Lecture 9
XML and HTML

Dennis Sun
Stanford University
DATASCI / STATS 112

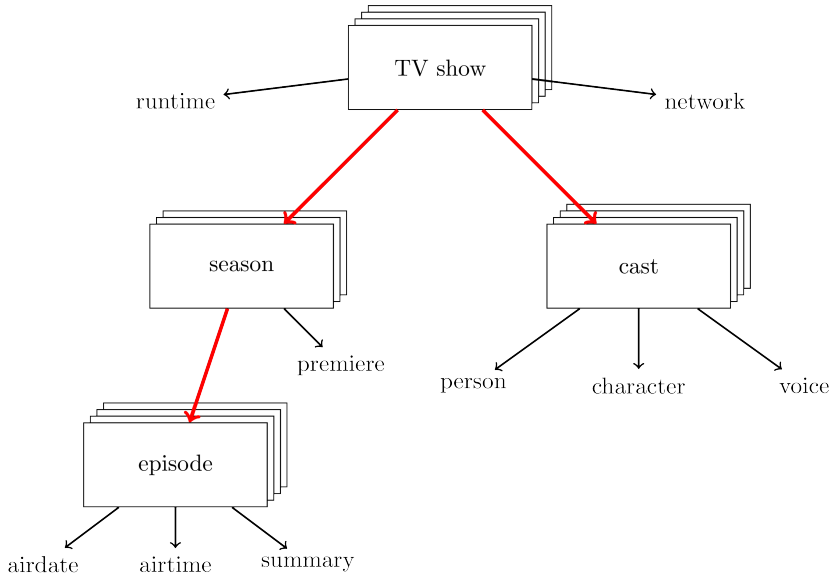
February 1, 2023

1 XML

2 HTML

3 Reminders

Hierarchical Data



We've seen that JSON is one way to represent data like this.

```
[{'name': 'Girls',
  'runtime': 30,
  'network': {'name': 'NBC', ...}
  'cast': [{'person': {'name': 'Lena Dunham', ...},
            'character': {'name': 'Hannah Horvath', ...},
            'voice': False},
            ...],
  'seasons': [{'premiereDate': '2012-04-15',
               'episodes': [...]},
               ...]},
  ...]
```

eXtensible Markup Language (XML)

XML is another way to represent data like this.

- Fields are represented by tags.
- Every tag has an open `<cast>` and a close `</cast>`.
- Children are represented by nested tags.
- Repeated fields are represented by repeated tags.

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <show>
    <name>Girls</name>
    <runtime>30</runtime>
    <cast>
      <person>...</person>
      <character>....</character>
    </cast>
    <cast>
      ...
    </cast>
    <season>
      <episode>...</episode>
      <episode>...</episode>
      ...
    </season>
    <season>
      ...
    </season>
  </show>
</root>
```

Processing XML Files

- We use a Python library called “Beautiful Soup 4”.
- We read the XML data into a **`BeautifulSoup`** object, which represents the data as a tree.
- You can navigate this tree using **`.parent`** (**`.parents`**) and **`.children`** (**`.descendants`**).
- You can search for a tag using **`.find_all()`** or **`.find()`** (returns first tag found).

Using Beautiful Soup

Read in the XML using Beautiful Soup.

```
from bs4 import BeautifulSoup
import requests
response = requests.get("https://dlsun.github.io/pods/data/tvshows.xml"
soup = BeautifulSoup(response.text, 'xml')
```

Which show had the most episodes?

```
show_names = []
show_episodes = []
for show in soup.find_all("show"):
  show_names.append(show.find("name").string)
  show_episodes.append(len(show.find_all("episode")))
```

Show the results in a `DataFrame`:

```
import pandas as pd
pd.DataFrame({
    "name": show_names,
    "episodes": show_episodes
}).sort_values("episodes")
```

	name	episodes
4	Florida Girls	10
6	Derry Girls	12
8	Bomb Girls	19
2	Good Girls	26
0	Girls	63
5	Chicken Girls	76
7	The Powerpuff Girls	82
3	The Powerpuff Girls	119
9	Gilmore Girls	153
1	The Golden Girls	181

In-Class Exercise

Let's do another example in Colab.

1 XML

2 HTML

3 Reminders

What is HTML?

HyperText Markup Language (HTML) is an XML-like language used to specify the appearance of webpages.

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>List of United States cities by population - Wikipedia</titl
    ...
  </head>
  <body>
    ...
    <h1 id="firstHeading" class="firstHeading mw-first-heading">
      <span class="mw-page-title-main">List of United States cities by
    </h1>
    ...
  </body>
</html>
```

Web Scraping

You can read in HTML using Beautiful Soup as well.

First, you get the webpage using the requests library:

```
import requests
response = requests.get(
    "https://en.wikipedia.org/wiki/List_of_United_States_cities_by_popu
```

Next, you use Beautiful Soup to parse the string into a tree.

```
from bs4 import BeautifulSoup
soup = BeautifulSoup(response.text, "html.parser")
```

Now you can navigate this tree using the same functions that we used for XML (e.g., `.find_all()`, `.parent`)

This is called **web scraping**, and you will practice it for section tomorrow!

1 XML

2 HTML

3 Reminders

Reminders

- Finish the Colab on web scraping.
- Graded Exam 1 will be returned in section.
- We will have a guest lecture from Alok Pattani (Google) on Friday. He will talk about data science in sports.
- Assignment 3 will be released on Friday and due next Friday.