

POIR 613: Measurement Models and Statistical Computing

Pablo Barberá

School of International Relations
University of Southern California
`pablobarbera.com`

Course website:

pablobarbera.com/POIR613/

Scraping the web

Scraping the web: what?

An increasing amount of data is available on the web:

- ▶ Speeches, sentences, biographical information...
- ▶ Social media data, newspaper articles, press releases...
- ▶ Geographic information, conflict data...

These datasets are often provided in an **unstructured format**.

Web scraping is the process of extracting this information automatically and transforming it into a **structured dataset**.

Scraping the web: why?

- ▶ Copy & pasting is time-consuming, boring, prone to errors, and impractical for large datasets
- ▶ In contrast, automated web scraping:
 1. Scales well for large datasets
 2. Is reproducible
 3. Involved adaptable techniques
 4. Facilitates detecting and fixing errors
- ▶ When to scrape?
 1. Trade-off between your time today and your time in the future. **Invest in your future self!**
 2. Computer time is cheap; human time is expensive

Scraping the web: two approaches

Two different approaches:

1. **Screen scraping**: extract data from source code of website, with html parser and/or regular expressions
 - ▶ `rvest` package in R
2. **Web APIs** (application programming interfaces): a set of structured http requests that return JSON or XML data
 - ▶ `httr` package to construct API requests
 - ▶ Packages specific to each API: `weatherData`, `WDI`, `Rfacebook`... Check CRAN Task View on [Web Technologies and Services](#) for examples
 - ▶ More on APIs later this semester

The rules of the game

1. Respect the hosting site's wishes:
 - ▶ Check if an API exists or if data are available for download
 - ▶ Keep in mind where data comes from and give credit (and respect copyright if you want to republish the data!)
 - ▶ Some websites *disallow* scrapers on `robots.txt` file
2. Limit your bandwidth use:
 - ▶ Wait one or two seconds after each hit
 - ▶ Scrape only what you need, and just once (e.g. store the html file in disk, and then parse it)
3. When using APIs, read documentation
 - ▶ Is there a batch download option?
 - ▶ Are there any rate limits?
 - ▶ Can you share the data?

The art of web scraping

Workflow:

1. Learn about structure of website
2. Choose your strategy
3. Build prototype code: extract, prepare, validate
4. Generalize: functions, loops, debugging
5. Data cleaning

Three main scenarios

1. Data in table format



WIKIPEDIA
The Free Encyclopedia

Article [Talk](#)

Read [Edit](#) [View history](#)

Not logged in [Talk](#) [Contributions](#) [Create account](#) [Log in](#)

International court

From Wikipedia, the free encyclopedia

[Main name](#)

List of international courts [\[edit \]](#)

Name	Scope	Years active	Subject matter
International Court of Justice	Global	1945–present	General disputes
International Criminal Court	Global	2002–present	Criminal prosecutions
Permanent Court of International Justice	Global	1922–1946	General disputes
Appellate Body	Global	1995–present	Trade disputes within the WTO
International Tribunal for the Law of the Sea	Global	1994–present	Maritime disputes
African Court of Justice	Africa	2009–present	Interpretation of AU treaties
African Court on Human and Peoples' Rights	Africa	2006–present	Human rights
COMESA Court of Justice	Africa	1998–present	Trade disputes within COMESA
ECOWAS Community Court of Justice	Africa	1996–present	Interpretation of ECOWAS treaties
East African Court of Justice	Africa	2001–present	Interpretation of EAC treaties
SADC Tribunal	Africa	2005–2012	Interpretation of SADC treaties

Three main scenarios

2. Data in unstructured format

The screenshot shows the IPaidABribe website interface. At the top, there is a navigation bar with the IPaidABribe logo, a search bar, and a 'Register for updates' button. Below the navigation bar, there are several tabs: 'I PAID A BRIBE', 'I DID NOT PAY A BRIBE', 'I MET AN HONEST OFFICER', 'BRIBE HOTLINE', 'ALL REPORTS', 'NEWS', and a red 'REPORT A BRIBE' button. The main content area displays a list of reports under the 'I PAID A BRIBE' filter. Each report entry includes a title, a brief description, and a 'Read more' link. The filter sidebar on the right allows users to filter reports by city, department, and bribe amount, with a 'SUBMIT' button at the bottom. A promotional banner for 'INSPIRE OTHERS WITH YOUR STORY' is also visible, featuring a photo of a person in a blue helmet and a 'SEE HIS STORY' button. At the bottom of the page, there are three icons with text: 'Ever Paid A Bribe?', 'Report your Bribe Story!', and 'See action taken.'.

