Module 1 - Session 1 - Data Collection

Working effectively with data

CivicDataLab

2021/07/15 (updated: 2021-07-16)

Onboarding - Things to do

- Check the onboarding document
- Create an account on Rocket Chat
- Take the survey

Session Structure

- Data Collection - How to
- Collecting data from secondary sources
    - The data biography
    - Structured vs Unstructured data sources
    - Collecting (structured) data from secondary sources
        - Working with CSV files
        - Extracting data from PDF files
        - Collecting data from the web
- Dealing with (messy) public datasets
- Session 2 - Introduction
- Reading List

Data Collection - How to ?

Collecting it yourself

> The requirements for equitable data collection are complex. It’s not as simple as trying to ask everyone and not leave people out. Sample selection is important of course, but so is survey design, collector behaviour, scope and scale, cultural translation, collection mediums, data corruption, compatibility and fidelity and much more. It’s super worth doing, if for no other reason than your data will be more useful.[1]

Sourcing it from secondary data sources

[1] We All Count - Data Equity Framework

Secondary data sources

A few challenges we might face while using secondary data sources for research and analysis:

1. Data is not accessible.
2. Information and Knowledge gaps between data creators and data users.
3. Maintaining periodic datasets is hard and time consuming.
4. Cannot be used out of the box for research and analysis.

> Now, if you’re sourcing data, rather than collecting it first hand, instead of a jewel, you’re probably better off considering the data a steaming pile of garbage. At least until you know it’s not. [1]

[1] We All Count - Data Equity Framework

The data biography

Maintaining a *data biography* at an org level really helps in making secondary datasets more accessible.

> A comprehensive data biography – the where, why and how of any dataset – is absolutely crucial to equitable analysis. Get to know your data on the nitty-gritty, how-did-they-get-this, look-at-the-original-survey-wording, who-did-they-miss, level. When you really know your data and run it through the filter for potential bias and equity issues, you can begin to use facts and figures with confidence. You can maintain a buck-stops-here attitude towards ensuring inclusive, non-garbage, truthful data science. 1

A Data biography Template

Structured vs Unstructured

Difference Use-cases Types

Structured data is a lot easier to work with but often times we're limited with the scope of datasets. On the other hand, the process of converting **unstructured data** to a format that can be analysed is hard but it creates a lot of opportunities for research.

> All data has some structure, but ‘structured data’ refers to data where the structural relation between elements is explicit in the way the data is stored on a computer disk[1]

[1] Open Data Handbook - Glossary

Structured vs Unstructured

Difference Use-cases Types

1. Criminal 'Injustice': How Courts Use 'Remand' to Penalise the Poor - *Analysis of 153 remand orders*
2. Sharp Fall in Citation of Supreme Court Judgments by Foreign Courts After 2014, Study Finds - *Analysis of Judgements from 43 countries*

[1] Open Data Handbook - Glossary

Structured vs Unstructured

Difference   Use-cases   Types

Structured sources

- Census of India
- NCRB Database
- National Judicial Data Grid (NJDG)

Unstructured sources

- Laws and Judgements
- Newspaper Reports
- Ethnographic summaries

[1] Open Data Handbook - Glossary

Collecting (structured) data

Working with CSV files

Why CSV

- One of the most widely used data formats

Why CSV

- One of the most widely used data formats
- Synonymous with open datasets

Why CSV

- One of the most widely used data formats
- Synonymous with open datasets
- Offers a lot of flexibility for building tools that aid in research and analysis

Challenges with CSV's

- Not a standard way to create a CSV file.
    - CSVs generated from standard spreadsheets and databases as a matter of course use variable encodings, variable quoting of special characters, and variable line endings.

Challenges with CSV's

- Not a standard way to create a CSV file.
    - CSVs generated from standard spreadsheets and databases as a matter of course use variable encodings, variable quoting of special characters, and variable line endings.

- CSV files are desperately poor at providing contextual information that can aid in automated processing of the data that they hold or even in informing developers about how they should be interpreted.

