

ISA 401: Business Intelligence & Data Visualization

04: Scraping Webpages in R

Fadel M. Megahed, PhD

Associate Professor
Department of Information Systems and Analytics
Farmer School of Business
Miami University

Twitter: [FadelMegahed](#)



GitHub: [fmegahed](#)

Email: fmegahed@miamioh.edu

Office Hours: [Automated Scheduler for Virtual Office Hours](#)

Spring 2022

Quick Refresher from Last Class

- ✓ Subset data in .
- ✓ Read text-files, binary files (e.g., Excel, SAS, SPSS, Stata, etc), json files, etc.
- ✓ Export data from .

Going Over Assignment 02 Solutions

Q1

Q2

Q3

Q4

Based on the readr package, the following function is appropriate for the purpose of reading the file `patterson_cafe_yelp_reviews.rds`

✗ `read_csv()`

✗ `read_fwf()`

✓ `read_rds()`

✗ `readRDS()`

Going Over Assignment 02 Solutions

Q1	<u>Q2</u>	Q3	Q4
----	-----------	----	----

Read the file `patterson_cafe_yelp_reviews.rds` in RStudio. Based on the file, the data contains columns/variables.

```
if(require(tidyverse)==FALSE) install.packages("tidyverse")
cafe_yelp_reviews = readr::read_rds(file = "../Data/patterson_cafe_yelp_reviews.rds")
ncol(cafe_yelp_reviews)
```

```
## [1] 4
```

Going Over Assignment 02 Solutions

Q1 Q2 Q3 Q4

When you read the `patterson_cafe_yelp_reviews.rds` in R/RStudio, the column titled `review_date` is being recognized by R as

✓ character

✗ date

✗ datetime

✗ double

```
if(require(tidyverse)==FALSE) install.packages("tidyverse")
cafe_yelp_reviews = readr::read_rds(file = "../Data/patterson_cafe_yelp_reviews.rds")
class(cafe_yelp_reviews$review_date)
```

```
## [1] "character"
```

```
cafe_yelp_reviews_base = readRDS(file = "../Data/patterson_cafe_yelp_reviews_base.rds")
class(cafe_yelp_reviews_base$review_date)
```

```
## [1] "character"
```

Going Over Assignment 02 Solutions

Q1	Q2	Q3	<u>Q4</u>
----	----	----	-----------

What is the number of reviews with a reviewer score ≥ 4 ?

```
pos_review_count_approach1 = cafe_yelp_reviews$score >= 4  
sum(pos_review_count_approach1)
```


```
## [1] 76
```

```
pos_review_count_approach2 = cafe_yelp_reviews %>% filter(score >= 4)  
nrow(pos_review_count_approach2)
```

```
## [1] 76
```

```
pos_review_count_approach3 = which(cafe_yelp_reviews$score >= 4)  
length(pos_review_count_approach3)
```

Learning Objectives for Today's Class




- Understand when can we scrape data (i.e., `robots.txt`)
- Scrape a webpage Using 
- Utilize loops or `purrr::map` to download data from multiple webpages.

Web Technology

World Wide Web (WWW)

WWW (or the **Web**) is the information system where documents (web pages) are identified by Uniform Resource Locators (**URLs**)

A web page consists of:

-  **HTML** provides the basic structure of the web page
-  **CSS** controls the look of the web page (optional)
-  **JS** is a programming language that can modify the behavior of elements of the web page (optional)

Hypertext Markup Language (HTML)

- with the extension `.html`.
- rendered using a web browser via an URL.
- text files that follows a special syntax that alerts web browsers how to render it.

