# ISA 401: Business Intelligence & Data Visualization

## 04: Scraping Webpages in R

Fadel M. Megahed, PhD

Enders Associate Professor
Farmer School of Business
Miami University

@FadelMegahed
fmegahed
fmegahed@miamioh.edu
Automated Scheduler for Office Hours

Fall 2022

# Quick Refresher from Last Class

✅ Subset data in R.

✅ Read text-files, binary files (e.g., Excel, SAS, SPSS, Stata, etc), json files, etc.

✅ Export data from R.

# Going Over Assignment 03 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 03 Solutions

Q1 | Q2 | 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 03 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
class(cafe_yelp_reviews$review_date)
```

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

```
cafe_yelp_reviews_base = readRDS(file = "../../data/patterson_cafe_ye
class(cafe_yelp_reviews_base$review_date)
```

# Going Over Assignment 03 Solutions

Q1 Q2 Q3 Q4

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] 82
```

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

```
## [1] 82
```

```
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 

# Web Technology 


Source: Slides 5-15 are from Dr Earo Wang's STAT 220 Web Scraping Slides, which were adapted from Dr Emi Tanaka's "Communicating with Data" course.

# World Wide Web (WWW)

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

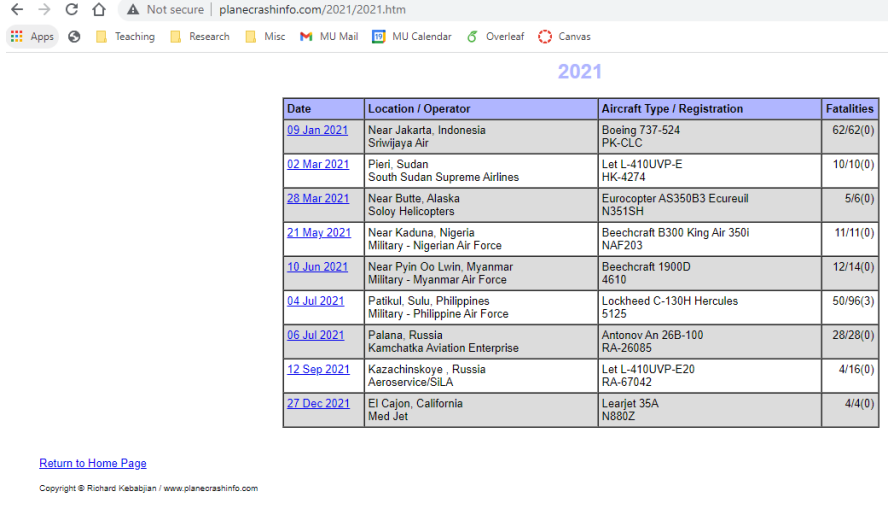
A web page consists of:

- **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



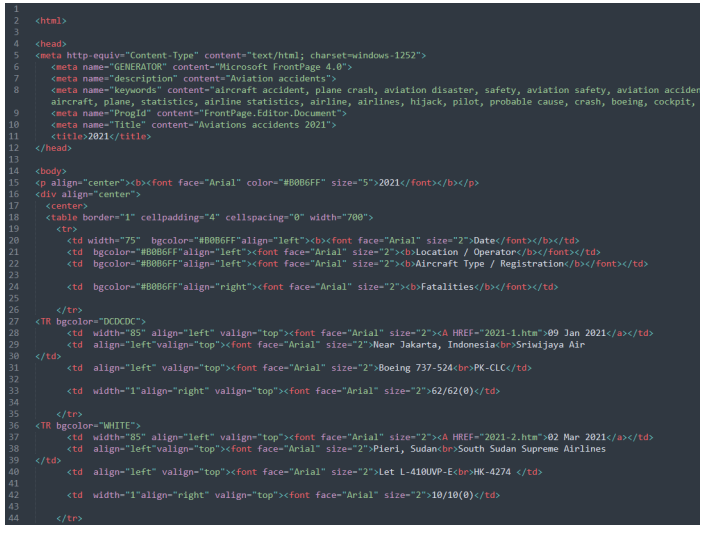

2021

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

Return to Home Page



## via a text editor