A CSV file lacks context

> CSV is insufficient for tabular data on the web, with independent generic viewers, because it lacks this context. Not knowing which columns contain numeric data makes it impossible to sort correctly. Not knowing which values are links makes it impossible to connect the data into the web. Crucially, from an open data perspective, this lack of expressivity means there is no obvious way to express the licence under which CSV data can be reused. [1]

Focusing on the data entry and storage aspects, we can follow a set of best practices for organizing spreadsheet data to reduce errors and ease later analyses [2]

[1] 2014: The year of CSV

[2] Karl W. Broman & Kara H. Woo (2018) Data Organization in Spreadsheets

Core principles

- Be consistent

Core principles

- Be consistent
- Build a tidy dataset

Core principles

- Be consistent
- Build a tidy dataset
- Create a data dictionary

Core principles

- Be consistent
- Build a tidy dataset
- Create a data dictionary
- No calculations in the raw data files

Core principles

- Be consistent
- Build a tidy dataset
- Create a data dictionary
- No calculations in the raw data files
- Don't use font color or highlighting as data

Core principles

- Be consistent
- Build a tidy dataset
- Create a data dictionary
- No calculations in the raw data files
- Don't use font color or highlighting as data
- Use data validation to avoid errors

Be consistent

- Use consistent codes for categorical variables
- Use a single fixed code for any missing values
- Use consistent variable names[1]
- Use a single common format for all dates

[1] Choose good names for things

A tidy dataset

A **tidy dataset** is a standard way of mapping the meaning of a dataset to its structure. A dataset is messy or tidy depending on how rows, columns and tables are matched up with observations,variables and types.

In tidy data:

- Each variable forms a column.
- Each observation forms a row.
- Each type of observational unit forms a table.

Building a tidy dataset

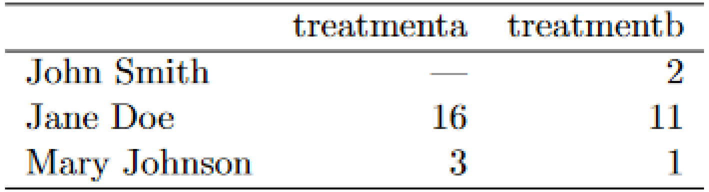
Tidy Dataset | Table 1 | Table 2 | Tidy Dataset | Variables vs Observations

A dataset is a collection of values, usually either numbers (if quantitative) or strings (if qualitative). Values are organized in two ways. Every value belongs to a variable and an observation. A **variable** contains all values that measure the same underlying attribute (like height, temperature, duration) across units. An **observation** contains all values measured on the same unit (like a person, or a day, or a race) across attributes. [1]