via a web browser

← → ↻ 🏠 🔒 Not secure | plane crash info.com/2021/2021.htm

📱 Apps 📁 Teaching 📁 Research 📁 Misc 📧 MU Mail 📅 MU Calendar 🌿 Overleaf 🎨 Canvas

2021

Date	Location / Operator	Aircraft Type / Registration	Fatalities
09 Jan 2021	Near Jakarta, Indonesia Sriwijaya Air	Boeing 737-524 PK-CLC	62/62(0)
02 Mar 2021	Pieri, Sudan South Sudan Supreme Airlines	Let L-410UVP-E HK-4274	10/10(0)
28 Mar 2021	Near Butte, Alaska Soloy Helicopters	Eurocopter AS350B3 Ecureuil N351SH	5/6(0)
21 May 2021	Near Kaduna, Nigeria Military - Nigerian Air Force	Beechcraft B300 King Air 350i NAF203	11/11(0)
10 Jun 2021	Near Pyn Oo Lwin, Myanmar Military - Myanmar Air Force	Beechcraft 1900D 4610	12/14(0)
04 Jul 2021	Patikul, Sulu, Philippines Military - Philippine Air Force	Lockheed C-130H Hercules 5125	50/96(3)
06 Jul 2021	Palana, Russia Kamchatka Aviation Enterprise	Antonov An 26B-100 RA-26085	28/28(0)
12 Sep 2021	Kazachinskoye, Russia Aeroservice/SILA	Let L-410UVP-E20 RA-67042	4/16(0)
27 Dec 2021	El Cajon, California Med Jet	Learjet 35A N880Z	4/4(0)

[Return to Home Page](#)

Copyright © Richard Kebabjian / www.plane crash info.com

via a text editor

```
1 <html>
2
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
5 <meta name="GENERATOR" content="Microsoft FrontPage 4.0">
6 <meta name="description" content="Aviation accidents">
7 <meta name="keywords" content="aircraft accident, plane crash, aviation disaster, safety, aviation safety, aviation accident,
8 aircraft, plane, statistics, airline statistics, airline, airlines, hijack, pilot, probable cause, crash, boeing, cockpit,
9 <meta name="ProgId" content="FrontPage.Editor.Document">
10 <meta name="Title" content="Aviations accidents 2021">
11 </head>
12
13 <body>
14 <p align="center"><b><font face="Arial" color="#B086FF" size="5">2021</font></b></p>
15 <div align="center">
16 <center>
17 <table border="1" cellpadding="4" cellspacing="0" width="700">
18 <tr>
19 <td width="75" bgcolor="#B086FF" align="left"><b><font face="Arial" size="2">Date</font></b></td>
20 <td bgcolor="#B086FF" align="left"><font face="Arial" size="2"><b>Location / Operator</b></font></td>
21 <td bgcolor="#B086FF" align="left"><font face="Arial" size="2"><b>Aircraft Type / Registration</b></font></td>
22 <td align="right" colspan="3"><b><font face="Arial" size="2">Fatalities</b></font></td>
23 </tr>
24 <tr>
25 <td align="left" colspan="4"><b><font face="Arial" size="2">2021</b></td>
26 </tr>
27 <tr>
28 <td align="left" colspan="4"><b><font face="Arial" size="2">09 Jan 2021</b></td>
29 <td align="left" colspan="4"><b><font face="Arial" size="2">Near Jakarta, Indonesia</b></td>
30 <td align="left" colspan="4"><b><font face="Arial" size="2">Boeing 737-524</b></td>
31 <td align="right" colspan="4"><b><font face="Arial" size="2">62/62(0)</b></td>
32 </tr>
33 <tr>
34 <td align="left" colspan="4"><b><font face="Arial" size="2">02 Mar 2021</b></td>
35 <td align="left" colspan="4"><b><font face="Arial" size="2">Pieri, Sudan</b></td>
36 <td align="left" colspan="4"><b><font face="Arial" size="2">Let L-410UVP-E</b></td>
37 <td align="right" colspan="4"><b><font face="Arial" size="2">10/10(0)</b></td>
38 </tr>
39 <tr>
40 <td align="left" colspan="4"><b><font face="Arial" size="2">28 Mar 2021</b></td>
41 <td align="left" colspan="4"><b><font face="Arial" size="2">Near Butte, Alaska</b></td>
42 <td align="left" colspan="4"><b><font face="Arial" size="2">Eurocopter AS350B3 Ecureuil</b></td>
43 <td align="right" colspan="4"><b><font face="Arial" size="2">5/6(0)</b></td>
44 </tr>
```

