Text Analysis and the Humanities

Lucia Michielin, Digital Skills Training Manager

Centre for Data, Culture, & Society

Our Mission

To support, facilitate, promote and inspire data-led and applied digital research across the arts, humanities and social sciences

Key Activities

- Research support
- Community Building
- Training

Training Programme

Various levels of **applied digital skills** training for CAHSS Researchers that cover **different stages** of apply digital method to research

Pathways of learning to guide researchers in the in the world of digital skills

Support researchers in their self-learn with informative **material and asynchronous training/support**

Reusable and Accessible Material

Training Programme

- Good Practices of Digital Research
- Data Wrangling and Data Visualisation
- Intro to Programming
- Structured Data Analysis
- AI
- Geographical Data
- Digitised Documents and Text Analysis

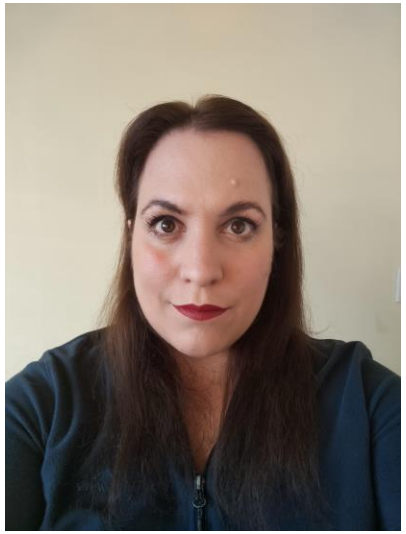
Introductions

Lucia Michielin

Digital Skills Training Manager

PhD in Computational Archaeology

Specialised in: Webscraping, Text and Data Analysis, Data Visualisation, GIS, 3D reconstructions, Photogrammetry



Yiqing (Eric) Liu

Second-year PhD student at the School of Informatics specializing in complex network dynamics

His current research focuses on modelling information diffusion across social networks..

Introductions

Why are you interested in text analysis?

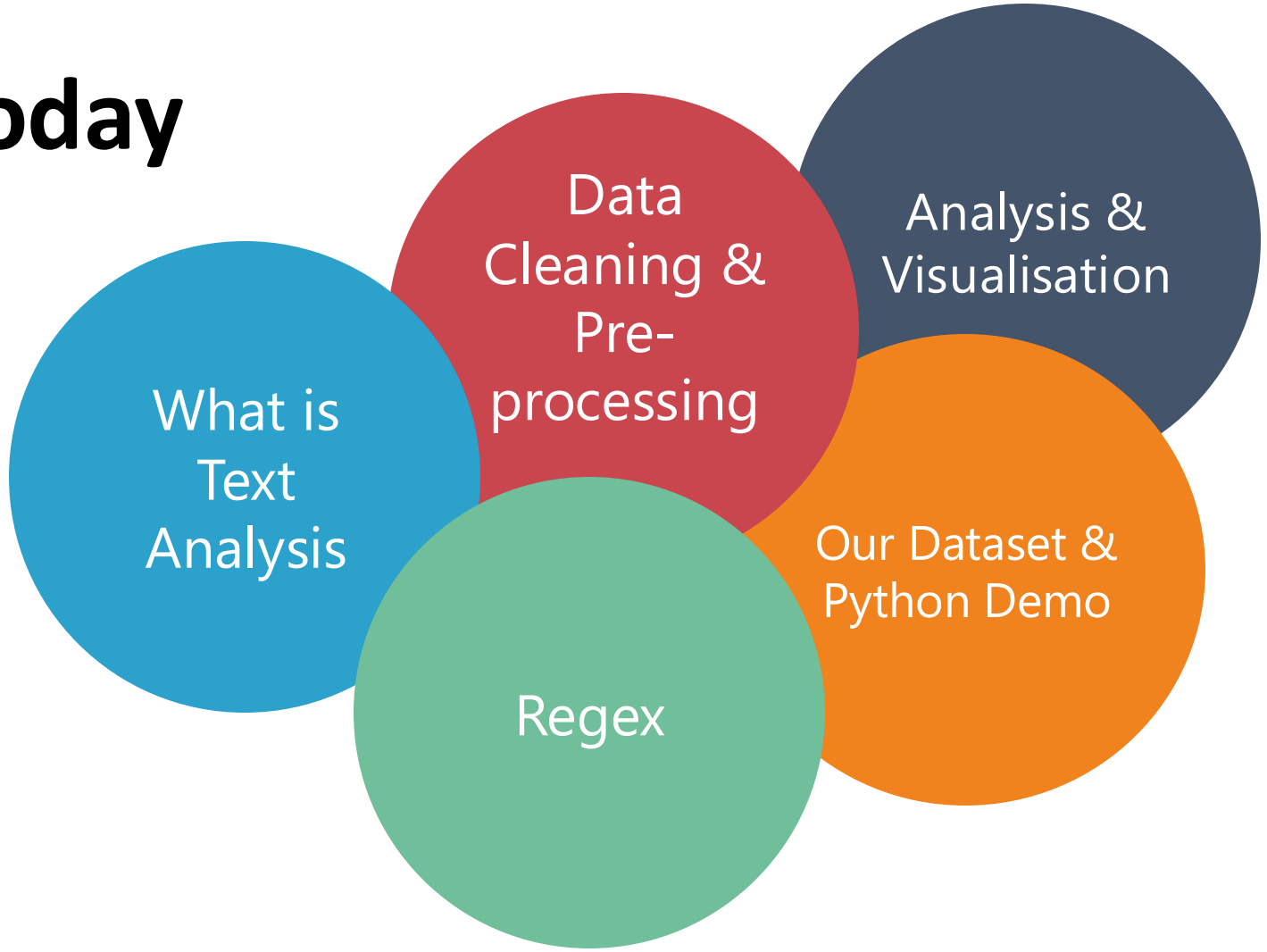
Have you used Python before?

Have you used Jupyter Notebooks before?

Have you used Regular expressions before?

Have you used NLTK before?