[1] Tidy data, Hadley Wickham

Building a tidy dataset

Tidy Dataset | Table 1 | Table 2 | Tidy Dataset | Variables vs Observations

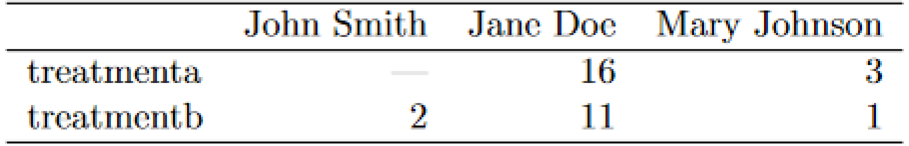


	treatmenta	treatmentb
John Smith	—	2
Jane Doe	16	11
Mary Johnson	3	1

[1] Tidy data, Hadley Wickham

Building a tidy dataset

Tidy Dataset Table 1 **Table 2** Tidy Dataset Variables vs Observations

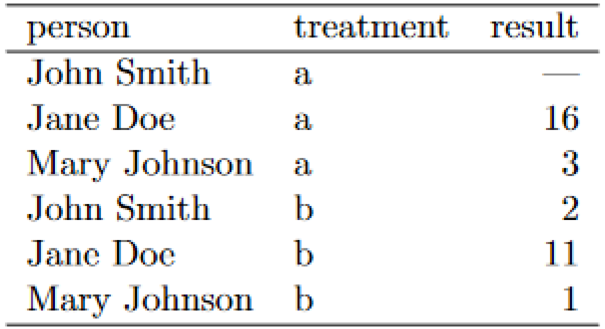


	John Smith	Jane Doe	Mary Johnson
treatmenta	—	16	3
treatmentb	2	11	1

[1] Tidy data, Hadley Wickham

Building a tidy dataset

Tidy Dataset | Table 1 | Table 2 | Tidy Dataset | Variables vs Observations



person	treatment	result
John Smith	a	—
Jane Doe	a	16
Mary Johnson	a	3
John Smith	b	2
Jane Doe	b	11
Mary Johnson	b	1

[1] Tidy data, Hadley Wickham

Building a tidy dataset

Tidy Dataset | Table 1 | Table 2 | Tidy Dataset | Variables vs Observations

A general rule of thumb is that it is easier to describe functional relationships between variables (e.g.,z is a linear combination of x and y,density is the ratio of weight to volume) than between rows, and it is easier to make comparisons between groups of observations (e.g., average of group a vs. average of group b) than between groups of columns.

[1] Tidy data, Hadley Wickham

Create a data dictionary

It’s helpful to have a separate file that explains what all of the variables are

Such a “data dictionary”[1] might contain:

- The exact variable name as in the data file.
- A version of the variable name that might be used in data visualizations.
- A longer explanation of what the variable means.
- The measurement units.
- Expected minimum and maximum values, perhaps.

[1] Data Dictionary - Justice Hub Docs

No calculations in the raw data files

Often, the Excel files include all kinds of calculations and graphs. The primary data file should contain just the data and nothing else: **no calculations, no graphs**.

Your primary data file should be a pristine store of data. Write-protect it, back it up, and don’t touch it.

If you want to do some analyses in Excel, make a copy of the file and do your calculations and graphs in the copy.

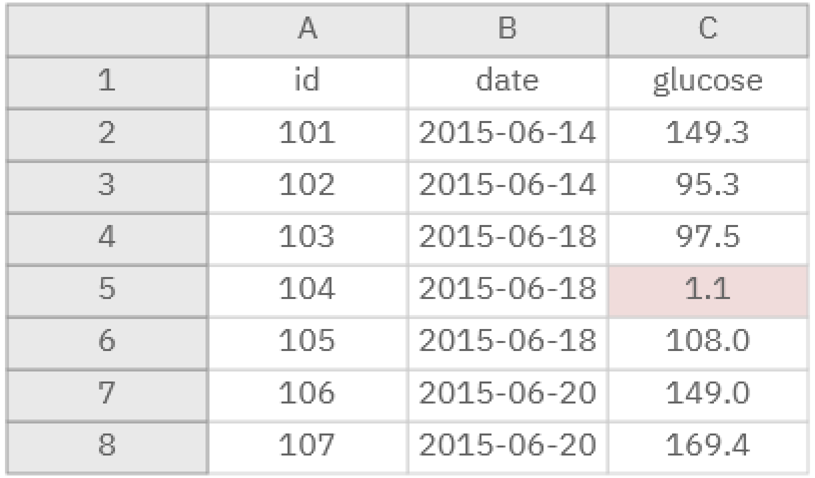
Don't use font color or highlighting as data

You might be tempted to highlight particular cells with suspicious data, or rows that should be ignored. Instead, add another column with an indicator variable (for example, "trusted", with values TRUE or FALSE)