India English 11,072,800 Visitors

SEARCH Register for updates

I PAID A BRIBE I DID NOT PAY A BRIBE I MET AN HONEST OFFICER BRIBE HOTLINE ALL REPORTS NEWS REPORT A BRIBE

All Reports > I Paid A Bribe

ALL / I PAID A BRIBE / BRIBE FIGHTER / HONEST OFFICER / BRIBE HOTLINE

I PAID A BRIBE 1 day ago 76 views

POLICE NILO GHUSS (bribe)

Passport | Police Verification for Passport | Paid INR 5,000

Reported on January 17, 2016 from Bankura, West Bengal | Report #89544

What will happen to this country..police mamu's govt income: 30,000 per month. Per day GHUSS income 5000 (per passport verification). Imagine they t...[Read more](#)

How to Get a Passport Verified in Ghaziabad

I PAID A BRIBE 1 day ago 104 views

Corruption due to vague rules

Police | Traffic Violations | Paid INR 500

Reported on January 16, 2016 from Mumbai, Maharashtra | Report #89509

At Chembur near Eastern Expressway traffic cop stopped me and started checking docs..all was fine buy puc expired..then he pointed out film.. He took...[Read more](#)

Things to Know on Traffic Offences and Respective Penalties

I PAID A BRIBE 2 days ago 105 views

Bribe collected by Staff of Enrollment agency

Municipal Services | Aadhaar or UID Related | Paid INR 120

Reported on January 16, 2016 from Mysore, Karnataka | Report #89467

UIDAI has to take a stand on fees to be paid to enrolment agencies for processing Aadhaar

INSPIRE OTHERS WITH YOUR STORY

Manik Tanega, a sports enthusiast, wrote against a custom officer on ipaidabribe.com, for cough up a hefty bribe by a Customs official at Bengaluru airport.

SEE HIS STORY

Ever Paid A Bribe?

Report your Bribe Story!

See action taken.

www.ipaidabribe.com/reports/paid

Three main scenarios

3. Data hidden behind web forms



The screenshot shows the 'MONITOR LEGISLATIVO' website interface. At the top, there is a navigation bar with icons for home, user profile, news, candidates, national assembly, reports, and contact. Below this is a search bar with the text 'RESULTADOS DE LA CONSULTA'. The search criteria are set to 'Seleccione' and 'Partido', with a 'BUSCAR' button. A purple banner below the search bar reads 'DIPUTADOS ENCONTRADOS'. Below the banner, there are six candidate profiles, each with a photo, name, and state. The candidates are: Julio Ygarza (Estado: Amazonas), Mauligmer Baloa (Estado: Amazonas), Nirma Guarulla (Estado: Amazonas), José Brito (Estado: Anzoátegui), Chaim Bucarán (Estado: Anzoátegui), and Richard Arteaga (Estado: Anzoátegui). Each profile includes the 'Unidad' logo.

MONITOR LEGISLATIVO

INICIO | PERFIL IDEAL | NOTICIAS | CANDIDATOS | ASAMBLEA NACIONAL | ABUSOS | CONTACTENOS

RESULTADOS DE LA CONSULTA

Seleccione | Partido | BUSCAR

DIPUTADOS ENCONTRADOS

Julio Ygarza
Estado: Amazonas

Mauligmer Baloa
Estado: Amazonas

Nirma Guarulla
Estado: Amazonas

José Brito
Estado: Anzoátegui

Chaim Bucarán
Estado: Anzoátegui

Richard Arteaga
Estado: Anzoátegui

Candidates on 2015 Venezuelan parliamentary election

Three main scenarios

1. Data in **table** format
 - ▶ Automatic extraction with `rvest`
2. Data in **unstructured** format
 - ▶ Element identification with `selectorGadget`
 - ▶ Automatic extraction with `rvest`
3. Data hidden **behind web forms**
 - ▶ Automation of web browser behavior with `selenium`

HTML: a primer

Hypertext Markup Language (HTML): hidden standard behind every website.

- ▶ HTML is text with marked-up structure, defined by **tags**:

- ▶ `<!DOCTYPE html>`

```
<html>
```

```
<body>
```

```
<h1>My First Heading</h1>
```

```
<p>My first paragraph.</p>
```

```
</body>
```

```
</html>
```

- ▶ What you see in your browser is an interpretation of the HTML document

HTML: a primer

- ▶ Some common tags:
 - ▶ Document elements: `<head>`, `<body>`, `<footer>`...
 - ▶ Document components: `<title>`, `<h1>`, `<div>`...
 - ▶ Text style: ``, `<i>`, ``...
 - ▶ Hyperlinks: `<a>`
- ▶ An example: www.pablobarbera.com

Beyond HTML

- ▶ **Cascading Style Sheets (CSS)**: describes formatting of HTML components (e.g. `<h1>`, `<div>`...), useful for us!



- ▶ **Javascript**: adds functionalities to the website (e.g. change content/structure after website has been loaded)

Parsing HTML code

First step in webscraping: read HTML code in R and **parse it**

- ▶ Parsing = understanding structure
- ▶ How? `rvest` package in R:
 - ▶ `read_html`: parse HTML code into R
 - ▶ `html_text`: extract text from HTML code
 - ▶ `html_table`: extract tables in HTML code
 - ▶ `html_nodes`: extract components with CSS selector
 - ▶ `html_attrs`: extract attributes of nodes
- ▶ How to identify relevant CSS selectors?
`selectorGadget` extension for Chrome and Firefox.