Our Plan for today

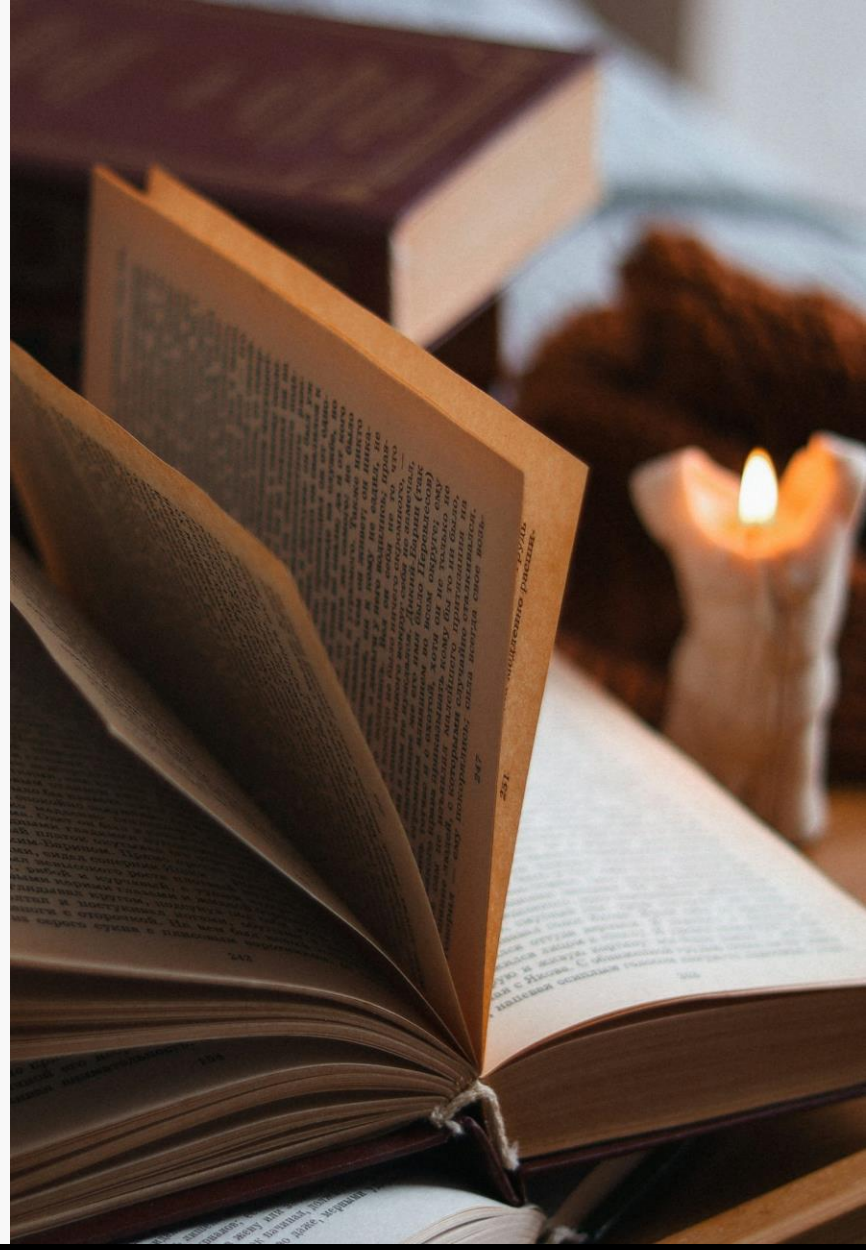


What is Text Analysis

- Computationally evaluating, investigating, and exploring textual (natural language) data
- Aims to identify patterns and trends in natural language data
- Methods from machine learning, linguistics, statistics, information studies, and other disciplines
- Advanced Techniques: Topic modelling, named entity recognition (NER), sentiment analysis, text classification
- **Some caveats:**
  - Training data matters
  - Computers “read” differently than we do
  - Potential for quantitative/qualitative mismatch in tools, results, statistical tests, etc.

Why would I use text analysis?

- To summarize HUGE amounts of text
- To study the evolution of an author's vocabulary throughout their publication history
- To study the differences in vocabulary or concepts in different groups of texts
- To understand how positively/neutrally/negatively people are writing book reviews or discussing certain topics
- To identify places or organizations named in a dataset or *corpus* (collection of texts)
- To extract certain types of information from a dataset
- To examine patterns in a text or corpus



General Terminology

Natural Language – Human Language (e.g. English, French etc) – unstructured data

NLP – Natural language processing – Text Mining – analysing textual data computationally

NLTK – Python Library that uses NLP principles to pre-process and analyse textual data

Distant Reading/ Text Analysis/ Text Mining – Doing Computational Text Analysis – What kinds of questions can you ask when you can use a programming language to study hundreds, thousands even millions of pages of text

Corpus – collection of texts, dataset used for the analysis

Standard Steps

Find Resources

Data Cleaning

Standardisation and pre-processing

Analysis

Data Sources for Text Analysis

- Books
- Newspapers
- Magazines
- Websites
- Transcriptions of audio
- Social media

NB! Always read the licensing/copyright information and terms of use
Text and data mining for non commercial research exception

Finding Text Sources

Libraries – NLS Data Foundry (data.nls.uk)

Project Gutenberg (Gutenberg.org)

Hathi Trust Digital Library (hathitrust.org)

Websites – Internet Archive (archive.org)'s Wayback Machine, UK Web archive(webarchive.org.uk)

Newspaper archives (Universities often subscribe to them)

Social Media data – More difficult now but still options

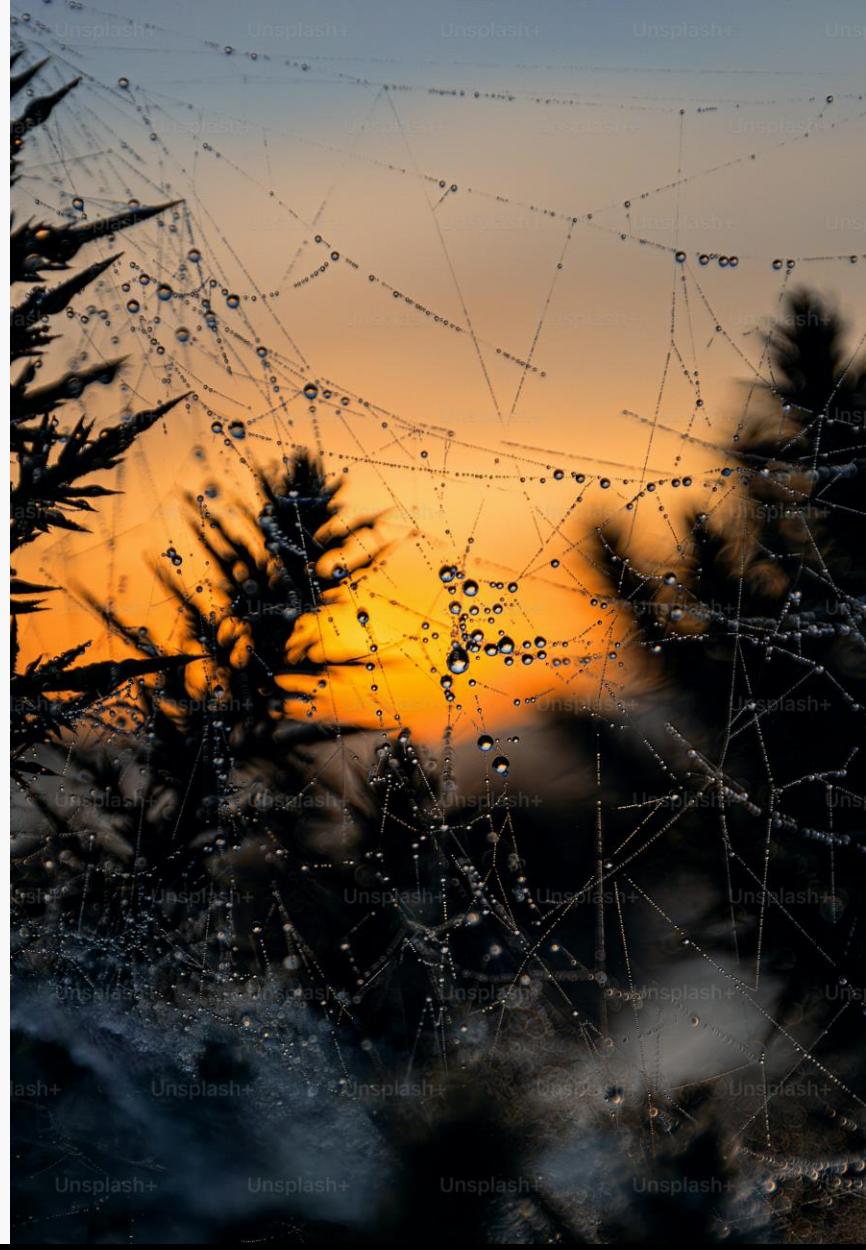
Finding Text Sources

OCR (optical character recognition) **and HTR** (handwritten text recognition)

