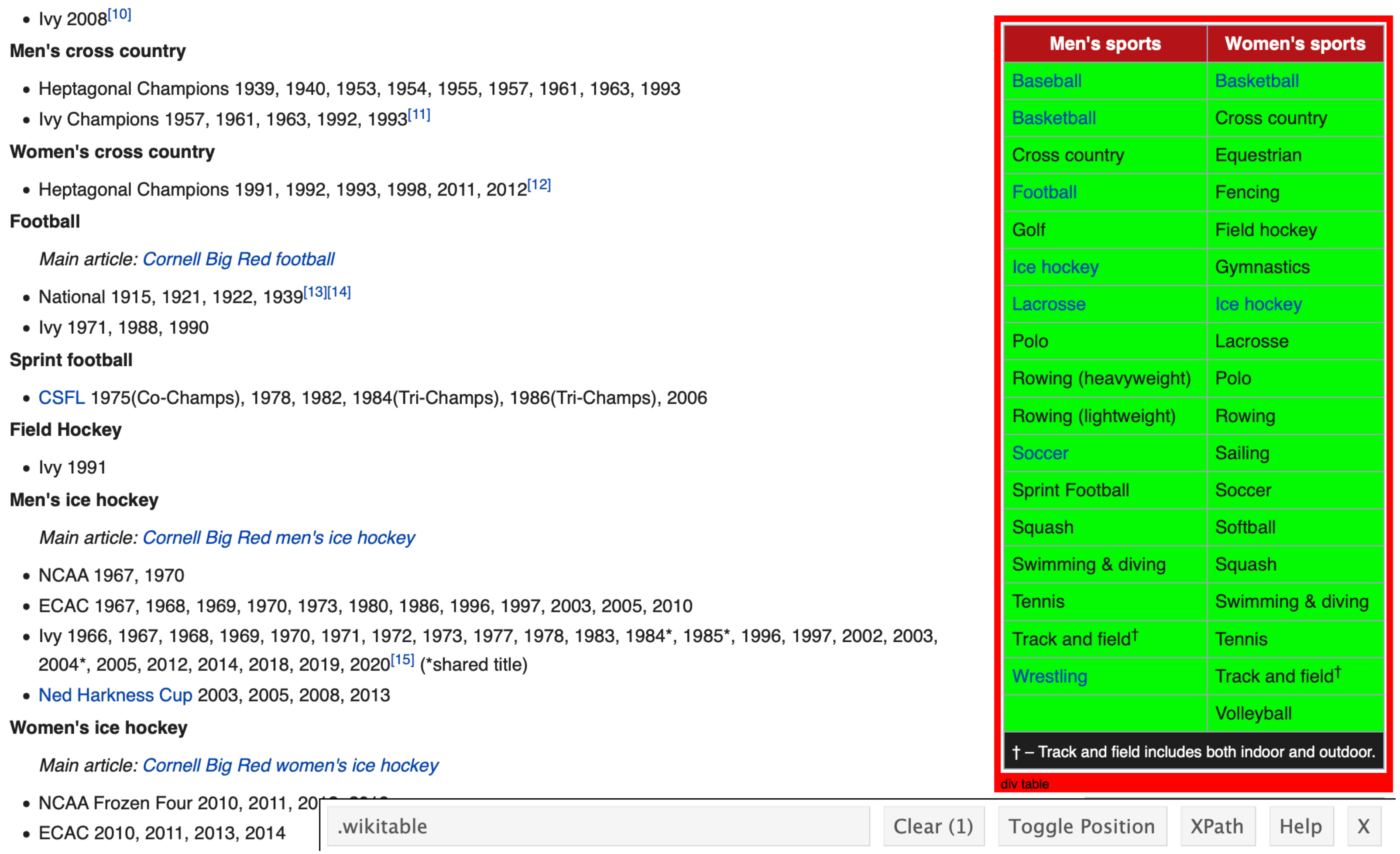
See also: Collegiate wrestling, Eastern Ir