Dealing with data formatting issues

Formatted Table

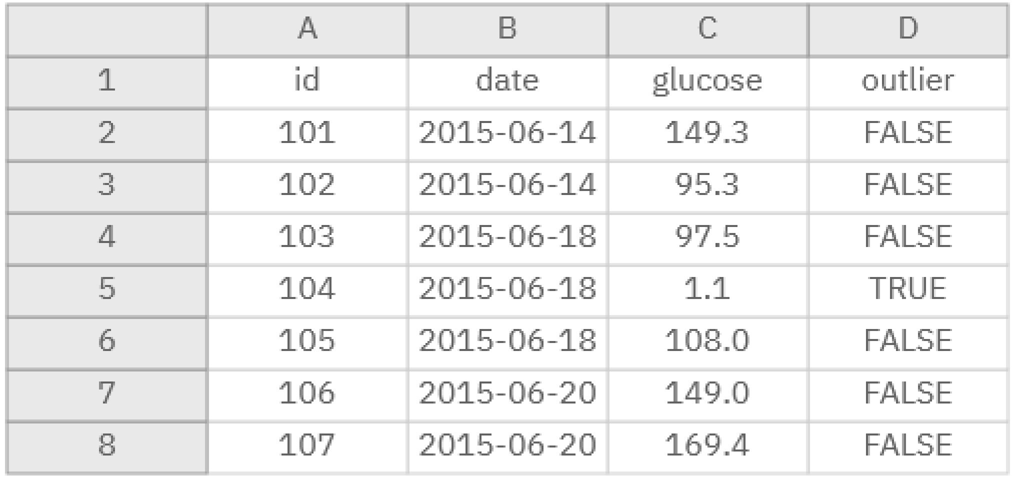
Better alternative



	A	B	C
1	id	date	glucose
2	101	2015-06-14	149.3
3	102	2015-06-14	95.3
4	103	2015-06-18	97.5
5	104	2015-06-18	1.1
6	105	2015-06-18	108.0
7	106	2015-06-20	149.0
8	107	2015-06-20	169.4

Dealing with data formatting issues

Formatted Table | Better alternative



	A	B	C	D
1	id	date	glucose	outlier
2	101	2015-06-14	149.3	FALSE
3	102	2015-06-14	95.3	FALSE
4	103	2015-06-18	97.5	FALSE
5	104	2015-06-18	1.1	TRUE
6	105	2015-06-18	108.0	FALSE
7	106	2015-06-20	149.0	FALSE
8	107	2015-06-20	169.4	FALSE

Use data validation to avoid errors

CSV looks easy, but it can be hard to make a CSV file that other people can work with. CSVLint,[1] helps you to check that your CSV file is readable. And you can use it to check whether it contains the columns and types of values that it should.

This service is maintained by an open source community and hosted by the Open Data Institute (ODI).

[1] CSVLint

Common warnings and errors

Errors Warnings

- Invalid encoding: if there are any odd characters in a file which could cause encoding errors
- Line breaks: if line breaks are not the same throughout the file
- Undeclared header: if you do not specify in a machine readable way whether or not your CSV has a header row
- Ragged rows: if every row in the file doesn't have the same number of columns
- Blank rows: if there are any blank rows
- Stray/Unclosed quote: if there are any unclosed quotes in the file
- Whitespace: if there is any whitespace between commas and double quotes around fields

Common warnings and errors

Errors Warnings

- Encoding: if you don't use UTF-8 as the encoding for the file
- Check options: if the CSV file only contains a single comma-separated column; this usually means you're using a separator other than a comma
- Inconsistent values: if any column contains inconsistent values, for example if most values in a column are numeric but there's a significant proportion that aren't
- Empty column name: if all the columns don't have a name
- Duplicate column name: if all the column names aren't unique
- Title row: if there appears to be a title field in the first row of the CSV.