- Creates **machine-readable** documents that can be searched, edited, and analysed computationally both code and code free options
- https://github.com/DCS-training/Image-to-Tech-Text-Extraction- (OCR), https://github.com/DCS-training/Transkribus (HTR)

Webscraping (web crawling and dynamic web pages scraping)

- Two main techniques: web crawling (for static website e.g. forum or news sites) and "social media scraping" (API used to be the most common for scraping dynamic pages)
- Intro to Beautiful Soup Python, Web scraping with R, Web Data Research Assistant (code free)



Out the Box Tools

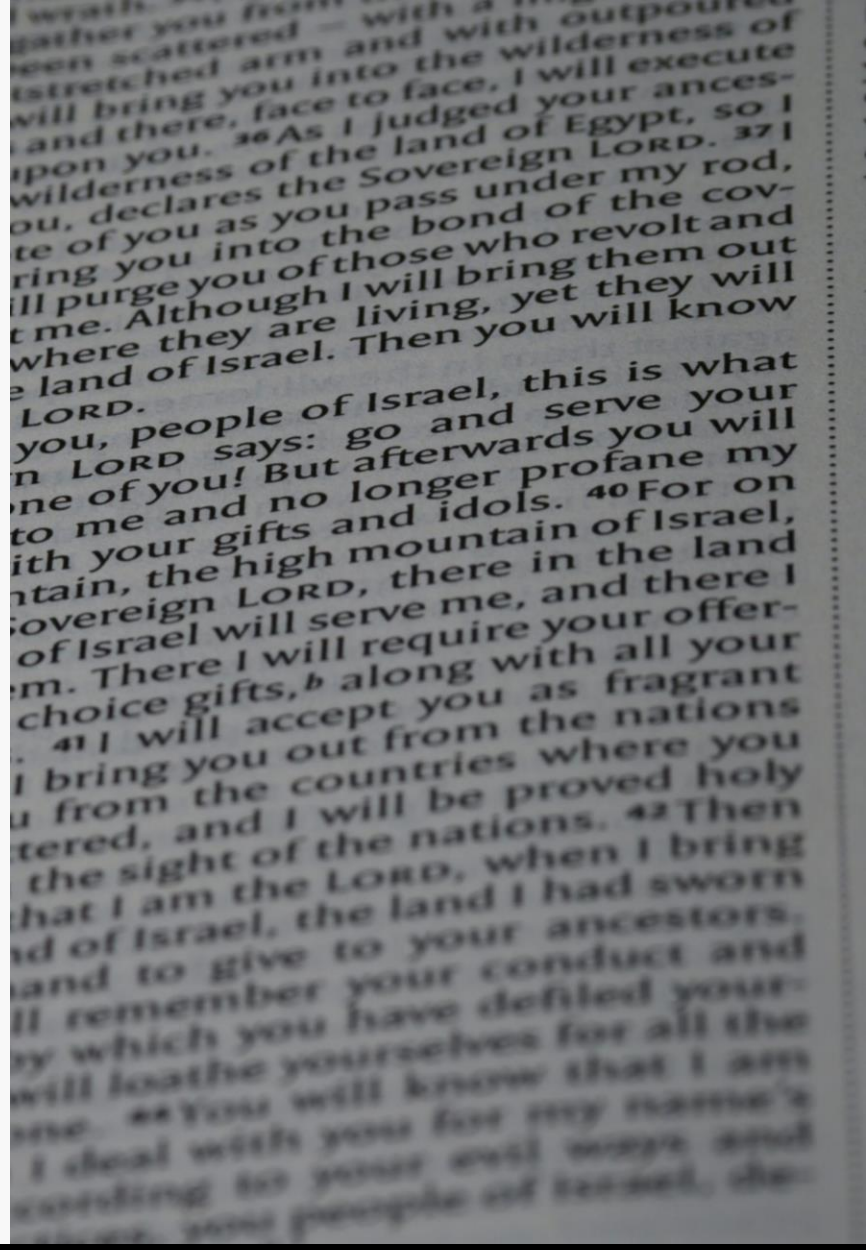
Voyant Tools - a beginner-friendly online platform for basic visualisations

Gale Digital Scholar Lab – Explore UoE primary sources + analyse code-free textual data

Corpus Analysis with Antconc – a user interface for text analysis that doesn't require any coding (with many of the capabilities you'll find in RegEx and Python NLTK), also open source

LancsBox - software platform for text analysis developed by Lancaster University

Tools from media labs, such as the **Digital Methods Initiative, University of Amsterdam** or the **SciencesPo MediaLab**



Code Based Tools

Python Based Packages - Python's **NLTK** is a popular text analysis library with many online resources, and it's open source (free!) There are also libraries like spaCy and Gensim which are usable alongside NLTK for many advanced text processing operations, such as topic modelling and NER.

Gensim is a Python library for unsupervised topic modelling and natural language processing. It is good at working with large text corpora, and at vector space modelling and document similarity analysis

R Based Packages - **QUANTEDA** (Quantitative Analysis of Textual Data) is a popular text analysis package for R users, which has a variety of models and sample corpora built in for text analysis.

The **TIDYTEXT** package in R is a tool for text mining using tidy data principles. It helps you manipulate and analyse text data by converting it into a "tidy" format — where each row is one word (or n-gram, sentence, etc.), and each column contains a variable (e.g., word, document ID, sentiment score, etc.)

Regex

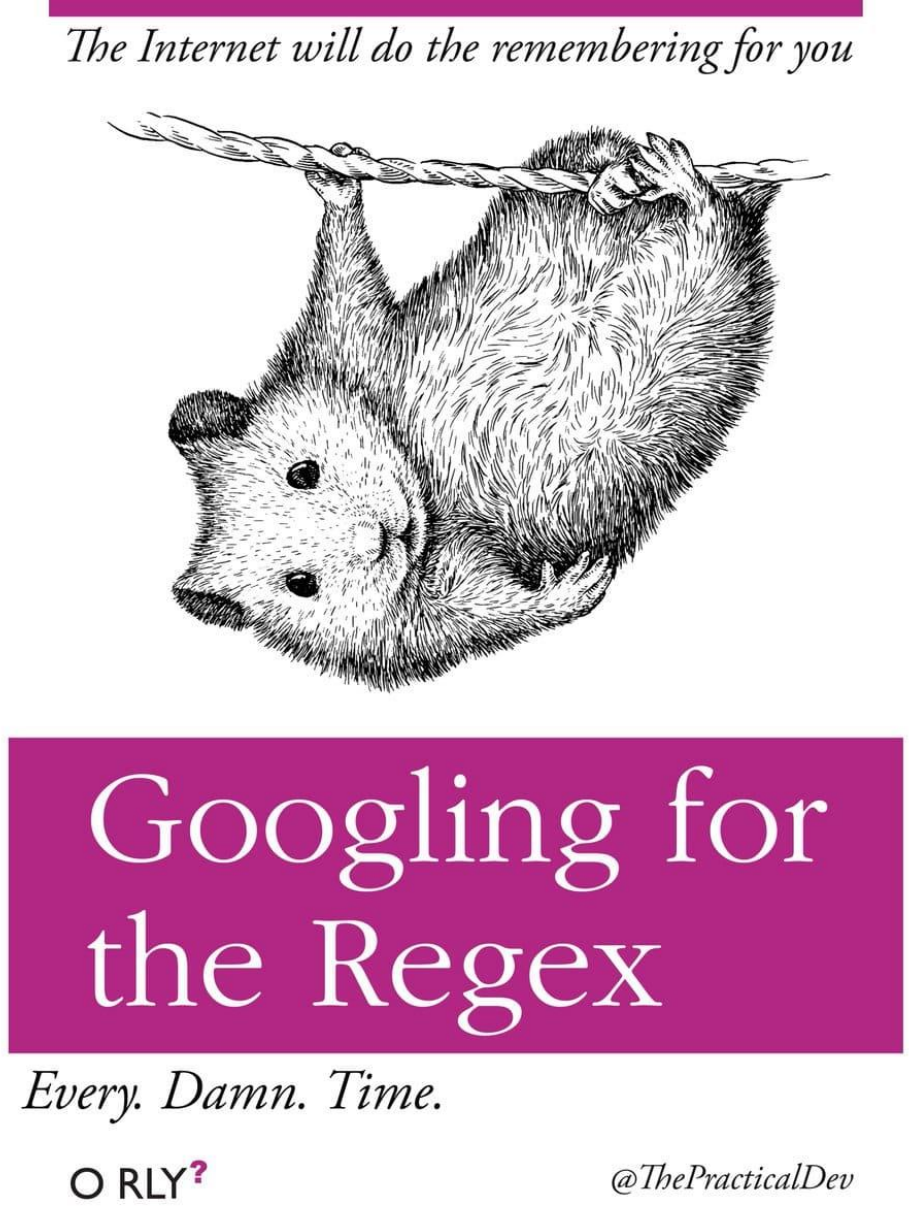
Regular Expressions (Regex) are a powerful tool used in programming for text processing. They provide a concise language for specifying text search strings