HTML Structure

```
<!DOCTYPE html>

<html>
  <!--This is a comment and ignored by web client.-->
  <head>
    <!--This section contains web page metadata.-->
    <title>ISA 401: Business Intelligence and Data Viz</title>
    <meta name="author" content="Fadel Megahed">
    <link rel="stylesheet" href="css/styles.css">
  </head>

  <body>
    <!--This section contains what you want to display on your web page.-->
    <h1>I'm a first level header</h1>
    <p>This is a <b>paragraph</b>.</p>
  </body>
</html>
```

HTML Syntax

`Author content` Author content

start tag: `Author content`

end tag: `Author content`

content: `Author content`

element name: `Author content`

attribute: `Author content`

attribute name: `Author content`

attribute value: `Author content`

Not all HTML tags have an end tag, for example:

`` → 

HTML Elements

block element:	<code><div>content</div></code>
inline element:	<code>content</code>
paragraph:	<code><p>content</p></code>
header level 1:	<code><h1>content</h1></code>
header level 2:	<code><h2>content</h2></code>
italic:	<code><i>content</i></code>
emphasised text:	<code>content</code>
strong importance:	<code>content</code>
link:	<code>content</code>
unordered list:	<code> item 1 item 2 </code>

Cascading Style Sheet (CSS)

- with the extension `.css`
- 3 ways to style elements in HTML:
 - **inline** by using the `style` attribute inside HTML start tag:
`<h1 style="color:blue;">Blue Header</h1>`
 - **externally** by using the `<link>` element:
`<link rel="stylesheet" href="styles.css">`
 - **internally** by defining within `<style>` element:

```
<style type="text/css">  
h1 { color: blue; }  
</style>
```

By convention, the `<style>` and `<link>` elements tend to go into the `<head>` section of the HTML document.

CSS Syntax

```
<style type="text/css">
h1 { color: blue; }
</style>
<h1>This is a header</h1>
```

This is a header

selector:	h1 { color: blue; }
property:	h1 { color: blue; }
property name:	h1 { color: blue; }
property value:	h1 { color: blue; }

You may have multiple properties for a single selector. ➡

```
h1 {
  color: blue;
  font-size: 16pt;
}
```

CSS Properties

```
<div>Sample text</div>
```

background color:	<code>div { background-color: yellow; }</code>	<div>Sample text</div>
text color:	<code>div { color: purple; }</code>	<div>Sample text</div>
border:	<code>div { border: 1px dashed brown; }</code>	<div>Sample text</div>
left border only:	<code>div { border-left: 10px solid pink; }</code>	<div>Sample text</div>
text size:	<code>div { font-size: 10pt; }</code>	<div>Sample text</div>
padding:	<code>div { background-color: yellow; padding: 10px; }</code>	<div>Sample text</div>
margin:	<code>div { background-color: yellow; margin: 10px; }</code>	<div>Sample text</div>

CSS Selector

<code>.classname</code>	selects all elements with the attribute <code>class="classname"</code> .
<code>.c1.c2</code>	selects all elements with <i>both</i> <code>c1</code> and <code>c2</code> within its class attribute.
<code>.c1 .c2</code>	selects all elements with class <code>c2</code> that is a descendant of an element with class <code>c1</code> .
<code>#p1</code>	selects all elements with the attribute <code>id="p1"</code> .

```
<h1>This is a sample htr
```

```
<blockquote>
```

```
<p>Maybe stories are jus
```

```
<footer>-Brene Brown</footer>
```

```
</blockquote>
```

```
<div id="p1" class="parent">
```

```
  Hmm
```

```
  <p>Hi!</p>
```

```
  How are you?
```

```
  <div class="child nice">
```

```
    <p>Hello!</p>
```

```
  </div>
```

```
</div>
```

```
<p>Household 1</p>
```

```
<div class="parent">
```

```
  <p>Hi!</p>
```

```
  <blockquote class="child rebel">
```

```
    <p>Don't talk to me!</p>
```

```
  </blockquote>
```

```
</div>
```

```
<span class="child">
```

```
<span class="parent child rebel">
```

```
  <p>Clean your room!</p>
```

```
</span>
```

```
</span>
```

Unlike `class`, you can only have one `id` value and must be unique in the whole HTML document.