```
<html>

<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
  <meta name="GENERATOR" content="Microsoft FrontPage 4.0">
  <meta name="description" content="Aviation accidents">
  <meta name="keywords" content="aircraft accident, plane crash, aviation disaster, safety, aviation safety, aviation accider
  aircraft, plane, statistics, airline statistics, airline, airlines, hijack, pilot, probable cause, crash, boeing, cockpit,
  <meta name="ProgId" content="FrontPage.Editor.Document">
  <meta name="Title" content="Aviations accidents 2021">
  <title>2021</title>
</head>

<body>
<p align="center"><b><font face="Arial" color="#B0B6FF" size="5">2021</font></b></p>
<div align="center">
  <center>
  <table border="1" cellpadding="4" cellspacing="0" width="700">
    <tr>
      <td width="75"  bgcolor="#B0B6FF"align="left"><b><font face="Arial" size="2">Date</font></b></td>
      <td  bgcolor="#B0B6FF"align="left"><font face="Arial" size="2"><b>Location / Operator</b></font></td>
      <td  bgcolor="#B0B6FF"align="left"><font face="Arial" size="2"><b>Aircraft Type / Registration</b></font></td>

      <td  bgcolor="#B0B6FF"align="right"><font face="Arial" size="2"><b>Fatalities</b></font></td>

    </tr>
<TR bgcolor="DCDCDC">
      <td  width="85" align="left" valign="top"><font face="Arial" size="2"><A HREF="2021-1.htm">09 Jan 2021</a></td>
      <td  align="left"valign="top"><font face="Arial" size="2">Near Jakarta, Indonesia<br>Sriwijaya Air
</td>
      <td  align="left" valign="top"><font face="Arial" size="2">Boeing 737-524<br>PK-CLC</td>

      <td  width="1"align="right" valign="top"><font face="Arial" size="2">62/62(0)</td>

    </tr>
<TR bgcolor="WHITE">
      <td  width="85" align="left" valign="top"><font face="Arial" size="2"><A HREF="2021-2.htm">02 Mar 2021</a></td>
      <td  align="left"valign="top"><font face="Arial" size="2">Pieri, Sudan<br>South Sudan Supreme Airlines
</td>
      <td  align="left" valign="top"><font face="Arial" size="2">Let L-410UVP-E<br>HK-4274 </td>

      <td  width="1"align="right" valign="top"><font face="Arial" size="2">10/10(0)</td>

    </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:

<img height="40px" src="https://tinyurl.com/rlogo-svg"> → 

# HTML Elements

| | |
|---|---|
| block element: | `<div>content</div>` |
| inline element: | `<span>content</span>` |
| paragraph: | `<p>content</p>` |
| header level 1: | `<h1>content</h1>` |
| header level 2: | `<h2>content</h2>` |
| italic: | `<i>content</i>` |
| emphasised text: | `<em>content</em>` |
| strong importance: | `<strong>content</strong>` |
| link: | `<a href="https://github.com/fmegahed/isa401">content</a>` |
| unordered list: | `<ul>`<br>`<li>item 1</li>`<br>`<li>item 2</li>`<br>`</ul>` |

# 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: | `div { background-color: yellow; }` | Sample text |
| text color: | `div { color: purple; }` | Sample text |
| border: | `div { border: 1px dashed brown; }` | Sample text |
| left border only: | `div { border-left: 10px solid pink; }` | Sample text |
| text size: | `div { font-size: 10pt; }` | Sample text |
| padding: | `div { background-color: yellow; padding: 10px; }` | Sample text |
| margin: | `div { background-color: yellow; margin: 10px; }` | Sample text |

# CSS Properties

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

| | | |
|---|---|---|
| center align text: | `div { background-color: yellow; padding-top: 20px; text-align: center; }` | Sample text |
| font family: | `div { font-family: Marker Felt, times; }` | Sample text |
| strike: | `div { text-decoration: line-through; }` | ~~Sample text~~ |
| underline: | `div { text-decoration: underline; }` | Sample text |
| opacity: | `div { opacity: 0.3 }` | Sample text |