Core Functions

- **Search:** Locate specific sequences of characters within text
- **Match:** Check if a part of text meets a specific pattern
- **Replace**: Substitute designated patterns in text with new text
- **Split**: Break a string into pieces according to patterns

Key Uses of Regex

- **Validation**: Check formats like email addresses, phone numbers, URLs
- **Parsing**: Extract information like dates, IDs, specific codes from texts
- **Data Cleaning**: Remove unwanted characters or formats from text data



Regex

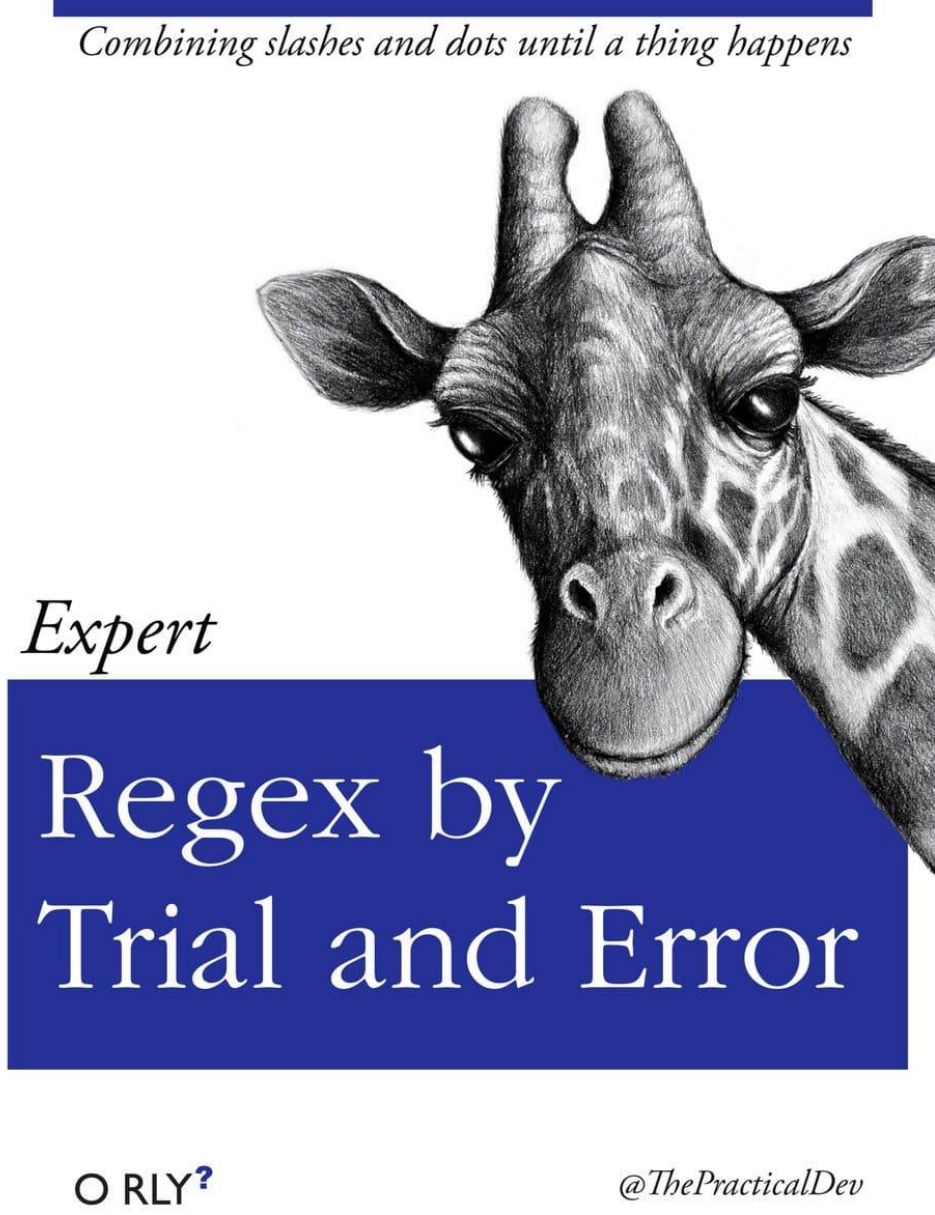
Regular Expressions (Regex) are a powerful tool used in programming for text processing. They provide a concise language for specifying text search strings

Core Functions

- **Search:** Locate specific sequences of characters within text
- **Match:** Check if a part of text meets a specific pattern
- **Replace**: Substitute designated patterns in text with new text
- **Split**: Break a string into pieces according to patterns

Key Uses of Regex

- **Validation**: Check formats like email addresses, phone numbers, URLs
- **Parsing**: Extract information like dates, IDs, specific codes from texts
- **Data Cleaning**: Remove unwanted characters or formats from text data