Resources on dealing with CSV files

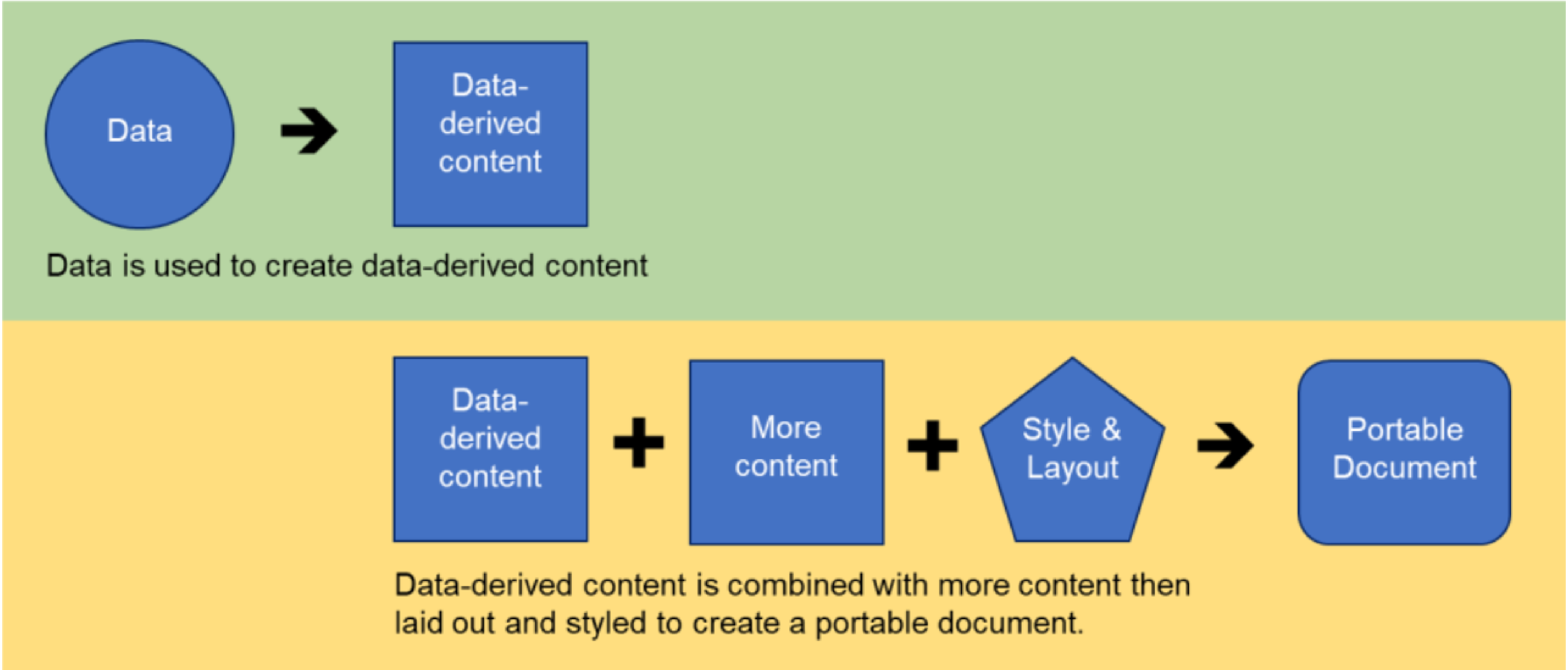
- Leek Group guide to data sharing](https://github.com/jtleek/datasharing)
- Data Carpentry lesson on using spreadsheets
- Releasing statistics in spreadsheets (pdf), by UK Government Statistical Service
- Video of Hadley Wickham talking about tidy data
- EP White et al. (2013) Nine simple ways to make it easier to (re)use your data.
- 3 common bad practices in sharing tables and spreadsheets and how to avoid them

Extracting data from PDF files

The essential feature of a portable document is that it looks the same (with exceptions for accessible views or views on mobile)wherever and whenever it is viewed or printed. This is a basis for reliable communication, when a document is sent, published, archived, or distributed, as the basis for further discussion[1]

[1] Best Practices for PDF and Data:Use Cases, Methods, Next Steps

PDF for data and content



Data is used to create data-derived content

Data-derived content is combined with more content then laid out and styled to create a portable document.