JavaScript (JS)*

- JS is a programming language and enable interactive components in HTML documents.
- 2 ways to insert JS into a HTML document:
 - **internally** by defining within `<script>` element:

```
<script>  
document.getElementById("p1").innerHTML = "content";  
</script>
```

- **externally** by using the `src` attribute to refer to the external file:

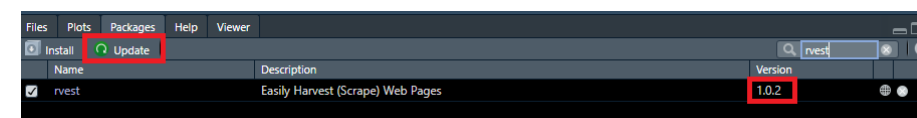
```
<script src="js/myjs.js"></script>
```

Web Scrapping

rvest: Step 1 - Reading Static HTML Pages in R



Use `{rvest}` \geq v1.0.2 (if not, update)



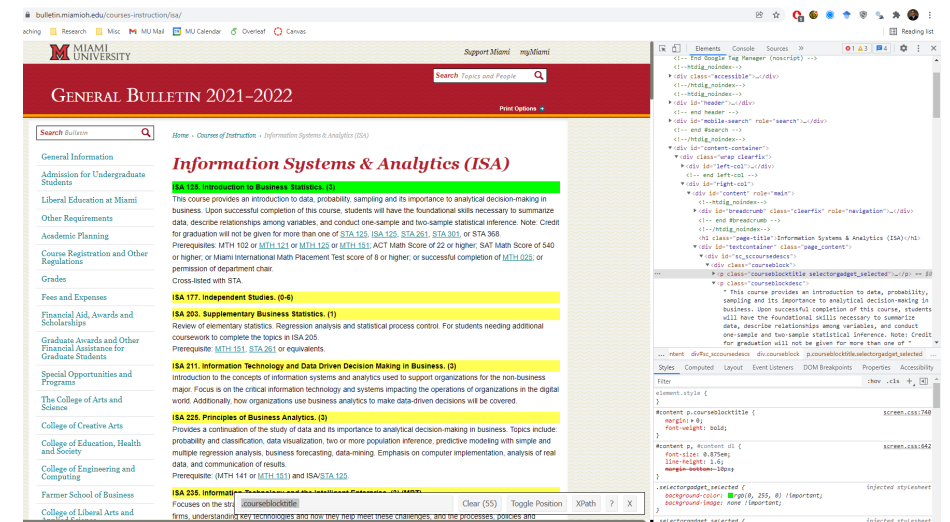
```
if(require(pacman)==FALSE) install.packages("pacman")
pacman::p_load(rvest)
isa_courses = read_html("http://bulletin.brown.edu/undergraduate/academic-requirements/requirements-for-the-degree-in-science/")
isa_courses
```

```
## {html_document}
## <html xml:lang="en" lang="en" dir="ltr">
## [1] <head>\n<title>Information System ...
## [2] <body>\n\n\n\n\n\n\n\n<!-- Google Tag ...
```