# CSS Selector

| | |
|---|---|
| `*` | selects all elements |
| `div` | selects all `<div>` elements |
| `div, p` | selects all `<div>` and `<p>` elements |
| `div p` | selects all `<p>` within `<div>` |
| `div > p` | selects all `<p>` one level deep in `<div>` |
| `div + p` | selects all `<p>` immediately after a `<div>` |
| `div ~ p` | selects all `<p>` preceded by a `<div>` |

```
<h1>This is a sample html</h1>

<blockquote>
<p>Maybe stories are just data with a soul.</p>
<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>
```

# CSS Selector

| | |
|---|---|
| `*` | selects all elements |
| `div` | selects all `<div>` elements |
| `div, p` | selects all `<div>` and `<p>` elements |
| `div p` | selects all `<p>` within `<div>` |
| `div > p` | selects all `<p>` one level deep in `<div>` |
| `div + p` | selects all `<p>` immediately after a `<div>` |
| `div ~ p` | selects all `<p>` preceded by a `<div>` |

```
<h1>This is a sample html</h1>

<blockquote>
<p>Maybe stories are just data with a soul.</p>
<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>
```

# CSS Selector

| | | |
|---|---|---|
| `*` | | selects all elements |
| `blockquote` | | selects all `<blockquote>` elements |
| `div, p` | | selects all `<div>` and `<p>` elements |
| `div p` | | selects all `<p>` within `<div>` |
| `div > p` | | selects all `<p>` one level deep in `<div>` |
| `div + p` | | selects all `<p>` immediately after a `<div>` |
| `div ~ p` | | selects all `<p>` preceded by a `<div>` |

```
<h1>This is a sample html</h1>

<blockquote>
<p>Maybe stories are just data with a soul.</p>
<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>
```

# CSS Selector

| | |
|---|---|
| `*` | selects all elements |
| `div` | selects all `<div>` elements |
| `div, p` | selects all `<div>` and `<p>` elements |
| `div p` | selects all `<p>` within `<div>` |
| `div > p` | selects all `<p>` one level deep in `<div>` |
| `div + p` | selects all `<p>` immediately after a `<div>` |
| `div ~ p` | selects all `<p>` preceded by a `<div>` |

```
<h1>This is a sample html</h1>

<blockquote>
<p>Maybe stories are just data with a soul.</p>
<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>
</span>
</div>

<span class="child">
<span class="parent child rebel">
  <p>Clean your room!</p>
</span>
</span>
```

# CSS Selector

| | |
|---|---|
| `*` | selects all elements |
| `div` | selects all `<div>` elements |
| `div, p` | selects all `<div>` and `<p>` elements |
| `div p` | selects all `<p>` within `<div>` |
| `div > p` | selects all `<p>` one level deep in `<div>` |
| `div + p` | selects all `<p>` immediately after a `<div>` |
| `div ~ p` | selects all `<p>` preceded by a `<div>` |

```
<h1>This is a sample html</h1>

<blockquote>
<p>Maybe stories are just data with a soul.</p>
<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>
```

# CSS Selector

| | |
|---|---|
| `*` | selects all elements |
| `div` | selects all `<div>` elements |
| `div, p` | selects all `<div>` and `<p>` elements |
| `p div` | selects all `<div>` within `<p>` |
| `div > p` | selects all `<p>` one level deep in `<div>` |
| `div + p` | selects all `<p>` immediately after a `<div>` |
| `div ~ p` | selects all `<p>` preceded by a `<div>` |

```
<h1>This is a sample html</h1>

<blockquote>
<p>Maybe stories are just data with a soul.</p>
<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>
```

# CSS Selector

| | |
|---|---|
| `*` | selects all elements |
| `div` | selects all `<div>` elements |
| `div, p` | selects all `<div>` and `<p>` elements |
| `div p` | selects all `<p>` within `<div>` |
| `div > p` | selects all `<p>` one level deep in `<div>` |
| `div + p` | selects all `<p>` immediately after a `<div>` |
| `div ~ p` | selects all `<p>` preceded by a `<div>` |

Ignores inline elements like `span`, `i`, `b`,...

```
<h1>This is a sample htr

<blockquote>
<p>Maybe stories are just data with a soul.</p>
<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>
```