Source

Types of PDF

1. Machine Readable
2. Non Machine Readable (Scanned)

Data extraction from content

Open source tools:

1. Tabula
2. Camelot
    - Excalibur - A web interface for Camelot library

Working with Tabula

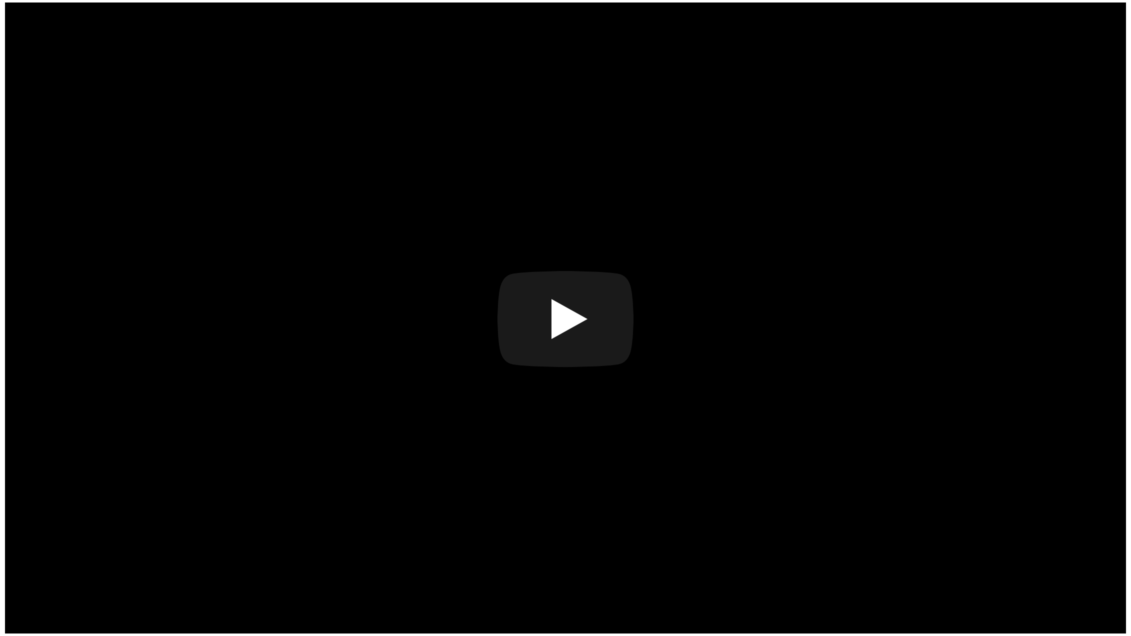
Objective - *Collect data to analyse state level trends in victim compensation*

Source - NALSA

Input Dataset - Link

Output Dataset - Link

Learning to use Tabula



Excalibur Demo

Collecting data from the web

Often times data is not made accessible using common file formats, but is present on the web in the form of HTML pages. To collect such datasets, a few approaches are widely adopted:

1 - Using API (Application Programming Interfaces) - These are URL's through which the data can be downloaded as per the user requirements. Not all data present on the web can be accessed via an API. The responsibility of creating an API is on the publisher/owner of the platform where the data resides.

2 - Human manual copy-and-paste

3 - **HTML Parsing**

Web Scraping 101

What Why Use-cases

Web scraping is the process of collecting the data from the World Wide Web and transforming it into a structured format. Typically web scraping is referred to an automated procedure, even though formally it includes a manual human scraping. [1]

[1] An introduction to Statistical Programming Methods in R

Web Scraping 101

What Why Use-cases

- A vast majority of data are user-generated content, presented in unstructured HTML format.
- The data are typically dynamic and vary over the time.
- All information is located on the Internet in human-readable format.
- The problem is not in accessing the data, but how to convert this information into the structured (think of tabular or spreadsheet-like) format.