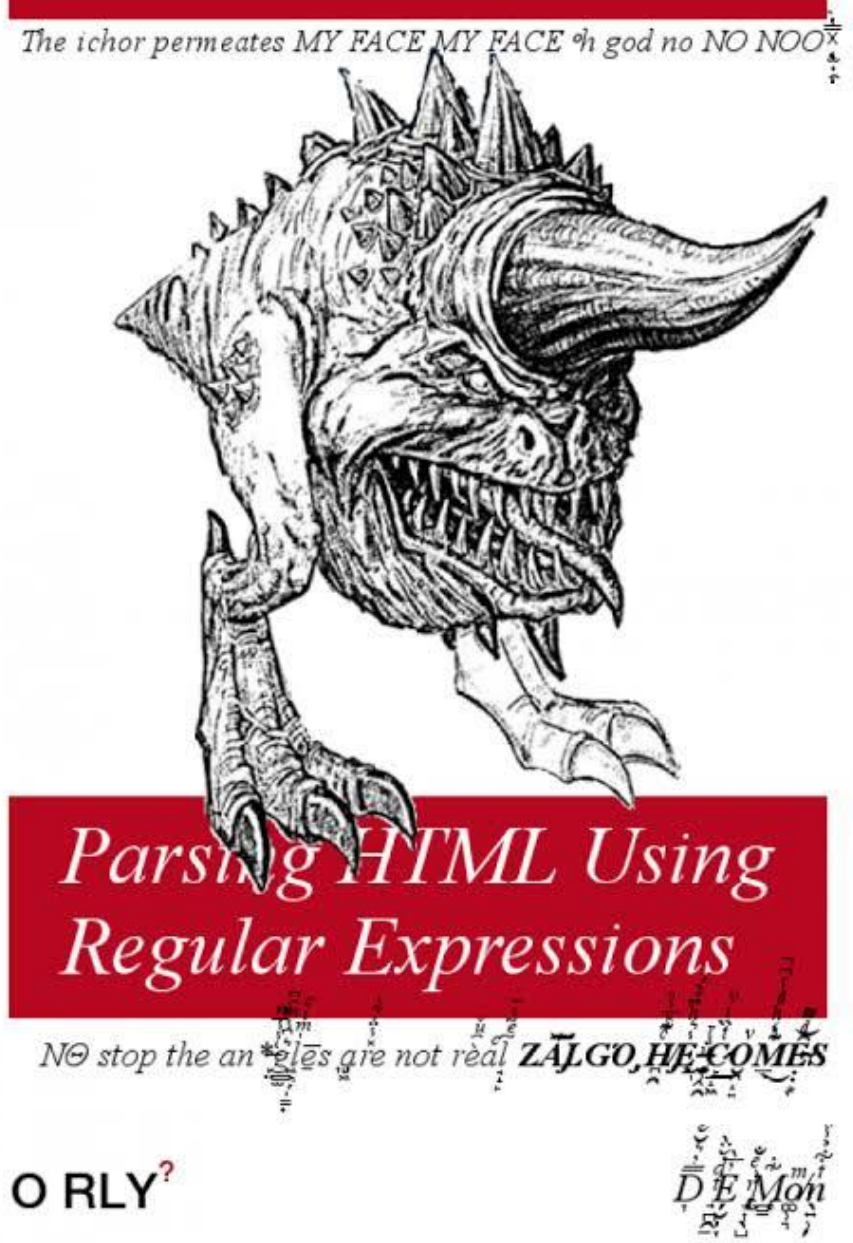


Components of Regex

Ordinary characters (e.g., a, 1, %) that represent themselves in searches unless specially treated

Metacharacters: Characters with special meanings (e.g., *, +, ?, ^, $, ., |, (), [], {}) that guide the regex engine to find different patterns

Quantifiers: Specify how often an element in a pattern should appear (e.g., * for 0 or more times)



Regex Cheat Sheet

. matches any character.

\d matches any single digit.

\w matches any part of word character (equivalent to [A-Za-z0-9]).

\s matches any space, tab, or newline.

\ used to escape the following character when that character is a special character

^ is an “anchor” which asserts the position at the start of the line.

$ is an “anchor” which asserts the position at the end of the line

\b asserts that the pattern must match at a word boundary. (e.g. bound not unbound)

| or (e.g. gray|grey)

? The question mark indicates zero or one occurrences of the preceding element

* The asterisk indicates zero or more occurrences of the preceding element.

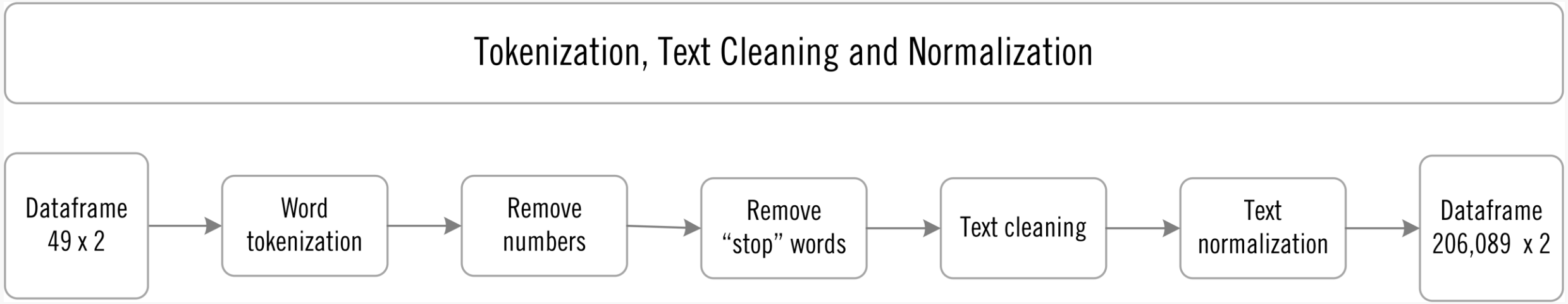
+ The plus sign indicates one or more occurrences of the preceding element.

{VALUE} e.g. \d{1,9} will match any number between one and nine digits in length

Data Cleaning & Pre-processing

Cleaning & Organise – Often the Data you will star from will need cleaning (mostly done via Regex) and Organising

Pre-Processing – Even if your dataset is already cleaned and organised you would need to pre-process it before analysis



Data Cleaning, Organising

Structure your corpus/dataset – Depend on the tool (normally csv or collection of txt). How complicated this step is depend on the starting dataset

Clean texts – If you are working with Ocr'd material that can be cleaned programmatically

Collect and prepare your metadata – Either separately or as part of the csv you need to think to which information you will need for your analysis (e.g. year, author …)

NB this step can be very quick or incredibly long depending on what are your question, where are the data located, how messy they are. They are also not standardised steps so what to do will change each time

Pre-Process

Formatting text for analysis and removing extraneous information and normalise it

Workflows vary depending on research objective, field, and dataset

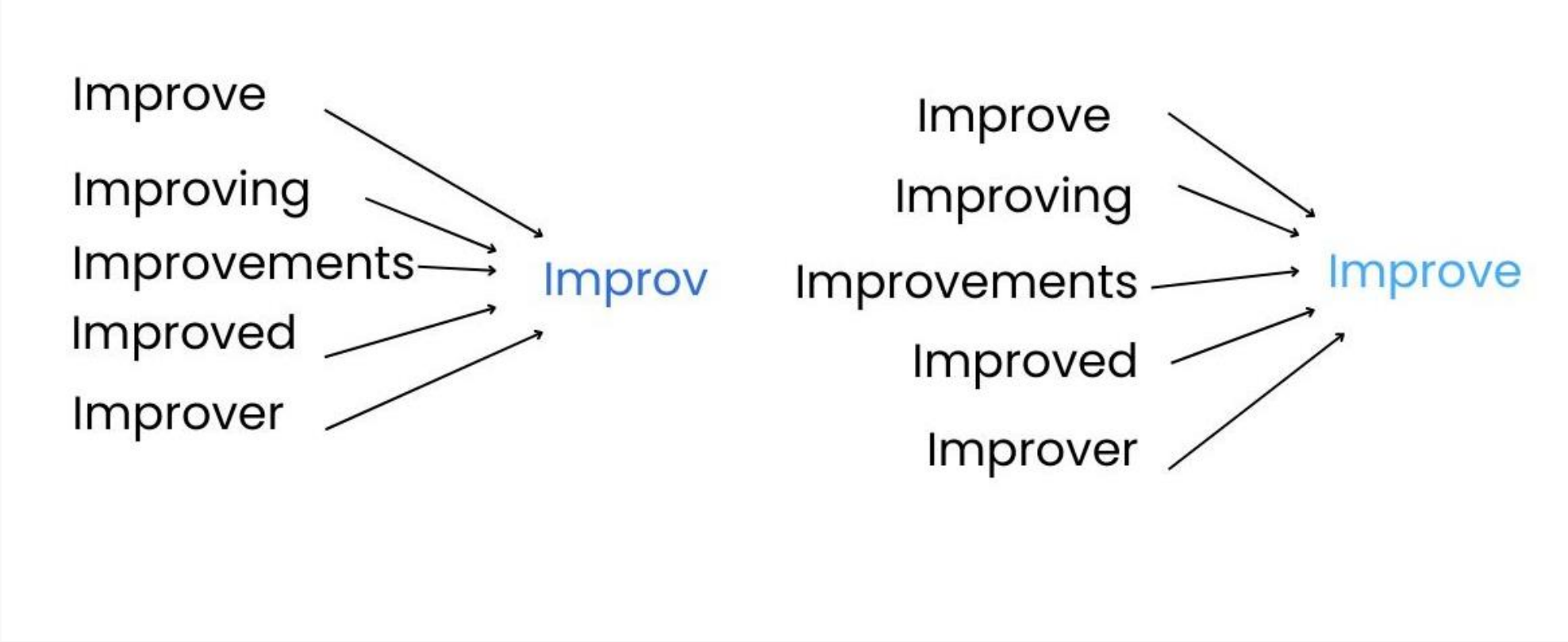
Common steps include standardising capitalisation, removing URLs and symbols, stop words removal, tokenisation, stemming, and lemmatization