# CSS Selector

| | |
|---|---|
| `*` | selects all elements |
| `div` | selects all `<div>` elements |
| `div, p` | selects all `<div>` and `<p>` elements |
| `div p` | selects all `<p>` within `<div>` |
| `div > p` | selects all `<p>` one level deep in `<div>` |
| `div + p` | selects all `<p>` immediately after a `<div>` |
| `div ~ p` | selects all `<p>` preceded by a `<div>` |

Ignores inline elements like `span`, `i`, `b`,...

```
<h1>This is a sample htr

<blockquote>
<p>Maybe stories are just data with a soul.</p>
<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>
```

# CSS Selector

| | |
|---|---|
| `*` | selects all elements |
| `div` | selects all `<div>` elements |
| `div, p` | selects all `<div>` and `<p>` elements |
| `div p` | selects all `<p>` within `<div>` |
| `div > p` | selects all `<p>` one level deep in `<div>` |
| `div + p` | selects all `<p>` immediately after a `<div>` |
| `div ~ p` | selects all `<p>` preceded by a `<div>` |

```
<h1>This is a sample html</h1>

<blockquote>
<p>Maybe stories are just data with a soul.</p>
<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>
```

# CSS Selector

| | |
|---|---|
| `.classname` | selects all elements with the attribute `class="classname"`. |
| `.c1.c2` | selects all elements with *both* `c1` and `c2` within its class attribute. |
| `.c1 .c2` | selects all elements with class `c2` that is a descendant of an element with class `c1`. |
| `#idname` | selects all elements with the attribute `id="idname"`. |

```
<h1>This is a sample html</h1>

<blockquote>
<p>Maybe stories are just data with a soul.</p>
<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>
```

# CSS Selector

| | |
|---|---|
| `.parent` | selects all elements with the attribute `class="parent"`. |
| `.c1.c2` | selects all elements with *both* `c1` and `c2` within its class attribute. |
| `.c1 .c2` | selects all elements with class `c2` that is a descendant of an element with class `c1`. |
| `#idname` | selects all elements with the attribute `id="idname"`. |

Note some offspring do not inherit class from their parents.

```
<h1>This is a sample htr

<blockquote>
<p>Maybe stories are just data with a soul.</p>
<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>
```

# CSS Selector

| | |
|---|---|
| `.classname` | selects all elements with the attribute `class="classname"`. |
| `.child.rebel` | selects all elements with *both* `child` and `rebel` within its class attribute. |
| `.c1 .c2` | selects all elements with class `c2` that is a descendant of an element with class `c1`. |
| `#idname` | selects all elements with the attribute `id="idname"`. |

```
<h1>This is a sample html</h1>

<blockquote>
<p>Maybe stories are just data with a soul.</p>
<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>
```

# CSS Selector

| | |
|---|---|
| `.classname` | selects all elements with the attribute `class="classname"`. |
| `.c1.c2` | selects all elements with *both* `c1` and `c2` within its class attribute. |
| `.parent .rebel` | selects all elements with class `rebel` that is a descendant of an element with class `parent`. |
| `#idname` | selects all elements with the attribute `id="idname"`. |

```
<h1>This is a sample html</h1>

<blockquote>
<p>Maybe stories are just data with a soul.</p>
<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>
```

# CSS Selector

| | |
|---|---|
| `.classname` | selects all elements with the attribute `class="classname"`. |
| `.c1.c2` | selects all elements with *both* `c1` and `c2` within its class attribute. |
| `.c1 .c2` | selects all elements with class `c2` that is a descendant of an element with class `c1`. |
| `#p1` | selects all elements with the attribute `id="p1"`. |

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

```
<h1>This is a sample htm

<blockquote>
<p>Maybe stories are just
<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>
```