- EIWA champions 1910, 1912–1917, 192
- Ivy League champions 1957–1960, 1962
- NCAA Runner-up 2010, 2011[34]

Other teams [edit]

- Equestrian
- Women's Fencing
- Men's Golf
- Gymnastics
- Men's Squash

Option 2: use `.wikitable` tag to get table



- Ivy 2008[10]

Men's cross country

- Heptagonal Champions 1939, 1940, 1953, 1954, 1955, 1957, 1961, 1963, 1993
- Ivy Champions 1957, 1961, 1963, 1992, 1993[11]

Women's cross country

- Heptagonal Champions 1991, 1992, 1993, 1998, 2011, 2012[12]

Football

Main article: Cornell Big Red football

- National 1915, 1921, 1922, 1939[13][14]
- Ivy 1971, 1988, 1990

Sprint football

- CSFL 1975(Co-Champs), 1978, 1982, 1984(Tri-Champs), 1986(Tri-Champs), 2006

Field Hockey

- Ivy 1991

Men's ice hockey

Main article: Cornell Big Red men's ice hockey

- NCAA 1967, 1970
- ECAC 1967, 1968, 1969, 1970, 1973, 1980, 1986, 1996, 1997, 2003, 2005, 2010
- Ivy 1966, 1967, 1968, 1969, 1970, 1971, 1972, 1973, 1977, 1978, 1983, 1984*, 1985*, 1996, 1997, 2002, 2003, 2004*, 2005, 2012, 2014, 2018, 2019, 2020[15] (*shared title)
- Ned Harkness Cup 2003, 2005, 2008, 2013

Women's ice hockey

Main article: Cornell Big Red women's ice hockey

- NCAA Frozen Four 2010, 2011, 20
- ECAC 2010, 2011, 2013, 2014

Men's sports	Women's sports
Baseball	Basketball
Basketball	Cross country
Cross country	Equestrian
Football	Fencing
Golf	Field hockey
Ice hockey	Gymnastics
Lacrosse	Ice hockey
Polo	Lacrosse
Rowing (heavyweight)	Polo
Rowing (lightweight)	Rowing
Soccer	Sailing
Sprint Football	Soccer
Squash	Softball
Swimming & diving	Squash
Tennis	Swimming & diving
Track and field†	Tennis
Wrestling	Track and field†
	Volleyball

† – Track and field includes both indoor and outdoor.

.wikitable

Scraping tables using `.wikitable` tag

Step 1: use `read_html()` to read in html from the url of interest

```
big_red <- read_html("https://en.wikipedia.org/wiki/Cornell_Big_Red")
```

Step 2: use `html_element()` to extract the first table element

```
big_red |>
  html_element(".wikitable")    # extract the first .wikitable
```

```
## {html_node}
## <table class="wikitable" style="">
## [1] <tbody>\n<tr>\n<th scope="col" style="background-color:#B31B1B;color:#FFF ...
```

Scraping tables using `.wikitable` tag

Step 3: use `html_table()` to convert the table into a data frame

```r
big_red_table <- big_red |>
  html_element(".wikitable") |> # extract the first .wikitable
  html_table()                  # convert html to a data frame

head(big_red_table, 8)
```

```
## # A tibble: 8 × 2
##   `Men's sports` `Women's sports`
##   <chr>          <chr>
## 1 Baseball       Basketball
## 2 Basketball     Cross country
## 3 Cross country  Equestrian
## 4 Football       Fencing
## 5 Golf           Field hockey
## 6 Ice hockey     Gymnastics
## 7 Lacrosse       Ice hockey
## 8 Polo           Lacrosse
```