Stopwords include words like “a,” “the,” “of,” “an” that don’t add meaning to the dataset

Tokenisation

- Token= unit of analysis
- Tokens can be **individual words**, **sentences**, or even **characters**, depending on the level of granularity desired.
- Tokenisation helps in **standardising** and **organising** text data, making it easier to analyse and process.
- Word-based tokenisation breaks down text into individual words, treating each word as a separate token.
- **Bag of Words Approach** – if our object of analysis is the word we lose grammar, words order etc.

Stemming & Lemmatisation



Analysis

- **Keyword Search/Term Frequency** – Counts the occurrences of a specific word within a document

- **Inverse Document Frequency (TF-IDF)** – Measure that assesses how crucial a word is to a document within a corpus, reducing the weight of commonly used words

- **Key-word in Context (KWIC)** – Displays occurrences of a specific word alongside its surrounding words, offering insights into the usage and context of the word within the document.

- **Bigrams & Ngrams** – Refers to sequences of two (bigrams) or 'n' (ngrams) consecutive words used for statistical analysis of text patterns, capturing more contextual information than single-word analysis.

- **Collocation Analysis** – Analyses the frequency and tendency of pairs or groups of words appearing close together more often than by chance, identifying commonly co-occurring terms and phrases.

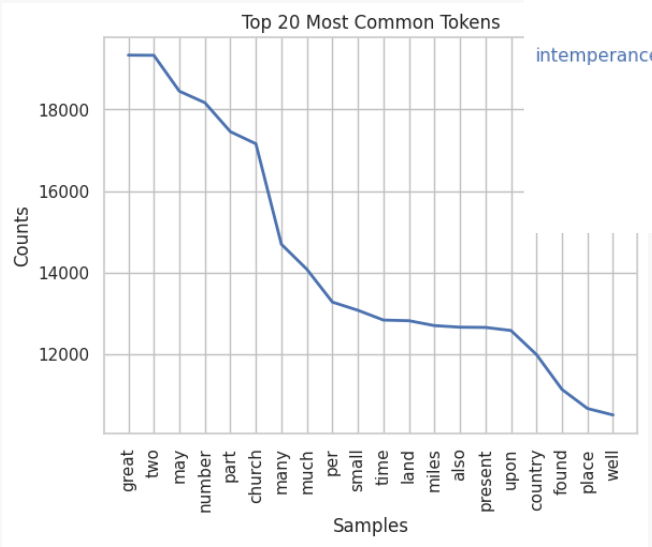
Advanced Techniques

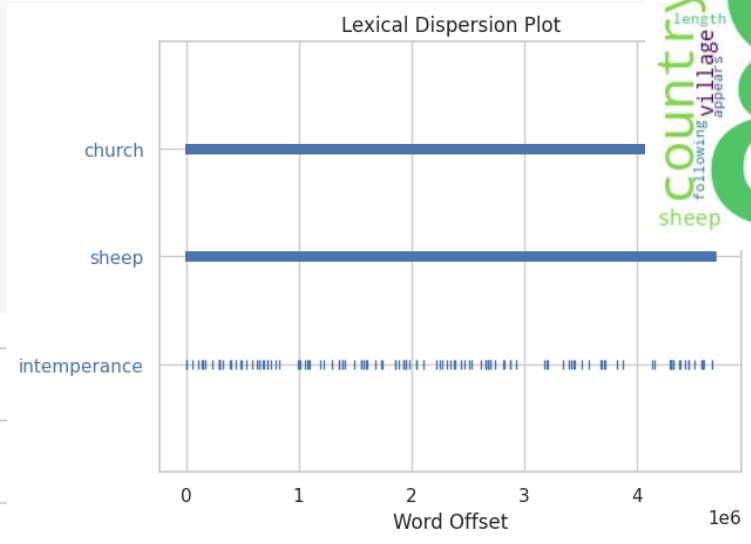
- **Part of Speech tagging** – identify language structures (More info here, try it here)
- **Named entity recognition** (NER) –to find people, places, organizations, etc. (More info here)
- **Topic modelling** – unsupervised Machine Learning technique to identify topics in large corpus (video introduction/examples here)
- **Word embedding** – to study meanings and how word relate to each others (Try it out here)
- **Classification** – e.g. Sentiment analysis, each word is associated with a +/- value (try it out here)