# 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:

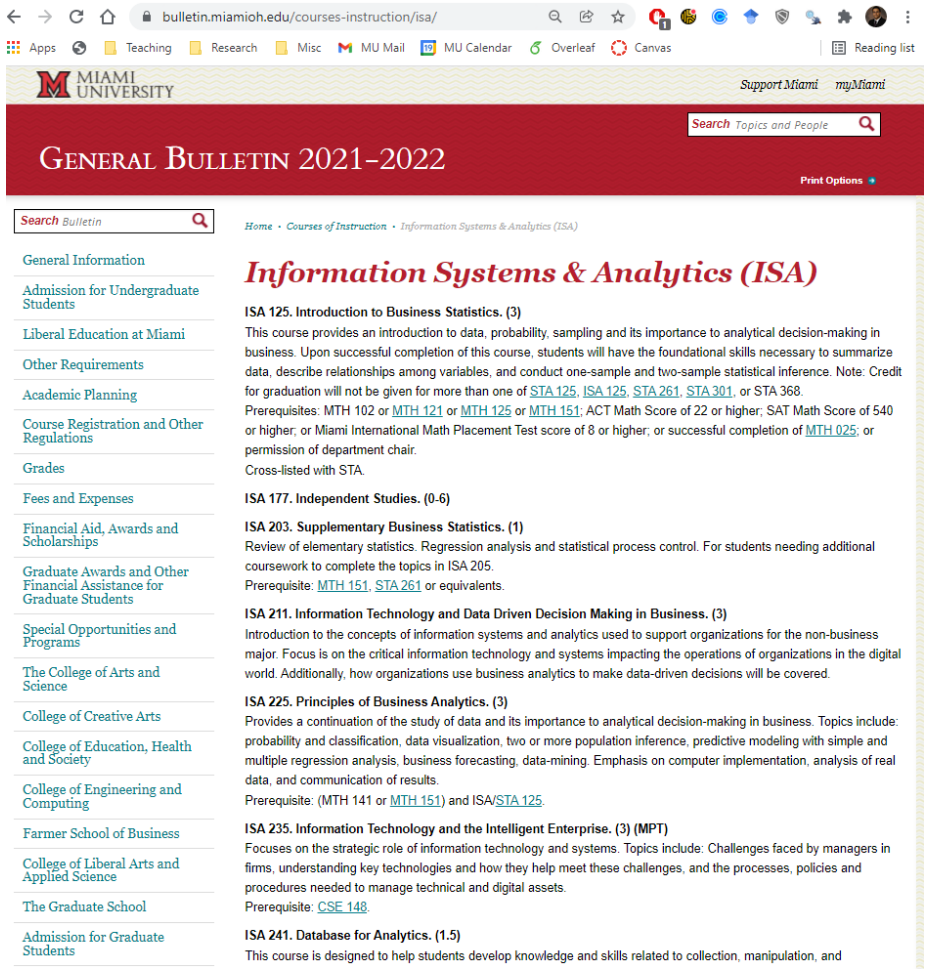
```
<script>
document.getElementById("p1").innerHTML = "content";
</script>
```

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

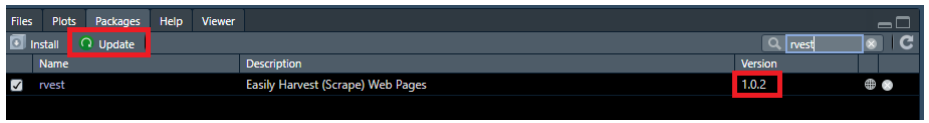
```
<script src="js/myjs.js"></script>
```

# Web Scraping 

# rvest: Step 1 - Reading Static HTML Pages in R




Use {rvest} `>= v1.0.2` (if not, update)

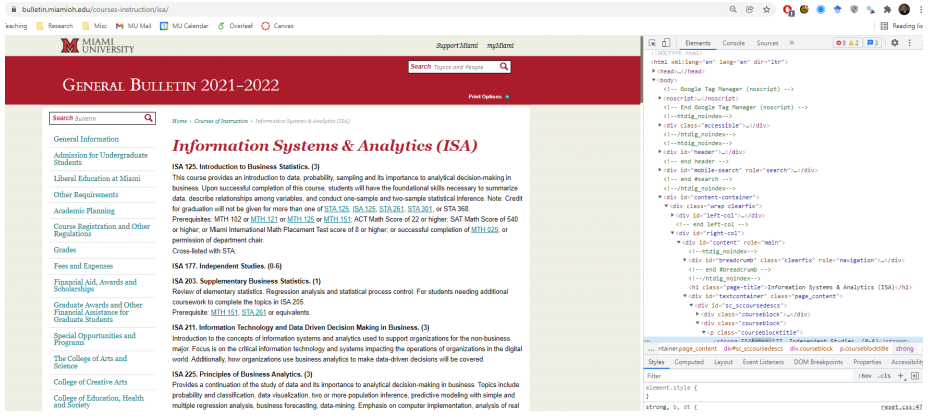



```
if(require(pacman)==FALSE) install.packag
pacman::p_load(rvest)
isa_courses = read_html("http://bulletin.
isa_courses
```