Scraped data frames are data frames

```
tidy_big_red <- big_red_table |>
  pivot_longer(everything(), names_to = "gender", values_to = "sport") |>
  filter(sport != "" & !str_detect(sport, "^†")) # remove things that aren't sports

tidy_big_red
```

```
## # A tibble: 35 × 2
##    gender          sport
##    <chr>           <chr>
##  1 Men's sports    Baseball
##  2 Women's sports Basketball
##  3 Men's sports    Basketball
##  4 Women's sports Cross country
##  5 Men's sports    Cross country
##  6 Women's sports Equestrian
##  7 Men's sports    Football
##  8 Women's sports Fencing
##  9 Men's sports    Golf
## 10 Women's sports Field hockey
## # ℹ 25 more rows
```

Scraped data frames are data frames

What function(s) could we use to determine how many gender category-sport pairs there are in `tidy_big_red`?

```
tidy_big_red |>
  count()
```

```
## # A tibble: 1 × 1
##       n
##   <int>
## 1    35
```

```
tidy_big_red |>
  nrow()
```

```
## [1] 35
```

(Or we could have gone back one slide to look at the tibble header...)

Scraped data frames are data frames

What function(s) could we use to determine how many distinct sports there are in `tidy_big_red`?

```
tidy_big_red |>
  distinct(sport) |>
  count()
```

```
## # A tibble: 1 × 1
##       n
##   <int>
## 1    25
```

```
tidy_big_red |>
  select(sport) |>
  n_distinct()
```

```
## [1] 25
```

Scraped data frames are data frames

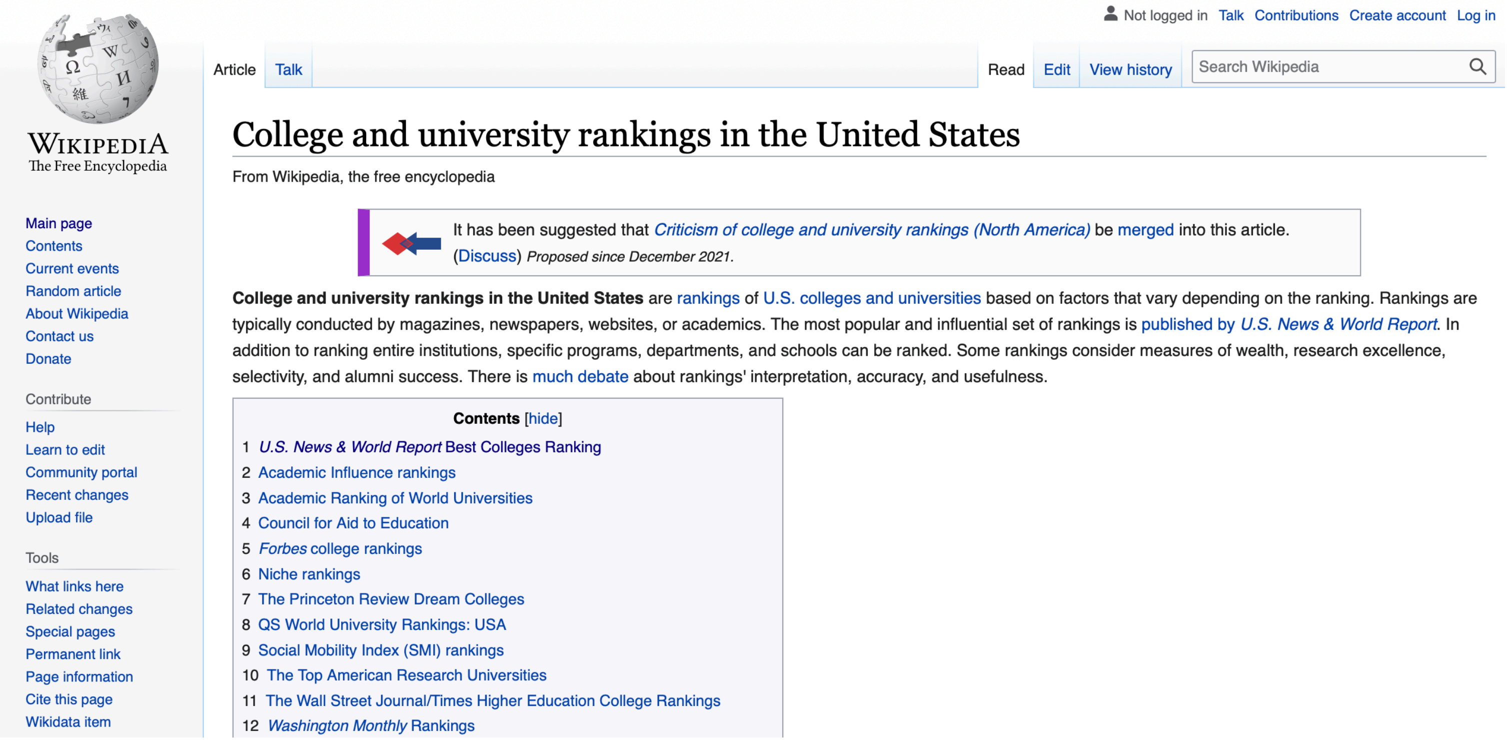
What function could we use to determine how many distinct sports are there for each gender category?

```
tidy_big_red |>
  count(gender)
```

```
## # A tibble: 2 × 2
##   gender              n
##   <chr>           <int>
## 1 Men's sports       17
## 2 Women's sports     18
```