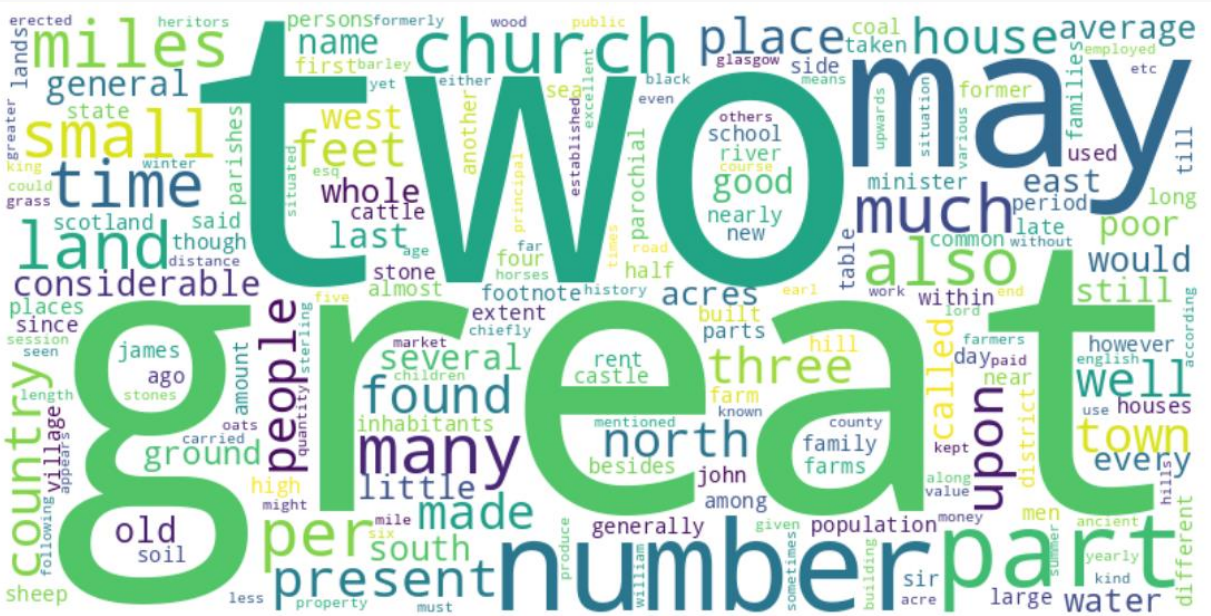
Visualisation

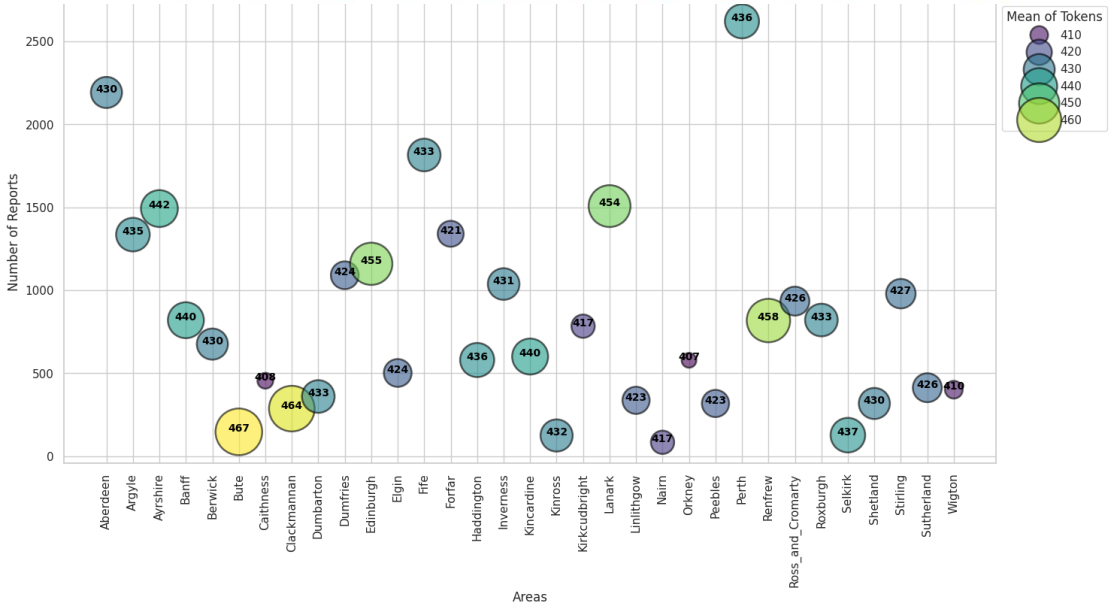
Standard visualisations

- Explore dataset
- Word-cloud
- Frequency plots
- Dispersion Plots





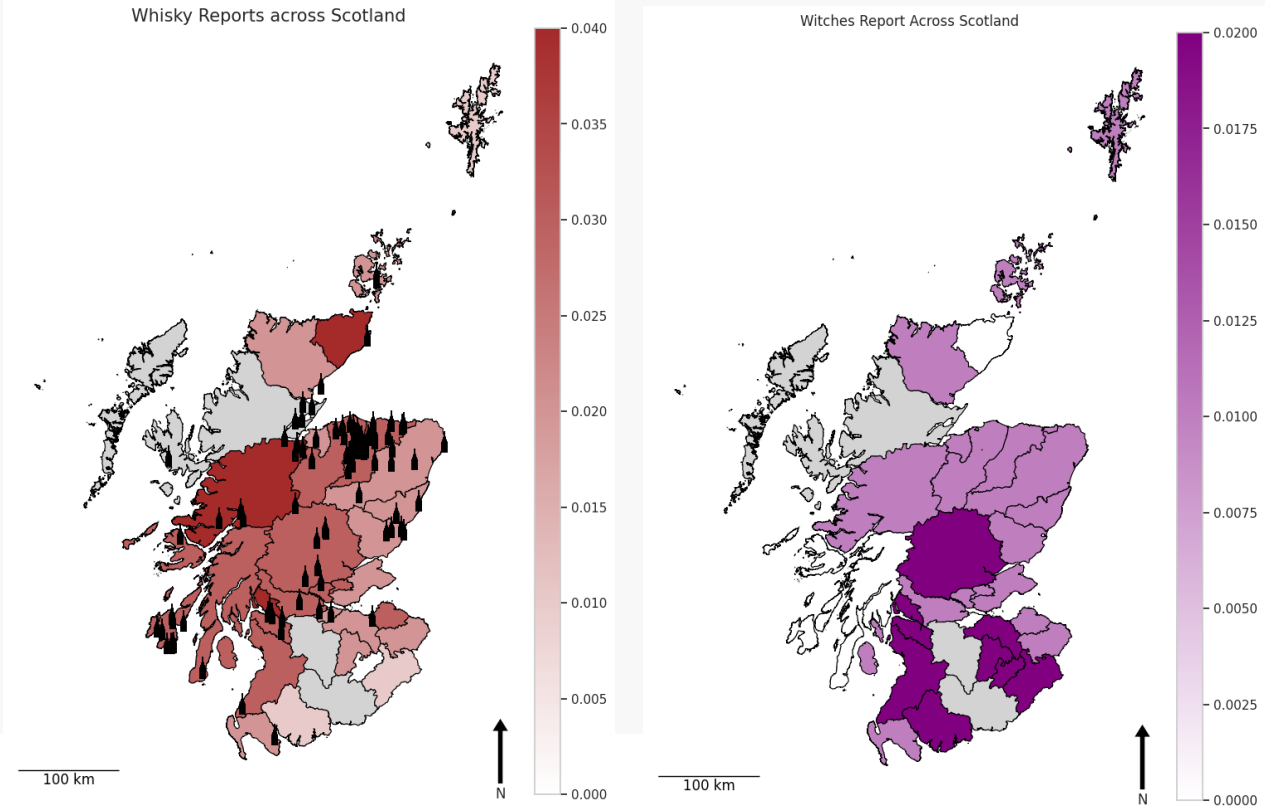


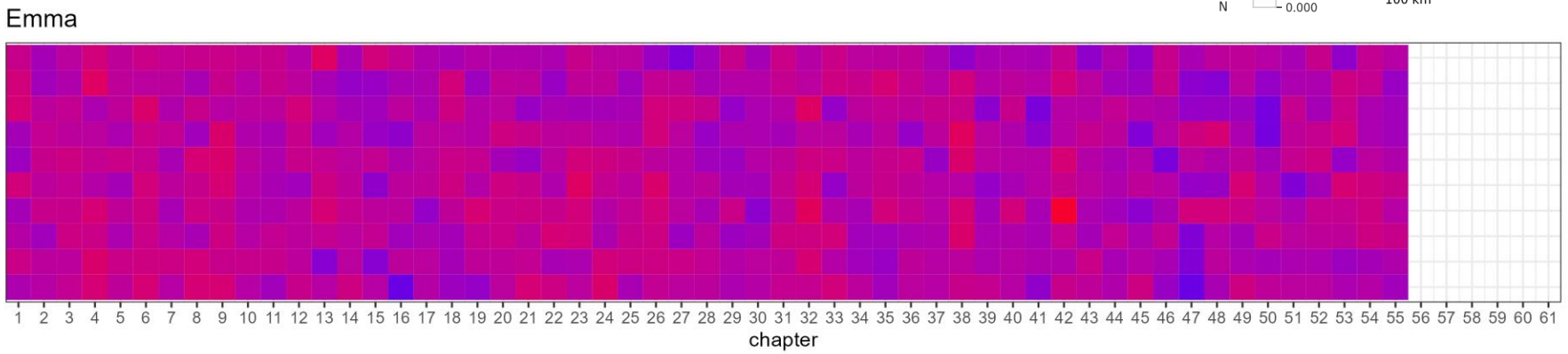


Visualisation

Advanced visualisation

- Geographical plots
- More artistic outputs

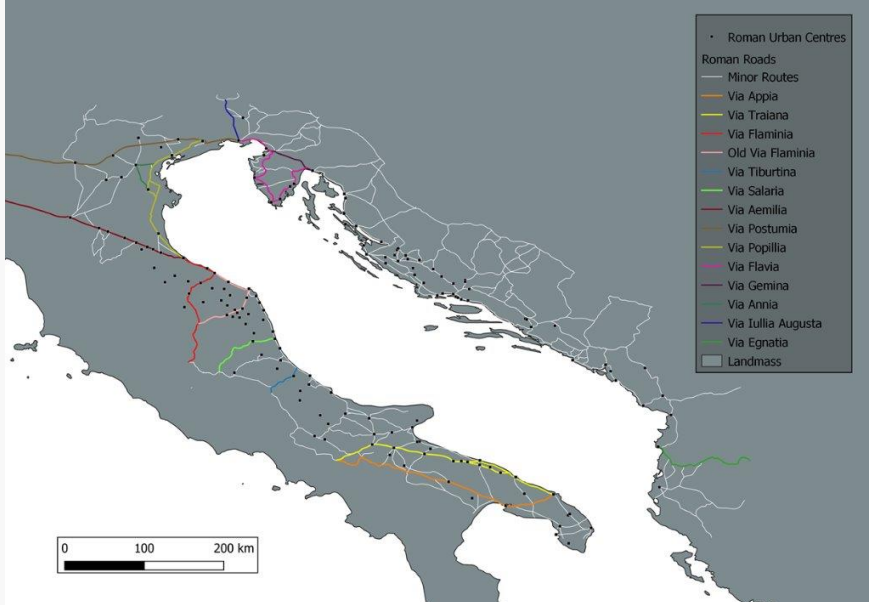


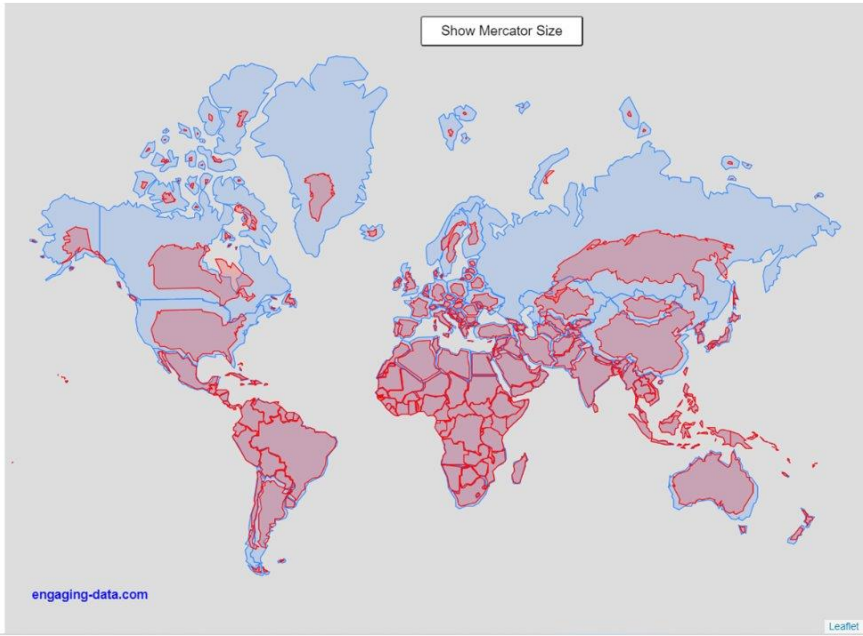


Geographical Data

Vectors

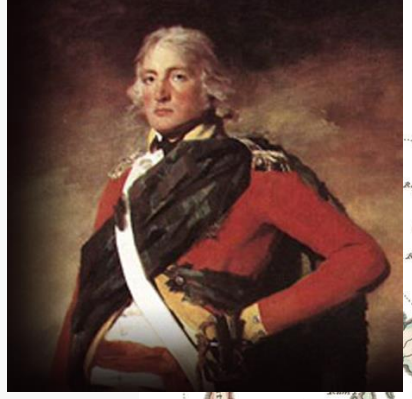
- Generalised representation of the real world
- Composed of XY coordinates (Lon Lat) = vertices
- Based on Reference System (each one is different)
- Points, lines or polygons
- Normally in .gpkg (Geopackage) format or .shp (Shapefile) format

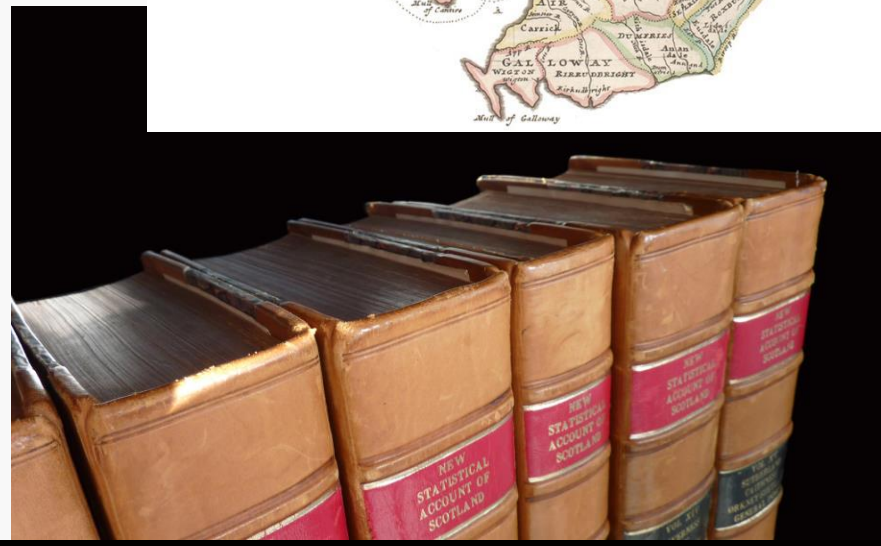




Our Dataset

- The ‘Old’ *Statistical Account* (1791-99), under the direction of Sir John Sinclair of Ulbster, and the ‘New’ *Statistical Account* (1834-45) are reports of life In Scotland during the XVIII and XIX century
- They offer uniquely rich and detailed parish reports for the whole of Scotland, covering a vast range of topics including agriculture, education, trades, religion and social customs
- https://stataccscot.edina.ac.uk/static/statacc/dist/home
- Everything from changing fashions in dress to the different attitudes to smallpox inoculation and resulting high infant mortality between the north and south of Scotland
- Our datasets are **29,083 .txt files** corresponding to single reports from the statistical accounts





Our Demo

The Links you needs are

Our Repositories (where the notebooks and data are)

- https://github.com/DCS-training/Text-Analysis-and-the-Humanities (Notebooks)
- https://github.com/DCS-training/StatAccountScotland (Data)

Running the Code

- https://noteable.edina.ac.uk/login
- https://colab.google/

Time for Python

Questions?


1729
1870
·I·WAS·A·STRANGER·
·AND·YE·TOOK·ME·IN·
PATET
OMNIBUS
·I·WAS·SICK·AND·
·YE·VISITED·ME·