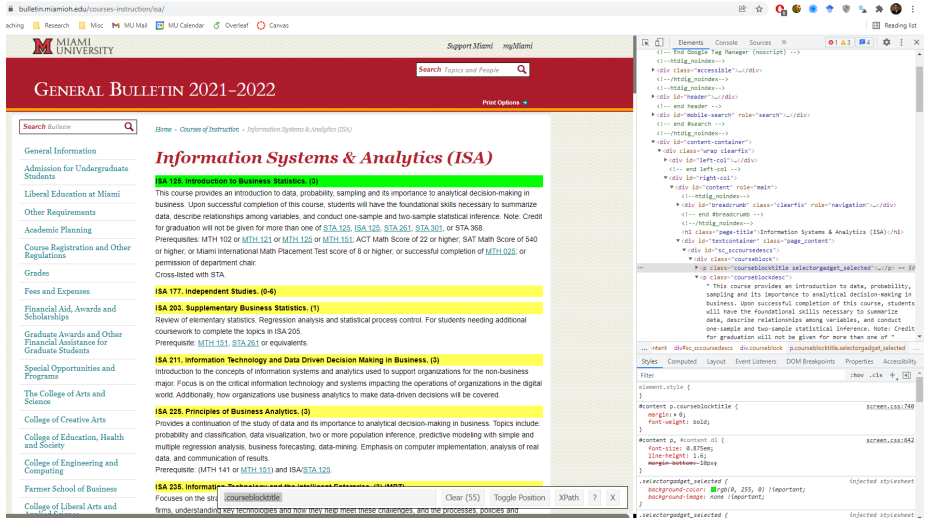
```
## {html_document}
## <html xml:lang="en" lang="en" dir="ltr
## [1] <head>\n<title>Information Systems
## [2] <body>\n\n\n\n\n\n<!-- Google Tag
```