[1] An introduction to Statistical Programming Methods in R

Web Scraping 101

What Why Use-cases

- Collecting data from websites like Wikipedia (There are ways to get access to structured information from Wiki Pages. Check DBPedia).
- Collecting data from court websites which don't often provide access to case laws, orders and judgements. Projects such as Court Listener and IndianKanoon work in the direction of making such datasets more accessible.
- Building time-series datasets from dashboards which are updated on a periodic basis.
- Collecting data from government dashboards and websites.

and a lot more ....

[1] An introduction to Statistical Programming Methods in R

Scraping data from the web - Ethically

- Read the robots.txt file.
- Analyzing Sitemap files.
- Analyse crawl delay.
- Check site ToS/T&C before scraping.
- Contact the site owner if you plan on doing a large amount of scraping.
- Introduce some delay between page scrapes, even if the site does not have a specific crawl-delay entry.

Also check:

- Ethics in Web Scraping - More general rules for both the scraper and the site owner

Web Scraping using Google Sheets

IMPORTDATA IMPORTXML IMPORTHTML IMPORTFEED

For retrieving structured datasets like CSV/TSV

Eg: Exploring a dataset from the JusticeHub

Syntax - `IMPORTDATA(url, delimiter, locale)`

Query - `=IMPORTDATA("https://justicehub.in/dataset/e48994b7-e77a-4fa1-869f-923dec3e4636/resource/02005017-cd9d-4fec-94d4-243b5e4f9fcb/download/top_sections.csv",",")`

Web Scraping using Google Sheets

IMPORTDATA IMPORTXML IMPORTHTML IMPORTFEED

For extracting specific content from HTML pages

Eg: Getting a list of all present judges from the Madras High Court

Syntax - `IMPORTXML(url, xpath_query, locale)`

Query -

```
=IMPORTXML("http://www.hcmadras.tn.nic.in/prejudge.html","/html/body/div[1]/div[3]/div/ul/li")
```

Web Scraping using Google Sheets

IMPORTDATA IMPORTXML **IMPORTHTML** IMPORTFEED

For extracting tables/lists from HTML

Eg: Extracting the list of states and union territories of India by crime rate, from Wikipedia

Syntax - `IMPORTHTML(url, query, index, locale)` Query - `=IMPORTHTML("https://en.wikipedia.org/wiki/List_of_states_and_union_territories_of_India_by_cri`

Web Scraping using Google Sheets

IMPORTDATA IMPORTXML IMPORTHTML **IMPORTFEED**

For retrieving RSS feeds

Eg: Getting the list of the latest 50 blogs published on the Vidhi website

Syntax - `IMPORTFEED(url, [query], [headers], [num_items])` Query - `=IMPORTFEED("https://vidhilegalpolicy.in/feed/","items","TRUE",50)`

Public data is messy

Here is a guide that might help you in dealing with a few common problems that we associate with datasets.

> Most of these problems can be solved. Some of them can’t be solved and that means you should not use the data. Others can’t be solved, but with precautions you can continue using the data. In order to allow for these ambiguities, this guide is organized by who is best equipped to solve the problem: you, your source, an expert, etc. In the description of each problem you may also find suggestions for what to do if that person can’t help you. [1]

[1] The Quartz guide to bad data

Session 2 - Introduction

Discussion Topics

- Data standards

Session 2 - Introduction

Discussion Topics

- Data standards
- When to scrape data vs when to build data standards

Session 2 - Introduction

Discussion Topics

- Data standards
- When to scrape data vs when to build data standards
- Case Studies
    - Akoma Ntoso

Session 2 - Introduction

Discussion Topics

- Data standards
- When to scrape data vs when to build data standards
- Case Studies
  - Akoma Ntoso
  - Data standard for publishing crime statistics

Reading List

1. The Daily Shaping of State Transparency: Standards, Machine-Readability and the Configuration of Open Government Data Policies
2. Responsible Data Handbook
3. A brief history of open data
4. Science friction: Data, metadata, and collaboration
5. From open data to information justice
6. THE DATA EQUITY FRAMEWORK