Ida's Python Group

Sam, Braden, Rakshan, Selina

Winter 2025

Purpose: Making a Dataset

This project aims to **collect and analyze news articles from O Globo**, focusing on gang violence and police violence (raids) in Rio de Janeiro.

The goal is to study the impact of violence exposure on educational outcomes, using web scraping and classification to create a dataset with historical violence data in Rio.

Where we started

We inherited a scraper and classification script from the previous Research Assistants.

- It scraped only 29 articles.
- It wasn’t super efficient in going back further.

For these reasons, we decided to start from scratch.

Step 1: Web Scraper

- Crawls articles on web page from O Globo's Rio-specific sections.
    a. Separated into two links to scrape, one ranging from 2015 - 2020 and one from 2020 - 2025
- Extracts links to each article and saves in a text (.txt) file
- **Library**: Beautiful Soup 4 and Selenium

Step 2: Data Processing

- Parses through given text file, and scrapes each article
- Identifies and categorizes violence-related articles.
- Extracts metadata such as location, violence type, and number of victims.
- Queries articles for the specific Rio neighborhood of the incident.

Step 3: Data Storage

- Saves data in structured formats (CSV/Database) for analysis.
- Has columns for article URL,
- Outputs violence likelihood metrics and summaries for visualization.

CSV

This initial scraper is now able to collect around 500 + articles from the O Globo website.

	Newspaper	Crime Date	Coordinates	Location	Description	Number of Victims	Gender of Victims	Level of Violence
1	O Globo	2025-01-22	22.9110137, -43.2093727	Rio de Janeiro	8 A Polícia investiga ho...	0	8M, 0F	Low
2	O Globo	2025-01-22	22.9247351, -43.2327165	Tijuca	Empresário é morto em...	0	0M, 2F	High
3	O Globo	2025-01-22	19.7999954, -41.7138807	Ipanema	M:22/01/2025 - 12:21 C...	0	0M, 0F	Medium
4	O Globo	2025-01-21	22.9247351, -43.2327165	Tijuca	/:16/01/2025 - 22:31 As...	0	0M, 0F	High
5	O Globo	2025-01-22	22.9110137, -43.2093727	Rio de Janeiro	ma empresa de Três Ri...	0	0M, 2F	Medium
6	O Globo	2024-10-22	22.9110137, -43.2093727	Rio de Janeiro	ro dos presídios do Rio ...	0	0M, 0F	High
7	O Globo	2025-01-22	22.9247351, -43.2327165	Tijuca	nília de Piruinha planeja...	0	0M, 1F	High
8	O Globo	2022-09-27	22.9110137, -43.2093727	Rio de Janeiro	a mulher pode ser subm...	0	0M, 24F	High

Challenge #1 - Getting Blocked on Sites

- A common scraping challenge is with getting blocked from the sites
- We solved this by using timeout out methods that mimic human behavior
- OGlobo has a strict paywall that prevents access to articles.

Challenge #2 - Reached a threshold

- Our first scraping method reaches a threshold after 2 years
- We solved this by using the search feature on **oglobo.globo.com** and filtered it to a customer time frame
- We later realized that the time frame also didn’t have the complete frame we needed

Challenge #3 - OGlobo

oglobo.globo.com - digitized version of print news in Brazil

Upon querying the URLs to bypass the threshold, we realized that O'Globo actually does not have articles visible before 2023.

Solution:

We decided to instead use **G1**, Grupo Globo's digital news portal of Brazil. This website actually had data for decades past. As a start, we queried the website for news between 2015 and 2025. Then we separated into the respective date ranges mentioned earlier.

Challenge #4 - Scraping Library

We initially used **BeautifulSoup4**, a popular web-scraping library that parses the HTML of a web page.

However, after changing the source website to **G1**, our existing scraping script ran into errors and found no pages. This was due to the backend of G1 using Javascript and buttons for page iteration.

Solution:

We edited our script to use Selenium, a Python library that automates web browser interaction, such as dynamic scrolling and button-clicking, solving this JavaScript problem by

Let’s revisit the updated code!

Summary Statistics

How far do we go back? **Until 2015**

How many entries in our CSV so far? **1200 → cleaned to ~950**

How many entries on average per month? **ranges from 9-40 articles**

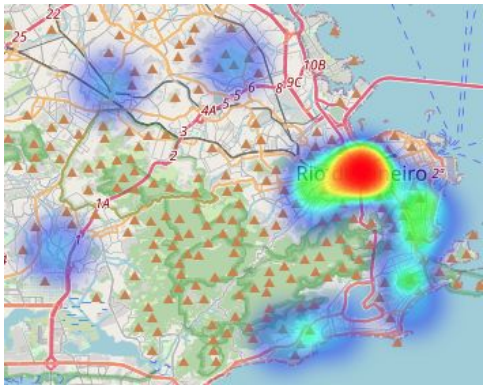
How many entries per year? **ranges from**

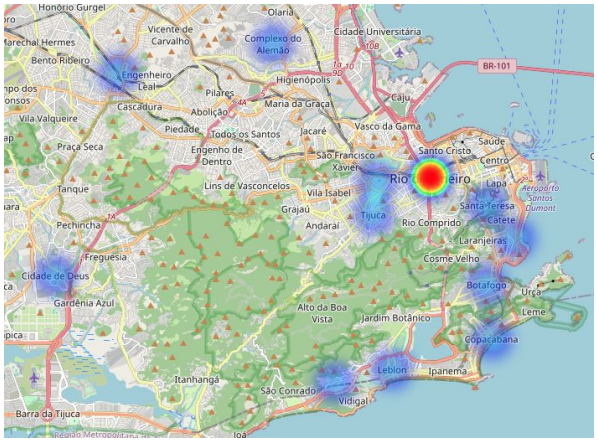
```
#per year
df.groupby("Year")["Newspaper"].count()
```

```
Year
2015     75
2016     83
2017    168
2018    152
2019    114
2020     12
2021     73
2022     57
2023    108
2024     85
2025    260
Name: Newspaper, dtype: int64
```

Future: Improve Location Precision

Right now, the locations we are getting are not specific enough. We would like to map out these articles onto specific neighborhoods. Right now, we are still largely getting Rio de Janeiro.





Future Plans - Fetching Content/Data Cleaning

As of now, we have been able to create code that allows for the code to click the “Veja Mais” button through Selenium Chrome Webdriver. This is successful pulling out around 1000 links.

However, we are having trouble fetching the content and verifying content data from these links.

In the future we will work on:

1.) Potentially adding more data points if feasible

2.) Cleaning the dataset we have using pandas, and verifying the accuracy of the column data

Conclusion

As you can see scraping is a very challenging task, given the complexity of websites, especially when they are in another language.

Luckily, it seems like a majority of us will come back next quarter so we can continue the progress we have made.

Although we have run into challenges, these were inevitable, and the future weeks should bring exciting results.