# rvest: Step 2 - Selecting HTML Elements

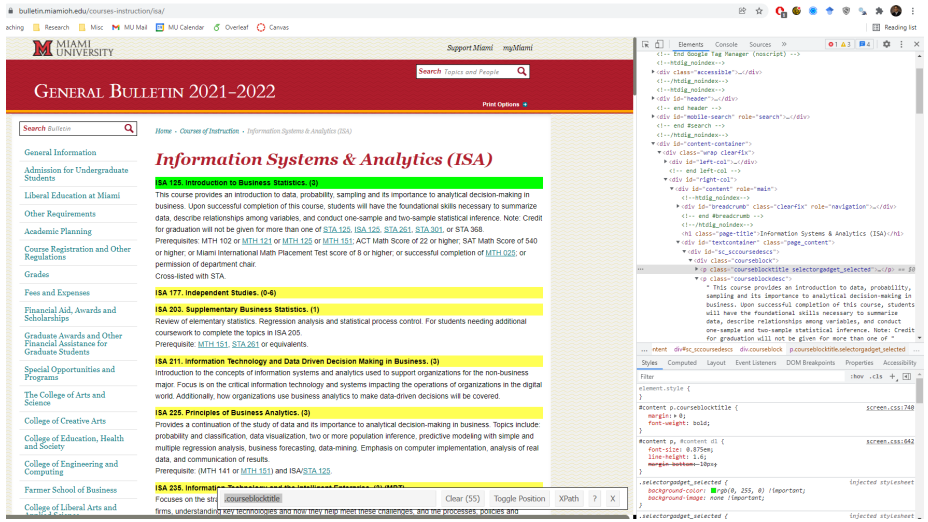
## Inspector




## Selector Gadget



# rvest: Step 2 - Selecting HTML Elements

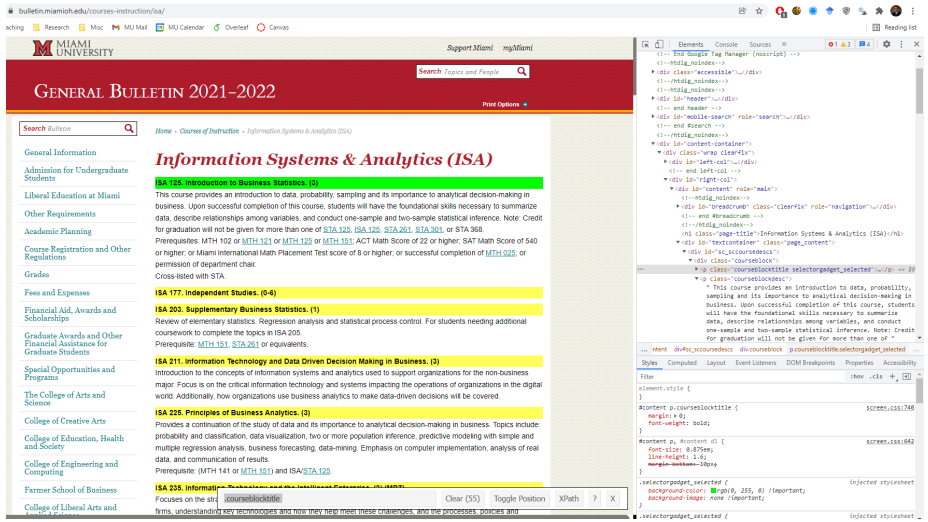



```
isa_course_titles = isa_courses %>%
  html_elements(css = "p.courseblocktitle

isa_course_titles
```

```
## {xml_nodeset (52)}
##  [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 Busines
##  [2] "ISA 177. Independent Studies. (0
##  [3] "ISA 203. Supplementary Business
##  [4] "ISA 211. Information Technology
##  [5] "ISA 225. Principles of Business
##  [6] "ISA 235. Information Technology
##  [7] "ISA 241. Database for Analytics.
##  [8] "ISA 242. Programming for Analyti
##  [9] "ISA 243. Database and Programmin
## [10] "ISA 245. Database Systems and Da
## [11] "ISA 250. Basic Math for Analytic
## [12] "ISA 277. Independent Studies. (0
## [13] "ISA 281. Concepts in Business Pr
## [14] "ISA 291. Applied Regression Anal
## [15] "ISA 301. Business Data Communica
## [16] "ISA 303. Enterprise Systems. (3)
```

# 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

04:00

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)

# Non-Graded Class Activity

04:00

Activity | Your Solution | My Solution

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

# Non-Graded Class Activity

04:00

Activity | Your Solution | My Solution

**Please refer to our discussion in class**

# Demo 2: Scraping all Plane Crashes 2013-2022

- We will build on the previous example and we will scrape all the plane crashes that were recorded in the plane crash database between 2013-2022.
- 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.

# 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  = "2022/", 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