Example 2: College rankings on Wikipedia

How could we scrape college rankings?



College and university rankings in the United States

From Wikipedia, the free encyclopedia

It has been suggested that *Criticism of college and university rankings (North America)* be merged into this article. (Discuss) *Proposed since December 2021.*

College and university rankings in the United States are rankings of U.S. colleges and universities based on factors that vary depending on the ranking. Rankings are typically conducted by magazines, newspapers, websites, or academics. The most popular and influential set of rankings is published by *U.S. News & World Report*. In addition to ranking entire institutions, specific programs, departments, and schools can be ranked. Some rankings consider measures of wealth, research excellence, selectivity, and alumni success. There is much debate about rankings' interpretation, accuracy, and usefulness.

Contents [hide]

The site has changed over time, so we will scrape an archive from The Wayback Machine. One of web scraping's many challenges!

Use `.wikitable` tag to get the first table

```r
rankings <- read_html("https://web.archive.org/web/20220405170508/https://en.wikipedia.org/wiki/Col

first_table <- rankings |>
  html_element(".wikitable") |> # extract the first .wikitable
  html_table()                  # convert html to a data frame

first_table
```

```
## # A tibble: 21 × 5
##    Top national universit…¹ `2022 rank` ``    Top liberal arts col…² `2022 rank`
##    <chr>                          <int> <lgl> <chr>                        <int>
##  1 Princeton University               1 NA    Williams College                 1
##  2 Columbia University                2 NA    Amherst College                  2
##  3 Harvard University                 2 NA    Swarthmore College               3
##  4 Massachusetts Institute…           2 NA    Pomona College                   4
##  5 Yale University                    5 NA    Wellesley College                5
##  6 Stanford University                6 NA    Bowdoin College                  6
##  7 University of Chicago              6 NA    United States Naval A…           6
##  8 University of Pennsylva…           8 NA    Claremont McKenna Col…           8
##  9 California Institute of…           9 NA    Carleton College                 9
## 10 Duke University                    9 NA    Middlebury College               9
## # ℹ 11 more rows
```

