Ethics and web scraping

Warmup

Work on the warmup activity (handout), then we will discuss as a group.

Ethical restrictions on web scraping

- As a general rule, respect websites' Terms and Conditions
- Terms prohibiting web scraping have become more prevalent with advent of LLMs
- Be *polite* when scraping: check that you have permission to scrape, and don't send too many requests in a short space of time
- Application Programming Interfaces (APIs) provide an alternative to web scraping for some websites

Other restrictions on web scraping

- While sites often prohibit scraping in their terms and conditions, they do “allow” some portion of their site to be accessed by web crawlers for things like search engine indexing etc.
- These limitations are expressed in a site’s `robots.txt` file

`robots.txt`

```
https://www.imdb.com/robots.txt
```

```
User-agent: *
Disallow: /OnThisDay
Disallow: /*/OnThisDay
Disallow: /ads/
Disallow: /*/ads/
Disallow: /find
Disallow: /*find
...
```

In R:

```r
library(robotstxt)
paths_allowed("https://www.imdb.com/find/")
```

```
[1] FALSE
```

```r
paths_allowed("https://www.imdb.com/list/ls055592025/")
```

```
[1] TRUE
```

`robots.txt`

This means that *everyone* is disallowed from *everything*:

```
User-agent: *
Diallow: /
```

robots.txt

https://www.imdb.com/robots.txt

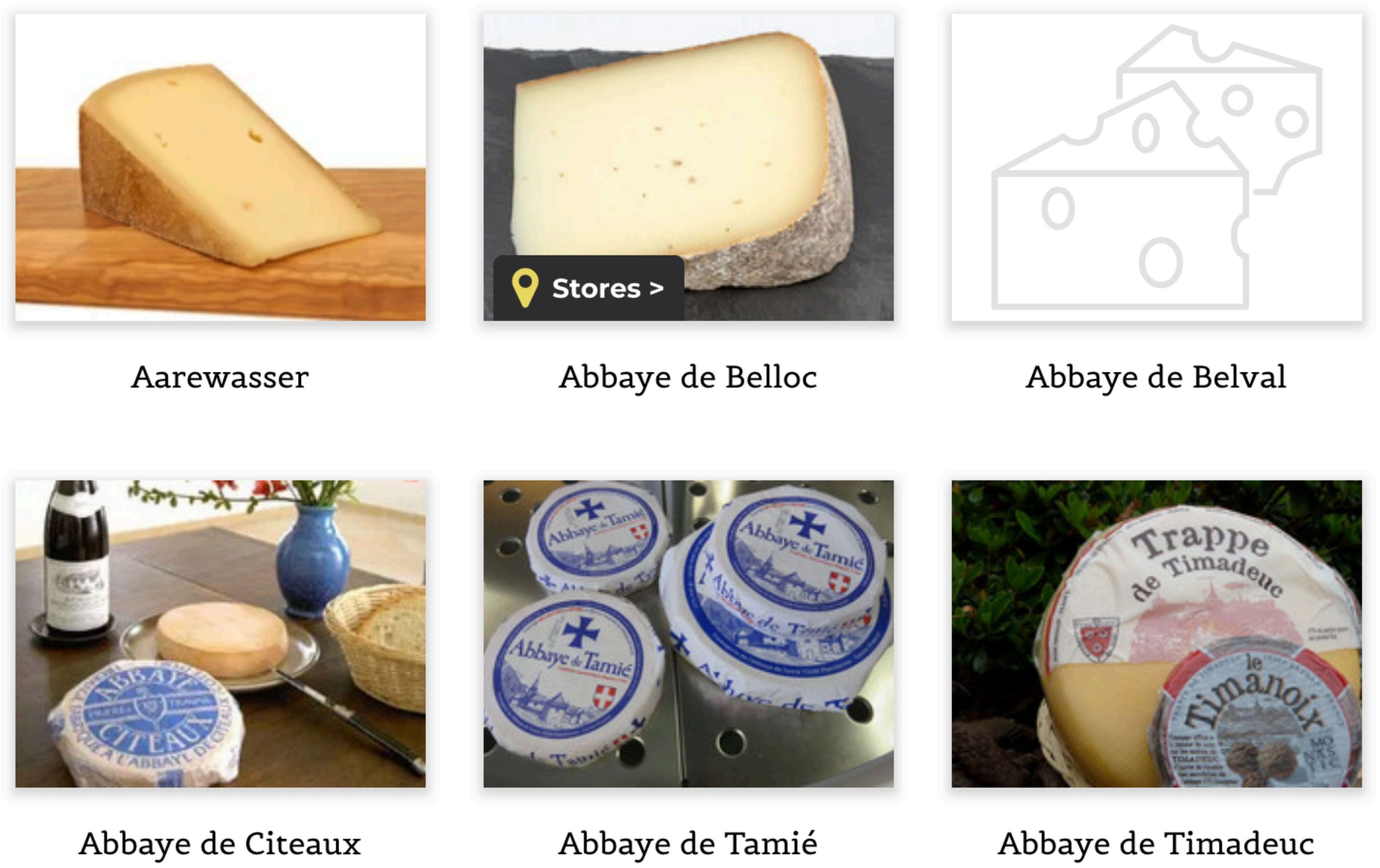
```
User-agent: anthropic-ai
Disallow: /
User-agent: Claude-Web
Disallow: /
User-agent: CCbot
Disallow: /
User-agent: FacebookBot
Disallow: /
User-agent: Google-Extended
Disallow: /
User-agent: GPTBot
Disallow: /
User-agent: PiplBot
```