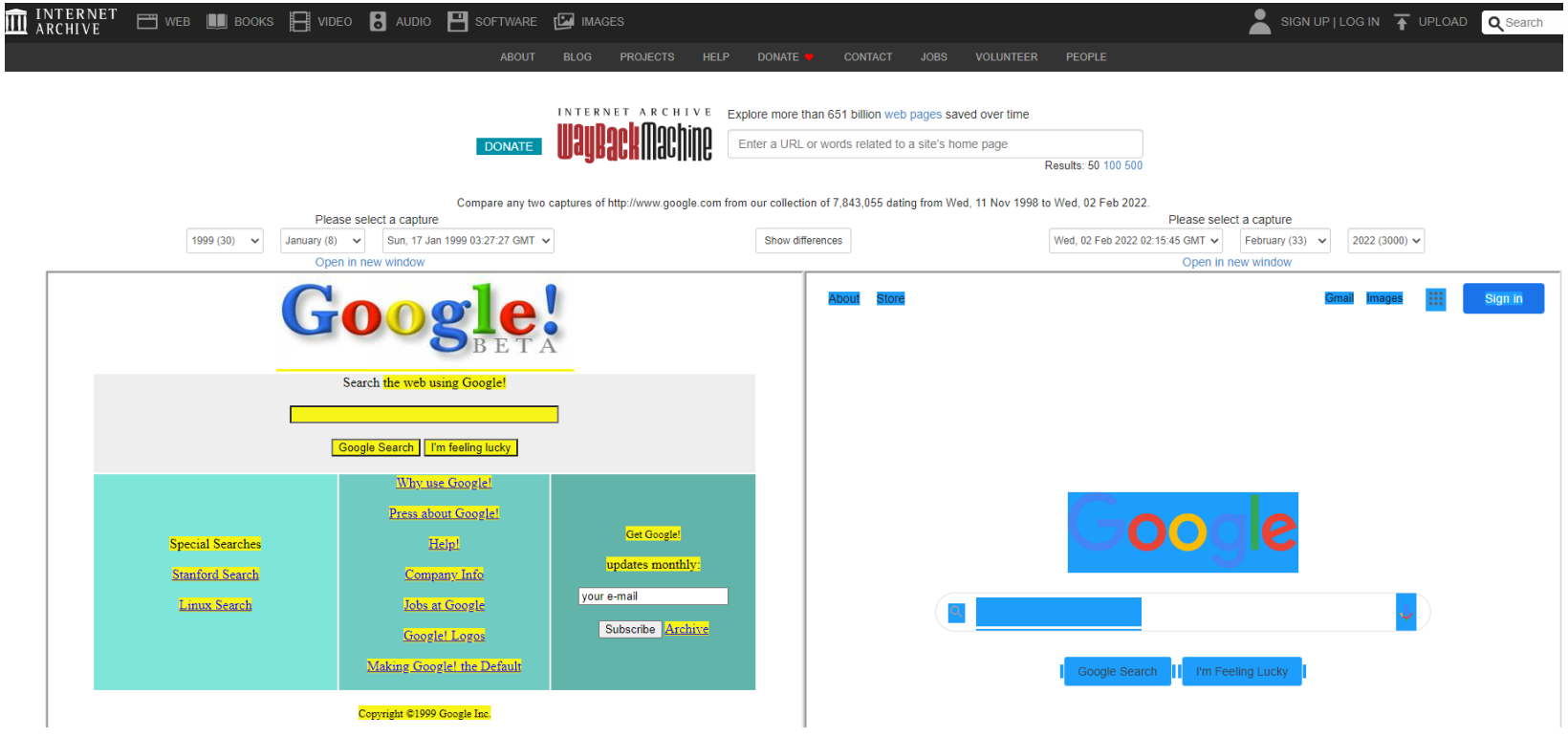
- **Use of publicly available reviews as a part of your service:** Would you classify the Yelp vs Google Feud as such an example?

# 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

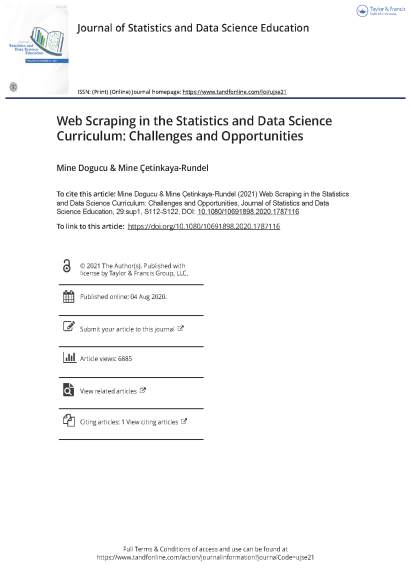
# Summary of Main Points

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

- Understand when can we scrape data (i.e., `robots.txt`)
- Scrape a webpage using R

# Things to Do to Prepare for Next Class

- Go over your notes, read through the supplementary material (below) and complete Assignment 04 on Canvas.



Journal of Statistics and Data Science Education

Web Scraping in the Statistics and Data Science Curriculum: Challenges and Opportunities

Mine Dogucu & Mine Çetinkaya-Rundel


- PDF of Published Paper
- ePub of Published Paper
- Selector Gadget
- Getting Started with rvest