Scraped data frames are data frames

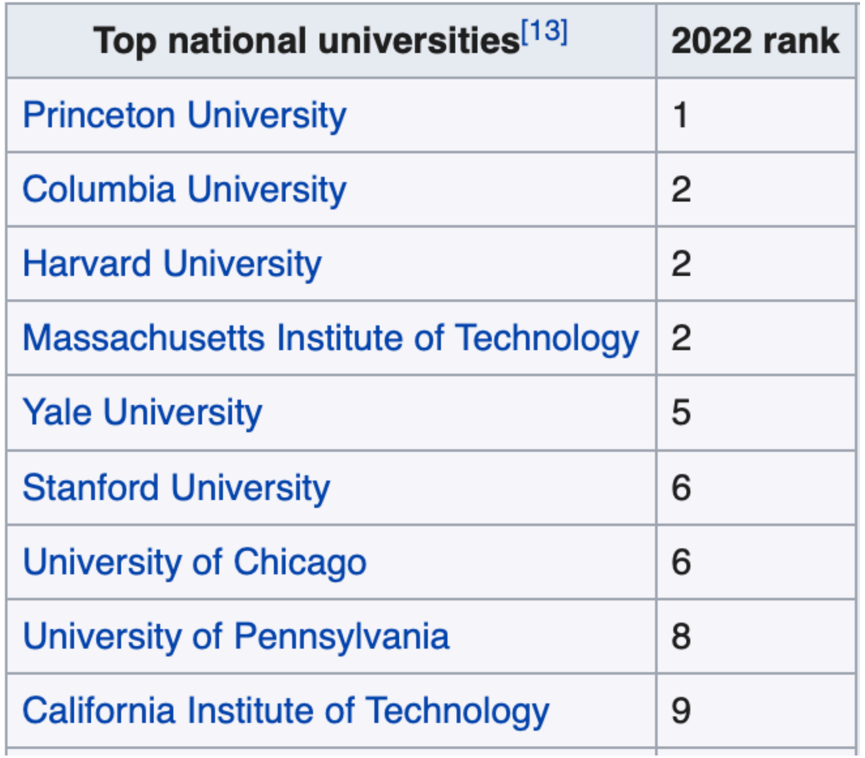
How does Cornell stack up?

How could we find it within a table with many other schools?

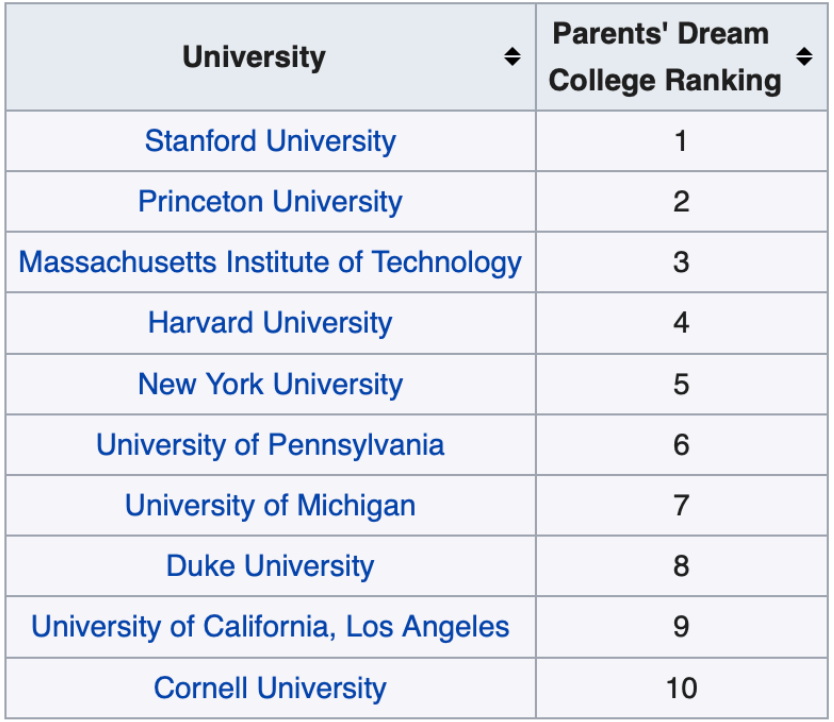
```
first_table |>
  select(uni = 1, rank = 2) |>       # select and rename the first two columns
  filter(str_detect(uni, "Cornell")) # use pattern matching to find Cornell
```

```
## # A tibble: 1 × 2
##   uni                 rank
##   <chr>              <int>
## 1 Cornell University    17
```

What if CSS selectors match multiple tables?