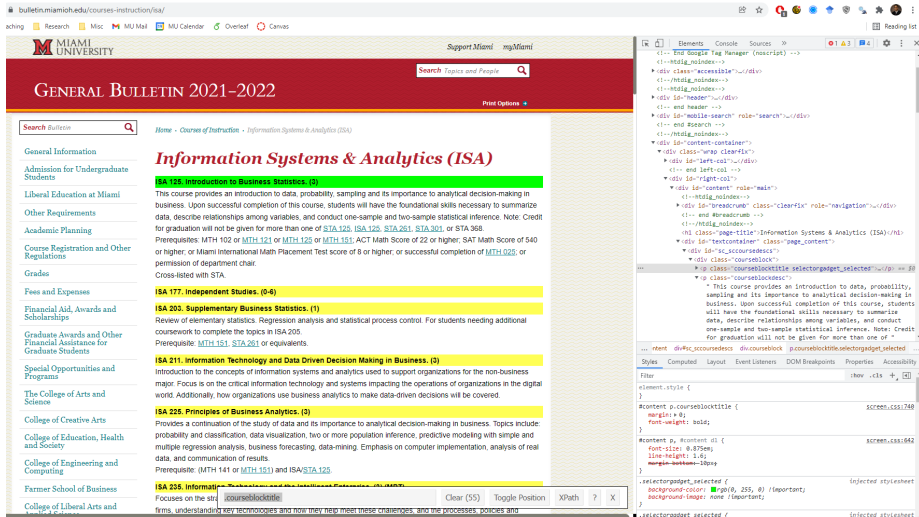
rvest: Step 2 - Selecting HTML Elements

 Inspector

 Selector Gadget



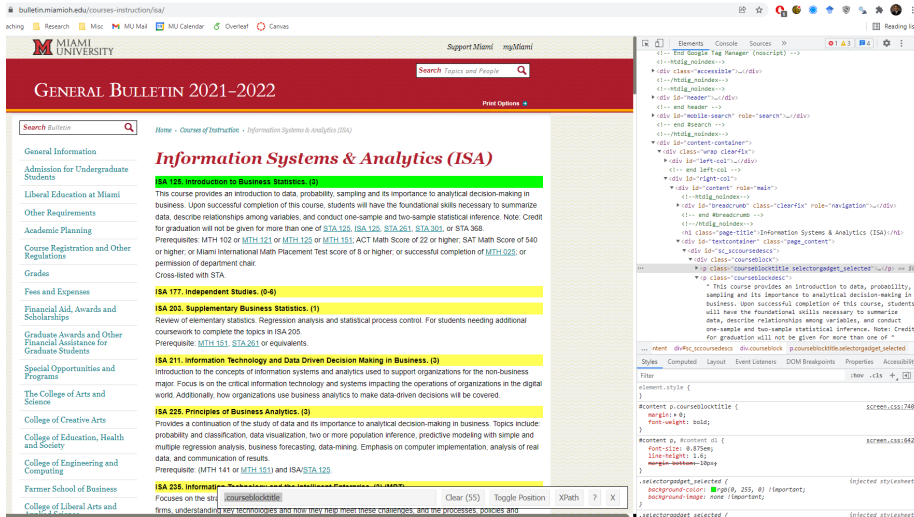
rvest: Step 2 - Selecting HTML Elements



```
isa_course_titles = isa_courses %>%  
  html_elements(css = "p.courseblocktitle")  
isa_course_titles
```

```
## {xml_nodeset (55)}  
## [1] <p class="courseblocktitle"><str ...  
## [2] <p class="courseblocktitle"><str ...  
## [3] <p class="courseblocktitle"><str ...  
## [4] <p class="courseblocktitle"><str ...  
## [5] <p class="courseblocktitle"><str ...  
## [6] <p class="courseblocktitle"><str ...  
## [7] <p class="courseblocktitle"><str ...  
## [8] <p class="courseblocktitle"><str ...  
## [9] <p class="courseblocktitle"><str ...  
## [10] <p class="courseblocktitle"><str ...  
## [11] <p class="courseblocktitle"><str ...  
## [12] <p class="courseblocktitle"><str ...  
## [13] <p class="courseblocktitle"><str ...  
## [14] <p class="courseblocktitle"><str ...  
## [15] <p class="courseblocktitle"><str ...
```

rvest: Step 3 - Getting HTML Text



```
isa_course_titles_en = isa_course_titles  
html_text2()  
  
isa_course_titles_en
```

```
## [1] "ISA 125. Introduction to Business Statistics.  
## [2] "ISA 177. Independent Studies. (0-6)"  
## [3] "ISA 203. Supplementary Business Statistics.  
## [4] "ISA 211. Information Technology and Data D  
## [5] "ISA 225. Principles of Business Analytics.  
## [6] "ISA 235. Information Technology and the In  
## [7] "ISA 241. Database for Analytics. (1.5)"  
## [8] "ISA 242. Programming for Analytics. (1.5)"  
## [9] "ISA 243. Database and Programming for Anal  
## [10] "ISA 245. Database Systems and Data Warehou  
## [11] "ISA 250. Basic Math for Analytics. (3)"  
## [12] "ISA 277. Independent Studies. (0-6)"  
## [13] "ISA 281. Concepts in Business Programming.  
## [14] "ISA 291. Applied Regression Analysis in Bu  
## [15] "ISA 301. Business Data Communications and  
## [16] "ISA 303. Enterprise Systems. (3) (MPT)"
```

Demo 1: Scraping the Course Descriptions

- We will build on the previous example and we will scrape the **course descriptions** associated with these courses.
- Then, we will create a **tibble** containing **both** the **course titles** and **descriptions**
- Then, we will **export the results to a CSV** so that we can analyze that in a separate program if we wanted to.