Polite web scraping: the `polite` package

- Automatically ask permission to scrape (by checking `robots.txt` file)
- Don’t ask for the same information twice
- Limit rate of requests

Example: cheese!

```
https://www.cheese.com/alphabetical/a/
```



Aarewasser

Abbaye de Belloc

Abbaye de Belval

Abbaye de Citeaux

Abbaye de Tamié

Abbaye de Timadeuc

Example: cheese!

```r
library(polite)
session <- bow("https://www.cheese.com/alphabetical/a/")
session
```

```
<polite session> https://www.cheese.com/alphabetical/a/
    User-agent: polite R package
    robots.txt: 0 rules are defined for 1 bots
   Crawl delay: 5 sec
  The path is scrapable for this user-agent
```

`bow`: establish initial connection to the page

Example: cheese!

```r
library(polite)
session <- bow("https://www.cheese.com/alphabetical/a/")
session |>
  scrape()
```

```
{html_document}
<html lang="en">
[1] <head>\n<meta http-equiv="Content-Type" content="text/html; charset
...
[2] <body>\n    \n    \n\n    <!-- Header -->\n    <div id="header">\n
...
```

This is the polite version of:

```r
read_html("https://www.cheese.com/alphabetical/a/")
```

```
{html_document}
<html lang="en">
[1] <head>\n<meta http-equiv="Content-Type" content="text/html; charset
...
[2] <body>\n    \n    \n\n    <!-- Header -->\n    <div id="header">\n
...
```

Example: cheese!

What if we want to look at another page on the website?

```
https://www.cheese.com/aarewasser/
```

```r
current_page <- session |>
      nod("/aarewasser")
current_page
```

```
<polite session> https://www.cheese.com/aarewasser
    User-agent: polite R package
    robots.txt: 0 rules are defined for 1 bots
   Crawl delay: 5 sec
  The path is scrapable for this user-agent
```

```r
current_page |>
      scrape()
```

```
{html_document}
<html lang="en">
[1] <head>\n<meta http-equiv="Content-Type" content="text/html; charset
...
[2] <body>\n    \n    \n\n    <!-- Header -->\n    <div id="header">\n
...
```

Example: cheese!

```r
current_page <- session |>
    nod("/aarewasser")
current_page |>
    scrape()
```

`nod` allows us to modify the URL without having to establish a new connection (i.e. don't have to `bow` again)

This is the polite version of:

```r
read_html("https://www.cheese.com/aarewasser/")
```

Why be `polite`?

- Many requests in a short period of time uses the website's resources, and can look like a malicious attack
  - This could also result in the website blocking you
  - Be polite: ask slowly
- Requesting the same information repeatedly is a waste of resources
  - Be polite: establish the connection once (`bow`), and modify when you need a different page (`nod`)
- It is particularly important to be polite when you are scraping many different pages on a website

Class activity

Work on the class activity. Render your HTML and submit on Canvas at the end of class.