Top national universities[13]	2022 rank
Princeton University	1
Columbia University	2
Harvard University	2
Massachusetts Institute of Technology	2
Yale University	5
Stanford University	6
University of Chicago	6
University of Pennsylvania	8
California Institute of Technology	9



University	Parents' Dream College Ranking
Stanford University	1
Princeton University	2
Massachusetts Institute of Technology	3
Harvard University	4
New York University	5
University of Pennsylvania	6
University of Michigan	7
Duke University	8
University of California, Los Angeles	9
Cornell University	10

What if CSS selectors match multiple tables?

Multiple options:

1. **Tweak CSS selectors to uniquely identify element (if possible)**

2. **Scrape all of them, then use familiar R tools to extract data**

Let's try option 2

Scrape all the tables

Use `html_elements()` to extract all matching elements

```
all_tables <- rankings |>
  html_elements(".wikitable") |> # extract all the .wikitables
  html_table()                   # convert html to a data frame
```

```
class(all_tables) # we get a list of tables
```

```
## [1] "list"
```

```
length(all_tables) # 11 tables, to be exact
```

```
## [1] 11
```

How could we extract individual tables?

```
## # A tibble: 3 × 2
##   `Top national universities[13]` `2022 rank`
##   <chr>                                 <int>
## 1 Princeton University                      1
## 2 Columbia University                       2
## 3 Harvard University                        2

## # A tibble: 3 × 2
##   University                            `Students' Dream  College Ranking`
##   <chr>                                                              <int>
## 1 Stanford University                                                    1
## 2 Harvard University                                                     2
## 3 University of California, Los Angeles                                  3

## # A tibble: 3 × 2
##   University                            `Parents' Dream  College Ranking`
##   <chr>                                                             <int>
## 1 Stanford University                                                   1
## 2 Princeton University                                                  2
## 3 Massachusetts Institute of Technology                                 3
```

String matching again!

```
# use str_detect() to search for tables with "Parents"
str_detect(all_tables, "Parents")
```

```
##  [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE
```

```
# or use str_which() to get position of matching object(s)
str_which(all_tables, "Parents")
```

```
## [1] 8
```

You are fulfilling your parents' dreams

```r
# now extract table(s) with "Parents"
# below we use `[]` syntax to extract the table by index
# this is because because all_tables is a list, not a data frame
all_tables[str_detect(all_tables, "Parents")]
```

```
## [[1]]
## # A tibble: 10 × 2
##    University                            `Parents' Dream  College Ranking`
##    <chr>                                                             <int>
##  1 Stanford University                                                   1
##  2 Princeton University                                                  2
##  3 Massachusetts Institute of Technology                                 3
##  4 Harvard University                                                    4
##  5 New York University                                                   5
##  6 University of Pennsylvania                                            6
##  7 University of Michigan                                                7
##  8 Duke University                                                       8
##  9 University of California, Los Angeles                                 9
## 10 Cornell University                                                   10
```

robots.txt

robots.txt

So far we have scraped data from Wikipedia without much on legality or ethics

We acknowledged that scraping is costly to "us" and "them"

But how do we know what we "should" and "should not" scrape?