Non-Graded Class Activity

Activity	Your Solution	My Solution
----------	---------------	-------------

- Go to [this database on plane crashes](#)
- Scrape the HTML table. **Note the difference from text elements:**
 - The CSS selector for `html_elements()` will be different.
 - You will extract a table (in its **entirety**) and hence:
 - we will use `html_table()` instead of `html_text2()`
- Store the scraped data in an appropriate location on your computer (e.g., within the data folder for ISA 401)

04:00

Non-Graded Class Activity

Activity

Your Solution

My Solution

Over the next 4 minutes, use an R script file to perform the tasks outline in the activity panel.

04:00

Non-Graded Class Activity

Activity

Your Solution

My Solution

Please refer to our discussion in class

Demo 2: Scraping all Plane Crashes 2012-2021

- We will build on the previous example and we will scrape all the plane crashes that were recorded in the [plane crash database](#) between 2012-2021.
- Then, we will create a single **tibble** for all crashes. It will contain the fields in the individual tables as well as the year of crash.
- Then, we will **export the results to a CSV** so that we can analyze that in a separate program if we wanted to.

Demo 3: Downloading the Lecture PDFs (if time allows)

If time allows, we will download the [PDFs of the ISA 401 lectures from GitHub](#).

Legal and Ethical Issues with Web Scraping

Robots.txt

When scraping/crawling the web you need to be aware of `robots.txt`.

The robots exclusion standard, also known as the robots exclusion protocol or simply robots.txt, is a standard used by websites to communicate with web crawlers and other web robots. The standard specifies how to inform the web robot about which areas of the website should not be processed or scanned. --- [Wikipedia](#)

Using the excellent `robotstxt`  to check if scraping/crawling a specific directory is allowed.

```
if(require(robotstxt)==FALSE) install.packages("robotstxt")
robotstxt::paths_allowed(paths = "2021/", domain = "planecrashinfo.com", bot = "*")
```

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

Terms of Service

Most large companies have **terms of service** that supplement what is permitted and/or disallowed on their `robots.txt` file. Examples include:

- [Yelp's US Terms of Service](#)
- [LinkedIn Terms of Service](#)

Ethical/Legal Considerations

- **Use of publicly available reviews as a part of your service:** Would you classify the **Yelp vs Google Feud** as such an example?



Jeremy Stoppelman 
@jeremys

Wow Google, congrats on a new low. Consumer searches for Yelp gets "reviews" which are Google Ads.

About 7,020,000 results (0.84 seconds)

HVAC pros serving San Francisco Sponsored ⓘ

The Appliance Repair Do.. 4.6 ★★★★★ · See reviews ✓ Google guaranteed Alameda (510) 871-3938 Open now	Healthy Duct Cleaning S... 4.9 ★★★★★ · See reviews ✓ Google guaranteed Daly City (415) 993-1965 Open now	Atlas Trillo Heating & Air 4.5 ★★★★★ · See reviews ✓ Google guaranteed San Jose (408) 915-7800 Opens Tue at 8 AM
---	--	--

→ More HVAC pros in San Francisco

Heating & Air Conditioning/HVAC in San Francisco - Yelp
<https://www.yelp.com/c/sf/hvac> ▼
The Best Heating & Air Conditioning/HVAC in San Francisco on Yelp. Read about places like: Air Flow Pros Heating And Air Conditioning, Kohler Heating, ...