The Robots Exclusion Protocol

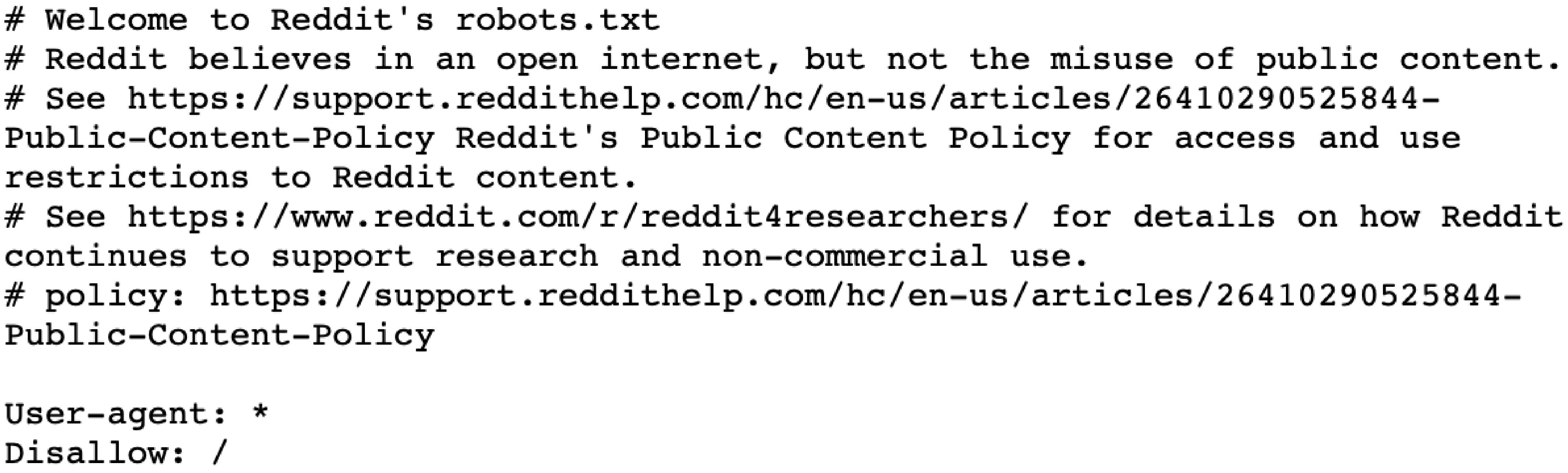
The Robots Exclusion Protocol is a *voluntary* standard for web scraping

Site owners provide guidance via `robots.txt` files

Let's take a look at some examples

Does Reddit want us to scrape data?

Reddit's robots.txt says:



```
# Welcome to Reddit's robots.txt
# Reddit believes in an open internet, but not the misuse of public content.
# See https://support.reddithelp.com/hc/en-us/articles/26410290525844-
Public-Content-Policy Reddit's Public Content Policy for access and use
restrictions to Reddit content.
# See https://www.reddit.com/r/reddit4researchers/ for details on how Reddit
continues to support research and non-commercial use.
# policy: https://support.reddithelp.com/hc/en-us/articles/26410290525844-
Public-Content-Policy

User-agent: *
Disallow: /
```

Does Wikipedia want us to scrape data?

https://en.wikipedia.org/robots.txt

For users like us (`User-agent: *`), they say that:

> "Friendly, low-speed bots are welcome viewing article pages, but not dynamically-generated pages please."

Does Yahoo Finance want us to scrape data?

https://finance.yahoo.com/robots.txt

For scraping bots like `anthropic-ai`, they disallow the entire site

For others, like us, they disallow specific pages

We can safely scrape content from other pages within the site

Prelim 2

Prelim 2 overview

Prelim 2 will cover material from **Weeks 7 through 14**

Format: paper exam, closed-book (-notes, -computer, etc.)

It will stress concepts more than syntax (but will test both)

We will have multiple question types, including but not limited to:

1. improve this data visualization
2. explain how you would approach a coding task
3. explain what this function call would return
4. explain whether/why this code would fail
5. write code snippets

Prelim 2 preparation

I will provide practice questions before Thanksgiving Break

We will host extra office hours leading up to the test:

- Monday, Dec 1: the TAs will have regular open office hours
- Tuesday, Dec 2: review session in class
- Thursday, Dec 4: no class (optional TA office hours)

Questions?