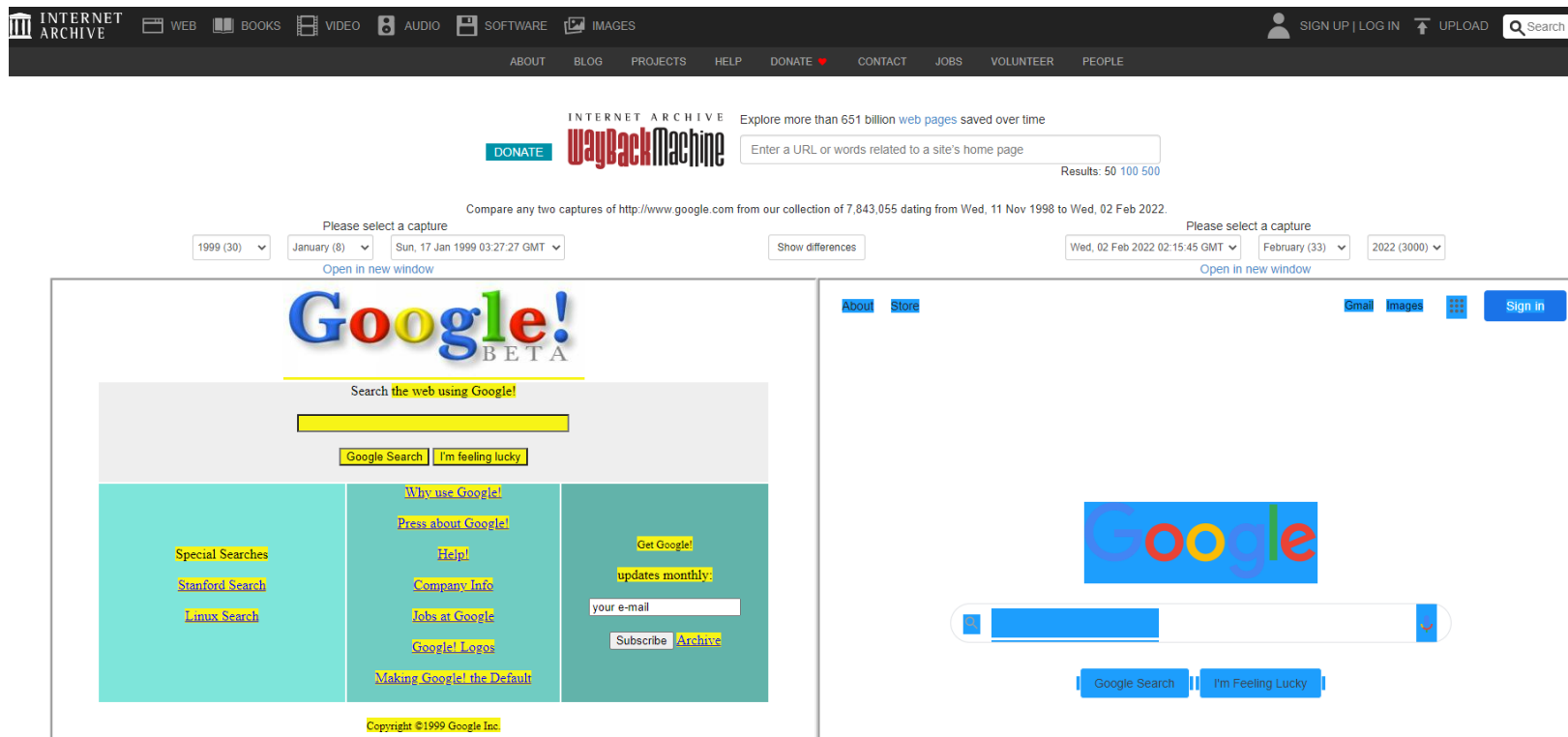
Best Hvac contractors in San Francisco, CA - Yelp
https://www.yelp.com/search?find_desc=hvac+contractors&find_loc=San+Francisco+CA

Ethical/Legal Considerations

- **Use of publicly available profiles as a part of your service:**
 - LinkedIn vs Hiq Labs: Ninth Circuit Decision in 2019
 - Revival of Case in 2021 by Supreme Court

Ethical/Legal Considerations

- What about scraping entire websites/webpages for the purpose of archiving the internet?




The evolution of the home page for Google per the Wayback Machine

Recap

Summary of Main Points

By now, you should be able to do the following:

- Understand when can we scrape data (i.e., `robots.txt`)
- Scrape a webpage Using 
- Utilize loops or `purrr::map` to download data from multiple webpages.

Supplementary Reading



- PDF of Published Paper
- ePub of Published Paper

- Selector Gadget
- Getting Started with rvest
- Practical Web Scraping in R